Global Energy Conservation in Large Data Networks

Lisa J. Durbeck

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Joseph G. Tront, Chair
C. Jules White, co-Chair
Nicholas J. Macias
Devi Parikh
Dhruv Batra

December 4, 2015
Blacksburg, Virginia

Keywords: Energy Conservation, Networks, Computing, Communication

Global Energy Conservation in Large Data Networks

Lisa J. Durbeck

(ABSTRACT)

Seven to ten percent of the energy used globally goes towards powering information and communications technology (ICT): the global data- and telecommunications network, the private and commercial datacenters it supports, and the 19 billion electronic devices around the globe it interconnects, through which we communicate, and access and produce information. As bandwidth and data rates increase, so does the volume of traffic, as well as the absolute amount of new information digitized and uploaded onto the Net and into the cloud each second. Words like *gigabit* and *terabyte* were needless fifteen years ago in the public arena; now, they are common phrases. As people use their networked devices to do more, to access more, to send more, and to connect more, they use more energy—not only in their own devices, but also throughout the ICT. While there are many endeavors focused on individual low-power devices, few are examining broad strategies that cross the many boundaries of separate concerns within the ICT; also, few are assessing the impact of specific strategies on the global energy supply: at a global scale. This work examines the energy savings of several such strategies; it also assesses their efficacy in reducing energy consumption, both within specific networks and within the larger ICT. All of these strategies save energy by reducing the work done by the system as a whole on behalf of a single user, often by exploiting commonalities among what many users around the globe are also doing to amortize the costs.

Dedication

may this benefit its readers and the larger society.

Acknowledgments

This work was supported in part by the U.S. Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program, by a Hume Center Graduate Fellowship, and by a P.E.O. International Scholars Award.

This would not have been possible without many people at Virginia Tech who played a part in the process, including my committee members Jules White, Joe Tront, Devi Parikh, Nicholas Macias, and Dhruv Batra, whom I thank for both practical assistance and for interesting discussions along the way. I also am indebted to Peter Athanas for engaging me and encouraging me along the many stages of this research. I also would like to acknowledge the Bradley Department of Electrical Engineering department head and office and IT staff, including graduate coordinator Cynthia Hopkins for many critical *saves* at various junctures; Academic Advisor Leslie Pendleton; Director of IT John Harris and Systems Administrators Calvin Winkowski and Branden McKagen for their knowledge and dedication; Fiscal Technician Melanie Gilmore; Virginia Tech Graduate School Advisors Gwen Ewing and Priscilla Wright; and the Virginia Tech library staff and all the subscriptions the library maintains and indexes.

I would particularly like to express my gratitude to the ones who made it possible for me to see this through. I thank members of my family and my second family—Karen Akerlof, Mike Berigan, Liam, Cormac and Kieran Berigan, Mike Cutlip and Ed Bertz—for providing me material support and companionship at key points in my PhD. I would also like to properly acknowledge my parents,

Dr. Patricia Kenney Bertz and Dr. Donald C. Durbeck. When I reflect on what went into achieving this milestone, I appreciate that their role in reaching this is likely no smaller than mine. I also thank my sisters Kathleen Suher and Heidi Hershock for continuity, fellowship, engagement, solace, advice, humor and entertainment all along.

I also am grateful to the many individuals who posted useful information on the web that got me out of many a stuck place, in Stack Overflow and other community forums, and for research tools such as search engines, IEEE Explore, the ACM Digital Library, and Google Scholar. Doing this research twenty years ago would have been significantly harder and more tedious, or done in a larger vaccuum.

Lisa J.K. Durbeck

Preface

William McDonough, coauthor of *Cradle to Cradle: Remaking the Way We Make Things*, gave a talk at the Jefferson Center in Roanoke a few years ago. As his book is an important addition to the conceptual base of Western thinking on environment, I attended his talk. One of the tenets of his book is to embody a view of this world as one of *abundance*, not *scarcity*, and yet to endeavor to make human production beneficial to the earth: beneficial to all inhabitants of all species in all of the future. The near-koan it gives is to make human production and manufacturing systems like a cherry tree.

One of the notions at which McDonough took direct aim in his talk was the notion of saving the planet by greater *efficiency* of human endeavors—such as the *energy efficiency* work presented in this dissertation. This is the privilege and the mantle of visionaries: to show you how short of the mark conventional wisdom falls. If you are trying to get to Atlanta from here, he said, does it make sense to drive to Washington D.C.—only *slower*? You are never going to reach Atlanta that way. Doing the wrong thing, only slower, is not the answer.

In the absence of a better idea, you *remain closer* to Atlanta by driving slower to Washington D.C.. And that is the hope that motivates this work: by increasing the energy efficiency of one of the major systems in use today, the backbone of digital culture, we will give ourselves *longer to think*, and give future generations a planet not as close to the brink, providing more resources, time and potential for the *sea change* needed to decide to go to Atlanta instead.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

An enormous amount of research has been devoted towards low-power Internet devices and appliances, whether they are in our offices, idle at home, or in our pockets. The primary focus of these efforts has been on reducing energy costs, extending battery life, lowering heat emissions, based on individual-centered metrics. Similarly, many have shown that energy can be saved by splitting the execution of applications between leaf nodes and the cloud. In many of these instances, node energy is saved at the expense of much greater energy effort within the cloud. The purpose of this work is to take a more holistic look at a community's data network and examine strategies for reducing the *overall energy consumption*. The goal of this work is ultimately to trim the cost in power and infrastructure to transfer content, based on informed knowledge about the specific network in question: its architecture, topology, physical infrastructure; its customers; and their behavior.

While a great deal of research has gone into network topology and routing, and into network equipment and infrastructure, with the goals of low latency and satisfying quality of service agreements—and into improving device size, weight, area, and power across the board—to provide a form of conservation of energy per bit, the focus in this work is instead on uncovering the benefits from an energy conservation standpoint of various strategies for *conservation of bits* transmitted across the

network, including entropy reduction, more efficient distribution of popular content transmission points across the network, and more efficient content delivery methods, such as multicast delivery. This examination provides an often overlooked contribution to global energy conservation in communities due to the cross-cutting nature of global conservation within such a complex structure as the ICT, that has so many separated concerns and entities involved.

Chapter 4 examines conservation of bits within the message from an energy standpoint, while Chapters 3, 5, and 6 examine energy savings attributable to various methods that reduce not the size of the message but the number of times it is repeated. This work aims at assessing the energy efficiency achievable through how content and messages are replicated within the communication path traveled by the communication between sender and receiver.

1.1 Motivation

The entire web of information and communication technology on the globe today (ICT) is estimated to consume up to 7–10% of the energy used on the planet [1–3]: roughly 1–3% by the telecommunications network [1, 4], 1.3% by data centers [5], an undetermined amount by in-house institutional networks, and as much as 4–7% consumed by the access network and all of the electronic devices and computer equipment attached to it [2, 6]. Network traffic is growing exponentially [7]; network energy consumption could also grow exponentially, by one estimate encompassing 75% of global energy supply in 2025—as the number of devices added to the network continues to increase and, with it, communications—if the technologies involved were frozen at their 2010 levels of energy efficiency [8]. Far from signaling the collapse of global energy supplies and a global energy crisis, what this estimate reveals instead is the remarkable pace of innovation that has kept up with the growth in ICT to date so that energy use is matched with greater energy efficiency.

This rapid growth in demand for ICT is due to a number of simultaneous trends in digitized and shared information, such as widespread ownership of networked devices and the use of increasingly sophisticated mobile devices, the proliferation of sensors and of automation, and the past four decades of gradual transfer of commercial activity from local transactions—done on paper or on a standalone accounting system—to increasingly merged, shared, computer-based and networked systems for order, delivery and payment. While the underlying activities have in many cases not fundamentally changed, the way that people engage in those activities has become *digital* and *online*. The question addressed here is how to quantify the energy performance of various activities so that more efficient solutions can be identified for adoption, slowing the growth in the total *metabolic rate* by which ICTs consume energy.

The phenomenon of the Internet—the global data communication network, or, more precisely, the merged electronic, optical, and radio communication systems and computer networks through which digitized information produced or collected, encoded, communicated, analyzed, and utilized—will access and transmit more information this year than in all prior years, following an exponential growth curve [7, 9]. This phenomenal growth has its roots in the Second World War with the rise of encoding, communicating, and decoding secret messages by such legendaries as Alan Turing, followed in 1948 by Bell Lab's publication of Claude Shannon's seminal work on Information Theory—work that fostered many of the electronic communication technologies that constitute the Internet of today [10]. What began then has snowballed, after a half century of considerable technical and financial investment in mass digitization of the processes and products of industry and culture.

Energy consumption of personal electronic devices such as mobile handhelds receives a great deal of attention so that devices' utility to customers increases while their battery life also improves; adding new capabilities while improving size, weight, power and area are perpetual concerns within the deliberate evolution of electronic devices. In a sense, then, the motivations are

already sufficient in this arena to bring about significant improvement in energy efficiency.

There is considerable economic impetus for new strategies to accommodate new growth, so that information and communications networks continue to grow in size and capacity while shrinking in unit materials cost and energy usage, in signal transmission costs, in end-to-end latency, and other desiderata that lead to greater resource productivity or efficiency. Given the current capabilities of networks, and looking to future networks, are there ways of, say, organizing a computation on a network with the global objective of reducing energy expenditure? This research looks at achieving greater efficiency by changes at the data layer, which is to say not by redesign of channels or transmission media, or transmitters or receivers or other equipment, but by shepherding the bits, optimizing where and how a bitstream is processed and transformed on a given network, considering the effects in terms of power consumption, latency, and lower transmission and bandwidth requirements.

To illuminate the tradeoffs of a computation that performs entropy reduction on its inputs, for example, and to understand entropy reduction's role in energy consumption on large data networks, for instance, one might compare the entropy of two different processing chains on the same hardware, such as two scenarios for delivery of home movies using two video compression techniques where one more computationally intensive technique results in a more-compressed video stream. What are the net energy effects of trading the increase in computational energy and latency of the more aggressive compression method for the transmission bandwidth reduction due to data reduction, or greater entropy? The static aspects of this question are directly addressed in Chapter 4, for the more generalized notion of entropy reduction of content delivered to home residences through the application of compression techniques to the data streams. The energy used at the leaf nodes is directly compared to the energy used within the network by various compression schemes.

How might these changes play out at full network scale with millions of users, changing for instance the network traffic load and the energy grid draw and device battery draw from greater

computational energy use on the network periphery? The problem changes when one examines the energy profiles of an ensemble of users; Chapters 3, 5, and 6 look at these higher-level effects arising from an ensemble of users.

While commercial data center facility energy use has shown remarkable progress, as is discussed in more detail in Chapter 6, other ways to reduce the energy used on behalf of Internet devices within the global telecommunications network have not yet been used to their full benefit. One pertinent example is exploiting the economies of scale that exist due to the remarkable overlap between what individuals are using the Internet to do, creating some definite patterns and commonalities among what individuals watch or download at any given moment, thereby giving a high degree of collective redundancy to Internet transmissions. Redundant transmissions are wasted bits; by some measures, fully capitalizing on this wasted redundancy could reduce the number of transmissions at individual network nodes by as much as two orders of magnitude, as estimated by Aaltonen [11].

The work presented here on energy performance is applicable to communities, cities, or nation states; it crosses the boundaries and confines of the separate businesses and industries that contribute to computer and network systems and power generators and suppliers. It can be used to enact policy aimed at reducing energy consumption across the many divides and concerns that amount to network infrastructure, such as Internet service providers, as well as application infrastructure, such as content providers. It also theorizes in Chapter 4 on longer term trends in network and application energy performance, and how they intersect over time.

A specific goal is to analyze the *energy consequences* of recent FCC rules that violate a prior policy of *net neutrality*. If, as a result of the FCC's reversal, content providers such as Netflix decide to fundamentally change their content delivery framework to avoid paying Internet Service Providers (ISPs) premium prices, what might be the *energy consequences* of the new content delivery structures and methods currently under consideration? Verizon is reportedly considering development of a multicast-based content delivery structure as the recent competition pressure mounts from

mergers of telecommunications companies with television and other media companies [12]. Others have proposed that Netflix adopt a bitTorrent model of peer-to-peer content sharing and delivery to outwit ISPs increasingly eager to externalize the cost of delivery of streaming video content to customers [13].

The goal of this work is to answer whether this would be far more, or far less, energy intensive than current ways that Netflix and others deliver content, such as private or commercial content delivery networks (CDNs). While Netflix's corporate charter restricts it to answering this question only in the limited sense in terms of how it aids the goal of maximizing return to shareholders, here we look at it from the perspective of the global demand for energy and the need to conserve energy as the network and its uses grow exponentially, and where policy may be set to temper global demand for energy.

The general hypothesis pursued here is that the global minimum energy is lower than the sum of individual energies of subparts more narrowly defined. How this plays out varies based on the aspect of the ICT under consideration; one example pursued here is that the global minimum transmission cost in energy to deliver content to n network users is significantly lower than the cost to deliver the content to each of the n users individually, by leveraging work done within the network for prior requests.

The structure of the remaining document is as follows. Chapter 2 gives some background common to all aspects of the work. Chapters 3 through 6 each present forays into the large space of possible approaches to achieve energy efficiency; each takes a different approach to energy savings within the global ICT. For each approach that is presented, the approach is described; the plausibility of this approach as a major source of savings is argued; an estimate of the energy savings possible from that approach is presented, based on examining one or more cases. Chapter 7 assesses the large-scale impact of these techniques in practice when used in the global ICT. The final chapter, Chapter 8, provides a conclusion to the work that integrates the results across the multiple areas

explored, gauges the broader impact of this work, and identifies areas for future work.

1.2 Defining the Domain of This Research

There are several ways in which the present work views a different problem from those typically confronted by network research. The concern here is with energy performance in Joules, a timeless unit. This work does not focus on power or other time-based measures of the rate of energy flow, yet the results can readily be extended and related to other time-based metrics like throughput or speedup. Working in the realm of energy potential rather than energy use rate renders this analysis independent of the current state of the art and provides a framework for analyzing energy profiles in future networks.

The reciprocal gain for Joules spent is considered in the realm of *information*, in the sense meant by Shannon, in the estimation of the efficiency of productive work, when gauging how much is gained for energy spent. This is in contrast with other research focused on packet-level optimizations of either devices or algorithms and protocols. The focus on information-level optimizations rather than packet-level optimizations reveals the non-productivity of various system aspects and allows the assessment of the *information-efficiency* of traffic and transmission patterns and other aspects of the ICT.

The methods used in this work extend beyond networking *per se*, and this should not be categorically labeled as network research. Rather, it is research performed upon networks, or research the subject of which is networks, utilizing low-level activity- and energy experiments on hardware and software deployed within the global ICT. It likely belongs in the field of systems analysis, mathematical modeling, and constraint-based optimization of systems, utilizing low-level activity- and energy experiments on hardware and software deployed within the global ICT.

Chapter 2

Background

2.1 The Journey of a Bit through a Network

This section introduces terminology useful to the subsequent material by tracing the journey a typical bit takes through the Internet. A *bit* or binary digit is the smallest unit of a digital or electronic file such as a photo taken with a digital camera, or a photo scanned into a PDF using a flatbed scanner. The binary number representation contains only two values, 0 and 1, and complex information such as the color of each pixel of an image is digitized—that is, encoded into binary digits—before being transmitted across the network.

A photo is not a photo the way we think of it as it traverses the network; rather, it is *encoded* into a pattern of ones and zeros, and then serialized. The encoding is an issue unto itself; it really is more of an art that has been under near-constant pressure to evolve since the origin of electronic or digitized information. The evolutionary pressures have largely been to accurately represent the *analog* world in the binary representation, and to retain the meaning of the original object or message as it traverses the physical/electrical/chemical transforms of networked systems to reach its final destination or recipient or action or goal. The unwanted stuff of reality not quite meeting the abstraction mixes with the original information along the way, adding *noise* that makes the original meaning less clear.

A *user* of the system might take a photo with the camera of her smart phone or with a digital camera; the light hits a plane of sensors that transforms the light of different wavelengths into colored pixels. The photo is a transform of light into a 2D image that is a 2D array of pixel values. This encoding of a scene into a rectangle of pixel values is often encoded a second time, via a *compression algorithm* running on the smartphone, that reduces the original encoding, replacing a line of white pixels with a symbol and a length of bits, for example, when compressed via gzip, or by reducing the number of colors used to represent the original scene, and associating one color from this smaller set with as large a rectangular region as possible in the original image, to produce something like a JPEG photo file. This second encoding is to reduce the size of the image stored on the device—stored on the hard drive of a laptop, or in the flash memory of a small handheld device. This is a measure taken largely to reduce the cost associated with taking pictures or the inconvenience of running out of storage space or disk space.

The user might decide this image is worthy of sharing with others, and post it to a social media or photo-sharing web site. For this, the digital camera or smart phone or tablet or computer or laptop—the general term for which is *electronic device*—transmits the photo over the web. To do so, the photo is converted into one long thin message, a *bitstream*, that is sent following a particular protocol such as TCP, as a series of individual *data packets* that are sent across the network, which contains a piece of the original photo and a *header* containing metadata such as the web address of the intended recipient, such as the Instagram web server for uploads from users in North America, which is an Instagram-owned or -leased computer (*machine*) sitting somewhere, attached to the Internet) somewhere on the Internet, running particular *computer programs* or *software* that is sitting waiting for things like this photo upload, which it treats as a task to perform.

Each packet travels in loose ordering over the network, not held back waiting for those before it to be received; as each one is received, the *receiver*—which is a term for both the intended recipient machine and for the helper network *nodes* along the way that have any processing capability at

all (some don't, such as *switches*)—checks the transmitted package for typical *transmission* errors that result in the *flipping* of a one bit to a zero bit or a zero to a one. If the message contained within the packet acknowledges the receipt of the packet by transmitting a single *ACK* 320-bit or 40-*byte* message packet back, addressed to the original sender. For TCP the final intended recipient at Instagram—the machine to which the packet headers are addressed—starts accumulating these messages and reordering them to be in perfect order. If any packets are missing, after a certain deadline it sends a retransmit message back over the network to the *sender*, which then retransmits the missing packet or all of the packets a second time.

These packets that constitute the image the user wants to post can travel along different paths, but generally, because *shortest-path routing* is the most typical form of *route* information that the *intermediary nodes* between the source and destination have, the packets travel the same path. By and large, networks are too large for every node to know how to reach every other node; they know the *IP addresses* of some useful nodes, and they forward messages along the best path toward its final destination while themselves blind to what happens to the packet beyond a certain *hop distance* that is as far as they can see of the larger network. The set of networks is organized by numbers that are assigned by a *domain-name service* (DNS) that maintains *tables* or lists of domain names that can be used by intermediary nodes to look up a destination.

The path from a user device to a server goes through several physical/functional/logical substructures of the Internet. The user sitting in an office or at home uses *wireless (wifi)* radio-based transmission medium, or an *Ethernet* cable that connects to a modem typically, or could be sitting at a machine on a local area network (LAN) in a private network at a company or institution. The home user modem is attached to a system owned by a phone or cable company that provides an *access network* for users—a network of machines and other hardware and physical or wifi *links* that gives their customers access to the larger web or the Internet. These often traverse an access network to an edge router that bridges between the outer periphery of the Internet, comprised of many access

networks, to another large set of interior, or *core networks*, to reach either a commercial or private network where machines such as the Instagram photo upload server sit. Increasingly, such servers sit at *data centers*, commercial sites that combine computational and storage and software access and fast network access to businesses such as Instagram that might not want to invest in their own physical infrastructure, but rather *lease* it from another company such as Amazon, IBM or Google providing *cloud hardware, applications and services*. Many of the core networks use fast, *high bandwidth* devices for their network nodes, and fast, high-bandwidth physical links such as *direct optical links* for their network connections; the machines are often not general-purpose computers that one encounters day-to-day but rather specialized devices such as *routers* and *switches*. Routers contain some *processing* capability to read the headers and sometimes the contents of the packet *payload* and take a series of actions based on what is contained in parts of the packet, whereas switches contain no real processing capability, just the ability to learn which of the nodes to which they are connected to send packets of header type A, B, C, D, and so on.

Once the packets are reordered and assembled into the original transmission, the Instagram server lower-level software sends the full message up to an application layer, a higher layer of the *TCP/IP stack*, that knows what actions to perform in response to the photo uploade contained in the message. The message contains not only the photo but the user ID and the *password* or a *cookie* or other authentication information that both identifies the user and account and authorizes the photo to be posted on that user's behalf; the photo is then *posted* by giving it a universal record identifier *URI* that uniquely identifies it, or a universal record locator *URL* that uniquely identifies its current location on the web. The user sends this URL as a link to her friends via email, TXT, or social media site to invite them to look at the photo. When they click on the link, their device sends an *http GET* message that is transmitted as an http request through the Internet. This request travels in a very similar way to the photo described above, although it takes far fewer packets to represent the http request. A second Instagram machine with software running to act as a photo download

server receives the request and, in response, transmits the photo—again as a series of packets with headers, but now addressed to the device upon which the friend who requested to see the photo made the request. These traverse the network from the *autonomous system (AS)* on which the Instagram photo download server is sitting toward the network location of the receiver, typically a *host* sitting at a *leaf node* of the global telecommunications network, that is, on the outer periphery of the network, attached to the Internet by an *access router* or access network.

2.2 The Predominant Forms of Network Traffic

Examining the contents of network traffic over the global ICT, there are two independent observations of the traffic commonly reported:

> *The vast majority of transmissions are general web traffic and video, not voice.*
>
> *Most exchanges are point-to-point, or unicast.*

The predominant delivery method for network transmissions is *unicast* transmissions, or a point-to-point connection made from source to receiver [14]; unicast transmits the requested content is transmitted on a per-request basis from source to sink, addressed specifically to the receiver, and transmitted through the Internet core and a peripheral access network on behalf of an individual receiver serviced by the access network.

Arising likely out of both its popularity and also its much larger size than other media such as music or books or web pages or email, *video* traffic is already a huge percent of all traffic. In commercial cases Kilper estimates 86% of all traffic is video [15], and larger overall than smartphone data or voice, and expected to continue to increase exponentially, as is Internet traffic in general [7].

To provide a sense of the traffic volumes and mixture here, for example, in a study comparing recent historical trends across North American carriers in order to estimate both the mixture and

growth within service classes, based on the volumes of traffic of individual carriers reported for several years and projecting them forward based on trends, Kilper estimates the mixture for 2015 as 3.72 terabytes total within North America, of which, in decreasing significance, 3.19 is general web traffic, .324 is video, .152 is peer-to-peer, .0426 is mobile data, and .00327 is mobile voice. Looking further at traffic over access networks, Kilper finds that 2009 the majority of traffic is video since 2009; the highest rate of increase is in mobile data and video; and the highest volume is video traffic [7].

Chapter 3

Conservation of Energy through Efficient Content Distribution

Redundant data can be found on many levels of the ICT, not just within a single transmission, but also across transmissions. The larger sandbox for this work is *conservation of bits* within large networks. The goal of this work is not *data reduction* itself but the resultant *energy reduction* for large networks. As such, carefully considering the transmission energy of various data reduction schemes can be viewed as one *strategy* for achieving the goal of energy reduction. Within the large sandbox of *conservation of bits* within networks, one realm is conservation of bits by greater *sharing of bitstreams*. Examples include peer-to-peer, multicast, data replication within a CDN, and other forms of reducing not the redundancy of information within a bitstream but the number of redundant copies of the bitstream floating through the network throughout the day, or week, as well as the number of hops those copies traverse.

The research within this chapter aims at energy efficiency through how content is stored and replicated within the network. How to save energy across the entire set of activity is a complicated question; this chapter examines one aspect of the collective behavior of the system; namely, the pool of energy that arises from many individuals using the network to download the same thing—some significant fraction of a reported 500 billion video downloads per day by Facebook users

alone [16], a perfect example of which can generally be thought of by the reader based on today's big story shared and commented-on by many.

Duplication of a piece of content to multiple sites throughout the Internet is currently done for several reasons. It allows better network performance in handling multiple simultaneous requests for the material by spreading out the traffic to better fit the bandwidth of the cyberinfrastructure involved; it lessens the impact of failure in one particular hosting site; and the set of locations of hosting sites can be optimized to minimize the average path length of all requests for commonly requested material over an arbitrary length of time.

The shorter path from source to sink costs less energy as well: fewer hops from from source to destination means fewer nodes employed and less work done as a whole to satisfy the request. Energy-efficient content distribution over networks, as defined here, takes full advantage of redundancy among requests and transmissions to minimize all forms of duplication—duplicate transmissions as well as duplicate servers, caches, buffers, and other copies—to yield the highest overall energy efficiency. How much energy this could save is of particular interest in this chapter, as well as the dependence on traffic volume, traffic patterns, and network topology.

While private CDNs and in-network caching may be the contemporary solutions to delivering popular content efficiently—and a number of proposals are afloat to rearchitect the IP-oriented network either at a fundamental level or an application level to construct content-centric, or information-centric, networking approaches (CCNs or ICNs), as part of revisioning the Internet—we take a step back, here, and ask the simple question: how much energy is at stake in this, and what important qualifiers are there on that number–for instance, what characteristics does this energy have, and what key factors does the energy savings depend upon? The goal of this work is ultimately to trim the cost in power and infrastructure to supply content, based on informed knowledge about the specific network in question: its architecture, topology, physical infrastructure; its customers; and their behavior.

Exact numbers for the energy consumption of content distribution methods remain elusive, particularly for complex domains or autonomous systems within the Internet. This work examines energy-efficient distribution of static content within real networks. Prior efforts at estimating the energy efficiency of content distribution methods have not taken specific, real, large-scale router-level network topologies into sufficient consideration; topology's known effect on performance in other areas of network estimation raises interest in its effect on this question as well.

Questions answerable from the pursuit of this basic question include: what might the power budget be for designers of next-generation network architectures, routers and switches to directly address duplicate transmissions; what the numbers-to-beat are for new approaches to efficient static content delivery, such as ICNs; how different network topologies fare in their handling of duplicates; and where to place duplicates: whether it is better to store content at a few key nodes, or to distribute it throughout a network at a certain distance from receivers. This last question has certainly been asked before, but rarely from a whole-network perspective and comparing the effect at nodes. Further, most studies have used synthetically generated networks with statistical properties that fit statistical correlations in IP-level networks rather than for real, observed, large networks defined at the physical hardware- and physical link level, as it is approached here [17].

3.1 Background

Broadly defined, redundant data can be found within transmissions, across transmissions, and across nodes of the network. It is this third–and secondarily this second–sense that efficient content distribution addresses. The object of data replication is reducing either the number of transmissions over a group of receivers and senders, or reducing the average path length or maximum path length of any particular content request. Strategies such as multicast share transmission streams among receivers to avoid the replication that is associated with transmitting the same content at the same time to multiple separate receivers. Other strategies replicate content instead, placing content at

better positions within the network from which receivers can be served more conveniently. Often this is achieved by creating n distributed replicas of the data throughout the communications network and load-balancing multiple requests to the least occupied or closest of these servers. These replicas can be established within a content distribution network (CDN) by a third-party commercial data center or service provider such as Amazon Cloud Services; whereas peer to peer networks are a subscriber-based method of establishing a set of alternate content sites.

3.1.1 Content Distribution Methods

The concept of the rise of the *content-centric Internet* was addressed, and several existing content distribution methods—mainly CDNs and peer-to-peer networks (P2P)—in recent surveys by Passarella, and by Conti et al—who also include the practical realities in their assessment of where global telecom is headed [18, 19]. These models are briefly described here as well as some other proposals that fall into this same general vision of a content-address rather than the Internet Protocol (IP) IP-address-centered worldwide retrieval process.

Content Distribution Networks

CDNs explicitly replicate content at multiple distinct servers to reduce network congestion associated with delivery and receipt of the content—typically popular content, such as movies, books, music, and other media provided by a particular publisher or distributor. As such, they are perhaps a more network topology-aware approach to the general strategy of server replication, beyond prior strategies such as having several download sites geographically distributed as is common for software bundle downloads, or using multiple servers at a data center to service a large number of simultaneous requests. As much as 60% of Internet traffic goes through CDNs or through Content Providers networks (CPs) from commercial companies such as Netflix, Akamai, and Google, much of it video traffic from sites such as Facebook and YouTube [20]. CDNs tend to be privately

owned and managed for the sake of specific content providers, rather than the more comprehensive and general approaches studied within academia.

A strategy for redistributing traffic within a network to avoid the hot spot produced by content sitting at a single server might put replicas of the content on a set of network *edge servers* that are fewer hops from a population of user devices than the original source, such as a server sitting central within the network. Requests for content are redirected to or intercepted by these proxies for the original server. Fewer hops translates to lower latency for user downloads of movies and other media.

Fewer hops could also translate to *lower energy* for user downloads of movies and other media. Although the original motivation for such replication has been load balancing for overall throughput and also performance from the user's perspective—to improve leaf-node device receipt of content achieving lower latency, higher transfer rate, lower congestion or contention—these motivations have had little to do with energy efficiency. Yet energy efficiency may also be well-served by content replication. Here, content replication at multiple sites is considered in terms of its effects on global energy use within the ICT and telecommunications network.

Here the term *content replication* will be used rather than ambiguities that may arise from the term *data replication* in that the replication referred to here is not replication of more general information flowing through the network, nor at the level of packet payloads or messages: it is content from the user's perspective, measured in movies, books, articles and television broadcasts.

A newer method that appears promising to bring about energy conservation on a global scale is caching *content*, within *information-centric networks*, an early example of which is content-centric networks (CCNs). Similar to the caching associated with processors accessing memory, the first request for the content is expensive, going to the original source for the content, but near-term future requests for the content are cheaply fulfilled by a local copy.

In-network caching is a well-developed area of research, starting with web caching over a decade ago, and having a rennaissance the past six years as part of proposals for the Future Internet based on CCNs; however, the goals have been to accelerate response times and reduce network congestion rather than to save energy. Much more recent work on content-centric caching strategies aims specifically at the energy savings potential and at arriving at cache locations with the minimal energy [21–25]. Li et al [22] empirically compare average number of hops per transmission and per-bit energy consumption within simulated network topologies for a novel popularity-based scheme to one in which all routers cache all content they encounter and find a 70% energy savings when an aged content popularity metric is used. Choi et al [21] use several optimization techniques to arrive at energy efficient global placement of cache contents and find improvement over CDNs for small registries and very popular content. Llorca et al [24] compare global omniscient to online local optimization schemes to tune caching strategies to energy efficiency of devices used, capacity of channels, global benefit expected, and popularity of the content. Imai et al consider both transmission cost and power consumption due to specific device caching in their optimization study [23].

The majority of these studies focus on methods for determining the ideal location for caches or the ideal cache size and contents; here the object is instead aimed at contextualizing such work by much broader analysis focusing on the relative gains of CCNs' in-network router-based caching to the set of other contemporary candidates such as CDNs, multicast, unicast, and P2P. One possible outcome of such work is recommendations as to which approaches appear most promising for further improvement and deployment; and even further, comparing the relative gains of the best of these to gains from lowering information entropy.

Information-Centric or Content-Centric Network Designs

Another proposal afloat is to overhaul the current IP-address-based Internet and replace it with a fundamentally different design. One of the chief energy benefits of the proposals for CCN or content-centric networking is the prospect of caching contents closer to recipients. This arises within CCN because requests are satisfiable by any publisher of the same information, or by information recently cached at intermediate routers, allowing them to respond to requests directly. In the traditional IP model used in unicast and multicast, the message content is hidden from intermediary nodes and the data request is associated with the IP addresses of the sender and receiver. If the topic of the conversation is not known by the intermediaries, as is the case with traditional IP routing, then intermediaries' ability to substitute equivalent information is exceedingly limited.

The IP model has some similarities with an old model of communication by letters—such as the ones described two hundred years ago, in accounts of monarchs and nobility. In this ancient model, the meaningful information exchange is a private communication, the content of which is completely hidden from outside observation. The intermediaries are simply messengers or deliverers, traveling one leg of the journey from sender to receiver by ship or horseback, conveying a physical, written message bound up and sealed with wax imprinted by the sender's recognizable mark, to be broken open only by the intended recipient; interception of the message or counterfeit messages were fairly easily detected by tampering with the seal, or inauthenticity of the seal. Senders secured authenticity by securing the physical marking device with which they imprinted the hot wax seal.

In stark contrast, in the CCN model, rather than a message, there is a request for information made by the sender. *There is no recipient of the sender's request*: there is the request itself, put out into the network as a request for information that is to be satisfied by any node that can satisfy it. It is hashed or otherwise converted to a unique identifier, and that identifier is sent out through a

typically hierarchical network structure in which successive rungs have access to a larger scope of identifier locations, within which the content is found. If for instance the CCN architecture utilizes distributed hash tables utilizing an XOR-based progressive lookup scheme such as the one devised by Maymounkov and Mazieres [26], then each link in the path is traversed based on the XOR of the next field within the address, and all traversals make progress toward the content. The inequivalence of these two models suggests that both are necessary within global telecommunications, and indeed CCNs have been put forward to replace many of the functions of the Internet rather than of the larger telecommunications network. CCNs are not the ideal structure for private, creative exchange between two communicators; rather, they provide a universal library or retrieval scheme for both archived and new information in which *publishers* produce information, and *subscribers* consume it.

Examining the modern pool of communications, while *conversation* among individuals is common and widely supported within the digital framework—whether in the form of voice conversations, or text, or video—the modern digital communications system also handles a huge volume of economic activity, either trading directly in ephemeral or lasting information, or by the attendant traffic of trading: from the functions of recordkeeping, or by other monitoring functions. Records, whether ephemeral such as video feeds or permanent such as birth records, fit into the library, indexed model. Even physical electronic devices can be catalogued by unique IDs and referred to by ID within the burgeoning *Internet of Things* (IoT). Thus much of the network traffic could be handled within a CCN model of addressed content.

In the present work, CCNs are of interest because they can contribute directly to one of the identified problems of interest by providing a means to reduce the number of messages transmitted over the global telecommunications network via sharing across common requests. Roughly how much energy could be saved by CCNs' ability to recognize and dispatch *identical* requests from different requesters? Of specific interest is the energy savings to be had from avoiding duplication; that is,

from designing in strategies for cost savings over the volume of traffic generated by a single piece of content. Also of interest here is whether there are any mechanisms built into existing CCNs or attachable to their designs that would consider these costs: cache replacement strategy does so only indirectly.

To date, this question has not been directly answered by groups developing CCNs; in particular, the balance between adding cached copies or additional publishers to the network to reduce the average number of hops traversed by data requests has not been analyzed for the net energy cost for each specific proposal. In this area Guan compares CDN networks to a CCN approach, as well as to optical bypass routing, and finds that CCN is more energy efficient for popular content, and optical bypass routing for unpopular content, over traditional CDNs. Their approach was to construct and analytical model over a synthesized intra-AS network consisting of 24 nodes, whereas the present work targets synthesized traffic over real, observed inter-network router-level topologies for nine ASes within the Internet on order of several hundred to several thousand nodes [25].

Unlike CDNs and P2P, CCNs are not yet generally implemented within networks. The past five years have seen a number of large, multi-institution government-funded projects such as SAIL (€20M) to define what is being called the *Future Internet*, many within the European Seventh Framework Program, such as SAIL/NetInf, CONVERGENCE, COMET, and PURSUIT, and others within the US Future Internet Architecture Program and prior domestic funding, such as MobilityFirst, NDN, and DONA. The similarities and differences among these models were recently summarized by Xylomenos [27]. There are fundamental differences between the candidate architectures and models, and there is as yet no single, predominant model and no recognized standard. Nor is it a given that the Future Internet will be based in part or entirely upon a global content and device addressing scheme, as recently analyzed from an economic standpoint by Zhang [28].

Data Replication within P2P

Replication of content at multiple sites within a P2P network has been used to maintain performance as the number of users increases. Indeed, one of the benefits touted of P2P networks is that they scale by adding content hosts as well as clients. Data replication for P2P *with the goal of reducing global energy consumption* has not been widely studied for the Internet or global telecommunications.

Content popularity plays a role in most all of the policies and optimizations studied, optimizing the common case. A nice summary of work to date on content replication for P2P was done recently by Wu and Lui [29]. They state that, for general content over P2P networks, the main criteria for replication policy refinement has been file availability within the network as seen by P2P clients. In the specific case of video-on-demand content, Wu and Lui point out that past work has focused on minimizing the server's workload; their contribution was to explicitly support video quality (QoS) in their policy [29]. Tewari and Kleinrock [30] also examined video quality for bitTorrent support of live video to 100K peers in terms of peer group size and content fragment size. Recent work by Tan et al investigated optimizing for maximizing the peer's uplink bandwidth utilization [31].

Earlier work to find good content replication patterns focused on synthetic networks and non-DHT search strategies. For synthetic networks using blind search strategies, Cohen et al examined ideal replication locations. Tewari and Kleinrock also examined replication strategies for two search patterns typical of early P2P networks, flooding and random walk and characterized the average search distance for these as a function of the frequency of file replication for synthetic networks [32]; however, these results are intimately tied to the search algorithm and thus do not hold for DHT-based search.

The present focus is on typical deployed P2P network protocols such as bitTorrent. BitTorrent has the location of content indexed by distributed hash tables (DHT), often by schemes derived from

the efficient XOR-based progressive lookup scheme of Maymounkov and Mazieres [26]. Typical traversal times for such DHTs to retrieve a particular item can be $O(log(n))$ in the number of peers, although there is flexibility based on the goals and algorithm. DHT-indexed P2P networks have been a subject of intense study for at least a decade. The goal here is to retain the short tree traversal of Maymounkov and Mazieres within a larger policy of replication that seeks lowest energy for the ensemble of content and peer groups across the global set of P2P peer networks. Minimizing server workload or maximizing QoS is not likely to also produce the lowest energy solution across the global ICT: more likely it creates extra copies of content within the P2P network, the hosting of which requires storage and energy.

3.1.2 The Structure of the Internet

The typicalness, or representativeness, and scale of networks used in the current study is for several reasons an important consideration that directly affects the broad applicability of this work, dilating or shrinking the pool of circumstances to which the conclusions drawn here can be usefully applied.

Direct experimentation and data-gathering of the relevant information within the Internet is arguably the best source of data for this, particularly since it permits direct observation of the phenomena of interest without interference. Yet the direct approach is thwarted from many fronts; such an effort is largely procluded by the protections in place to secure systems from network attacks, protect the privacy of users, and preserve corporate proprietary knowledge from the prying eyes of competitors. As such the Internet and the global ICT represent the agglomeration of many separate networks, many of which are hidden from view.

To partially surmount this research difficulty, researchers occasionally obtain detailed usage datasets from commerical network providers under nondisclosure agreements; a more common approach to the problem, however, is to set up small laboratory research networks, or to work with synthesized networks derived from models of the Internet. It is impossibly expensive to replicate the complex-

ity and structure of the Internet in these research networks or models; they are, necessarily, simple subsets of the full Internet. For these to be of high utility in predicting the applicability of the results obtained when deployed within real networks out in the field, they must accurately embody the important qualities of the real networks, while somehow representing large networks such as the Internet with orders of magnitude fewer nodes and edges.

The physical and logical topologies of the Internet have been studied and conjectured extensively by research groups interested in accurately modeling the properties of the Internet for research and development purposes. Groups have produced a number of popular contradictory and sometimes hotly contested theories as to the underlying structure of the Internet. The differences are based mainly on the research methods of the researchers involved. The two main camps that have been produced over time are the proponents of cost models and the proponents of emergent statistics.

Structural/Cost Models

The first widely used network graph generation method was based on the multipoint routing model by Waxman [33] that specifies how to build up a series of connections between a set of nodes based on optimizing against known costs to add each possible edge. Waxman graphs are often referred to as random graphs, in reference to the Erdös-Rényi formulation for probabilistic edge-setting within sets of vertices to form *random graphs* [34]; however, the edge-setting function in Waxman's model is not probabilistic but rather a search-based cost function, successively choosing paths that connect two nodes such that path minimizes cost of communicating between the two nodes while not violating bandwidth constraints for each edge. Waxman based his approach on prior work on Steiner trees within graphs by Gilbert and Pollack [35], shown by Karp to be an NP-complete problem of minimizing a cost function over a set of vertices [36].

The nonconformity of Waxman networks to the structural patterns of modern engineered computer *internetworks*—simple enough at the time that they could be drawn by hand by network

managers—triggered a follow-on class of topology generators a decade later by Zegura and Calvert [37], and Doar [38]. Their goals included ensuring that the network reflected the larger structure or superstructure of existing router network designs; guaranteeing the connectedness of the resultant network; and better reflecting the real policies for adding new physical links in computer internetworks, which included adding links for redundancy as well as to connect new nodes, and producing a connected network. Their three-tiered hierarchical models worked by partitioning the space into transit domains and building up randomly generated subgraphs around each transit domain center, attached by another type of specialized node that serves a local area network (LAN) [39]. A popular network topology generator produced with these types of models is Thomas and Zegura's Georgia Tech Internetwork Topology Models (GT-ITM) simulator [40, 41].

Nearly a decade later, a new class of cost-based network models was developed based on a general cost-optimization heuristic for constructing networks that satisfy certain constraints by Carlson and Doyle [42]. This HOT, or heuristically optimal technique, was first applied to the construction of router networks by Fabrikant [43]. The application to data networks was developed in more detail by Li [44] utilizing cost constraints developed along the lines of earlier work by Gendron [45] and Grötschel [46]. Cost models included such objectives as maximizing the throughput of the network for a given traffic demand, and solving for the traffic that goes through each router, while minimizing the dollar cost of links, considering customer connectivity, and honoring the contractual level of service guaranteed, while maximizing the throughput of the network for a given traffic demand, with each router constrained by a range of realistic values for each router's traffic capacity and node degree, or its number of links with other routers [47]. Alderson presented a step-by-step approach to modeling the Internet using HOT techniques [48]. A significant component of the process is initially locating the access nodes geographically before running the constraint-based optimization with specific cost models. The goal is not to find the optimal network but a good design as judged by heuristic measures, expectations and past experience.

Follow-on work by Bowden resulted in the COLD data network topology generator that combines cost optimization for a PoP-network level similar to HOT with a method of generating a router-level network using a range of patterned designs based on Parsonage [49, 50]. The objective function for the optimization problem chooses the minimum-cost graph that has sufficient capacity to carry the traffic assumed. The cost optimization can be controlled by the setting of three parameters to vary the average degree of nodes and the coefficient of variation of node degree, the network diameter, and the clustering coefficient. Different settings for these favor spanning trees, minimum spanning trees, cliques, or hub-and-spoke networks. The resulting network is not the optimal solution for the cost optimization but rather the result of a genetic algorithm that randomly searches for improvements to the original seed set of networks, which are provided at the PoP level based on real examples collected by Knight [51]. The original PoP graph is defined over a two dimensional rectangular region of space.

Statistical Graph Properties-based Models

The second camp contains a variety of theories about Internet structure and a variety of means of generating networks with similar properties to those of the Internet; however, these form a collection in the sense that they use similar data to form conclusions about the structure of the Internet. In all cases, they base their models on statistical fit to network traces. These statistical properties of graphs are formed via tracing actual pathways through the Internet from various vantage points throughout the network using the network diagnostic tool $\boldsymbol{traceroute}$. This plies the router-level connectivity of devices running IP, giving not a graph of the layout of the entire Internet, but a spanning graph from the source node from which the trace was launched. The idea is to repeat this process at a sufficient number of vantage points within the Internet to encounter all routers. In practice, this is unachievable, due in main to the inaccessible regions of the Internet protected behind firewalls.

The pioneering study that launched the statistical camp was performed by Pansiot and Grad, who also were the first to note various limitations of the approach that were later come to be known as the *IP aliasing problem*. [52]. Their idea was to obtain information about the shape and variation of multicast trees supported within the Internet in terms of route length and node degree so that they could "calibrate" graph generators used to simulate and validate network multicast protocols, and their interest was in sparse multicast groups of a thousand receivers for which the average distance between receivers is high. They conducted their study by modifying Jacobson's well-known ***traceroute*** tool in several ways that improved the performance sufficiently to make a large-scale experiment querying many hosts within the Internet practical.

They then worked to clean up the results to form a router graph, eradicating duplicates of particular physical nodes that arose from unknowns within the ***traceroute*** results. The problems include cases in which some hosts have multiple IP addresses, and others in which multiple hosts share a single IP address. For all IP addresses they derived a method of determining in some cases if name and address correspond with the same host, or if two different addresses represented two different hosts or two interfaces on the same host, or if two different names pointed to the same host. To get around these address-resolution difficulties they devised a partial solution that they do not claim captures all instances; nor were these entirely satisfactorily resolved by other researchers who later took different approaches [53]. Similar to ***traceroute***, their method sends a UDP packet with an unused port number to all IP addresses they obtained; because the return *ICMP Port Unreachable* message generally fills in the source address with the address of the emitting interface, rather than the arrival address of the interface where the original packet arrived, this can be used to resolve multiple addresses by checking whether the ICMP source address is different from destination address of the UDP packet. They also note oversampling of edges in this ***traceroute***-based sampling method: that distinct edges produced by this do not necessarily correspond to distinct media links, giving the example for a broadcast network such as Fiber Distributed Data Interface

(FDDI) of three edges connecting three routers that actually correspond to the same link. They also eliminated from their sample the approximately 10% of hosts (network leaf nodes or intermediate nodes such as routers) that returned with anonymous IP addresses for which such address resolution is not possible.

They performed their extended $\boldsymbol{traceroute}$ protocol to capture hosts and routes using as a starting sample IP addresses in their campus network accounting database in the summer of 1995. They reasoned that these could plausibly form a multicast group, being from hosts that previously communicated with their network. From this they gathered around 30,000 traces. Using twelve randomly chosen hosts as source nodes, they pulled out 15,322 routes that originated from these sources, towards 1,270 destinations, with a maximum route length of 35 hops; path length to sources varied widely within the pool of networks they examined. They were careful to note that these routes do not necessarily capture all routes between the nodes. This collection of routes formed a graph of 3,888 host nodes and 4,857 links. They reported a graph diameter of 31, radius of 16, and reported a maximum node degree of 35, and an average node degree of 2.5, while the average node degree of central nodes in the graph was 4. Later work by Achlioptas critiques the sampling bias from the limited purview of $\boldsymbol{traceroute}$-based sampling such as this, in which some routes are oversampled and many undersampled, within the underlying system of many orders of magnitude more routes [54]. Ways of sufficiently broadening and democratizing the pool of vantage points have been attempted by large distributed efforts such as DIMES that take the SETI approach of asking volunteers worldwide to run and report their $\boldsymbol{traceroute}$ information [55]. It is, however, chronically hard to validate the datasets obtained because of the partially obscured nature of the Internet.

Looking further at node degree, and using the full traceroute results of 30,000 traces, they found two nodes of high degree: 45 and 37. Investigating the networks involved, they report that in both cases, the high degree is a result of the underlying technology and represents abstract links

in a sense: non-physical links within networks that use IP over a switched circuit technology, and they note that, with such technologies, there is no limit on the degree of a node even if the number of physical interfaces is limited. All routers connected by such a fabric appear as direct neighbors in the capture method used in their study which uses IP addresses to ascertain routing. This effect, from Multiprotocol Label Switching (MPLS) tunnels, appeared in 30% of routes, in a $traceroute$ study by Donnet [56].

As a result of these complications, of their graph Pansiot and Grad make the qualifying statement that the links are IP links that may or may not represent a direct physical link between two graph nodes. Within the global ICT some are physical, point to point links wiring the two nodes together; others are links within a broadcast network such as an Fddi or Ethernet LAN or wireless radio broadcast; others multiplex the underlying physical technology, such as a non broadcast multi-access (NBMA) network and use connection-oriented schemes, including connection-oriented protocols such as MPLS for packets or Asynchronous Transfer Mode (ATM) or X.25 or Frame Relay for frames, or circuit-switched networks. MPLS, ATM, and Frame Relay are designed to abstract the network's router- and link topology from IP beyond knowledge of the edge routers of the network, providing what appears at the IP level to be direct connectivity between n pairs of physically distant routers—which also gives the edge router the appearance of a node with very high degree. The term *IP graph* is used in the ensuing discussion below to describe the graph resulting from their method because of the imprecise mapping from IP address to physical device that they enumerate; in contrast, the term *router graph* is used in the ensuing discussion to denote the actual physical reality that remains partiallyunknown or obscured from observation.

A widely cited study by Faloutsos [57] generalized the results of Pansiot and Grad, incorporating them into a general theory of the structure of the Internet that included other logical or physical Internet structure as well, showing a remarkable similarity across the graphs of various aspects of the Internet that sparked great interest, as well as a new class of network generators that reflect the

theory. As far as the theory's applicability to the router-level graph of the Internet is concerned, Faloutsos' analysis was based entirely on the IP graph of Pansiot and Grad [57].

Faloutsos provided a new way to characterize the topology of the Internet, through the fit of power laws to graph metrics such as the number of nodes and edges and the average size of neighborhoods within h hops of a node. A power law curve of the form $y = x^d$ defines a straight line in log-log space; Faloutsos fit power law curves to various graphs associated with the Internet, including the graph of separate autonomous systems or AS graph, the worldwide web graph formed by hyperlinked web pages, or WWW graph, and the IP graph of Pansiot and Grad. In all four datasets they examined, graphs of different aspects of the Internet, they found that the *distribution of node degree* exhibits a 96% or higher fit to a log-log plot of node degree versus node frequency of appearance/occurrence within the graph, or *rank*. More precisely, Faloutsos states: first, that the outdegree d of node v is proportional to its rank r raised to the power of a constant c_R, or $d_v \propto r_v^{c_R}$, and second, that the frequency f of nodes with outdegree d is proportional to the outdegree raised to the power of a constant c_O, or $f_d \propto d^{c_O}$.

Two other aspects of the graphs they examined also had a good fit to a power law: the total number of pairs of nodes within h hops followed a power law where the number of hops is small relative to the diameter of the graph, within which the number of pairs within the graph G at hopcount h is proportional to h raised to some constant, or $|P(G_h)| \propto h^{c_H}$. They found also that, for each of the four graphs they inspected, the eigenvalues of the adjacency matrix were proportional to the graph order, or number of vertices $|G|$, raised to an exponent.

Ensuing work on network topology generators created networks that replicate the power law relationships described by Faloutsos [58]. Some examples of these are: 1) the BRITE topology generator by Medina, which also includes elements of the earlier structural models like GT-ITM [59, 60]; 2) a method of generating graphs that is parameterized by logsize and log-log growth rate, by Aiello [61]; 3) two algorithms by Palmer and Steffan that generate power law networks, one using a

power law to guide graph construction, the other based on a recursive probability distribution [62]. In modeling the Internet AS and router graphs, Yook proposed replacing randomly generated small subgraphs with fractal-generated small subgraphs as part of a larger movement toward preferential attachment methods by Barabási and Albert [63, 64].

Work also continued on developing the $\boldsymbol{traceroute}$ approach to discovering network topology, by Burch and Cheswick [65], Govindan and Tangmunarunkit [66], Spring [53], and Claffy [67,68], toward improving the resolution of IP aliases for a single router, and by Burch and Cheswick, Spring, and Claffy on enlarging the view of the Internet seen by their experiments. whereas Pansiot and Grad applied it initially to the much smaller problem domain of multicast groups from a single node. Spring applied $\boldsymbol{traceroute}$ to detect the structure of an entire ISP for several U.S.A. Internet service providers in a tool called Rocketfuel. The connectivity and underlying structure of the Internet has been probed in a many-year study by Shavitt and Shir [55], using a similar citizen science structure of volunteered computing resources to the one pioneered by Anderson et al with the SETI project [69]. Dolev and Shavitt [70], probing Layer 2 or the IP layer of the Internet for its support for multicast, confirm prior work as well as concurrent work showing a power law distribution of node degree within the network [57, 68] and a Gamma distribution of node degree and distance from various low-degree sender nodes to leaf nodes. Roughan and Willinger point out many of the shortcomings even now to these methods stemming from the fundamental uncertainty of the IP aliasing problem from ambiguities in $\boldsymbol{traceroute}$ and sometimes intentional network operator obfuscation of such network structures as MPLS tunnels and clouds [71]. Donnet finds that 30% of traceroutes go through obscure clouds such as MPLS or ATM circuit technologies [56]; Augustin discusses studies that report that between 35-95% of peer-to-peer AS connections are not discovered [72], Roughan and Willinger and Li trace the historical and modern problems with alias resolution [44, 71].

The networks used for this work are from a new method developed by Pansiot and colleagues

that is not so dependent on ***traceroute***. It employs ***traceroute***-based methods to establish certain otherwise undiscoverable connections within a network, but to extract the topology within a network domain it employs a different method of querying nodes that is based on sending out ***mrinfo*** requests; it takes advantage of the fact that responses to IGMP messages report the list of multicast conversation partners for the node. This is convenient for producing a network that more accurately represents the physical, router-level connectivity, according to Pansiot, who reported on a study of its confidence levels in a paper that first described the technique [73].

The technique gives relatively high confidence for those nodes that respond to ***mrinfo*** requests; this, however, does not include nodes that employ IGMP filtering or that do not have multicast enabled [73, 74]. Mérindol, Marchetta and other members of the research team moderated these effects, enabling much larger network topologies to be discovered, by employing ***traceroute*** to fill in missing connections within the physical-router-level structure exposed by ***mrinfo*** campaigns [75, 76]. The project, named Merlin, ran a path-traversing form of ***mrinfo*** to trace at each stage the neighbors of the prior edge of discovery, improved by a second-stage employing Paris Traceroute and Ally, which were used to reach disconnected subnets where IGMP filtering prevented routers from responding to ***mrinfo***. The Merlin project used six separate launch sites—including one hosted by CAIDA—at once, as vantage points from which to expand the discovery when querying the intra-domain topology of a specific network. The network topologies produced by this effort are publicly available [77]. These give the physical-network definition needed for the present work, which follows bits through routers and switches, and cannot make use of the more abstract IP-level network topologies or the AS-connection networks produced by many of the other methods developed to date.

3.2 Related Work

Caching and web-caching have been examined before, and in ways that compare and contrast approaches, and the results have in some cases been reported in terms of energy efficiency. Baliga and Guan examined global-scale content placement relative to the users of the content [25, 78]. The tradeoff between transmission energy and storage energy in placing content in a heterogeneous network of data centers, architectures, and physical link options has many boundary conditions that need to be mutually optimized, several of which are enumerated by Kilper [15]. The tradeoff has been examined for CCNs by Lee, Guan, and Hasegawa; for CDNs by Baliga, Guan, and Osman; and for video-on-demand for Internet Protocol Television by Jayasundara [25, 79–84].

Spring and Weatherall compared object-level caching to another technique, inter-packet redundancy elimination, and found that inter-packet redundancy elimination was more effective; however, the applicability of this result is less relevant as time proceeds. Spring and Weatherall's results were from a traffic log of incoming and outgoing traffic from a single private enterprise network. This was also fifteen years ago, when offices were far less likely to be distributed, and work was much more often done with largely local resources, rather than utilizing both data and computing resources scattered across the global ICT; the bandwidth and reliance on the ICT, and many network use patterns were markedly different fifteen years ago from those of today [85]. Contemporary approaches to inter-packet redundancy elmination, for redundancy found within a large dataset of network traffic from twelve sites, including one of the networks examined in this Chapter, was assessed by Anand, who found that the ideal placement of redundancy elimination was the client hosts generating the traffic, at the network periphery [86].

The primary focuses of this work are first, the assessment method used, in contrast with integer-linear, mixed-linear, and integer-nonlinear global optimization methods and analytic models of previous work by Llorca, Araujo, Jayasundara, Mandal, Guan, Osman, Bektaş and Modrzejew-

ski [24, 25, 83, 84, 87–90]; second, the amenability of the method to either abstract models or specific traffic from real networks; and third, the use of specific observed networks that are reasonably complex, and that span the structures at different functional levels of the Internet; fourth, using methods that can be composed to conduct a similar assessment but at a larger scope, such as traffic between multiple networks.

3.3 Methods & Assumptions

3.3.1 Energy Assessment

The traffic assessment algorithm uses full global knowledge about the complete communication pattern, across nodes, and across time. It has a complete history of transmissions that occurred over the network, and it can parse or peruse this history as many times as necessary to compute the set of refinements that will give the global minimum energy usage. Only requests for the single item of interest are considered, not any unrelated and potentially competing network traffic.

The traffic is presented as a time-ordered series of complete, end-to-end transmissions from content source to the receiver node. Any earlier transmission of the static content item is permitted to substitute for a later transmission within the series. Thus, the effect of the assessment algorithm is to trim from transmissions the upstream portion above a particular node, proposing instead to source the content from this intermediate node.

Figure 1 shows a very simple example of the concept in action with the colored nodes being ones at which traffic redundancy has been eliminated by re-sourcing the content. Which node is chosen as the trim-node is, at each step, the one that produces the most energy savings. An example is a node that appears in many transmission paths: greater efficiency may be achieved by seeding the store at the node with the initial transmission, and then satisfying later requests for the item

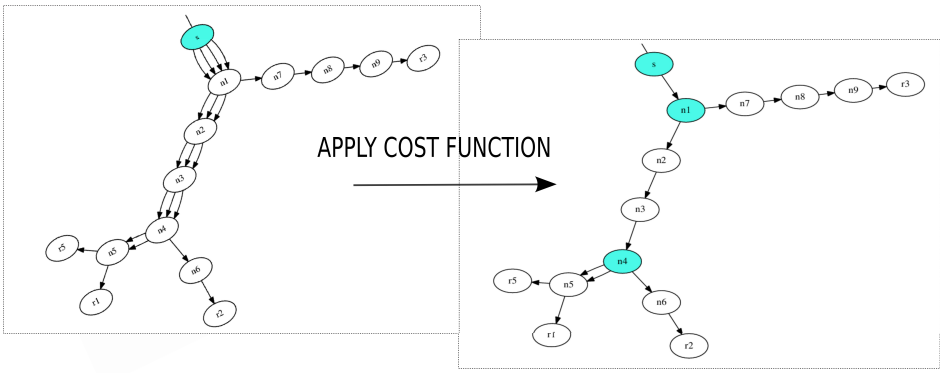


Figure 1: Illustration of the Redundancy Assessment. Elimination of the most costly forms of redundancy in network traffic. Example communication graph showing transmissions as edges and routers and switches as nodes. Graph is refined with a particular cost function for transmission energy versus storage energy $c = f(t, s)$.

from the locally cached copy. The best-energy node is chosen at each step, in an exhaustive but successively pruned search; at each stage, the transmission history entries are shortened.

< this is a snippet of the traffic from Figure 1>

s -> n1 -> n2 -> n3 -> n4 -> n5 -> r1

~~s -> n1 -> n2 -> n3 ->~~ n4 -> n5 -> r5

~~s -> n1 -> n2 -> n3 ->~~ n4 -> n6 -> r2

~~s ->~~ n1 -> n7 -> n8 -> n9 -> r3

< and so on >

A cutoff of .1% of the best node's energy savings was chosen; it is an arbitrary number well beyond what is likely to be energy-efficient, used to capture a lot of information about the network energy

distribution of redundancy[1]. In practice, a threshold based on the known cost of the store, cache, or other re-source method is a practical choice for termination. We emphasize that this is a study, using the benefit of hindsight, and of choosing such nodes *a posteriori*; it is the intent of this work to promote future protocols and other proposals that can provide accurate predictors of such nodes, with minimal impact on quality of service. This work shows what is achievable if optimal predictives protocols could be enacted.

As stated earlier, the refinement occurs with global knowledge of the traffic pattern over the network, over time; the goal being to extract and measure the ideal achievable benefit of efficient delivery. The traffic assessment method developed for this work can be modified based on assumptions as to the localness of the decisionmaking, or other factors, such as a lower traffic sampling rate.

The observed traffic used as a starting point is full, ordinary unicast transmissions under conditions where all requests are sent all the way to the original source and all the way to the individual requester: no caching occurs at any intermediate nodes in the initial traffic sample. The goal is to independently arrive at the ideal re-sourcing pattern rather than to observe and interact with a network's existing re-sourcing pattern.

Table 1: Sample Networks Used in Experiments. From P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J. Pansiot, *Topology discovery at the router level: a new hybrid tool targeting ISP networks,* Selected Areas in Communications, IEEE Journal on, vol. 29, no. 9, pp. 17761787, 2011. Used under fair use, 2015.

Tier 1 networks (AS, Name)		**Transit networks**		**Stub networks**	
1239	Sprint	680	DFN	109	Cisco System EU
2914	NTTC	1267	IUNet	137	GARR
3356	Level3	3292	TDC	224	Uninett
3549	Global Crossing			59	U.Wisc Madison

[1]A cutoff of .01% was used in initial runs but proved too time-consuming for large networks.

3.3.2 Networks

Traffic assessment is done here for several significant autonomous systems from various classes of networks, defined at the router level. A great deal of research has gone into the difficulties of accurately obtaining the structure of real networks, and assessing the larger structure of the Internet at the AS-level, as well as overcoming the shortcomings of existing methods. The majority of methods derive from pioneering work by Pansiot and Grad, who adapted Jacobson's `traceroute` to explore multicast trees within the Internet [52]. Work also continued on developing the ***traceroute*** approach to discovering network topology, by Burch and Cheswick [65], Govindan and Tangmunarunkit [66], Spring [53], and Claffy [67,68], toward improving the resolution of IP aliases for a single router, and by Burch and Cheswick [65], Spring [53], and Claffy [68] as well as by Shavitt and Shir [55] with the explicit goal of enlarging the view of the Internet seen through their experimental methods. These traceroute-based methods suffer an Achilles Heel in what is known as the IP-to-router aliasing problem that has never been completely resolved. Donnet finds that 30% of traceroutes go through obscure clouds such as MPLS or ATM circuit technologies [56]; Augustin discusses studies that report that between 35-95% of peer-to-peer AS connections are not discovered [72], Roughan and Willinger and Li trace the historical and modern problems with alias resolution [44, 71].

This difficulty recording network topology down at the physical level is an important problem to surmount, because prior work by a number of groups in performance assessment has suggested the dependence of performance on network topology, by Palmer and Steffan, Tangmunarunkit, and Lorenz, suggesting that network scale, structure and connectivity affect results [25, 58, 62, 91].

This work is grounded upon the necessity of hardware-level network definitions, so that actual energy of nodes and links can be assessed. This need is met by a set of networks obtained by Marchetta and Mérindol, using an alternate technique [75–77]. These are router-level topolo-

gies obtained by probing over the Internet using a combination of Paris `traceroute` tracing campaigns along with Pansiot's IGMP probing based on `mrinfo`, which gives a high level of confidence as to the exact hardware router-level topology [73].

We use networks from Marchetta and Mérindol that cover many important structural classes of networks present in the Internet; the set of networks is shown in Table 1. The networks span the three major classes of network function (Tier 1, Transit, and stub). These consist of up to 10,000 routers and switches, and many more links. In addition, a small campus network included in Marchetta and Mérindol's dataset was also examined. This campus network permitted exploring assessment instability from having a small sample of receivers. Autonomous System 1267 and 137 were eliminated early on from experimentation: their xml files did not generate a traffic log in a reasonable amount of time.

3.3.3 Traffic

The network traffic examined is comprised of the predominant form of transmission in the Internet, unicast messages traversing the shortest path from the original source to a receiver requesting the content. The traffic was generated using a simple model of receiver location distributions, rather than traffic collected over network observation. The collection of traffic used to examine the effect of variance in transmission ordering was generated as 100 random samples of traffic, in which the j^{th} element in the traffic log was randomly chosen from a pool of unicast transmissions from each sink/source pair. This does not model receivers' requests directly, because receivers are not explicitly represented in the network; it models and captures the path the requests take through this particular network, however.

The random samples of traffic were generated with 1 million requests. Most of the results are reported for the smallest tested subset of this random sample, which was 100,000 requests for the content item. The effects of both total traffic volume and specific traffic ordering on the network

characterization were investigated in tandem with these 100 1M-request random samples, by the construction of ten shorter samples from each of the 100 random samples of traffic. This has the intentional effect of preserving the ordering from the random pool across the experiments investigating the effect of the range of traffic volumes.

Exploring these variations leads to a large number of cases; for the sake of problem tractability, at most two content-source nodes within each network were chosen for examination. In almost all cases, the source node is chosen from those that had a short path to the network edge to reach the larger Internet, given the assumption that a randomly chosen content item is more likely to reside in the much larger realm outside the network than to come from within it. In cases where the networks were not simply connected, the source was chosen to involve the largest subnet. Thus, with the possible exception of stub networks for which a single externally connected node provides all Internet traffic to the network, it is important to qualify that these numbers do not represent a complete energy assessment of content delivery across the network, only for one source within the network. As such, the numbers should not be used to compare the networks with each other or to rank-order their efficiency.

Content traveling from a single original source point in the network was the object of the assessment, and its efficient distribution in the network the goal. The single source nodes [2] were chosen from those nodes in the network with fairly direct access to externally connected nodes with the idea that this network is but a small subgraph of the larger network; thus, more traffic comes from the outside than is generated inside the network. Of these, the ones that gave the largest spanning tree were chosen as source nodes [92]. For two networks where both these criteria were not well-met, two separate source nodes were tested. Table 2 lists the specific nodes within the Merlin dataset that were used for these experiments along with their designator or label in the Merlin network definition file.

[2]The reciprocal node is interchangeably designated as the sink, receiver, or requester node.

Information about the locations of the host nodes, or end receivers, for these networks is very limited. It is not explicitly present in these network definitions, which are limited to the core and edge; the access network and leaf nodes are extrinsic to the network, and are only implicit in the requests that enter these core networks through access networks. For networks that have a simple stub structure, the access network can be inferred; for more balanced networks such as transit networks, few assumptions can be made as to the locations of receivers. Traffic generated for these networks is biased in the following way toward certain presumed receiver locations: for stub networks, a minimum of 73% of traffic originates at leaf nodes; for Transit networks, a minimum of 33%; for Tier-1 networks, no bias; the network classifications come from both the original study and from the Center for Applied Internet Data Analysis (CAIDA) autonomous system classification system [93].

Traffic refinement operates only on the traffic associated with delivering a particular single content item from the single source to each receiver; it then optimizes this for preferable content locations within the network, without regard for competing traffic or traffic loads.

3.3.4 Content Item

The analysis is performed for various content item sizes. The results reported here are for the transmittal of a web-page-sized content item based on the disk size of the common elements of the Google mobile home page served at `www.google.com`, which is one of the pages very commonly accessed; and for a video clip equal in size and number of packets to the average of videos on YouTube ($4'12''$, 55 packets per frame on average, just under 30 frames per second [94]). The packet size is fixed at 1500 bytes, which is both the maximum transmission unit of IPV4 and the biggest non-ACK-packet hump in the bimodal distribution of packet sizes observed by Sinha, the CAIDA project, and Murray [95–97]. End-to-end encryption is ignored within this analysis.

Table 2: List of Source Nodes. Traffic source nodes used for experiments

Network	**Source Node Label**	**Size of Communication Graph (Nodes)**
AS 1239	213.200.68.182	3,913
AS 224	24	2,486
AS 3549	129.250.9.118	2,134
AS 59	216.56.60.174	1,834
AS 2914	64.208.110.82	1,725
AS 3356	87	728
AS 3356.1	213.206.131.45	126
AS 3292	217.17.71.145	250
AS 109	217.149.33.94	239
AS 680	1	158
AS 680.1	212.122.56.25	142

The overall volume of single-item-related traffic is a function of the number of users of the network as well as the item's popularity. For all but the smallest, campus network, energy harvestable from redundant requests was calculated for a variety of total number of requests, ranging from 100,000 requests to 1 million requests; for the campus network, where the number of users is small, and limited to the number of faculty, staff, and students, the maximum number of requests for a common item was capped at the reported popularity of the most popular website, or 33% of campus users.

3.3.5 Router and Switch Energy Profiles

Using the vendor-provided wall power ratings for routers and switches is insufficiently accurate or detailed for the question this work asks; this work, in fact, does not concern the total accumulated cost of running each piece of network hardware; instead, it looks at the incremental cost of whether to transmit a particular message over an already-running device[3]. We therefore use the carefully gathered energy information for routers and switches by Vishwanath [98]. While previous work by Mahadevan, Chabarek, Hlavacs, Kharitonov, Heddeghem and others has established the power consumption of routers under zero load to full loads [98, 99], only Vishwanath has broken these numbers down to the level necessary here, which is the incremental costs of individual packets flowing through the router or switch. This is needed to assess the transmission cost of content based on its size in bytes and packets, rather than based on the load on the router—which is a traffic-aggregate.

We consider the specifications as being applicable to the class of routers and switches utilized within the autonomous systems considered; this can not be disproven, as their method has not as of yet been exhaustively applied to profile a wide number of commercial router models. Yet the method described here can readily be reapplied to specific router information for the specific

[3]The present work assumes the devices are not energy-proportional devices, which is a reasonable assumption for the datasets used here.

network of interest when available. We consider these numbers as proxies for the class of either *high-end* routers and switches suitable for the core- and edge routers and switches within the majority of the networks tested, or *enterprise-level* routers and switches suitable for networks such as the campus stub network.

Table 3: Hardware Profiles. Hardware Incremental Energy Profiles Collected Experimentally by A. Vishwanath, K. Hinton, R. Ayre, and R. Tucker, *Modeling energy consumption in high-capacity routers and switches,* Selected Areas in Communications, IEEE Journal on, vol. 32, no. 8, pp. 15241532, 2014. Used under fair use, 2015.

Class, Model	**Packet Processing Cost (nJ)**	**Store and Forward Cost (nJ/byte)**	**Idle Power** (Watts)
Enterprise Ethernet Switch, Huawei S5300	40	.28	36.2
Metro IP Router, Huawei CX600-X3	1375	14.4	352
Edge Ethernet Switch, ALU 7450-ESS7	1571	9.4	631
Edge IP Router, ALU 7750-SR7	1707	10.2	576

At every stage, the upstream cost of transmitting the packets and bytes associated with the content item is assessed, for each upstream router along the path in each traffic log, across all nodes in all paths, to find the router at which storing or otherwise re-sourcing the content would result in the largest energy savings. The energy consumed by network connections, or physical link energy, is not included in the calculation. Baliga and Tucker found that transport per-bit energy is an order of magnitude lower than that of routers and switches, and, for most link media technologies, and in particular for point-to-point optical network connections, as these core networks typically have,

the power consumed by the physical links is rate-invariant, and independent of load over most of the range [100, 101]. Therefore we do not include link energy as a part of the assessment of incremental energy usage associated with individual packets.

3.3.6 Data Collected

Several forms of data were gathered from the experiments to assess the achievable energy savings from content request redundancy. The initial and final weight of the communication pattern within the traffic log or history was taken, in the number of hops, or edge traversals. The energy efficiency of the initial communication pattern is defined in proportion to the difference in the original and final weight. This measure does not include the introduced cost of re-sourcing the content at the trim-node, which must be accounted for once the re-sourcing method is decided; rather, this should be viewed as the power budget available at the node for designing and implementing a specific solution. The specific pool of energy in Joules due to the traffic trimmed to and from upstream nodes was explicity calculated for each trim-node chosen at each stage of traffic assessment and refinement. These are presented as the *energy pools* at each node. The variation in pool size was further investigated for each network, as well as the variation from network to network in this distribution of pool sizes, as was the distribution of pool sizes. In future work the distribution of pool locations will also be investigated; this, however, is beyond the scope of this chapter. It is also quite possible that the energy expenditure of some nodes will rise, in order to seek an overall global energy reduction, from extra processing and storage.

3.4 Experiments & Results: 75% Savings of Transmission Energy

Table 4 shows the total energy associated with duplicate transmission paths from traffic associated with 100,000 requests for the same item, for a web page such as the Google mobile home page.

These range from .9-17 KJ depending on the network, and from .75-14 J per content-packet–meaning per packet of the web page which is the subject of the assessment of the energy associated with redundant transmissions, ones that take portions of the same path that might be suppressed via content storage at intermediate nodes. The final column shows energy at a finer granularity—at what may more normally be thought of as per-packet energy, albeit here it is *savings*, not *cost*—by dividing Column 3 by the number of requests; this shows energy savings on a per-request-packet basis within the range of tens to hundreds of microJoules ($\mathbf{10^{-6}}$).

The variation can be attributed to differences in network topology rather than any other difference in the experimental setup, which was controlled for other differences, with the possible exception of the different biases in terms of inferred locations of leaf nodes discussed earlier. An important point is that the topology that is crucial to the traffic is not directly the topology of the AS, but, rather, the topology of the spanning tree graph of nodes from the source node to all reachable nodes within the network.

The networks are listed in order of size; in all of the ensuing discussion, the size referred to is not the number of routers and switches or physical links within the network but rather the size of the spanning tree from the content source node, termed the *communication graph*. The networks denoted with a suffix after their names in Table 4, $\mathbf{680.1}$ and $\mathbf{3356.1}$, are ones for which a second node was also investigated as the origin of the content, or source node. Both sources are shown, along with the savings associated with each. The choice of source node is important as well, as is shown by the effect on the results for network 680 and network 3356. The source node was chosen in almost all cases as the one that had the most connectivity within the network dataset, and, in the case of stub networks, the largest of the few available that are connected to an external AS. In both cases, however, this may be attributable again to differences in network topology, because the two source nodes were quite different from each other in terms of the spanning tree of the communication graph.

Table 5 shows the total energy associated with duplicate transmission paths from traffic associated with 100,00 requests for a larger item, in this case for a video of the average length of YouTube videos. Here the range of energy savings is from 313 KJ to 6 MJ.

The total energy savings in a small campus network such as AS 59, University of Wisconsin-Madison, is shown in both tables separately, arising not from 100,000 requests, but from 18,413 requests. A hundred thousand requests is beyond the size of the user pool for this network. The popularity of the most popular website, Google at 35%, was used to assign a maximum population size of 35% of campus users to the largest TR pool. This pool is reported in all results associated with AS 59 because, at 1,841 requests, the sample size is already fairly small, and any statistics may suffer from small sample size. We would expect a greater amount of variability in the results for such a small sample of traffic, and this is apparent in examining the average energy pool for each node across the 100 trials; it also may be the reason for the larger difference in per-packet energy for the two sizes of content in Table 6. Here again with AS 59, however, the standard deviation with packet ordering is quite small, smaller than we expected, over the 100 random trials; the redundancy estimation method appears robust even for small samples of receivers.

The standard deviation for all networks across separate trials using different traffic is quite small across the sample set of networks, as seen in Table 4 and 5. Further work will more completely investigate the effect of traffic ordering on the results; however, initial results with between 25 and 100 trials per network suggest that the assessment method is robust to the specific time-ordering of requests from different receivers.

The total energy cost for the video clip is nearly perfectly linear with that of the web page case, giving nearly identical per-packet savings; indeed, the per-packet column is given in the table largely to show this linearity: it is a way to normalize the results over the range of content sizes. Table 6

Table 4: Energy Divident from 100K Requests for a Web Page. Energy dividend from redundancy within various networks for 100,000 requests for a web page; showing the energy available due to redundancy for 1.8K requests for AS 59, a campus network

Network	**Redundancy (J)**	**Per-Packet [102] (J)**	**Per-Bit [103] (KJ)**	**STD (%)**	**Per-Packet Per-Request** [104] **(μJ)**
AS 1239	17,187.11	14.2988	76.3	0.11	142.99
AS 224	11,184.77	9.3051	49.6	0.07	93.05
AS 3549	15,311.79	12.7386	67.9	0.07	127.39
AS 2914	16,694.35	13.8888	74.1	0.08	138.89
AS 3356	13,648.76	11.3550	60.6	0.10	113.55
AS 3292	5,244.38	4.3630	23.3	0.06	43.63
AS 109	17,067.51	14.1993	75.7	0.10	141.99
AS 680	905.70	0.7535	4.0	0.53	7.53
AS 3356.1	16,527.21	13.7498	73.3	0.10	137.50
AS 680.1	3,712.25	3.0884	16.5	0.08	30.88
AS 59	207.01	0.1722	0.9	0.53	93.53

Table 5: Energy Dividend for a Video Clip. Energy dividend from redundancy within various networks for 100,000 requests for a $4'12''$ video clip; showing the energy available from redundancy for 1.8K requests for AS 59, a campus network

Network	**Redundancy (J)**	**Per-Packet (J)**	**Per-Bit (KJ)**	**STD (%)**
AS 1239	5,994,480.37	14.4167	0.0769	0.13
AS 224	3,915,609.12	9.4170	0.0502	0.29
AS 3549	5,306,897.88	12.7631	0.0681	0.08
AS 2914	5,788,006.21	13.9201	0.0742	0.10
AS 3356	4,725,086.45	11.3638	0.0606	0.11
AS 3292	1,815,585.52	4.3665	0.0233	0.07
AS 109	5,905,860.88	14.2036	0.0758	0.09
AS 680	312,918.63	0.7526	0.0040	0.60
AS 3356.1	5,718,781.96	13.7537	0.0734	0.16
AS 680.1	1,285,037.08	3.0905	0.0165	0.09
AS 59	957,943.41	2.3039	0.0123	0.15

Table 6: Linearity of Dividends with Content Bytelength. A comparison of the per-content-packet energy dividend of a web page versus a $4'12''$ video clip

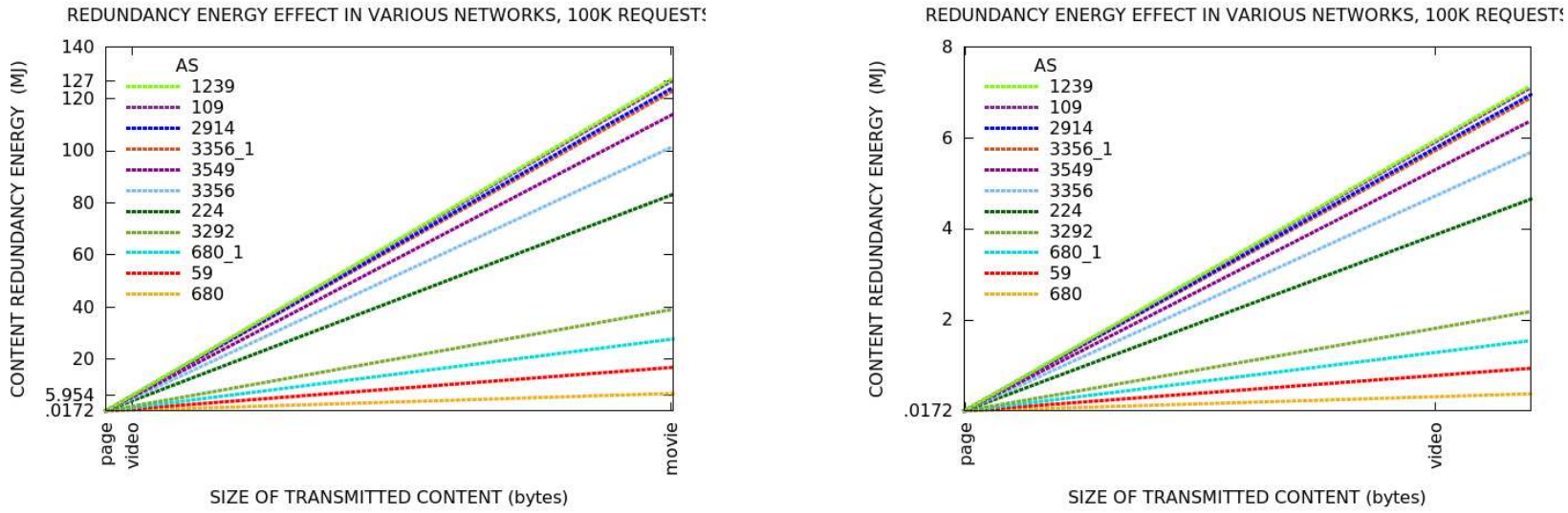
Network	**Energy Per-Packet Video (J)**	**Energy Per-Packet Web Page (J)**
AS 3549	12.7481	12.7386
AS 2914	13.8971	13.8888
AS 3356	11.3596	11.3550
AS 3292	4.3663	4.3630
AS 109	14.2036	14.1993
AS 680	0.7526	0.7535
AS 3356.1	13.7537	13.7498
AS 680.1	3.0905	3.0884
AS 59	1.8749	0.1722

shows the two side-by-side for comparison. Two networks, 224 and 1239, were omitted from this table due to variation in the termination conditions for these large networks that make their video- versus page- results differ slightly. The larger *amount* of energy from redundant transmissions in Table 5 for the video clip, vice Table 4, arises from the larger number of packets associated with the content, requiring more packet processing energy and more packet store-and-forward energy to transmit. Storing such content at intermediate nodes is also more expensive in terms of energy; here, the goal is to assess the energy pool both at specific nodes and within the network as a whole that could be viewed as an energy budget for a storage method. A variety of methods could be employed, from ICN overlays over the network, or caching at routers, or a peer-to-peer strategy within the network, or the provision of an in-network proxy server, for sufficiently large networks and sufficiently large content items; the goal here is to simply assess what makes sense in terms of the cost of transmission and the cost of storage of modern-day networks from a sample of typical network infrastructure in use today.

The per-packet energy[4] available network-wide (harvestable in various specific places) from the redundancy among 100,000 requests for the same content item from Table 4 is used in Figure 2 to graph the available energy from 100K-request-redundancy for a variety of content bytelengths. It ranges from 170 KJ or less for a web page to 127 MJ or less for a feature-length movie transmitted within typical current compression standards (1:30 length, 55 packets per frame on average, just under 30 frames per second [94]).

Per-bit energy savings is also provided for the sake of mapping this effect into the space created from other work on various tradeoffs and their per-bit energy cost in networks. This per-bit savings is not to be confused with per bit cost over the node or network. It is a pooling effect, refactored per bit, of the redundancy effect under study here.

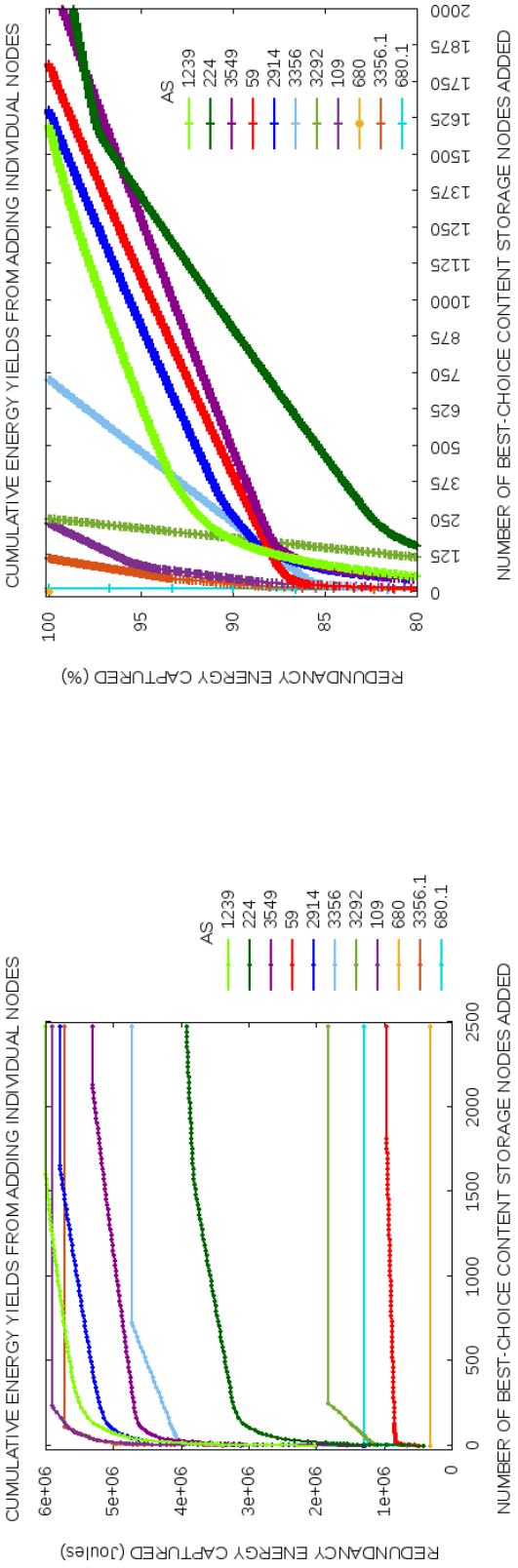
[4]This is not to be confused with more common uses of the term *per packet energy* over the network: this is savings across the network here, not energy spent at the node. More specifically, it is the pooling effect of redundancy, factored per packet, of the specific content under study. This is used in part to normalize the results across various content sizes, which is a more meaningful metric in some forms of the analysis.



(a) full content range

(b) experimentally tested range

Figure 2: Total Energy as a Function of Bytelength. Total capturable energy due to redundancy in each of the networks resulting from overlap of transmissions from 100,000 requests for a range of typical content lengths seen in Internet traffic: single web page download such as Google mobile home page; a 4'12" video clip download; a feature-length movie stream or download. The difference between networks is an increasingly important consideration for larger content. While a) shows the projected energy over the range of typical content bytelengths, b) shows the observed energy within the tested range, 180,025 to 623,700,000 bytes. The key is shown in order from the largest to smallest energy, corresponding with the per-content-packet energy in Table 4.



(a) cumulative energy on an absolute scale

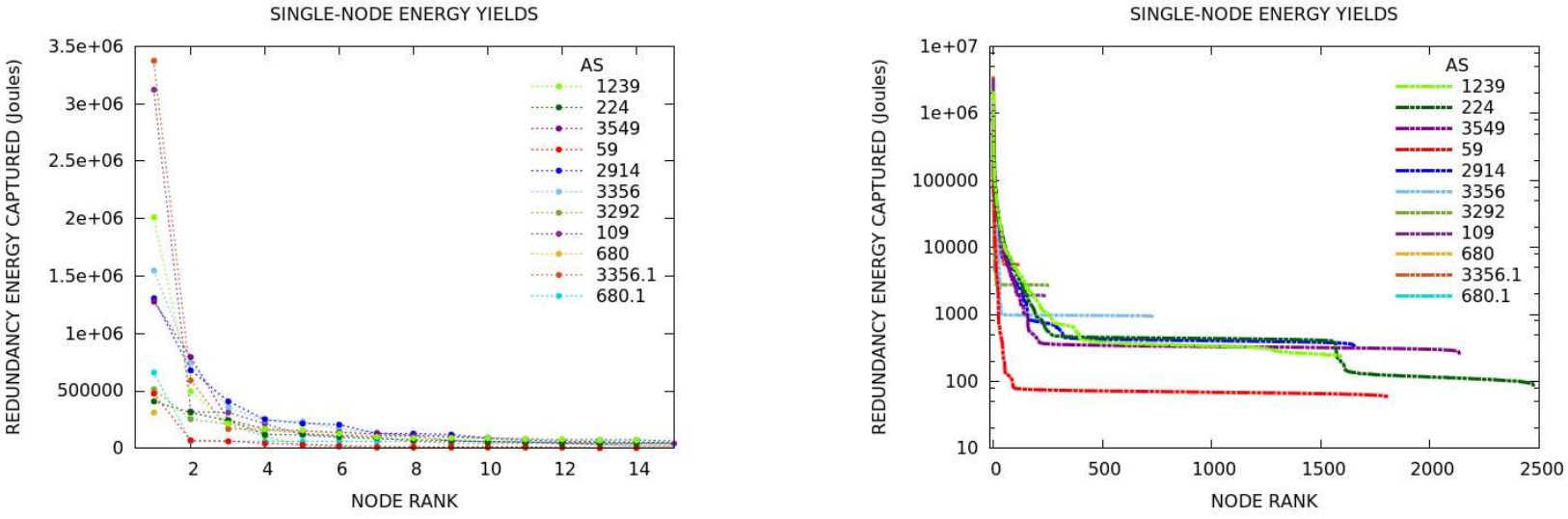
(b) cumulative energy shape on a percent-of-total scale

Figure 3: Redundancy Energy Accumulation Pattern. How the energy total grows with added nodes for the various networks, shown on an absolute and relative scale.

When looking at exploiting the effect of redundancy and how to harvest the energy it makes available, some important patterns arise both at the local and global level of the network. First, looking at the global level, as shown in Figure 3. When examining the underlying iterative process of adding duplicates at nodes that yield the most of the remaining energy, the redundancy energy arising from multiple requests exhibits a power-function pattern that roughly holds across the sample of networks, $y = ax^b, 0 \leq b \leq 1$. In words, the successive cumulative energy yielded by adding new nodes rises precipitously at first, then drops or tapers off as more nodes are added. The incremental effect of adding a node diminishes rapidly as more nodes are added in some cases; how rapidly the effect diminishes varies widely across the networks, as seen most clearly in examining the different curve shapes in Figure 3b, where the cumulative energy effects for the networks are expressed as a percentage of the total energy to align their results.

Figures 4a through 5d show similar information from a node-centered perspective rather than a total-energy perspective. This information is useful for deciding on a content storage strategy. One may be tempted to view this as energy directly available at the node for storage; however, it is more accurately viewed as network-wide energy freed because of storage at this node: the costs of transmission through routers upstream of this node along the path from source to sink are suppressed by storage of the content at this node. Figure 4a shows the top-energy-yielding nodes; 4b shows the full range of trim nodes that were selected for an energy yield of at least .1% of the energy yield of the best node in the network. The iterative, pruned search algorithm selects the best node at each iteration, starting with the node of rank 1; performs the communication trim; and proceeds to find the next-best node.

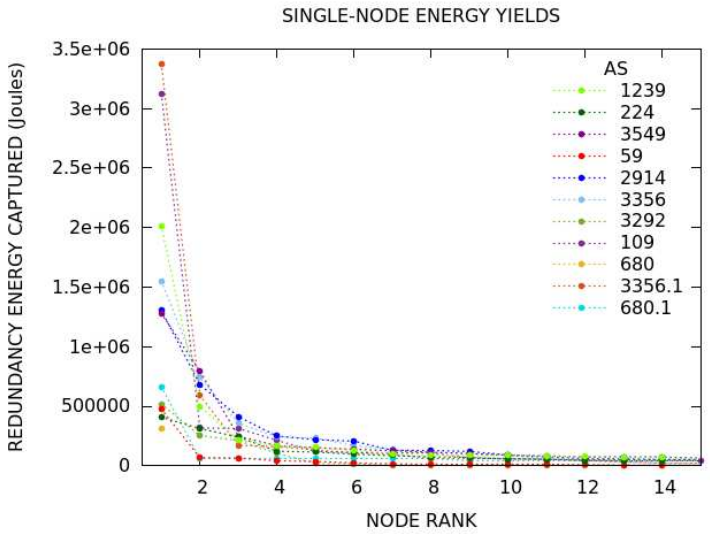
The numeric values for the top-ranking nodes are also provided, in Table 8. These are termed the *energy pools* at nodes; they are not literally energy stored at the node, but rather energy liberated by taking action at the node—savings resulting from storing the content at the node and re-sourcing from the local copy to satisfy subsequent requests.



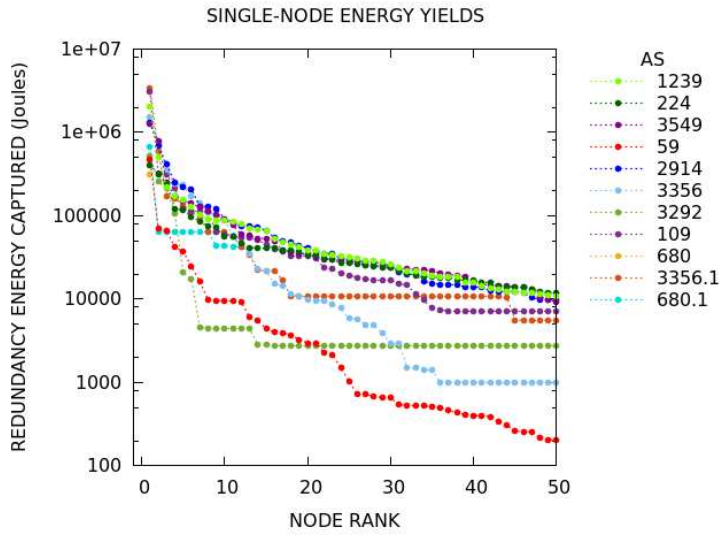
(a) the highest-ranking nodes

(b) all selected nodes, from full search range

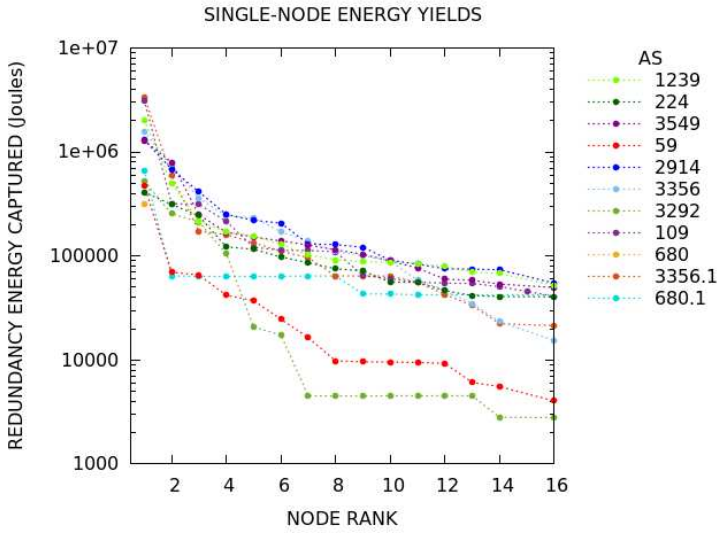
Figure 4: Individual Node Energy. Redundancy energy captured for each network at individual nodes, by taking advantage of redundancy. Figure a) shows the top-energy-yielding nodes; b) shows the full range of trim nodes that were selected for an energy yield of at least .1% of the energy yield of the best node in the network.



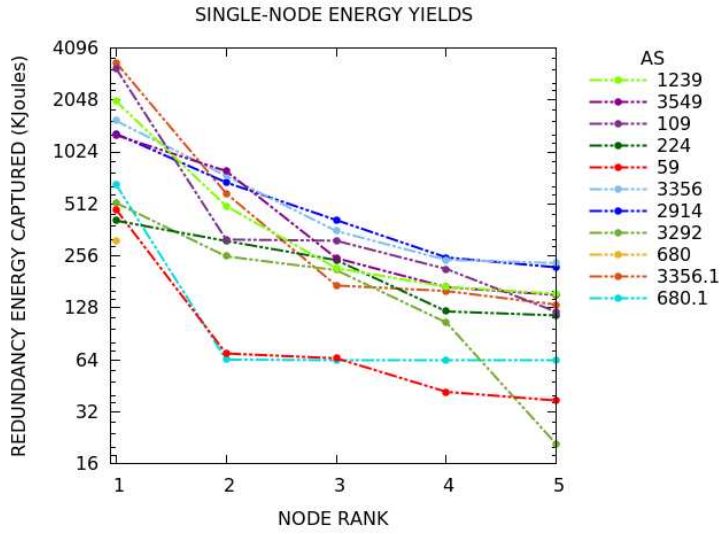
(a) the highest-ranking nodes, $\boldsymbol{y}$ axis on a linear scale



(b) nodes of rank 1-50, $\boldsymbol{y}$ axis on a log scale, allowing comparison of networks



(c) same as (a), but with $\boldsymbol{y}$ axis on a log scale



(d) just the top five nodes, allowing comparison of networks

Figure 5: Network Variation in Individual Node Energy. A closeup comparison of the same data from Figure 4: redundancy energy captured for each network at individual nodes, by taking advantage of redundancy, with all but (a) shown on a log scale to enhance the differences between nodes and to facilitate comparison of the networks with each other. Figure (a) again shows the top-energy-yielding nodes; (b) shows the top 50; (c) is (a) on a log energy scale; (d) is a closeup of the top five nodes.

Actually storing the content at this node *requires energy*, which is why it is important to cull nodes that do not add much to the total cumulative energy savings across the network. In this regard, the information in Figures 4a-5d is quite interesting and useful. It suggests that a liberal caching policy such as *Cache Everywhere* adopted within several ICN proposals is not energy-efficient. Instead, there is just a small handful of nodes for which storage saves energy on a network-wide scale. This also implies that an analysis should be performed periodically of each network as its node hardware and connectivity evolves, and from all source nodes, to identify what could be a very small set of nodes that yield the greatest energy efficiency in content distribution. A local-node caching policy suffers a lack of knowledge of the gains necessary to perform this analysis. This adds additional an additional intra-network domain of relevance to prior work by Modrzejewski [90] that demonstrated the benefit of a global energy analysis when looking for the best locations for proxy servers within a network.

Figure 6 provides a different way of looking at the distribution of energy pools and the relative size of the pools both within and across networks as a series of spectra. This is again for 100,000 requests for the content—18K for AS 59, where there are not 100,000 users—and is shown for the case of a video clip. Each energy pool available at a node is plotted as a dot. A notional dashed horizontal line indicates a threshold set for a particular storage scheme below which, storage at the node is energy-inefficient. As bytelength increases, only the y-axis values change in this spectrum plot, because of the linearity of total energy with bytelength and the static nature of the analysis, with hindsight, of a historical traffic log. The actual dashed-line threshold varies with the known cost of the storage method (cost per byte $\times$ content bytelength); different schemes could be employed at different set points on the energy scale. In practice, different storage methods could be invoked at different absolute energy values in $\boldsymbol{y}$, and these could be triggered by both the size of the content item and/or the estimated anticipated popularity of the content item.

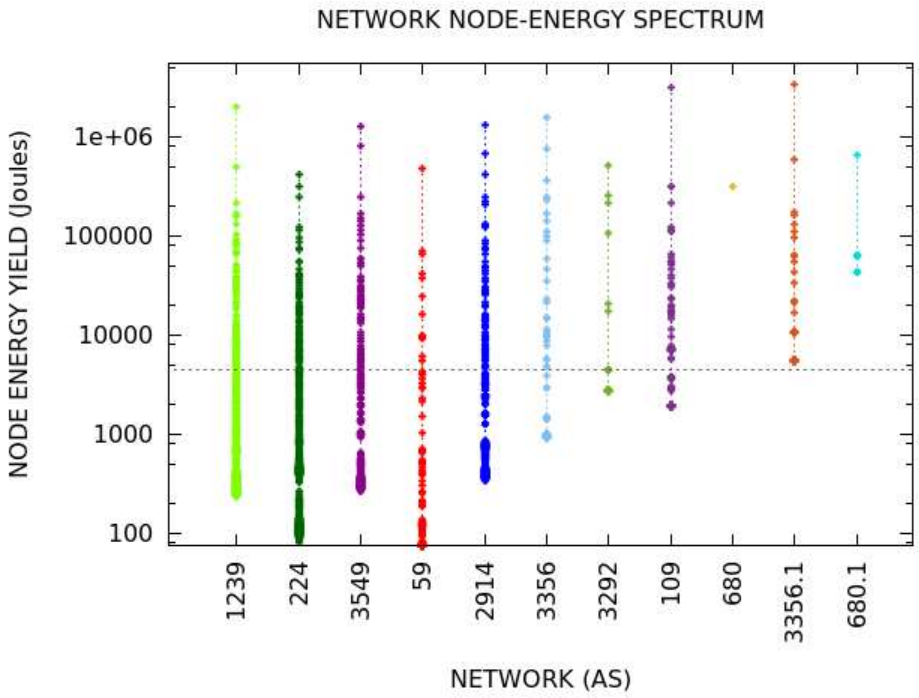


Figure 6: Spectrum of Node Energies for Each Network. Each energy pool available at a node is drawn as a dot, showing the distribution of energy pools and the size of each pool. These are plotted for each network side by side, allowing comparison between networks, from the largest-sized communication graph on the left to the smallest. Total energy derived from the redundancy within 100,000 requests for a $4'12''$ video clip within each network; 18K requests for network AS 59. A notional dashed horizontal line indicates a threshold set for a particular storage scheme below which, storage at the node is energy-inefficient; this could be set at the known cost of the storage method for content of that length and popularity; different schemes could be employed at different relative heights along the energy scale. As the bytelength of the content of interest increases or decreases, this changes only the y-axis values, no other aspect of this graph.

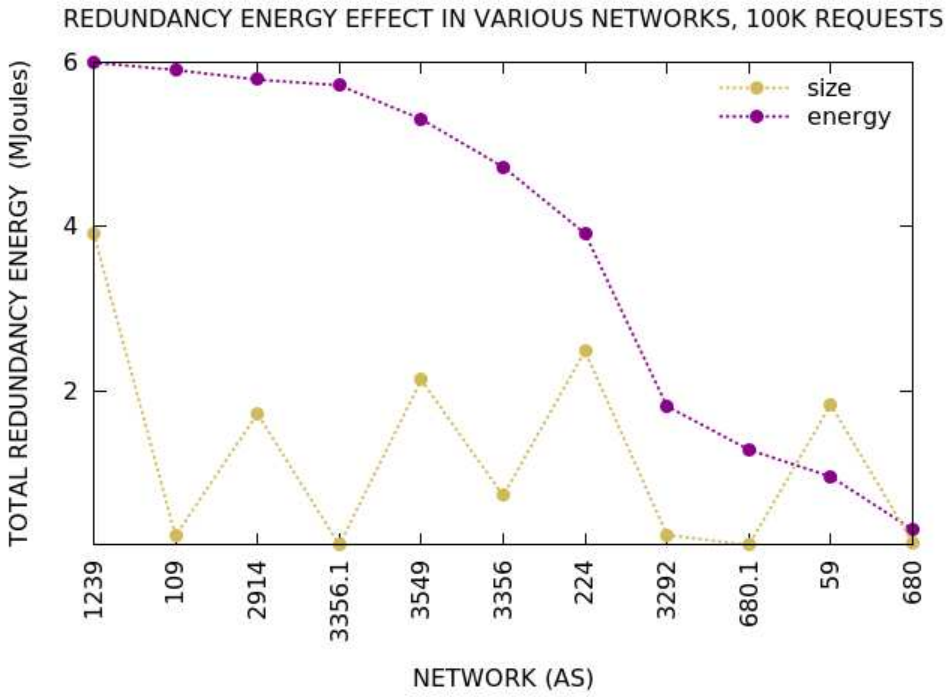


Figure 7: Total Energy vs Network Size. Total energy available from the redundancy within 100,000 requests for each network, in purple, alongside the network's size, in gold. AS 59 included as just an additional data point because its experimental conditions are different (18K requests).

As seen in Figure 7, the total harvestable energy associated with the redundancy within 100,000 requests does not appear to be a function of the size of the communication graph that is the number of nodes in the spanning tree from the source node to all receivers. There also appears to be no strong correlation between total energy, communication graph size, and the parameters of the power function of node accumulation from Figure 3b.

Table 7: Energy Dividend vs Item Popularity. The size of the energy dividend from redundancy varies fairly linearly with number of requests. The columns marked Deviation from Proportionality indicate the amount of deviation of the 500K and 1M cases from a perfectly linear relationship with the 100K case.

Network	**Energy Per Packet 100K (J)**	**Energy Per Packet 500K (J)**	**Deviation $\propto$ (J)**	**Energy Per Packet 1M (J)**	**Deviation $\propto$ (J)**
AS 3356	11.3550	56.8360	-0.0608	113.6580	-0.1076
AS 3292	4.3630	21.8302	-0.0150	43.6658	-0.0353
AS 109	14.1993	70.9886	0.0077	141.9871	0.0055
AS 680	0.7535	3.7580	0.0094	7.5133	0.0217
AS 3356.1	13.7498	68.7545	-0.0057	137.5037	-0.0061
AS 680.1	3.0884	15.4451	-0.0031	30.8906	-0.0066
AS 59	0.1722	0.9182	-0.0571	1.7735	-0.0513

Table 7 shows the deviation of the results from direct proportionality of total energy with the number of requests for the content item. Some are slightly above and some slightly below perfectly linear, and some vary from 500K to 1 million, but the numbers are very close to linear across a range spanning an order of magnitude. Total energy is expressed per packet of the content; nonetheless, it is total energy across the network available due to redundancy. The columns marked Deviation from Proportionality indicate the amount of deviation of the 500K and 1M cases from a perfectly linear relationship with the 100K case. The results presented in Table 7 were obtained over eight trial runs for each network at each of 100K, 500K, and 1 million requests for the same content, in this case of length equal to a web page. Results are not reported for several large

networks, for which the larger-size assessments are not available.

The gain shows linearity, not hyperlinearity, for larger pools of subscribers. The fact that more subscribers does not result in more gain may indicate that the topology, combined with the fixed shortest-path routing scheme, constrains the realizable effect.

Table 8: Comparing Full Cost to Efficient Content Distribution. A side-by-side comparison of the transmission cost of unicast with the transmission cost of the traffic after suppression of redundant transmissions for 100K requests for a $4'12''$ video clip. Networks listed in order of size, largest first. Also shown is the incremental energy savings resulting from replicates at each of nodes that resulted in the highest energy return: numbers are reported before accounting for content storage energy.

Network	**unicast cost (J)**	**min ECD cost (J)**	**max savings (J)**	**max savings (%)**	**nodes: 1st (J)**	**2nd (J)**	**3rd (J)**	**4th (J)**
AS 1239	7,057,081	1,191,532	5,865,549	83.1	2,014,094	497,090	217,599	169,569
AS 224	4,597,382	1,058,126	3,539,255	77.0	409,599	312,597	242,353	122,146
AS 3549	6,025,916	722,217	5,303,700	88.0	1,281,261	793,227	248,466	168,823
AS 59	1,247,162	352,008	895,155	71.8	475,942	69,741	65,466	41,843
AS 2914	6,533,060	747,885	5,785,175	88.6	1,304,700	681,895	412,787	250,719
AS 3356	5,442,846	712,274	4,730,572	86.9	1,549,079	744,902	356,682	242,832
AS 3292	2,524,117	708,866	1,815,251	71.9	519,489	254,659	211,979	105,309
AS 109	6,626,477	717,788	5,908,689	89.2	3,120,421	318,543	313,561	215,037
AS 680	2,674,362	2,362,111	312,251	11.7	312,919			
AS 3356.1	6,495,726	775,457	5,720,269	88.1	3,376,621	589,794	172,636	160,463
AS 680.1	1,991,771	707,211	1,284,560	64.5	664,026	64,265	64,217	64,216

Table 8 shows a side-by-side comparison of the transmission cost incurred by the original full-unicast transmission pattern within the traffic logs before and after redundancy within the traffic pattern was assessed. This shows the maximum energy deliverable by redundancy within the experiments—the actual energy savings requires an important additional expense, of storing the content, at a content- or at a packet-level, at the nodes selected as most-efficient positions, so that

requests for the item are satisfied locally. While this is not in effect stating the energy savings of efficient content delivery, it is setting the bounds for it: this is the maximum amount of energy that can be harvested from these network topologies by filtering unicast transmissions; the actual savings will most certainly be lower, once the strategy for storage is decided and the cost of storage is accounted for.

The numeric values for the top-ranking nodes are also provided, in Table 8. These are termed the *energy pools* at nodes; they are not literally energy stored at the node, but rather energy liberated by taking action at the node–savings resulting from storing the content at the node and re-sourcing from the local copy to satisfy subsequent requests.

3.5 Discussion

Further work is needed to tighten these estimates for the energy savings dividend from conserving duplicate transmissions in specific networks. The method seeks to establish the exact Joules that a transmission costs at a node; this relies on knowing the particular network hardware intimately, to know the incremental cost of packet processing and store-and-foreward cost. We have few detailed estimates for these from detailed hardware profiling and therefore used numbers as representative of the class of hardware nodes; however, this lacks the precision of utilizing the actual costs for the hardware deployed within the network. Added to this is separation of the service-layer packet processing cost from the hardware-layer processing cost; these costs are not separated out in the hardware profiles. This knowledge would aid in more closely estimating the specific router's cost to perform more complex analyses related to acting as a re-source of the content, such as deep packet inspection that may be required to intercept, cache, and/or redirect packets, rather than simply forwarding them.

Further work is also needed to properly assess the energy effect attributable to a reduction in

the cooling requirements of the routers and switches. As Kilper notes, rack systems are already thermally limited, and higher bandwidths require more cooling capacity than is present [15]; Ma and Harfoush argue that cooling energy demand is not linear with traffic load, but polynomial, greatly increasing the importance of load-effect on cooling for higher loads [105]. Cooling is also a significant energy cost, assessed but not directly measured as 100% again the idle power of the device by Vishwanath in their detailed studies of router and switch model energy consumption [98]. Given that the energy liberated by traffic optimization also directly reduces traffic load, and in precise and measured amounts, we intend to broaden the energy model to include the cooling energy.

3.5.1 Conclusion

There is a tendency among researchers to avidly optimize a thing without first asking the question whether it matters. The goal here is to assess the achievable energy savings due to the existence of redundant requests for content. It is the intent of this work to promote future protocols and other proposals that can save significant energy within networks, with minimal impact on quality of service. This work shows what is achievable if optimal predictive protocols could be enacted.

This work accumulates evidence toward an understanding of the energy pool available within complex real autonomous-system networks for more-efficient content delivery than achievable through unicast, cacheless transmission through the same network. The focus here is on teasing out the effect of network topology on the total energy pool, the distribution of separate energy pools at individual nodes, and the effect of the content popularity, volume of traffic on the size of the energy pools. This work also provides information about the sensitivity of the energy pool—and indirectly, the network topology—to request ordering.

The energy efficiency of observed patterns of satisfying multiple simultaneous users requesting the same item of static content over a wired data network was assessed for specific autonomous sys-

tems within the Internet, from intra-network scans that produced networks defined at the physical router level. A perfectly energy-efficient traffic pattern is one that cannot be further refined or optimized. For a fairly efficient communication pattern, few refinements can be found; whereas, for an inefficient communication pattern, there are large refinements, or many small refinements, that can be found. This work quantifies and estimates the effect, for the most common type of transmission protocol over the Internet; it also investigates the dependency of the quantification on traffic volume from to the popularity of the item, on the specific time-ordering of network transmissions, and on network topology.

The energy efficiency of a particular content delivery or communication pattern was defined in proportion to the difference between the original pattern and the post-refinement pattern. The energy available to specific nodes, usable to store and re-source content in the form of packets, was evaluated for the entire network, and nodes were ranked by the amount of energy available. The patterns of energy distribution across nodes were examined for a sample of a variety of networks, representing large Tier-1, Transit, and large- and small- stub networks. For all of these, for the sake of tractability, at most two content-source nodes were examined; thus, with the possible exception of stub networks, these numbers do not represent a full energy assessment of content delivery across the particular autonomous system.

An energy budget of approximately .74-14 Joules per packet was the result of assessments when looking at the redundancy across 100,000 separate requests for the same item in various networks. The total energy associated with the duplicate transmissions that could be eliminated by various means was found to be between .9-17 KJ for single web page-sized content items, and between 313 KJ and 6 MJ for a single average-length ($4'12''$) video clip when requested by 100,000 users. The variation that these ranges suggest arises from the different topologies of the autonomous systems examined.

The energy associated with redundancy is quite a large proportion of the total energy spent at

routers and switches on behalf of 100,000 requests for the same content. This work found that, on average, 75% of the total transmission cost is due to redundant transmissions across a range of networks. These wasted bits could potentially be eliminated by an efficient content distribution method within the network. Doing so requires a new strategy for encryption of copyrighted content other than point-to-point; however, this work showed that the majority of the energy benefit is achieved at a very small percentage of network nodes, therefore incurring little additional encryption cost, and contemporary point-to-point encryption techniques can be readily extended to a several-stage distribution method. This, however, is beyond the scope of this work.

Chapter 4

Conservation of Energy through Entropy Reduction

4.1 Introduction

This chapter focuses on the system-wide energy efficiency of one instance for which computation reduces the size of the bitstream being transmitted over the network; that is, computation on the bitstream itself to reduce signal entropy. This chapter examines the energy impact on the global telecommunications network as a whole of *entropy reduction* for content traveling between the outer periphery of the network and the center. The amount of energy required for *entropy reduction* at the edge of the network, on peripheral devices is compared to the energy gained in transferring the lower-entropy result through the network.

The choice to focus on the entropy reduction effects of data processing arises from the fact that the Shannon entropy of a signal generally decreases as it is processed, shedding noise, data errors, and useless bits, and lowering the channel capacity requirements of the signal from one processing step to the next. Among other things, this research aims to better understand the tradeoffs involved in choosing how to partition the data layer–the processing steps or processing chain from raw data to outputs—onto the network path from source node to sink nodes on a physical network, such that

these entropy effects are more fully exploited to conserve energy used per bit of useful information by reduction of channel capacity required. These effects can be readily modeled and compared from one processing chain/network node configuration to another.

To assess entropy reduction's role in energy consumption on large data networks, one might compare the entropy of two different processing chains on the same hardware, such as two scenarios for delivery of home movies using two video compression techniques where one more computationally intensive technique results in a more-compressed video stream. What are the net energy effects of trading the increase in computational energy and latency of the more aggressive compression method for the transmission bandwidth reduction due to data reduction, or greater entropy? The static aspects of this question are directly addressed in the current document, for the more generalized notion of entropy reduction of content delivered to home residences through the application of compression techniques to the data streams. The energy used at the leaf nodes is directly compared to the energy used within the network by various compression schemes.

Few researchers are examining broad strategies that cross the many boundaries of separate concerns and technologies involved in the ICT, and perhaps even fewer are projecting these analyses forward five to ten years from the set of base technologies improving at different rates. This work therefore also aims to provide a broader view of the potential energy savings from new technologies and techniques—broader both in the scale and temporal range considered [106].

4.2 Background

Information entropy for digital communications has been an area of research since the 1940s, with Shannon's formulation of *bits* and binary encoding, followed by his investigation of the limits to channel capacity, resulting in the Theory of Information [10]. Some areas of active innovation

within *data compression* or *source coding*[1] that often combine algorithms with hardware considerations include advances in video coding [107–111]; data encoding and hardware-aware customizations within the severe power constraints of wireless sensor networks, see for example [112–118]; and advances in source coding in the less-constrained but more-prevalent area of wireless mobile devices, see for example [119–121]. Source coding has also been proposed and developed for the CPU/network throughput mismatch seen particularly in high-performance computing [122]. Compression energy consumption has also been the subject of examination for servers [123].

4.3 Related Work

The application of entropy reduction examined here is to the original content as opposed to packet-level entropy reduction, and is neither within-packet entropy reduction, nor inter-packet redundancy elimination, the latter of which was originally proposed by Spring and Weatherall and demonstrated to reduce traffic considerably in a small set of test networks [85].

Prior work in two areas informs the present work. The first is analysis of network energy consumption trends for communications from any sender to any receiver over the global telecommunications network by Baliga that derived an energy associated with transmission of bits over the telecommunications network [8, 78]. The second area is a series of studies aimed at energy efficiency of applications run on low-power peripheral devices, beginning in 1999 with error correcting codes measured by Hayinga [124] and thereafter examining the energy efficiency of lossless compression techniques[2], first by Barr and Asanović in 2003 for low-power embedded microprocessors [125, 126]; followed several years later by customization and measurement of lossless compression techniques for severely constrained embedded devices within multi-hop sensor networks by Sadler and Martonosi [113]; followed by a recent set of studies similar to those by Barr

[1] *source coding* is used here interchangeably with *data compression*. The more familiar term depends on the field.

[2] source coding in which statistical redundancy is removed from the source signal without loss of any information contained within the signal

and Asanović done a decade later by Dzhagaryan and Milenkovic [127, 128], with new compression techniques, greater fidelity from higher sampling rate, and larger test sample, for several ARM embedded platforms. None of these directly compares the processing energy found to the network energy encountered by the resulting compressed file over large networks, which is a key aspect of the current work.

The most direct lineage of the entropy reduction work presented here is to an early effort aimed at energy efficiency measurement in wireless systems for error correcting codes by Hayinga in 1999 [124]; followed by an innovative and in-depth study of *lossless compression* [129] techniques on mobile-class wireless devices conducted by Barr and Asanović in 2003 [125,126]; followed several years later by an in-depth customization and measurement of lossless compression techniques for severely constrained embedded devices within multi-hop sensor networks by Sadler and Martonosi [113]; followed by a recent set of similar studies to those by Barr and Asanović done a decade later by Dzhagaryan and Milenkovic [127, 128, 130], with new compression techniques, greater fidelity from higher sampling rate, and larger test data, for several embedded platforms. None of these directly compares the processing energy found to the network energy encountered by the resulting compressed file over large networks, which is a key aspect of the current work.

Within this lineage, Barr and Asanović [125, 126] were the first to point out the disparity between processing energy and transmission energy. In their case, they looked to exploit the 1000:1 difference in energy cost of a single CPU instruction as compared with transmitting a single bit over an 802.11b wireless link on a dedicated channel for UDP packets or datagrams between a wireless sender and receiver. Also present in their work was the notion that this discrepancy does not in fact imply that any compression performed will result in a net energy savings, and they explore the reasons for this in depth, particularly within the algorithms' map to the hardware resources.

Sadler and Martonosi were the first within this lineage to extend this further into the network by quantifying the tradeoffs in processing energy versus communication energy for radio wireless

communication in a multi-hop sensor network and to determine the energy spent as a function of number of hops traveled within the sensor network for various communication methods [113]. One result was finding also in this arena a large discrepancy between processing and communication energy in sensor networks that gave them many options for customizing compression utilities to the platform. They were the first to conceptualize and demonstrate the accumulated benefit of compression when a multi-stage transmission network is considered. Theirs, however, was a radio sensor-to-sensor communication network; thus, in the present study, their network would fall out on the periphery outside the *access network* and does not address the network energy of interest here.

Kothiyal examined the relative cost of transmission of uncompressed data compared to that of processing and then transmitting the compressed data over ten core-network hops and found that the data transmission cost was within the realm of $\mathbf{10^{-7}}$, and that compression was a losing proposition, raising the total cost by an order of magnitude [131]. Our results, however, contradict this result, for both analytic approaches presented in this chapter and show a transmission cost within the realm of $\mathbf{10^{-6}}$, not $\mathbf{10^{-7}}$, for each node, from the models of Baliga [78] and from the energy profiles of the devices from Vishwanath [98] that are used here to assess specific networks.

The Dzhagaryan and Milenkovic studies [127, 128, 130] directly measure the energy spent communicating from device up to the access network, to a local host, for both a wireless LAN 802.11n and a wired Ethernet network connection. They also underscore the importance of communication energy as a consideration, in their use of this measurement as a threshold to indicate what compression or decompression utilities result in lower compression/decompression plus transmission than raw uploads and downloads, finding only a subset that meet this criterioe their analysis ends at the access point to the first network link, it measures a much more local energy profile that will be shown here to result in different estimations of compression utilities' benefit. Even more significant is the difference in viewpoint as to what defines energy efficiency taken in the current work.

Their metric is essentially the per-bit energy expenditure of the application when processing its input file—suitable for comparing the energy efficiency of the utilities one to another. The question asked here is broader. By defining energy efficiency as the Joules consumed per bit processed from the original file, Dzhagaryan and Milenkovic do not consider transmission costs directly, nor do they factor in the compression ratio achieved by the application as part of its energy efficiency. Here the definition of energy efficiency includes both transmission costs and compression ratio. These allow a broader definition of energy efficiency by identifying what is gained in exchange for the energy spent processing the file.

As seen in the present analysis, however, network transmission beyond the first link and through the global telecommunications network far overshadows the total magnitude of energy consumed at the mobile device. At this larger scope the recommendation could be to take the best application at each compression ratio in all cases, because compression reduces transmission energy sufficiently that the indication is to not reject but rather utilize the compression scheme. In short, network effects serve here as the dominant factor by which the best compression utilities from the previous study are reordered and by which the threshold is drawn to accept or reject a compression scheme. This is because the previous work defines transmission energy only as what was expended to reach the access network. The much larger relative cost of communication is revealed more clearly here, where the full network is considered. Previous studies have focused more on conserving individual device energy, or on conserving communication energy only within the set of devices, as is the case in Sadler and Martonosi [113].

To estimate communication energy per bit within the telecommunications network, the work presented here relies on results and future trends from a detailed analysis of network per-bit energy by Baliga [78] and further analysis of it by Tucker [8]. Yet the goals here are quite different, whereas the prior work is strictly considering the network infrastructure and technological advancements; Baliga and Tucker focus on the telecommunications network only, not device energy. Their work

falls into the category that considers *conservation of energy per bit*, whereas the present work falls into the reciprocal category of considering *conservation of bits*—separate concerns contributing to total ICT energy use.

The contribution of the current work stems from synthesis, essentially comparing software trade-offs in the application layer running on peripheral devices in terms of the balance of device energy lost and network energy gained by their use in large telecommunications networks. This extends past where either body of previous work considered energy effects: the network analysis considering only network infrastructure, and the compression analysis considering only peripheral devices and at most the first communication hop to reach the network. This allows for a projection of the footprint of various applications on networks that to our knowledge has never been done before in terms of energy. The potential benefit is large, since the holistic view comprises a much larger percentage of global energy supply than either part alone.

One can attempt to reduce data transfers from one computational node to the next—thus reducing communication energy—at the expense of the processing energy expended to remove redundant bits in the signal. The specific question of the computational and communication *divide* between two nodes, however, is considered here to be a local optimization problem for which there has been extensive study [132]. In this work, the focus is instead on the global issue of overall system energy; at this larger scope and focus, policies enacted by such local optimization may be found to be at odds with global energy conservation.

It is also important to credit the importance of many areas of active research within networks that lead to greater energy efficiency, either within network infrastructure, within data center networks, or for mobile networks, as recently summarized in [133–136]. These efforts do not consider the application layer and focus largely on reducing transmission costs—reducing the cost per bit of communications—rather than the application-layer methods and routing methods explored here that improve energy efficiency by reducing the number of bits transmitted over the network.

4.4 Methods

This work examines the overall energy performance of applications running on devices peripheral to the global telecommunications network such as smart phones and tablet computers that people throughout the globe use throughout the day. Energy performance is defined here in a holistic manner that includes not only the device energy consumed by the application but also network energy consumed on behalf of the application, primarily in the form of communications to and from the device. Network and device energy are treated as fungible goods here, exchangeable and often traded one for the other in the way an application is designed. In these designs, network energy is typically not viewed as being nearly so tightly constrained as device energy due to consumer demand for reasonably long battery life on mobile devices, but also likely due to the effects of *tragedy of the commons* [137], since no one entity bears the burden of network energy. The focus here is on a specific subset of applications and techniques that have the potential to significantly *reduce* network energy costs. An example is applications that lower the entropy of the information at the device before transmitting the information through the network.

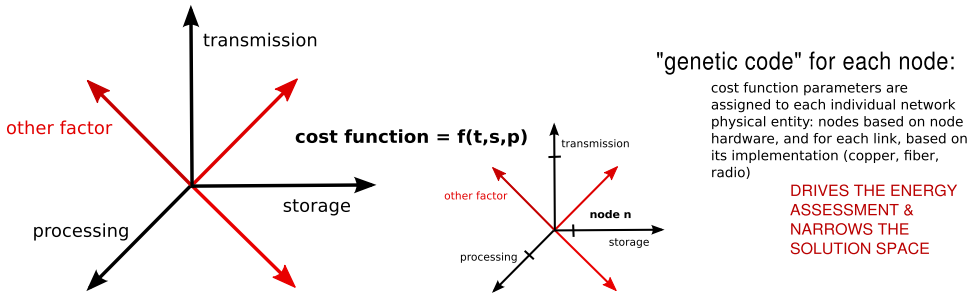


Figure 8: Illustrating the Basic Tradeoffs. The primary tradeoffs considered here: the relative cost of storage, processing, and transmission of information.

One main quantity is defined here and motivated, the *energy footprint* of a particular application, a whole-systems measure of the application's energy usage and energy efficiency both at devices and within the network. These are used to quantify the net energy gain or loss within the ICT from

reducing the redundancy within the data transmitted, which shortens messages. The joint energy is considered, from the union of three chief quantities:

- *processing energy*: how much energy the device uses (to reduce information entropy or to optimize transmissions), and how much energy network nodes use on behalf of the application
- *communication energy*: how much energy the subsequent transmissions of the information use within the network
- *storage energy*: how much energy the device or network nodes use to either cache or store in memory the content or packet-level representation in order to save communication energy elsewhere in the network

The joint energy from these three quantities is termed the *energy footprint* of the application. This footprint considers both the energy consumed on the peripheral device and the energy consumed within the network from using the application or technique. For some applications, processing energy is not fully circumscribed by the device, and in those cases the definition extends into the network and is the sum of the effort by the source and sink nodes and perhaps intermediate nodes involved with conditioning the data. The communication energy is the sum of energy required by the source, sink and all intermediate nodes within the network to process and forward each data packet in the path from the source to sink node.

As an example of how energy performance is framed, consider a video capture process running on a mobile device; assume the images are stored in the cloud. The question asked is, how much processing energy is used by this application to reduce information entropy, and how much communication energy is required by it as a result? This is the essence of the assessments described throughout Chapter 4 when examining entropy reduction. These are often described here in terms

of bits or binary digits of information—particularly the number of bits eliminated by processing the information to reduce redundancy, and the number of bits carried or conveyed by a network.

The question addressed in this work is centered around information theory rather than network technologies [10]. That is, given the seemingly inescapable fact that it costs something to transmit one bit of information over a network, there are two reciprocal quantities implied by this that require new methods and technologies; either or both can be addressed. They are: *conserving the cost per bit*, and *conserving the number of bits*. This work focuses on the latter; that is, it seeks energy benefits from reducing the number of bits transmitted over the network, rather than focusing on reducing the cost per bit through innovation in the physical infrastructure. Both are critical efforts to reducing the *metabolic rate* of resource consumption of large-scale networks. There remains much inefficiency in the number of excess bits sent over communications channels beyond the theoretical minimum, as well as beyond the practical limits of quality required for many activities. This fact can be used to benefit global energy use through new methods and technology applied to reduce the energy footprints of individual applications.

The concern here is again with energy performance in Joules, a time-less unit. This work does not focus on power or other time-based measures of the rate of energy flow, yet the results can readily be extended and related to other time-based metrics like throughput or speedup.

4.4.1 Energy footprint

Given the focus of the current work on *entropy-reducing applications*, the energy footprint of the application can simply be defined with two terms, energy primarily being either:

- *processing energy*: how much energy the device or other network nodes use to reduce information entropy
- *communication energy*: how much energy the subsequent transmissions of the information

use within the network, going from node to node along the path from source to sink

The *energy footprint* $\boldsymbol{f_e}$ is directly proportional to these quantities as accumulated along the path from source to sink node

$$\boldsymbol{f_e} \propto \sum_{i=0}^{n} (\boldsymbol{p_e} + \boldsymbol{c_e}) \tag{4.1}$$

where $\boldsymbol{n}$ is the number of nodes involved, $\boldsymbol{p_e}$ is the per-bit energy contribution of the processing or computation performed by the node to achieve entropy reduction of the data stream as described in the Introduction, $\boldsymbol{c_e}$ is the per-bit energy contribution of transmission or communication between nodes.

For all transmissions of the data stream $\boldsymbol{d}$, communication energy is assumed to be directly proportional to the size of the stream

$$\boldsymbol{c_e} \propto |\boldsymbol{d}| \tag{4.2}$$

where $|\boldsymbol{d}|$ is the number of bits in $\boldsymbol{d}$. In the cases examined for initial work, the data stream undergoes a single transformation at the source node, and the size of $\boldsymbol{d}$ is therefore unchanged as it is transmitted from node to node.

The initial work also assumed a simple, common framework for communication that involves no sharing between users of the data at any node, such as might be the case for unicast style of communication between server and client. One important embellishment in follow-on work is to consider other forms of communication that do involve data stream sharing, such as multicast, broadcast, and peer-to-peer bitTorrent-style sharing; in those cases, $\boldsymbol{c_e}$ is not in general derivable from static analysis and requires a more complex energy consumption model. For future work, a first pass at defining this more complex consumption model is done via analysis of simulations.

Both $\boldsymbol{p_e}$ and $\boldsymbol{c_e}$ are measured in Joules (J) and normalized as a per-bit expenditure (J/bit) for direct comparison. Although the energy costs of the two are due to different effects, energy is energy at this level, permitting comparison of energy spent in one place within the network versus another. This is appropriate because it is precisely this tradeoff of processing energy for communication energy upon which the present analysis focuses. Where this occurs in the node network is ignored within the initial work.

Equation (4.1) uses a more general definition of processing energy than is strictly required for the stated goal of the current analysis, at least as described in the Introduction, for which all processing is done on a device peripheral to the network. This is to allow for applications that perform successive steps to reduce the entropy of the datastream on one or more devices or nodes. The term *node* is used here to indicate any hardware with processing capability on the path from the sender or source to the receiver or sink node, including the sender and receiver. In the cases considered here, one terminal node—either the sender or the receiver—is a device on the network periphery such as a smartphone, laptop, tablet, HDTV, or game console; the other terminal node is a server sitting centrally on the network such as an Autonomous System-level server or cloud server. Because the roles of sender and receiver are exchanged in the experiments below, the term *device* will be used to distinguish the terminal node that is a peripheral device—which could be either the sender or receiver; the term *server* will be used to distinguish the other terminal node, and the term *node* will be used for intermediate nodes along the path from *device* to *server*.

In order to measure $\boldsymbol{p_e}$ and $\boldsymbol{c_e}$ there must be a specific electrical device upon which the current can be measured. Here that physicality is encompassed by analyzing energy use for a specific hardware/software/firmware instance. The term *component* is used here to attach energy measurements to a physical representation of the node, including what hardware platform is used, what software or firmware is used, what algorithm is implemented in the software or circuits—all of these to some extent contributing to the measured energy use for the activity.

This physicality stops at the terminal nodes in the present example, the *device* or *server*. While different network node physical devices and connectivity and routing layer can have a dramatic impact on communication energy, these are not the focus of this study. Even though a network-connected workstation and a wrist watch likely perform a particular computation at wildly different *rates*, it is assumed that the *amount of energy* required for a change in state of the information is constant across platforms. Within initial work, the network is instead assumed to be a fixed asset and part of the controlled variables within experiments conducted to analyze the impact of various device applications that perform information entropy reduction. In the initial study the device hardware is also fixed: only the application software is varied.

In future studies, the effect of hardware choice may also be examined for its effect on the energy footprint; however, this will be very limited in scope, limited to the leaf nodes, rather than considering changing the equipment used within the network. Another way of stating this is that physicality within the network is a necessary requirement for the low-level simulations planned for multicast and peer-to-peer sharing analysis; yet searching the space of equipment for lower-energy devices is not a part of this research. Thus, the physical devices considered are chosen arbitrarily rather than chosen for their energy efficiency. In this sense, the physicality of the network is consistently considered to be beyond the control or scope of this research.

Further developing this model, $\boldsymbol{c_e}$ in Equation (4.1) is the energy consumed downstream of the node $\boldsymbol{i}$, while communicating with the immediate downstream neighbor nodes. In the cases initially considered, the number of downstream neighbors is assumed to be one, or a path that takes a single branch from node to node. Given the type of applications considered initially, entropy reduction of the input stream via *lossless compression schemes* [129], there is a direct relationship between $\boldsymbol{c_{e_{i-1}}}$ and $\boldsymbol{c_{e_i}}$ through the change in $\boldsymbol{|d|}$ in Equation (4.2). This relationship is commonly called the *compression ratio* CR

$$\text{CR} = \frac{|\boldsymbol{I}|}{|\boldsymbol{O}|} = \frac{|\text{raw file}|}{|\text{compressed file}|} = \frac{|\boldsymbol{d_{i-1}}|}{|\boldsymbol{d_i}|} \tag{4.3}$$

where $\boldsymbol{I}$ is the set of bits entering the algorithm/hardware/software/firmware *component* as the input bit stream, and $\boldsymbol{O}$ is the set of bits exiting the component as the compressed bit stream at lower entropy, and the ratio of the two indicates the fractional size of the compressed file to the original file. In the case of decompression, the rightmost $\boldsymbol{d}$ term in Equation (4.3) is unchanged, but the $\boldsymbol{I}$ and $\mathbf{0}$ are flipped in the second term: compressed file in, raw file out. From this, the relationship of downstream communication energy from a node that performs entropy reduction is

$$\boldsymbol{c_{e_i}} \propto \frac{\boldsymbol{c_{e_{i-1}}}}{\boldsymbol{CR}} \tag{4.4}$$

where this step-energy reduction for $\boldsymbol{c_e}$ is achieved by the measured processing energy at the node $\boldsymbol{p_e}$. It is precisely this tradeoff that is of interest in the current study: does the energy spent at node $\boldsymbol{i}$ in entropy reduction come out as a net gain in energy expenditure downstream, in communication energy thereon, from node $\boldsymbol{i}$ to $\boldsymbol{n}$? This can be expressed as true when

$$\frac{\boldsymbol{c_{e_{i-1}}}}{\boldsymbol{CR}} \leq \boldsymbol{p_{e_i}} \tag{4.5}$$

for any sized network, although for large networks the cumulative effect downstream may not be captured by imposing this stricter local condition in the hop from $\boldsymbol{i-1}$ to $\boldsymbol{i}$ since at each hop, the communication energy is again saved. The cumulative benefit can be instead described as the accumulated energy reduction of the new regime where a data reduction step is taken at node $\boldsymbol{i}$, consuming $\boldsymbol{p_{e_i}}$, resulting in a fractional reduction of the original communication energy entering stage $\boldsymbol{i}$, $\boldsymbol{c_{e_{i-1}}}$ as

$$\sum_{\boldsymbol{j=i}}^{\boldsymbol{n}} \boldsymbol{c_{e_j}} \geq \boldsymbol{p_{e_i}} + \sum_{\boldsymbol{j=i}}^{\boldsymbol{n}} \boldsymbol{c'_{e_j}}$$

where c'_{e_j} represents the new regime in which processing step p_{e_i} occurs, and c_{e_j} represents the old regime, in which p_{e_i} does not occur. This captures the full benefit of performing p_{e_i} downstream from node i to the terminal receiver node n.

Whereas in this work Equation (4.5) represents the goal as well as the challenge to application-layer designers, insofar as others consider energy use at all, the typical definition of energy efficiency is simply the strict proportionality of the energy required to achieve a certain compression ratio with respect to other solutions, or

$$p_{e_i} \propto CR$$

The above definitions and assumptions constitute the basic working set; how these translate to a specific case is described in more detail in the Experiments and Results Section. In the experiments undertaken toward understanding the role of entropy reduction on global energy, the set of components analyzed is ones that perform *lossless compression and decompression* on the inflow using common utilities such as gzip; thus, the processing energy is used predominantly to transform the raw input file into the compressed output file or to re-inflate to the original file from the compressed file.

The question of interest in this analysis is, the relative magnitudes of the terms on the right-hand side of Equation (4.5) with respect to each other, meaning how much processing energy is spent in exchange for how much communication energy saved, and how large an accumulation of savings results from p_{e_i} in a large network where the data stream passes through many nodes on its hops to the *server*, such as the global telecommunications network. The decision of whether or not to perform the entropy reduction at all is whether the left-hand side is indeed greater than the right-hand side of Equation (4.5), meaning that there will be a net energy gain in the system as a result of the processing at node i.

Also of interest is comparing specific entropy reduction schemes in terms of their specific energy characteristics to arrive at a solution chosen for its superior energy performance in the sense of Equation (4.5) where the difference between the left-hand and right-hand sides is very large. In the initial work, the comparison is done by varying the software used to achieve entropy reduction on fixed *device* hardware and a fixed network setup for *device-server* communication.

Table 9 summarizes the quantities described above, their definitions, and their sources if outside the present work.

quantity	**symbol**	**definition**	**units**	**source**
processing energy	$\boldsymbol{p_e}$	$\boldsymbol{p_{e_{compression}}} + \boldsymbol{p_{e_{decompression}}}$	Joules/bit	[127, 128]
communication energy	$\boldsymbol{c_e}$	$\boldsymbol{c_e} \propto \boldsymbol{d}, \boldsymbol{n}$	Joules/bit	[8, 78, 138]
bitstream length	$\boldsymbol{d}$	|bitstream|	bits	
compression ratio	CR	$\boldsymbol{d_{original}} \div \boldsymbol{d_{compressed}}$		
energy footprint	$\boldsymbol{f_e}$	$\boldsymbol{f_e} \propto \sum_{i=0}^{n}(\boldsymbol{p_{e_i}} + \boldsymbol{c_{e_i}})$ (4.1)	Joules/bit	
nodes	$\boldsymbol{n}$	terminal or network node		
energy-efficient compression	$\boldsymbol{T}$	$\frac{\boldsymbol{c_{e_{i-1}}}}{\boldsymbol{CR}} \leq \boldsymbol{p_{e_i}}$ (4.5)	true/false	
common time unit	$\boldsymbol{u}$	refactoring for comparison	hours, minutes	[139]
bit rate	$\boldsymbol{b}$	number of bits per engagement	bits/$\boldsymbol{u}$	[8, 78]

Table 9: Defining Terms and Notation. Defining the quantities and symbols used for the energy analysis

4.5 Experiments & Results: 72% Savings

4.5.1 Experimental Setup

The broader context addressed is content delivered over the telecommunications network. The experimental goal was to derive the total energy footprint for a set of components within the context of information delivery over Internet-scale networks, compare their energy footprints, and compare the relative magnitude of the two main contributors to the energy footprint: 1) the energy used within the network for communication, and 2) the processing energy used by a particular component for reducing information entropy.

The current work builds on a processing energy dataset for a set of eight commonly available utilities that all achieve lossless file compression to a user-specified compression ratio using different algorithms. The dataset comes from a recent in-depth study by Dzhagaryan and Milenkovic [127, 128, 130] using an experimental setup described in more detail elsewhere [140]. All processing energy measurements were run on the same hardware under the same experimental conditions to measure their energy use in a controlled experiment. This experiment took a set of components performing the same task in which the hardware is fixed, and the software, algorithm, and implementation vary. The task was the compression of a stream approximately 66MB in length of concatenated input files of various file formats including structured and unstructured text, images and software binaries. The hardware platform used was a Texas Instruments Pandaboard designed for software development for mobile devices and smartphones and using a similar chipset to a number of popular mobile devices. Experiments and direct current measurements produced a dataset of measured compression and decompression energy used by each component in processing the input stream. Table 10 summarizes the key component features that were either controlled or varied in the experiments:

Applications	**Hardware**	**Operating System**	**Test Data**
gzip lzop pigz xz bzip2 pbzip2	Pandaboard from Texas Instruments (OMAP4430 chip set)	Linux (Linaro Ubuntu for ARM)	a 66MB concatenation of files in a variety of formats (text, executable code, bitmap image, structured text)

Table 10: Experiments. Processing energy measurement experimental setup for lossless compression utilities. For each utility, all available compression level settings were tested. From A. Milenkovic, A. Dzhagaryan, and M. Burtscher, *Performance and energy consumption of lossless compression/decompression utilities on mobile computing platforms,* in Proceedings of the 2013 IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems. IEEE Computer Society, 2013, pp. 254263. Used under fair use, 2015.

Sadler and Martonosi do not report absolute energy expenditure [113]; Barr and Asanović report results with likely lower in fidelity due to a much lower sampling rate [125, 126]. The key benefit to using the dataset generated by Dzhagaryan and Milenkovic is not the particular hardware it employs, or the specific energy-efficiency of that hardware, but rather the comprehensiveness of the dataset—the comprehensiveness with which a controlled experiment was extended to a large set of components serving a similar activity. In the present work, this is valuable to capture not the absolute energy expenditure but rather the relative energy expenditure for differing compression techniques and the range of energy use across these compression techniques.

The interest in obtaining processing energy measurements within the previous study was primarily

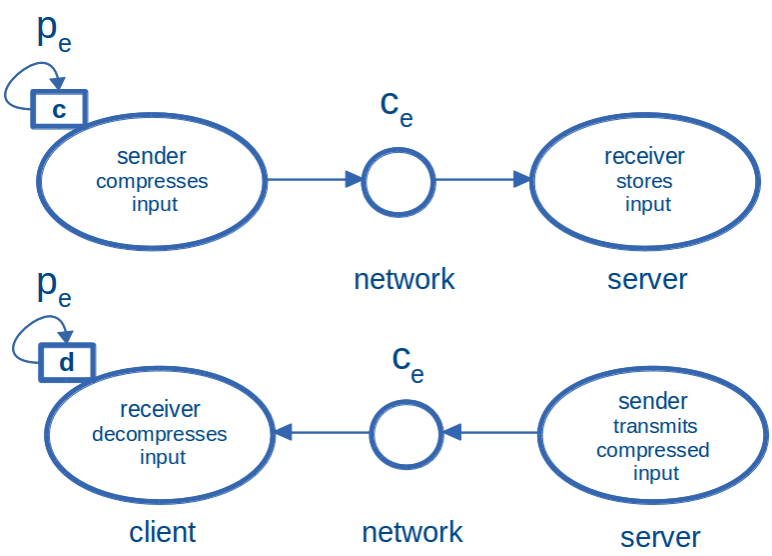


Figure 9: Experimental Scenarios Investigated. Experimental scenarios utilized in this study typical of streaming content from a server residing on the Internet, or cloud. In the upper scenario, a client such as a mobile device compresses content and transmits it to a server for storage, where it is stored as a compressed file. In the lower scenario, communication occurs in the opposite direction, from server to client. The client requests a compressed file, which the server retrieves and transmits. The processing task in the upper scenario is compression, indicated by the component labeled *c*; the processing task in the lower scenario is decompression. p_e and c_e show the sources of energy, processing energy and communication/transmission energy.

to determine which of n compression utilities achieved a certain compression ratio most efficiently. Here we take those results along the compression ratio continuum to ask what the gain is in a large-scale complex network in communication energy—assuming that the most processing-energy-efficient utilities discovered in the previous study are used on a similar device running as a peripheral device in a large network. In other words, the best-performing utilities from [127,128,130] are carried forward into the present analysis as a measure of the processing energy required to achieve a certain percentage reduction of the input file without loss of information content, or entropy reduction.

Two scenarios were considered in order to define the contributors to the energy footprint; these are illustrated in Figure 9. The first scenario is one in which an individual using a client device of

some kind uploads and transmits files to the cloud for storage; these files are first compressed on the client device using the compression utility under test and then transmitted to the cloud server sitting on the network somewhere, where they would be stored compressed (not read). The processing task is compression of the original data stream by the client-side device, and the experiments are intended to measure the relative contributions of processing energy and communication energy to the energy footprint for various compression utilities.

In the second scenario, an individual uses his or her device to access files stored on the cloud somewhere. The server transmits the requested compressed files to the client-side device, where they are decompressed for use.

These scenarios have sufficient similarity with the test conditions in [127, 128, 130] that their energy measurements for the Panda board can be used here for the client-side device. However, in both scenarios considered here, the file transmission is assumed to occur through the telecommunications network, and the results are reported for an average number of hops, 11.91, taken from a distribution from a large experiment on video traffic analysis [138]. This is a departure from the definition of transmission energy within the previous study in which a single hop to a desktop server was directly measured, when communicating through either an Ethernet or wireless router [127, 140].

The 54 components from Dzhagaryan and Milenkovic studies [127, 128, 130] were further analyzed here to obtain the energy footprint of each component, a measure of energy efficiency for which communication energy also plays a large role. The intention was both to define the separate contributions of processing energy and communication energy and to examine how this changes with compression ratio in order to determine the compression ratio's effect on energy footprint.

The energy footprint must be derived for the entire scenario or activity. It can be defined in general as the sum of energy footprints of all components used within the scenario, or

$$s_e = \sum_{i=1}^{m} f_e \tag{4.6}$$

where s_e is the energy footprint for the entire scenario, m is the set of components used within the scenario, and f_e is the energy footprint for a single component defined in Equation (4.1) as

$$f_e \propto \sum_{i=0}^{n} (p_e + c_e)$$

In both scenarios the server hardware and software need not be modeled because the server does no processing. In fact, the scenario can be captured via a single component, running the processing on the client, performing either compression or decompression, along with a single transmission c_e from client to server in Scenario 1 and from server to client in Scenario 2 through the network. These elements are pointed out in Figure 9. Thus the energy footprints for Scenario 1 and 2 are shown in Equation (4.7) and (4.8) respectively as

$$s_e = \sum_{i=1}^{1} (c_{e_{out}} + p_{e_c}) \tag{4.7}$$

$$s_e = \sum_{i=1}^{1} (c_{e_{in}} + p_{e_d}) \tag{4.8}$$

where there is a single node i, p_e is the energy used for compression in Scenario 1 and for decompression in Scenario 2, and c_e is the energy used for network transmission of the compressed file. Because the compression and decompression routines come in pairs in this domain[3], analysis can be further simplified into analyzing a single expression to derive a combined energy footprint for the component, combining the Equations (4.7) and (4.8) into the more general activity umbrella of client/server communication of compressed files,

[3]In this particular domain, there is no asymmetric compression such as possible with a scenario involving communication between two machines, each using the preferred method; here, the compression and decompression utilities come as pairs in that a gzipped file must be extracted by gzip.

$$f_e = \sum (c_e + p_{e_c} + p_{e_d}) \tag{4.9}$$

where, instead of scenario energy, f_e is a more general *component* energy footprint, derived from analyzing the scenarios and applicable to any component used to satisfy these two scenarios, p_{e_c} is the processing energy from compression, p_{e_d} is the processing energy used for decompression, and c_e is the energy used for transmitting the compressed file over the network.

In this, the p_e term from Equation (4.1) becomes energy from compression and decompression, and communication is counted once, according to a more general model in which the file is transmitted in compressed form, and the action performed on it is either compression by the sender or decompression by the receiver.

This treats the component as a combined compressor/decompressor and captures the desire to equally weight the energy utilized by the range of activities possible with this component: compression, decompression, and communication. The effect of an equal weighting is that decompression in some cases is underweighted given that it may in fact be a more common activity, at least for static files compressed infrequently and accessed frequently, such as web pages or feature-length movies. This requires a further analysis of the activity underlying the scenarios. However, this effect can be compensated for by attaching differing weights to p_{e_c} and p_{e_d}.

Using Equation (4.9), the energy footprint for each component in the pool of 54 combinations of application and compression level settings was calculated from a combination of sources, from known measured or estimated energy consumption of the components. The processing energy p_e was obtained from published measurements, along with the compression ratio achieved by each application for a variety of compression level settings on each application [127,128]. The transmission energy c_e over the network was estimated from a previous detailed analysis of per-bit energy on a 20-hop network for any-to-any communication within the global telecommunications network from [78] as presented in [8]. This per-bit network energy analysis provides a detailed analysis of

network energy and its constituent components in the year 2010, as well as technology-trend-based projections of changes in network energy use over time, from the time period of 2010–2025. The access network was not included in their analysis; this quantity is ignored in the estimations here. Only the effect that processing has within the telecommunications network is considered.

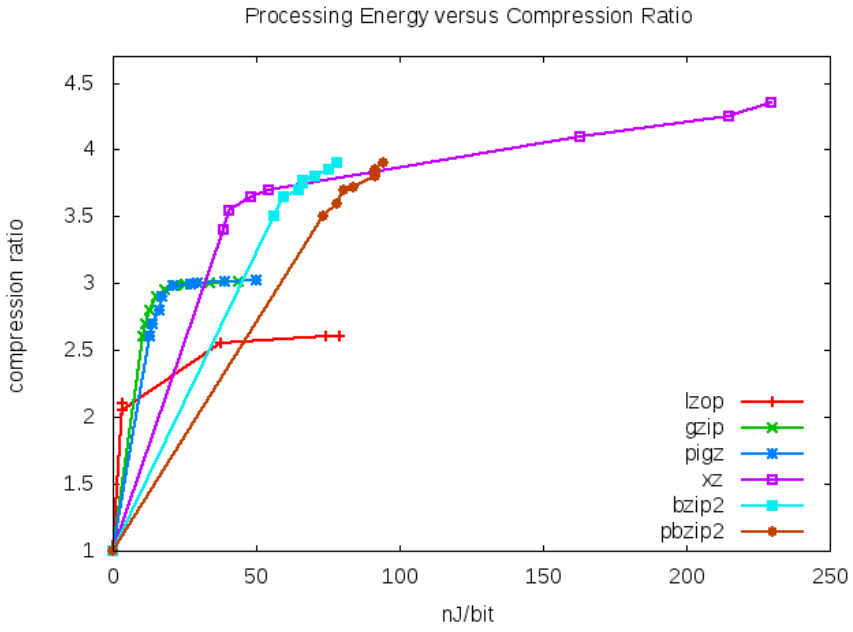
Given the network energy trends projection out to the year 2025 from a previous study [8, 78] it is possible to estimate changes in the energy footprint based on several factors influencing the rates of change in technologies. This was done holding the rate of equipment change constant at 15% based on a previous analysis [8] and assuming the same rate of energy efficiency improvement of the end user device as predicted for network equipment over the time period. From this a change in both total energy and in the relative contribution of processing energy and communication energy $\{f'_e, p'_e, c'_e\}$ was derived for the years 2015, 2020, and 2025.

4.5.2 Results

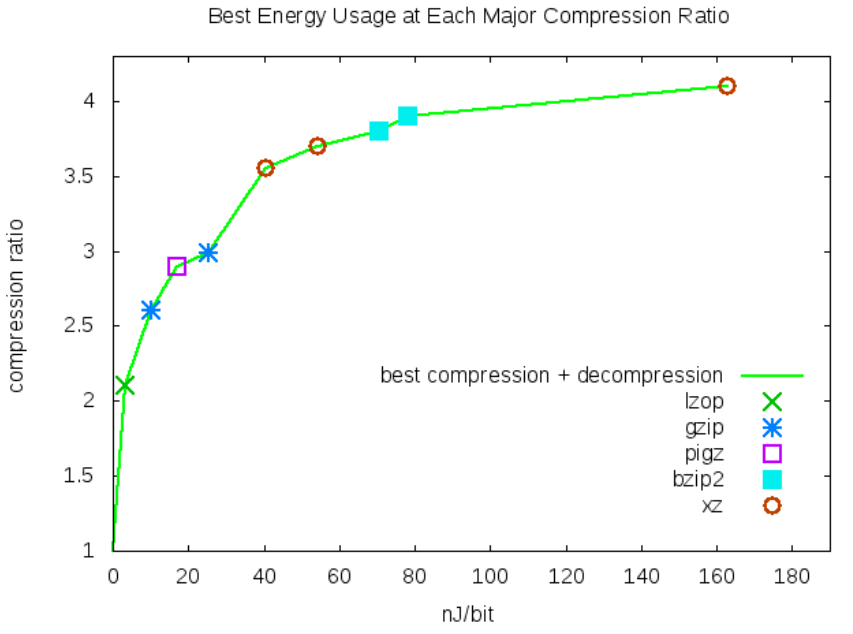
Per-Application Processing Energy

The processing energy results from [127, 128, 130] are plotted together and summarized in Figure 10. Processing energy is the sum of two separate quantities measured in the previous work, the nJ per bit utilized for decompression and compression. Each colored line represents a different application or compression utility, and points on the line represent the observed compression ratio achieved on the test input file by the different compression levels available for that application. A compression ratio of 1 utilizes no processing energy; it represents no compression or decompression of the file, just as would occur in transmission of the raw, uncompressed file.

Evident in Figure 10 is that the separate applications use different amounts of energy to achieve the same compression ratio and typically have differing profiles. As might be expected with mature algorithms and software, energy use increases with compression ratio; this is presumably due to



(a)



(b)

Figure 10: Processing Energy. Processing energy experimentally measured by Dzhagaryan and Milenkovic for the Panda board. Figure (a) shows all results from running six compression utilities, each at nine compression levels, as well as the compression ratio achieved by each utility in exchange for the energy expended. Figure (b) shows the nine most energy-efficient from (a) that were carried forward into the energy footprint analysis. The specific utility and compression level is named for each point.

the greater effort taken to compress the file to a smaller size. Also observable by the shapes of the curves, in many cases an application remains fairly efficient within a range of compression ratios but the compression gain, or energy efficiency, eventually decreases, showing large increases in energy use in achieving diminishing improvements in compression ratio.

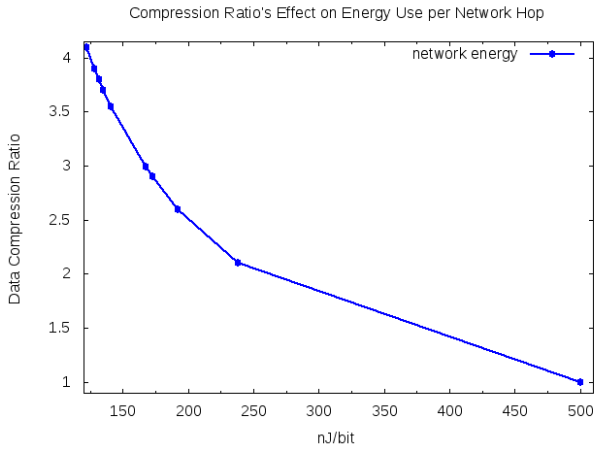
The nine best-performing utilities per compression ratio are shown on the right in Figure 10(b). The curve formed by these points represents the best obtainable processing energy for the given compression ratio. Only these are carried forward in the remaining graphs for simplicity, with processing energy hereon representing the best obtainable processing energy. Point clusters around major and minor compression ratios were resolved into a single selection.

No one compression utility stands out in terms of its energy performance. Comparing (a) to (b) in Figure 10, for a compression ratio of $\mathbf{2.5}$, lzop uses the least energy; up to $\mathbf{3.0}$, gzip is the most efficient, closely followed by pigz; around $\mathbf{3.5}$, xz is superior; as compression ratio reaches $\mathbf{4.0}$, bzip2 uses the least energy; beyond $\mathbf{4.0}$, the only option is xz.

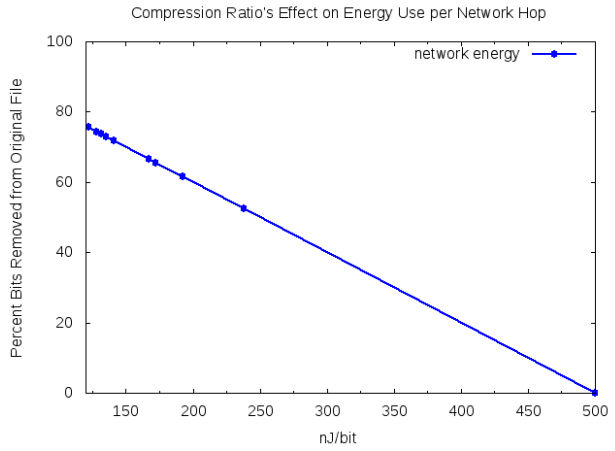
Communication Energy for 1 Hop

Whereas processing energy increases with compression ratio, an opposite effect holds for communication energy, as evident in Figure 11(a). This figure represents per-bit transmission energy for one hop within a network, as defined by Baliga [8,78], as it varies with compression ratio and with percent file reduction, which gives a linearly increasing y axis. Given the total energy within a 20-hop network presented as an energy per bit in [8], Figure 11 shows the average contribution of one hop to that total per-bit energy. This is then factored by the percent reduction in the transmission length that the compression ratio achieves to represent the reduced transmission size scaled down to the per bit regime under which all quantities are compared.

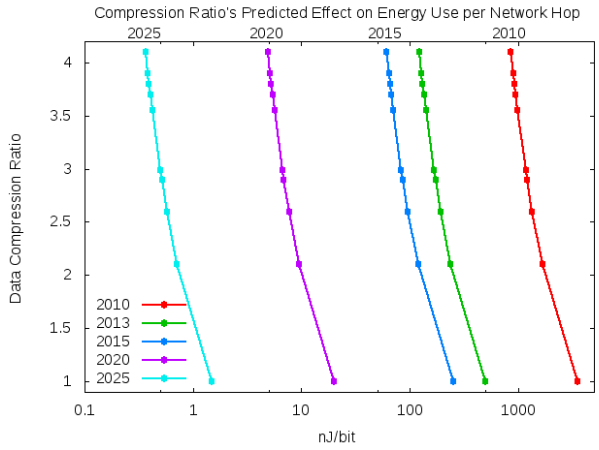
Figure 11(a) uses network per-bit energy for 2013 from projections made in 2009 [8, 78]. It does



(a)



(b)



(c)

Figure 11: Network Energy for One Hop. Energy consumed per bit as a result of network transmission for one hop. Shown as a function of compression ratio (CR) in (a) and as a percentage file reduction in (b); transmission length is 1/CR $\times$ raw or uncompressed file size. Figure (c) shows network per-hop energy use as a function of Compression Ratio at five-year increments from projected trends, as well as replicating the 2013 curve shown in (a). x-axis plotted on a log scale. This graph shows the order-of-magnitude reduction in network energy per five years as predicted by the model of Baliga. Network energy reduction is evident as the dominant effect by comparing the slope of the lines, attributable to Compression Ratio, to the relative x-position of the lines, attributable to network energy consumption.

this to align network energy to the compression study, which was conducted in 2013 [130]. Putting these in the same time period permits their combination to derive the energy footprint for each component.

Network Energy Trend Analysis

Figure 11(c) shows the one-hop network energy for 2013 along with projections for 2015, 2020, and 2025. This helps to illustrate the effect of predicted energy efficiency gains of roughly an order of magnitude every five years. These gains are predicted by technology trends analyzed by Tucker [8, 78].

Tucker [8] discusses two main factors behind this projected reduction in network energy per bit: 1) a combination of 15% per year efficiency gain in equipment energy use, as well as 2) increasing bit per second access rate combined with the fixed energy use of passive optical networks (PONs). Because PONs use a fixed amount of energy regardless of bit rate, their increased adoption means that the rise in network access rate decreases network energy use much faster than equipment efficiency gains alone [8]. If the same rate of equipment efficiency gains of 15% is assumed on all equipment, including peripheral devices, then the rate of efficiency gain in component processing is lower than the rate of efficiency gain in the network. This difference in improvement rate is examined further to see what impact it has on energy footprint.

4.5.3 Energy Footprint

The communication energy for each component was derived for 11.91 hops, the average number of hops found in a large study of Internet streaming video [138], assuming the energy consumption per hop from the top-down analysis of the telecommunications network [78]. Transmission length was the original transmission factored by the percent reduction in signal given by the compression

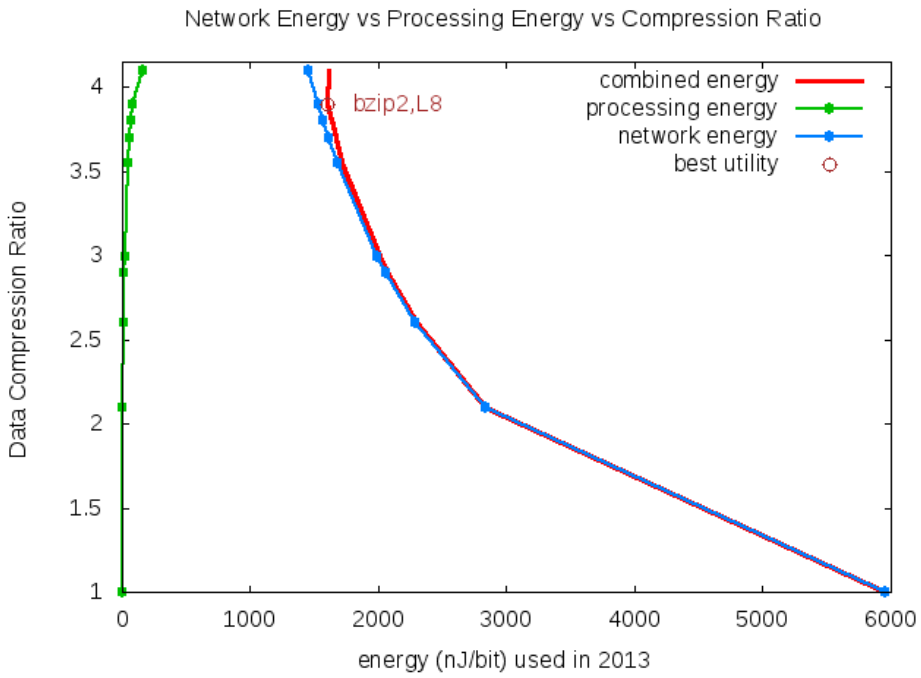


Figure 12: Energy Footprint. Energy footprint in red and its constituents in blue and green for the best utilities at each compression level. Network transmission over the full average number of hops (11.91). Processing energy and communication energy also graphed to show their relative magnitudes; communication energy labeled as network energy. The utility with the lowest combined energy is circled as well as the compression level used. Results for the year 2013.

ratio achieved by the component. Arriving at the communication energy per bit, this was added to the processing energy per bit to construct the energy footprint for each component. Just those for the best-performing utilities were graphed.

In Figure 12 the *energy footprint* of the best-performing utilities from Figure 10(b) as the sum of communication energy and processing energy are shown together as they vary with compression ratio. The energy footprint is shown as the sum of processing energy and communication energy; these separate contributors are also graphed to show their relative contribution. Immediately apparent is the correspondence between communication energy and the energy footprint. The contribution of processing energy to total energy is near zero for compression ratios below 3, as evidenced by the overlap of the combined energy curve with the network energy curve: the predominant contribution is communication or network energy, measuring two orders of magnitude higher than processing energy.

Energy footprint selects one component from the nine candidates at differing compression ratios. This one component has the best energy footprint. The application that gave the second-highest compression ratio, bzip2 at Compression Level 8, performs slightly better than the application with the highest compression ratio (xz at Compression Level 5) due to the effect of higher processing energy on the energy footprint. For lower compression ratios, processing energy has little effect on the energy footprint because of the order-of-magnitude higher cost of transmitting bits through the network versus compressing them on the device.

Another observation of Figure 12 is insightful for a cost/benefit analysis of the value of compression. It shows that the contributed cost of compressing the file is very small or negligible in all nine cases, shown in the small distance between the energy footprint curve in red and the network energy curve in blue, as well as by the x position of the green curve representing processing energy alone. This may not be borne out in the future, as suggested by the next set of figures.

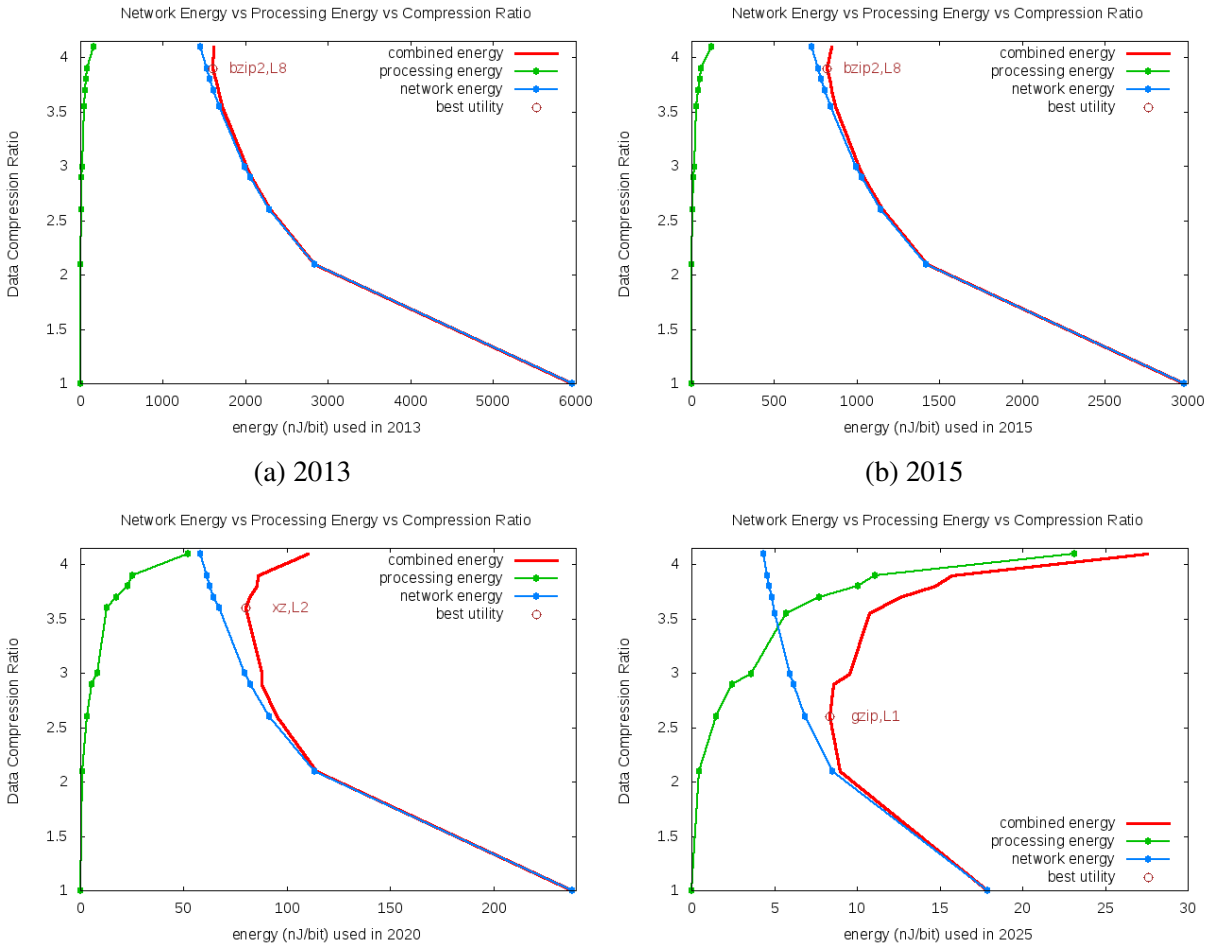
Despite the negligible cost of compression, evident in Figure 12 is the large benefit of compression on energy footprint. For instance, using bzip2 Level 8 to gain a compression ratio of **3.9** yields a savings of over 400 nJ/bit over the application giving a compression ratio of 3, and a 4,400 nJ/bit savings over uncompressed transmission (CR=1). For both 2013 and the projections for 2015, this represents an energy consumption increase at the client device of less than 2% to run bzip2 Level 8 in exchange for an energy consumption decrease of over 74% in the network to transmit the compressed file, or a net energy savings of 72% over no compression.

4.5.4 Energy Footprint Trend Analysis

Future projected energy footprints for the years 2015, 2020, and 2025 were also constructed based on forward projections of network and device energy improvements and are shown along with 2013 in Figure 13. The rise of the importance of processing energy and reduction of importance in communication energy are evident in comparison of the distance from the communication energy to the processing energy from 2015 to 2025. Whereas 2015 looks similar to 2013, 2020 shows a greater contribution from processing energy, and 2025 shows processing energy dominating for higher compression ratios. While these are long-term predictions based on today's improvements that do not take into account future innovation, the trend suggests that asymmetric network energy versus device energy improvements bring communication energy down to processing energy levels over the next decade.

Leaf cost versus Network Cost Analysis

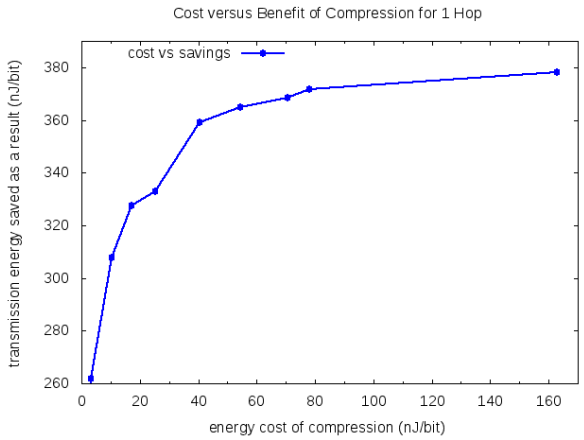
Would you spend $3 to receive $262 back a few hundred milliseconds later...and again receive $262 eleven more times? This is essentially what entropy reduction at the leaf node does, albeit in different units. Take for instance the leftmost point in Figure 14a associated with the compression utility lzop run at Compression Level 1. It costs 3.30 nJ/bit at the leaf and earns 262 nJ/bit



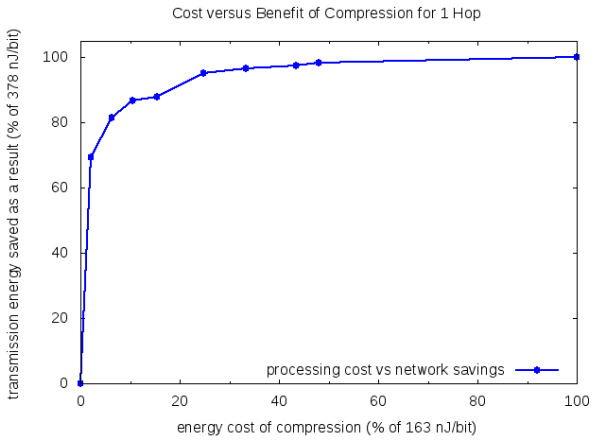
(a) 2013 (b) 2015

(c) 2020 (d) 2025

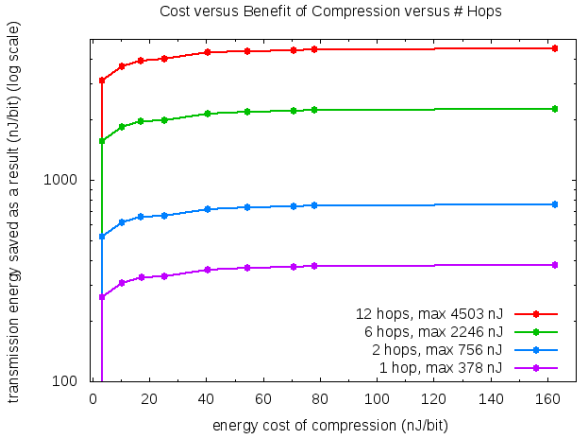
Figure 13: Changes in Energy Footprint Over Time. Energy footprint in red and its constituents in blue and green for the best utilities at each compression level for years 2013, 2015, 2020, 2025. Network transmission over the full average number of hops (11.91). Processing energy and communication energy also graphed to show their relative magnitudes and how they change; communication energy labeled as network energy. The utility with the lowest combined energy is circled as well as the compression level used. The more-rapid decrease in network energy versus equipment energy consumption brings communication energy and processing energy into the same range by 2025. The increasing contribution from processing energy–that is decreasing also, but at a slower rate—brings the choice of best compression ratio lower.



(a) nJ/bit



(b)



(c)

Figure 14: Leaf Node/Transmission Hop Energy Tradeoff. Amount of processing energy spent at the leaf nodes for compression versus amount of communication energy saved within a single hop within the network as a result, shown as a numeric value in (a) and as a percentage of the maximum values from the set in (b). Figure (a) shows that the range is from 3.3 nJ/bit spent at the leaf to save 262 nJ/bit in the network, to 163 nJ/bit processing energy spent at the leaf to save 378 nJ/bit in communication energy saved. Evident in (b) is the narrow range of savings for a wide range of utilities, and that the compression utility that spends 2% of the maximum spent saves 69% of the maximum saved. Figure (c) shows the same savings due to compression for 1 hop, half, and the full average network transmission distance. Log scale used for communication energy. The energy saved is quite large for 11.91 hops: between 3,120 nJ/bit and 4,503 nJ/bit.

back within the network within a single hop, and it accumulates a total of 3,120 nJ/bit over the average transmission distance of 11.91 hops. This is shown for the full range of best-performing compression utilities in Figure 14. The cost in energy at the leaf node spent in processing the bit stream to reduce entropy is compared directly to the savings in network energy that results from the subsequently smaller transmission length. Figure 14 show this direct apples-to-apples tradeoff of nJ/bit at the periphery for nJ/bit within the network. Figure 14(a) shows the numeric value of the energy spent versus saved; Figure (b) shows this instead as a percentage/percentage quantity, where the percentage of energy spent at the leaf node versus the maximum processing energy is plotted against the percentage of energy saved within the network versus the maximum communication energy. What these two figures show is the energy saved by information entropy reduction at the leaf nodes, which is evident in the difference in numeric value of energy along the y axis versus the x axis. The points along the curve express the nJ/bit energy expenditure at the leaf node processing the data stream as the sum of compression and decompression energy, compared to the nJ/bit energy expenditure transmitting the same, compressed data stream one hop within the telecommunications network. In practice, either compression or decompression will be performed, not both, increasing the savings further. However, as explained earlier in this section, the energy footprint combines the processing energy for compression and decompression, intended to capture the full cost of compression.

Figure 14(a) and (b) demonstrate the savings incurred just from one hop within the network; this is retabulated for the full 11.91 average network hops in Figure (c). For the full network transmission, the energy saved by information entropy reduction at the leaf nodes is a large sum, between 3,120–4,503 nJ/bit.

One interesting feature of the energy benefits of compression applications on peripheral devices—at least for the ones used in this study—is shown in the slope of the percentage curve in Figure 12. Even a small degree of compression captures most of the savings from compression. The

compression utility that gave the minimum compression ratio among those considered was lzop at Compression Level 1, achieving a compression ratio of $\mathbf{2.1}$; it spends only 2% ($\mathbf{3.3}$ nJ/bit) of the energy of the compression utility with the highest achieved compression ratio, xz at Compression Level 9, yet it reaps 69% of the benefit, in communication energy over the telecommunications network. This is the joint effect of processing energy at the leaf node interacting with communication energy saved due to compression ratio achieved for transmissions within the network...simplified into a simple exchange of nanoJoules at the leaf node exchanged for nanoJoules in the network. The more complex story is more evident in previous figures showing the full energy footprint and its relative components, Figures 6–9.

The general downward trend predicted in the cost of transmitting bits versus processing bits is shown in Table 11 showing its historical trend in observed results in 2003 versus 2013, and continuing with a predictive analysis for 2020[4]. The change from observed values in 2013 into the projected space of the analysis is graphed in Figure 15a. Built into this trend is the assumption that the same rate of energy efficiency improvement of the end user device holds as that predicted for network equipment over the time period [8]; actual improvement could be higher, with the result of flattening this curve. The table shows the observations from 2003 and 2013, moving to predicted trend in the difference of the energy cost of transmitting versus processing one bit in the ICT. Transmission cost is shown for 1 hop as well as the typical number of hops from server to client (and half the average of 20 hops from leaf to leaf [8, 141]). Predicted values are based on: 1) transmission energy prediction by Baliga [8,78]; 2) processing energy prediction based on 2013 mobile device-class hardware from experiments, projected along with typical 15% per year energy efficiency improvement from Tucker [8].

Figure 15b shows the downward trend over time in total energy consumed by the entire encoding/ transmission /decoding chain from the general scenario that this section deals with, as depicted in

[4]The ratio reported here was derived from *16-baseline* optimized compression code versus *no compression* in Figure 16 of Barr and Asanović [126].

(a) ratio of transmission energy to processing energy

(b) nanoJoules per bit

Figure 15: Total Energy Trend. The observed-in-2013 and the predicted future trend in the cost of transmitting bits through one network hop versus processing them on a typical peripheral processor, an OMAP4430, a common, low-end chipset commonly used in peripheral devices. Figure (a) shows the ratio of transmission energy to processing energy required to encode and decode the input file. Figure (b) shows the total cost of processing (coding, decoding) and transmitting one bit through the network at different points in time, and as it is predicted to change over time. The same assumptions hold for both graphs; however, the y axis is expressed in absolute energy in nanoJoules per bit rather than as a ratio.

Figure 9.

year	processing	1-hop transmission	full transmission	source
2003	1	350	NR	[126]
2013	1	6.4	76.2	
2015	1	4.4	52.7	
2020	1	.8	9.5	

Table 11: Transmission vs Processing Energy Trend. Observed values for 2003 & 2013, moving to predicted trend in the difference of the energy cost of transmitting versus processing one bit in the ICT. Transmission cost is shown for one hop as well as the typical number of hops from server to client. For 2003, the transmission cost was measured specifically for wireless transmission; for the rest, cost is reported based on average Internet transmission cost across various transmission media[4].

4.6 Discussion

The energy impact of *entropy reduction* in the global telecommunications network of was examined for content traveling from the outer periphery of the network towards the center as an example of the potential for positive impact on global energy by the actions of applications running at the network leaf nodes. The proposition that lossless data compression, or source coding, *saves energy* was examined for contemporary peripheral device hardware and telecommunications networks. The two chief tradeoffs in energy consumption due to compression were examined, and their magnitudes were compared. They are: a) compressed data with less redundant information cost more energy to produce from raw data, but b) the resulting data stream transmits at lower energy from its shorter length. This problem was framed as comparing the relative energy lost at the leaf node to the energy gained in the network. Two cases were examined: one in which the

sender is a peripheral device on the global telecommunications network, performing some degree of source coding before transmission, the other in which the receiver is a peripheral device on the global telecommunications network, performing some degree of decompression after transmission. In both cases the other node was assumed to be a server sitting centrally on the network, at an average transmission distance of 11.91 hops.

The space of contemporary lossless file compression schemes and compression ratios was explored to address this question using detailed energy measurements from a previous study, in which the Joules per bit consumed by a Texas Instruments OMAP4430 chipset were measured when running each scheme. The questions asked in the present work were 1) what does this space look like now, and five, ten, fifteen years from now; 2) where is the optimal point within this space, and what are the relative costs of processing and communication at that point; 3) does that point change over time, based on projections in new technology's effect on network energy use.

A full two-dimensional comparison of processing energy at the leaf node to communication energy within the network for the set of compression schemes at different compression ratios achieved was constructed and presented as the variation with compression ratio in the best achievable total energy footprint for compression/decompression for contemporary peripheral devices and telecommunication networks.

This analysis was carried forward to 2015, 2020, and 2025 to examine the competing trends that may change the effect of compression ratio on energy consumption within the global telecommunications network.

In 2013, a significant reduction in energy cost was observed from compression, and significant energy savings were observed within a wide range of compression ratios from $\mathbf{2.5}$-$\mathbf{3.9}$. The peak energy savings was found at a compression ratio of $\mathbf{3.9}$, where the total energy footprint was 26% that of uncompressed bitstreams due to improvement in communication energy. At higher

compression ratios than $\mathbf{3.9}$, the much greater processing energy involved reduced total energy savings, despite lower communication cost. Results overall suggest that the small cost of processing attributable to relatively high compression ratios ($\mathbf{2.5}$-$\mathbf{3.9}$) is more than compensated-for by the large savings in communication energy, given a fixed cost per bit to transmit the datastream through a large-scale telecommunications network such as the Internet.

The total positive energy effect of compression per bit ranges from 3,116 nJ/bit to 4,340 nJ/bit, after accounting for extra energy spent at the leaf node processing the input before transmission. For 2% of the processing cost of the most energy-expensive application, the least expensive application achieved 69% of the benefit in network transmissions energy reduction. This suggests that the benefit of entropy reduction via lossless compression sees diminishing returns, as the processing cost increases faster than the compression ratio decreases. This may be further exacerbated by effects present in the trend-on-trend analysis for 10- and 15-year horizons.

The cumulative energy impact of a global policy of using the compression utility with the smallest energy footprint (1605 nJ/bit) was significant, a 2% rise in leaf node energy in exchange for a cumulative savings of 74% system-wide, leaf nodes included. Transmission through the network was found to dominate the energy footprint in 2013; this effect continues in the near-term, based on projections for 2015. *Communication energy per bit was found to be two orders of magnitude larger than the device processing energy required for compression of the input stream on a peripheral network device*.

Communication energy is predicted to drop faster than processing energy, based on predictions and future projections from a previous study, due to the effect of scale invariance of network components such as passive optical networks that increase in efficiency as access bit rate continues to rise. If true, the relative cost of transmission drops steadily and at a faster rate than processing energy, bringing the communication energy down one order of magnitude every five years. With these assumptions, the communication energy fraction reduces by one, then two orders of magnitude in

2020 and 2025, bringing the best-case total energy footprint down under 10nJ/bit while increasing processing energy's contribution to nearly one half total energy at compression rates higher than 3.5:1. There are problems with any prediction, and this one ignores many potential means by which energy savings could be increased. However, the main contribution of initial work is to suggest that a) processing energy used for entropy reduction at peripheral devices at leaf nodes has a demonstrable net positive effect on communication energy and total energy use in global-scale telecommunications networks, b) the effect is currently dominated by the network component, but c) that a simple policy of always applying a high compression ratio to reduce communication energy has its limits even today, and may have much less benefit five and ten years from now.

4.7 Energy Savings within Specific Networks

This section assesses the specific energy that would result from a similar proposition but with slightly different assumptions than those of Figure 9 in 4.5. The content is assumed to be the size of the Google Mobile Home web page, 1.8MB, and may originate either from Google or from any location within the network: it could be from any source on the network, including a leaf node; the assessment is blind to the origin: it is simply somewhere outside the network of interest, or *extra-network source*. More motivation of the problem definition can be found in Chapter 3, as can a rigorous definition of the networks used, the hardware energy profiles, and the other aspects of the experimental setup that are not specific to entropy reduction itself. This allows the replacement of the network model in the previous results with particular networks—albeit ones that do not represent the full transmission path from source to recipient. The value of this is that the intra network path, and the specific hardware of each node, and the topology of nodes, are explicitly represented in this assessment, unlike in Section 4.5 where the actual path over the Internet is not explicitly modeled: it is not known, and is taken as an average path length from a distribution from a large study of video traffic, and an average hardware cost of transmission energy per bit.

To assess the energy savings associated with entropy reduction within each network over the set of sample networks, we take the example of transmission of a web page the size of the Google mobile website home page as a sample of content that is typically transmitted in raw, uncompressed form, and we compare the cost of transmitting the raw file to one that is losslessly compressed by the server and transmitted to all receivers, which decompress the file before reading it. The cost for transmitting the uncompressed page via unicast can be compared with the unicast transmission cost for the compressed page.

This calculation assumes that the most energy-efficient compression utility to use under this scenario is the best-compression utility for the representative mobile hardware, that is xz at Compression Level 5 experimentally measured by Dzhagaryan and Milenkovic as seen in Figure 10 [127, 128, 130], for the reason that the cost of *decompression* for xz is comparable[5], and the difference in *compression cost* at the source between the two utilities[5] is more than justifiable in the scenarios considered here—rendered insignificant, really—when the number of recipients is not one as it was in the prior analyses, but 100,000 to 1 million receivers.

Table 12 shows the energy difference between unicast transmission of the uncompressed and compressed web page. This table reports the savings for transmission only, excluding the energy costs of the processing required to perform the entropy reduction—within the scenario, that is the cost of compressing the Google Mobile Home web page on a server at Google each time it is revised, plus the cost of decompressing the compressed transmitted page at each receiver. The energy saved is expressed both in Joules and as a percent savings over uncompressed transmission.

As the consistency within the final column of Table 12 shows, the linearity of energy cost with bytelength causes the cost of transmission of the compressed content to be a fixed fraction of the cost of transmission of the content uncompressed; however, the total amount of energy saved

[5]The decompression energy is actually lower: 6.58 for xz at compression level 5 vs. 15.63 nJ/bit for bzip2 at compression level 8; the compression energy is higher: 156.25 nJ/bit for xz CL 5 versus 62.5 nJ/bit for bzip 2 CL 8.

Table 12: Comparing Full Cost to Entropy-Reduced Cost. A side-by-side comparison of the transmission cost of full versus compressed unicast transmission of 100K requests for a 1.8MB web page. Compression ratio = 4.1

Network	**uncompressed cost (J)**	**compressed cost (J)**	**savings (J)**	**savings (%)**
AS 1239	7,057,081	1,721,239	5,335,842	75.6
AS 224	4,597,382	1,121,313	3,476,069	75.6
AS 3549	6,025,916	1,469,736	4,556,181	75.6
AS 59	1,247,162	304,186	942,976	75.6
AS 2914	6,533,060	1,593,429	4,939,631	75.6
AS 3356	5,442,846	1,327,524	4,115,323	75.6
AS 3292	2,524,117	615,638	1,908,478	75.6
AS 109	6,626,477	1,616,214	5,010,263	75.6
AS 680	2,674,362	652,283	2,022,079	75.6
AS 3356.1	6,495,726	1,584,323	4,911,402	75.6
AS 680.1	1,991,771	485,798	1,505,973	75.6

is dependent on the network topology, as can be seen in variation among networks within the second-to-last column of Table 12, which reports savings in Joules.

Chapter 5

Conservation of Energy through Multicast Transmission

5.1 Introduction

In the cases considered in this chapter, the opportunity to conserve bits that arises from a different aspect of the system: the set of redundant bits that come from exploiting the commonality among separate individuals' simultaneous use of the network. Commonality along with simultaneity of requests for a particular URL or data object allows for such things as simultaneous broadcast of the desired information to many individuals, or multicast to a subscriber pool rather than to every device. This chapter examines the energy savings of harvesting this form of inter-message, inter-task redundancy to reduce the number of simultaneoush duplicate transmissions to receivers—when the individual transmission paths to separate simultaneous receivers overlap to some extent. These savings derive from using an alternative transmission mechanism to unicast: here, from a multicast form of communication.

The chapter does so from several different vantage points or series of assumptions, because there appears to be no one fixed way to answer the question of how applicable in reality multicast is, and the impact varies with scale; therefore, the question of multicast energy savings is addressed from

the perspective of both specific networks and of the general Internet structure as a whole; for the Internet structure, where there is again disagreement, two separate characterizations are employed in deriving different measures of multicast's energy efficiency; all efficiencies are reported relative to unicast transmission of one packet addressed to each recipient, by far the most common form of delivery in use today.

Unicast is the communication protocol assumed throughout the entropy reduction results presented in Chapter 4. Unicast transmission is a style within which the server communicates to each client through a unique path, there are other methods that are also commonly used. Multicast is a different style of communication, more similar to broadcast; broadcast used for television broadcasts and live news broadcasts does simultaneous transmission to all receivers akin in the analog electronics realm to a transmitter using an omnidirectional antenna to transmit simultaneously to $\boldsymbol{n}$ receivers. Multicast does so to a subset of all $\boldsymbol{n}$ receivers, such as a subscriber list. Multicast implemented over a data network minimizes the network traffic of unicast associated with serving all receivers independently. It does so by transmitting to all receivers via a treelike message duplication pattern. That is, where two or more messages share a path, all such transmissions are satisfied with a single message; where transmission paths diverge, the message is replicated and sent to each child node. Multicast minimizes the network traffic associated with broadcast by transmitting to just those receivers that have explicitly requested the content. This results in a transmission to the receivers via a treelike message duplication pattern over the unicast transmission medium. That is, where two or more messages share a path, all such transmissions are satisfied with a single message; where transmission paths diverge, the message is replicated and sent along each branch to each downstream node.

5.2 Background and Related Work

Multicast and content distribution, via content delivery networks or proxy servers, network caching, web caching, and other means are well-studied and frequently used means to improve network performance. While various metrics such as latency, node throughput, and server load are often considered, the amount of power they draw is less often considered.

Multicast style of communication has a long history of proponents and development as a more efficient method of communicating to multiple recipients within telecommunications networks emerged. Multicast publications started to pique interest within IT realms in the 1990s, such as the work of Deering [142]. Although multicast has been consistently proposed as a more efficient method of communication within data networks, there are few studies specifically comparing its energy consequences to other communication methods. Work has focused on generating the transmission tree efficiently, either computed from a centralized point with full network knowledge using some variant of Steiner trees, or in a distributed way out in the network using processing resources available at network nodes; these strategies are condensed and discussed by Lun et al in formulating a decentralized approach [143]. Meiling et al proposed managing the distributed microresources within a large power grid more efficiently using multicast and show not energy costs but link contention and average message delay for their methods over national and Europe-wide networks [144]. Valcarenghi and Castoldi report energy impacts of multicast on energy savings for invoking sleep mode within passive optical networks (PONs) for a limited set of considered conditions [145]. Fantini et al look at a variant of multicast in LTE cellular networks, comparing the total emitted RF power of the transmitting nodes comparing Type 1 relays to a multicast cooperative scheme using Type 2 relays [146].

5.3 Delivering the New York Times via Multicast

What if subscribers to the New York Times, the highest-circulation newspaper in the U.S., agreed to have it electronically delivered in the morning, and it were multicast to them over the Internet? To obtain a quick, back-of-the-envelope-type assessment of the energy efficiency of multicast delivery within the Internet, this work utilizes prior work by Dolev on the structure of the Internet [70]. A literature review describing a wide range of scholarship that has gone to plying and theorizing the structure of the Internet is presented on page 24 in Section 3.1.2.

The details of Dolev's work that are relevant to the current purposes are summarized here. Dolev analyzed real Internet data from Burch and Cheswick [147] and other sources to assess the underlying structure to support something such as Internet provider multicast structures within the Internet. Probing the structure explicitly, outward from a central, highly connected node at the network core they find a small number of rings of high degree nodes, and a strong power law relationship overall of node degree or connectivity to frequency. Leaf nodes lie on average 5–7.4 hops from a low degree originator node [70], and that most nodes were accessible within 3-5 hops from this structure for a low degree sender. Dolev and Shavitt also found a strong tendency for 16:1 relationship of highly connected nodes to leaf nodes when scanning the rings of structure outward from a highly connected central node, with one high degree node for each 16 leaf nodes. Node degree for one 10,000-node tree connectivity graph in their study ranged from 150 to 2, with nonoutliers obeying the power law falling in the range of 79 to 2. These numbers were obtained for the Internet structure in general as the support structure for multicast transmission: from multicast-style trees cut from the larger network graph.

Given Burch and Cheswick's map of Internet connectivity from 1999 [147], Dolev et al examined not actual multicast patterns of transmission but rather the shortest path observed in this connectivity for subset trees cut from the overall graph [70]. They found that, in the case of interest here,

a multicast tree intended to serve a million peripheral devices, the expected number of high degree nodes is 62,500. This can be approximated as a tree with arity 16 and depth of 5 hops from a central content server node. While the underlying structure of Internet connectivity in general shows a power law relationship between the rungs of the tree, this delivery structure can be viewed as a subtree of the overall connectivity and as the necessary and sufficient amount of resources at each outward ring of the connectivity. To reach the majority of leaf nodes within the Internet from a low degree node, Dolev et al observed 5-7 hops; assuming that the source is not a low degree node but rather a well-connected central node—for which connectivity in their study was in the range of degree 79–150—this reduces the distance from root to connected nodes by up to four hops. Therefore, reaching the majority of leaf nodes within 5 hops appears well within the connectivity for a central node.

To assess the relative cost of multicast routing over unicast, a multicast communication pattern is modeled and analyzed here based on the relevant data from Dolev et al [70] combined with an assumed scenario. The question asked here is, given the 1.1M subscribers to the digital edition of the New York Times, how can the basic structure above from observations of the Internet be utilized to achieve an efficient multicast transmission, via an overlay or dedicated resources or other means, to reach all 1.1M subscribers in the most energy-efficient way?

Given the expected number of high degree nodes of 62,500 from Dolev et al is sufficient to reach one million leaf nodes, a connectivity that satisfies the parameters can be approximated as a multicast tree with arity 16 and necessary depth of 5 hops from a central server node with degree ≥ 16. The maximum capacity of this tree is close to but just under the number of subscribers: 1.048M. This is sufficient for the present purposes of estimating energy roughly; however, to ground the results fully in terms of the case of delivering the New York Times, the calculations are done assuming two servers rather than one, so that the remaining subscribers are also accounted for in the energy assessment. Assuming here that the multicast tree is an intentional energy-savings measure

employed by the publisher or distributor, adding a second server and a second depth-6 tree is more efficient in terms of transmission energy than adding another hop to the first tree, because the geometric growth of the tree with depth outsizes it well beyond the newspaper subscriber pool at depth 7.

A second source can be added to reach the number of additional subscribers beyond the maximum capacity of this tree. With two such trees, the second source tree is used to reach the remaining subscribers beyond 1.048 million more efficiently than adding another hop to the first tree. This model is sufficient to obtain the number of hops per leaf node as well as a number of shared legs of the transmission path for tree-based replication of the content rather than unique transmissions for each client, or client-address-based replication of the content.

In practice, multicast trees may observe similar power laws to the more general Internet connectivity trees; however, in the interest in lining up the model with published results, the 16ary multicast tree achieves the expected ratio of leaf nodes to high degree nodes as well as the number of interior high degree nodes observed by Dolev et al. The effect may be to underestimate the number of required transmissions, since higher fan-out in the initial nodes of the tree in theory reaches more potential destinations yet with fewer redundant paths. Reaching all subscribers—even though all are known *a priori*—with a transmission mechanism not owned by the the owner or publisher of the media content may not allow for optimization of the multicast structure, regardless of this knowledge, and thus a more generalized transmission mechanism to any receiver is required. A more generalized transmission mechanism has a high-degree root node and a series of lower-degree intermediate nodes; however, aiming for the observed 62,500 high-degree nodes within the multicast tree still constrains this to *measured energy* similar to the one used here that is largely based on number of transmission paths. Given this structure of 16ary nodes, and ignoring the access network, it is possible to reach n receivers in $log_{16}(n)$ hops within a 16ary tree transmission pattern.

5.3.1 Energy Calculation Results: 58-91% Savings

If a similar multicast structure to the one described in Section 5.3 were used to deliver the digital version of the New York Times newspaper, which is the most popular newspaper in the digital realm, with a weekday subscription base of 1.1M readers, it is possible to reach nearly all of these readers, 1.049M, within 5 multicast hops, $log_{16}(1.049M)$. Dividing the pool of subscribers into two pools, each pool subscribing to one of two sources, provides full coverage of the readership within the 5 network hops estimated using Dolev et al's Internet analysis.

Delivering the New York Times newspaper electronically is an example of static, repeated daily transmissions to a large subscriber base. This type of transmission pattern is amenable to either unicast or multicast transmission: multicast synchronous delivery, for instance, could be achieved by running an application on the subscriber device that handles a scheduled delivery multicast to all 1.1M subscribers [148].

A multicast transmission from two separate servers to all subscribers is possible with 16ary multicast communication using two separate trees, each serving a maximum of 16^5 (1.048M) subscribers; the 1.1M subscribers are leaf nodes of one of the trees at tree depth 6.

The resultant multicast path length is 58% shorter than the average unicast path length from network-central servers of 11.91 hops reported in a large study of streaming video traffic [138] and 75% shorter than the 20-hop average path length reported by Tucker for global telecommunications traffic from any sender to any receiver within the network [8]. Assuming direct proportionality of energy consumption to the number of network hops through which a transmission travels, multicast from a New York Times newspaper server thus represents a 58% energy savings over a more traditional client/server unicast transmission mechanism. This speaks to the dramatic potential savings from a policy of multicast over unicast. This translates to a savings of 931–3,455 nJ/bit when combined with the range of entropy reduction available from the applications examined.

Communication energy was found to be far lower using a multicast style of communication over the original assumption of unicast delivery more typical of client/server communication. A multicast transmission pattern from two separate servers to all subscribers for content such as the New York Times daily edition saves 58% of the nJ/bit transmission costs via fewer hops traversed from server to leaf nodes—5 hops instead of 11.91 average for client/server style communication via unicast. Using the same basis for relating bit-hops to Watts to power plants in Chapter 4, this difference in usage translates to a savings of 931–3,455 nJ/bit when combined with the range of entropy reduction available from the applications examined.

Even greater energy savings comes from a different source, namely the sharing of data within the tree structure. All but the last hop to the leaf node are shared by multiple leaf nodes, requiring $\sum_{k=1}^{5} \mathbf{16}^{k}$ hops over a tree with the maximum of 1.048M subscribers at leaves; this effect results in a *91% reduction in the number of transmissions* over the network required to deliver the digital newspaper to all subscribers in the two-tree multicast arrangement versus the 13.10M hops required to reach each client via unicast over an 11.91-hop network (13.10M versus 1.17M hops).

5.3.2 Discussion

The calculations for multicast distribution of digital editions of the daily newspaper show a potential order-of-magnitude savings from utilizing content sharing mechanisms within multicast distribution of content. Regarding multicast, however, there is an important limitation to consider as well as its interplay with the chosen problem. In general, a global policy of multicast distribution of content does not map particularly well to the problem domain due to the multicast requirement that receivers receive the content synchronously. Most content access is done asynchronously over a long timeframe.

For the full savings from multicast, simultaneity of submissions is required: as in broadcast, a single transmission is received by all receivers, as if a live signal were transmitted via an antenna

using high energy, and picked up by all receivers simultaneously. This requirement for temporal synchronization could be met by running a background application on each leaf node such as a user's tablet PC that receives the daily transmission of the New York Times online edition, just as physical newspapers are dropped off at homes and stores each morning, long before many readers awaken. In the following section, other ways of achieving a hybrid effect—somewhere between unicast delivery to a specific IP address on demand and multicast delivery to all subscribers at 4 am GMT—are discussed within a more general strategy of strategic *caching* of data rather than replicating it anew.

For sufficiently small or short content that is sufficiently popular, it is important to point out that devices with plenty of storage may indeed benefit from a proactive multicasting policy such as multicast in case you want this later, at least from a global energy minimization standpoint. Delivery of today's movie releases to all subscribers may indeed be an appropriate service to which many people would subscribe, and for which a background application can be run on the device to receive movies synchronously. Just as ten years ago many people received a newsprint edition of their favorite or local newspaper, people would get the daily media package, containing news and other media—as soon as it is available rather than as soon as one requests it by clicking a link on the Web. The global effect of synchronous delivery of content to such a potentially large pool of subscribers could also be mitigated by staggering transmissions to independent trees using a delay scheme.

5.4 A More General Assessment of Multicast

5.4.1 Introduction

The above back-of-the-envelope-type estimate for the energy cost of delivering the New York Times via multicast transmission suggests that the current support for multicast trees within the

Internet is sufficient to produce dramatic savings in the energy cost of transmitting to a pool of fixed receivers. One key problem with this analysis, however, is that one cannot assume in general that the receivers are in fact all sitting at network positions five to seven hops away from the arbitrary-but-fixed source serving the New York Times. Other studies suggest that typical average distances are more like 11.91 hops, such as one in which Loguinov examined the general pool of video traffic in the Internet over a wide sample [138]. To give an idea perhaps of the far end of the range, where for example a peripheral device may be the source of the information of interest, and again the recipients being also peripheral devices, 20 hops may be the distance, taking from Tucker, who estimated the average distance somewhere in the timespane of 2009 to 2011 of any-to-any device communication of the Internet to be 20 hops [8]. This section goes into more depth on three issues: first, the more widely accepted or generally viewed cost savings of multicast from general studies of Internet structure; second, how often that savings can be expected to occur, based on degree of redundancy in transmissions at any given moment; three, how much that savings is reduced by the presence of multiple sources of the same item, rather than assuming a single server providing all of the Internet with the content item of interest. Results for a simple network with 1,000 receivers instantaneously retrieving items from a 100-item universe are presented to illustrate the effects [149].

5.4.2 Related Work

Transmission distance, expressed as a count of the number of hops a message traverses, is often used to compare one scheme to another; energy use is assumed to be proportional to the average or worst-case number of hops traversed. Chuang and Sirbu [150] were the first to express this cost using a baseline of unicast transmissions within the same network—a formulation that is useful in the current study, because this gives a common baseline to compare the limited aspect of content distribution of interest here to multicast. Their results and methods were later largely confirmed

by Phillips et al. and honored by the term *the Chuang-Sirbu [multicast] Scaling Law* [151]. Van Mieghem rederived an expression for small group sizes over random graphs and k-ary graphs, and cautioned that the Chuang-Sirbu Law may be insufficient to describe the phenomenon for very large group sizes greater than 10^6, but leave the law largely intact [152]. The present work also draws from the approach of Aaltonen et al. [11] and others in characterizing multicast group sizes. Aaltonen compared the tradeoff of multicast to unicast and derived a multicast gain of as much as two orders of magnitude by an analytical model of cell network cells; however, their aim was to characterize the retained cell capacity via multicast, and their model was limited to effects within a single cell or node.

5.4.3 Methods and Assumptions

Multicast conserves the number of transmissions by replacing identical transmissions over the same link by a single transmission; where paths fork, network nodes duplicate the transmission and send one message along each separate fork. The effective transmission path to r recipients from one source s can be the same pattern as in shortest-path unicast transmission, but the number of separate transmissions that pass along the initial h network hops or links is greatly reduced by the use of a single transmission for each of the common links in the path.

The goal is to quantify this benefit as well as the frequency of occurrence of the range of circumstances under which it holds. The basic approach of the argument and analysis below is as follows: it considers and assesses the potential gain provided by multicast within a network composed of single sources for each unique content item (film, book, etc). It then also looks at a regime in which there are multiple duplicate soures of the same content, so as to reveal the effect of content duplication on the potential gain from multicast. Background or related work is introduced as needed to underpin the method or analysis. The argument considers only instantaneous traffic, not temporal effects.

Multicast gain for a single multicast group

The economy of scale associated with multicasting a particular item simultaneously to a set of recipients r can be approximated as $|r|^{0.8}$ as compared with unicast to the same set, for which the cost is directly proportional to the number of recipients, or $|r|^{1.0}$. In other words, for a unicast transmission pattern that results in 1,000 simultaneous transmissions from source to receivers, transmission via multicast along the same routes requires only 250 transmissions. This measure of multicast's reduction in transmission costs was determined by Chuang and Sirbu after examining several Internet-like routing topologies and holds for a random distribution of recipients throughout the network [150].

Route length in multicast and unicast need not be the same. In more precise terms, given a multicast group of size M where $M = |r|$ from the set r of requesters, this benefit over unicast transmission to the same pool of recipients is defined as

$$cM^{0.8} \tag{5.1}$$

where c is an adjustment factor if different path lengths are employed for multicast versus unicast within the network; c is the ratio of the average depth of the distribution tree for multicast to the average path length associated with unicast routing, which is network-dependent [150]. In Chuang and Sirbu's analysis, the recipients are defined as leaf routers serving one or more local hosts and are assumed to be randomly distributed throughout the network.

Multicast gain for a representative set of network transmissions

The next step is to relate the above gain for a single multicast group to an overall gain within large data networks such as the Internet from multicasting to all such groups. This requires analyzing typical group sizes within Internet traffic.

For this analysis, the observable multicast group sizes can come from the observed distribution of commonality in web access requests for Internet traffic that has been heavily studied over the past twenty years, for which some of the earliest results and summaries come from Glassman [153] and Breslau [154]. These and more recent studies describe a Zipf-like distribution of access frequency by rank or *popularity* in which requests from a fixed pool of network users for the i^{th} most popular element of the access set occur with a probability $1/i^{\alpha}$; the appropriate value for α has been hotly contested by researchers for decades, and it varies from one network traffic trace to another [154].

Figure 19 shows the commonly reported Zipf-like distribution curves for Internet content requests for several values of the shape parameter that have been reported in the literature. If $\alpha = 1$, then the distribution follows a true Zipf law where the second most popular item is requested half as often as the first, the third a third as often, the fourth a quarter as often, and so on. If $\alpha < 1$, the less popular items are requested more often than this; whereas if $\alpha > 1$, they are requested less frequently than this. Reported values range from $0.63 <= \alpha <= 1.75$ depending on the study and the pool of requesters as well as the set or universe of requested items considered. The shape parameter α is here arbitrarily assigned to several values within the range of prior studies, and we report the effect of α on the results obtained.

For simplicity, each request is assumed to be drawn independently from this Zipf-like distribution. The precise definition of the distribution used here comes from Breslau [154] for a finite set of G items requested to by receivers in which the conditional probability of a request P for item i from the set G is

$$P_G(i) = \frac{\Omega}{i^{\alpha}} \tag{5.2}$$

where Ω is a normalizing constant defined as

$$\Omega = \left(\sum_{i=1}^{G} \frac{1}{i^{\alpha}} \right)^{-1} \tag{5.3}$$

Only the relative occurrence of groups of particular sizes matters here, and the size of the content item i in terms of number of packets or datagrams required to deliver it is considered immaterial to the present analysis, because the quantity this method aims to obtain is treated as a scaling factor—a relative, not absolute, number, applied to the stream of packets that satisfies a request for item i. Although by taking this approach we are explicitly enumerating the items of the set G, it is not actually necessary to define the specific content that is being requested in each case, only the commonality among requests.

Given this distribution of requests to access specific content from the finite set G, it is possible to construct the maximum gain available from multicast for the set G very simply, in terms of the group sizes for each of the items i. The actual benefit of multicast is heavily dependent on the grouping of requests, and we can estimate this grouping using the observed popularity distribution $P_G(i)$ in Equation (5.2) to obtain the number of simultaneous requests for the same item i for all such $i \in G$. The cost C of multicast is

$$C = c \times \sum_{i=1}^{G} M_i \tag{5.4}$$

where

$$M_i = (R \times P_G(i))^{0.8} \tag{5.5}$$

where M_i is the single group size of multicast from Equation (5.1), and it is being replaced in Equation (5.5) by the series of group sizes within the group G, c is the constant relating the average path length in a multicast tree to that of a unicast tree in the same network, R is the total number of requests, G is the total number of items in the set of content considered, and $P_G(i)$ is from Equation (5.2).

The cost of unicast is the sum of the costs of all unique transmissions from sources to each receiver that satisfies the set of requests $\boldsymbol{R}$; $\boldsymbol{U} \propto \boldsymbol{R}$. The specific cost of multicast is not derived here, but rather its cost relative to that of unicast—more precisely, the energy gain to be had over the use of unicast due to the significantly smaller total number of transmissions multicast uses. This energy gain is directly proportional to the number of transmissions conserved by the property of multicast represented in Equation (5.1).

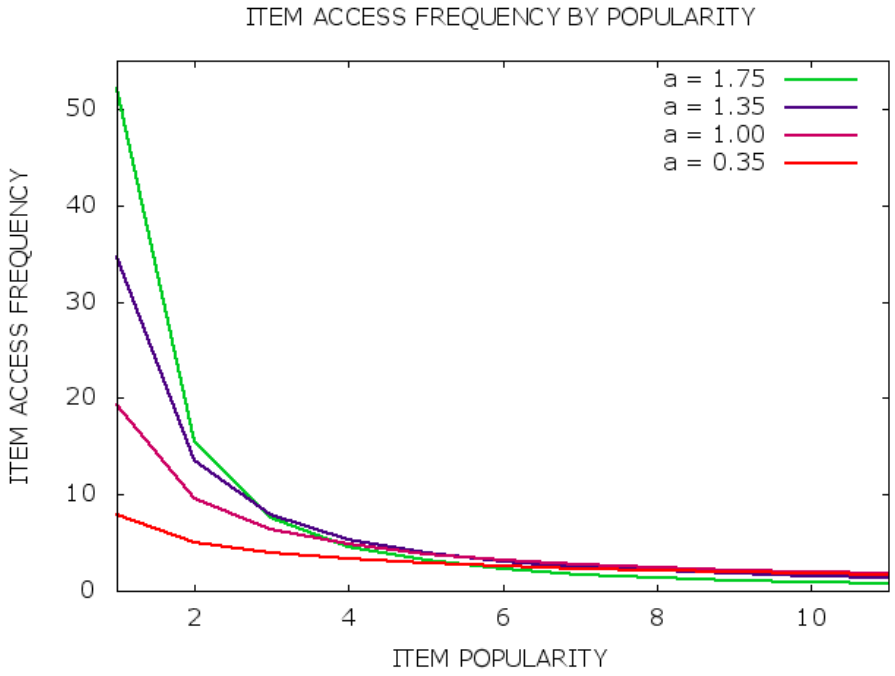


Figure 16: Zipf-like Distributions of Access Requests. Typical reported distributions of content access requests on the Internet. Only the most popular items are shown (the top 10%); beyond 10%, the values continue to drop to zero. The essence is that, depending on the exact value for the shape parameter $\boldsymbol{\alpha}$ characterizing the decay rate, between 35-90% of access activity is devoted to the top 10% most popular items in the universe of accessible objects, and overwhelmingly to the first few percent of those: 23-83% goes to the top 5% most popular items.

5.4.4 Energy Calculation Results: 47-52% Energy Savings

Table 13 shows the energy gains from using multicast by comparison of the unicast and multicast values when fielding 1,000 receivers' requests for 100 distinct items under different assumptions. Multicast cost is expressed as a fraction of the full cost of unicast; thus, a lower number is better: more energy-efficient. The results shown indicate two major effects on the gain to be had from multicast. The first three rows vary α to show the effect of the specific popularity distribution on the cost of multicast for a range of reported values of Zipf shape parameter α in various studies of Internet content collections: the higher the α, the more benefit to be gained from multicast delivery.

Table 13: Multicast Energy Cost. Multicast cost versus unicast cost for a variety of observed values of the shape parameter of the Zipf-like distribution of item popularity, α.

α	n	unicast	multicast
0.63	1	1	0.60
1.00	1	1	0.55
1.35	1	1	0.48

Figure 17 shows the relative cost of multicast and unicast under these assumptions, with energy along the y axis plotted on a log scale, and x values from one to ten plotted on a log scale. Shown is the energy efficiency of multicast versus the reference value M for unicast as a function of item popularity when fielding 1,000 receivers' requests for 100 distinct items for $\alpha = \mathbf{1.35}$, on a log-log scale appropriate to Zipf distributions.

The effect of wider content distribution in the network on the energy efficiency of multicast is also shown in Figure 17. This point is not crucial to the current topic, and is described only very briefly

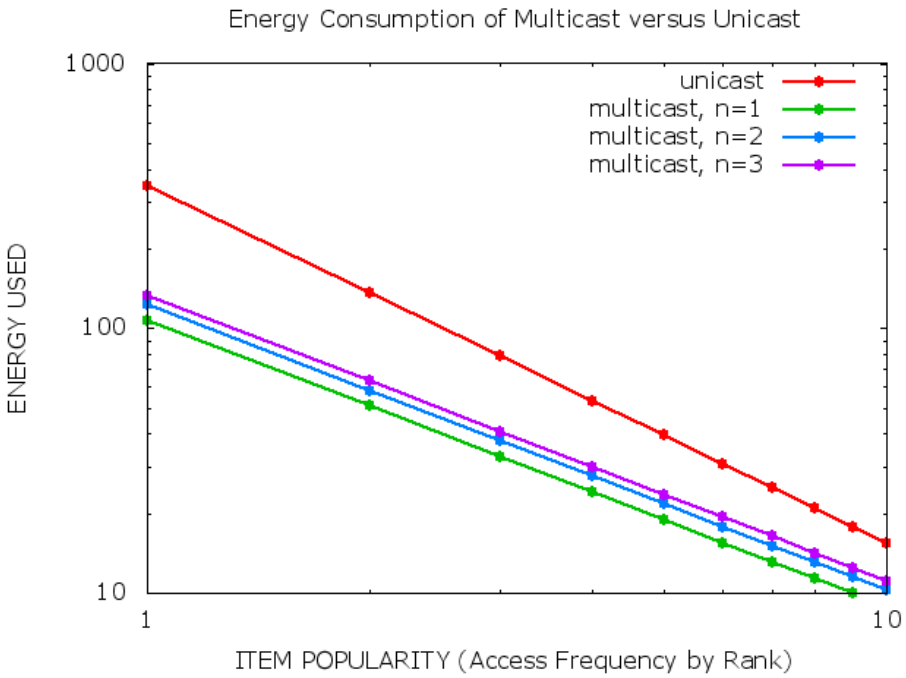


Figure 17: Multicast versus Unicast. The relative cost of unicast (red) and multicast (green) transmission of the set of items in G. The blue and purple lines show the higher energy requirements of partitioning multicast groups into multiple separate groups, for $n = \{1, 2, 3\}$ locations sourcing item i. Plotted on a log-log scale; only the top 10% most popular items are plotted. The costs graphed are relative rather than absolute. More specifically, the reference, unicast cost is not presented in Joules; rather, it is directly proportional to the points plotted, which are the number of requests for content item i, whereas the multicast cost is directly proportional to the points plotted, which are proportional to $cM^{0.8}$ of unicast cost, M. No reduction in path length due to multicast group partitioning was included, as may be the case of multiple servers utilized in the same data center.

here. What the three multicast curves show is that the main contributors to cost as well as gain are from aggregating the requests for the most popular items of the Zipf-like distribution of requested content—gain by the difference in slope of the unicast curve versus the multicast curves. In the case shown for a single source of each item i, denoted *multicast,* $n = 1$, the group size for each item i is equal to item popularity. This is not so for the case where i is duplicated at multiple sources s in the network—also shown in Figure 17 for $n = 2$ and $n = 3$ sources. The resultant reduction in multicast energy efficiency is due to the resulting smaller group size: the popularity of item i is unchanged, but group size is divided into $n = 2$ and $n = 3$ groups and distributed over the two or three distinct sources of item i. Shown is the maximum effect from a perfectly even division of receiver requests to n sources for all i in G; an uneven partition gives better energy performance due to the heightened contribution of the larger group. Only the ten-most popular items in the set G are graphed, as the popularity dwindles quickly on the log-log scale. At the limit, the sum of single-receiver multicast groups gives the same energy productivity as unicast transmission. The reader is referred to a paper that covers this fully [149].

5.4.5 Discussion

Internet studies of access frequency over the past decade suggest that there is a remarkable similarity in requests for content out of a large pool of content items. The derivable effect of multicast transmission to the pool of requesters was assessed here, given the known cost savings from multicast transmission over unicast transmission pathways within the Internet. The results suggest that, if multicast were used in all possible occasions to transmit content to receivers, there is sufficient duplication in receivers' requests that the energy savings would be roughly 40-52% of the cost to transmit to the same set of receivers via a unicast transmission pattern of full, independent transmissions. This assessment is based on viewing the duplication from observed similarity in large pools of Internet traffic showing a Zipf-like pattern of requests distributions for elements of

a finite universe such as a catalogue or set of URLs. The specific savings depends upon the actual distribution, for which estimates vary. The assessment here does not consider temporal effects or temporal variation in the distribution or the receiver locations throughout the network.

While the above reports the expected gain from fully using multicast to satisfy the pool of simultaneous subscribers requesting the same content, the actual, realized gain in networks is based on actual use of multicast in practice; not all opportunities to utilize multicast are currently exploited.

5.5 Multicast within Specific Networks

5.5.1 Introduction

Whereas the previous analyses of multicast used statistical information about the Internet from large but unavoidably imperfect samples suffering some of the issues with $\boldsymbol{traceroute}$-based methods described in the background given on pages 24 and 38 of Chapter 3, these provide a top-down analysis of multicast cost in networks.

The aim of the work described here is to arrive at an exact measurement of multicast transmission cost within real networks using a bottom-up approach that counts the cost of bits over routers and switches. This analysis uses the networks described in Chapter 3 that define real AS domain networks within the Internet for the main classes of network, Tier-1, Transit and Stub, topologically defined at the hardware (router- and switch) level.

5.5.2 Methods & Assumptions

All routers within the networks are assumed to be multicast-enabled and capable of performing multicast; this is fine for what we are mainly interested in, which is multicast cost under the salient conditions of real networks such as their topology; their size; their degree of connectivity; the

specific transmission paths within them; and the energy consumed by the hardware in processing, storing and forwarding packets.

Detailed analysis of the per-byte cost to process, store and forward packets for an appropriate class of switches and routers by Vishwanath [98] was used to assess the content transmission cost. The results reported here are for 100,000 requests for a particular video clip of average length of a YouTube video download, four minutes and twelve seconds.

The cost of unicast transmission over the network was assessed from the original traffic, that was generated at random plus a bias for request location based on the class of network in a similar manner as was done for the iterative assessments in Chapter 3; the only difference is that the algorithm walks each path, accumulating the transmission cost at each node in the transmission path, rather than assessing where the maximum benefit is to splice the path looking across the entire set of paths. The cost includes the cost of the request message traveling upstream along the same path, which is nominally assigned to be one 1500-byte packet long.

The costs are assessed for both unicast and multicast on one sample from the traffic history pool; which one is immaterial, since the ordering of traffic does not matter for either unicast or multicast assessment: all paths are aggregated into the total.

The simplifying assumption was made that the amount of traffic under consideration was sufficiently large and that random differences in requester location make it the case that the reasonable cost to assign to multicast is that for transmission to a random large set of nodes that can occupy any set of leaves—and thus, the average cost of multicast under these assumptions is the full cost of multicast to the spanning tree associated with the source node. The numbers reported for multicast are thus higher than they might be for a specific set of receivers, particularly a small set of receivers that is clustered within the network; however, the unicast cost also goes down in such cases, as well, preserving to some degree the relative difference between them. The calculation

described for unicast was run on the traffic log to capture the cost of just the request packages traversing the network on behalf of the requesters; this was then combined with the calculation run over the nodes of the spanning tree. As in Chapter 3, links were assumed to have no additional incremental cost associated with transmissions.

5.5.3 Experiments & Results: 99% Energy Savings

The results reported here are for 100,000 requests. Table 14 shows the results of the assessment for both unicast and multicast for each of the networks. For these networks, the typical cost of multicast was less than 1% that of unicast to complete the same task—on average the cost of multicast was 0.6% that of unicast.

5.5.4 Discussion

This detailed bottom-up assessment of several specific networks suggests that multicast over specific networks may cost substantially less than suggested by the statistical sample used to develop the Chuang-Sirbu Scaling Law. The sample set is far too small here to make any claims about multicast support in general within the Internet; however, the results certainly underscore the importance of assessing the cost of any contemplated delivery method on the target network itself for its energy efficiency before making a decision what to use. They also fall close to the ballpark of the estimate for delivering the New York Times via multicast, even though both methods of estimation use very different approaches to estimate the cost of multicast transmission in networks.

Table 14: Comparing the Energy Cost of Unicast vs Multicast. A side-by-side comparison of the transmission cost of unicast delivery and multicast delivery to satisfy 100,000 requests for a $4'12''$ video clip within a sample set of real networks

Network	**unicast cost (J)**	**multicast cost (J)**	**savings (J)**	**savings (%)**
AS 1239	7,057,081	54,735	7,002,346	99.2
AS 224	4,597,382	35,185	4,562,197	99.2
AS 3549	6,025,916	30,223	5,995,693	99.5
AS 59	1,247,162	34,616	1,212,546	97.2
AS 2914	6,533,060	24,360	6,508,700	99.6
AS 3356	5,442,846	10,335	5,432,511	99.8
AS 3292	2,524,117	3,561	2,520,555	99.9
AS 109	6,626,477	3,419	6,623,058	99.9
AS 680	2,674,362	2,259	2,672,103	99.9
AS 3356.1	6,495,726	1,811	6,493,914	99.97
AS 680.1	1,991,771	193	1,991,578	99.99

5.6 Conclusion

Broadcast television has long been an energy-efficient means of reaching many viewers, but it has given way to delivery of content over the Internet and through other carriers that reach consumers, such as cell phone service providers [7]. The Internet analog for broadcast is point-to-multi-point, or multicast transmission; like broadcast, it requires simultaneity among viewers; but unlike broadcast, these groups of viewers can form on demand. Given the large potential for saving energy via multicast delivery as underscored here through analysis of several different cases under varying assumptions, there is motivation to come up with strategies to incentivize users' behavior that aggregate users' requests and maximize the applicable energy savings using multicast delivery of content.

Many content distribution services like NetFlix use point-to-point encryption; however, contemporary encryption techniques can easily be extended to apply to a tree-like distribution. This, however, is beyond the scope of this dissertation.

Chapter 6

Conservation of Energy in Data Center Networks

6.1 Introduction

Chapter 3 examined specific domains within the Internet; this chapter examines a different class of networks, the network inside a data center, for its energy efficiency. In doing so, it goes significantly beyond the typical metrics of facility energy efficiency for data centers, which is blind to energy proportionality and other aspects of the efficiency within the computer- and network architecture, or IT portion, of the data center, by looking in more detail at the network, and at how it fits specific distributions of traffic [155].

6.2 Motivation

Open architectures like the one recently unveiled by Facebook allow a detailed assessment of the energy efficiency of commercial data centers. Data centers like the ones that run Facebook sites and services represent approximately 1.3% of total global energy use, and 2% of domestic energy use within the United States [5]. The network over which data center compute nodes communicate with each other as well as with the outside world consumes a substantial and increasing fraction of the

total energy budget of data centers, as servers improve. Abts reports that the network consumes 12-50%[1] of total power [156]. To put into clear perspective the environmental impact of choices made in designing and deploying data center networks, Dong calculates that the annual CO emission of 1 kW (i.e. two typical 500W router line cards) is the same as that of driving a car 13,000 kilometers – close to the average annual usage of an American driver [157]. Worse, this could become a fleet of cars, as Kilper estimates that the power per wavelength per card will exceed 1 kW in the future for 1 Tb per second transmission rates [15].

Data centers appear to be shifting over time toward higher-performance, higher-bandwidth networks through increasing the bisection bandwidth, raising the limit on the amount of information that can be moved from one side of a network to the other. There are a number of reasons behind this, keeping up with speedup of other aspects of the data center such as storage, memory, and CPU. One undesirable consequence to increasing the network bisection bandwidth pointed out by Abts is that it requires more switches, and each switch must have faster and more power-consuming links [156]; Kilper points out that racks are already at their thermal capacity; thus, increasing the number, rate, or bandwidth of switches also necessitates increasing their physical footprint or cooling requirements, to dissipate the greater heat [15]. Thus, there is an energy price to pay for higher-performance networks.

One example is the adoption of more meshlike network topologies, such as the topology recently adopted by Facebook in its new data center design. These meshlike topologies cost-effectively provide much greater bisection bandwidth[2] [156]. Greater bisection bandwidth has a chain of benefits: it lowers the oversubscription to routes, which in turn enables dynamic resource allocation across large pools of servers, freeing up communication-intensive jobs to run anywhere, rather than on a

[1]The reason for the range is variation in server and network utilization, with 12% reflecting 100% server utilization and 50% reflecting 15% server and network utilization along with the use of fully energy-proportional servers that power down when idle.

[2]For the exact-value bisection width of such structures we refer the reader to Arjona Aroca and Fernández Anta [158].

single rack or tier [159]. Several groups have demonstrated that these high-performance network resources can also be used to save energy, through dynamic invocation of energy-proportional behavior [156, 160]. Here we examine some properties of a meshlike communication topology between data center compute nodes for their energy efficiency running typical Internet traffic patterns.

This chapter explores the fit of Zipf-like distributions typical of network traffic, to updates of user pages and the entity graph, for the new Facebook data center network architecture.

6.2.1 Why Facebook

Facebook is one of the hubs of referral-based traffic within the Internet. It is the second-most visited website, at least from U.S. web use statistics, and its 1.44 billion users worldwide create 56% of all referrals, from sharing links, and 38% of the total subsequent volume of network traffic due to acted-upon referrals, from opening or following links [161–163]. To give an idea of traffic volumes, Bhardwaj reported that in 2011 Facebook users shared over 30 billion items per month, posting links to news articles and other articles, blogs, web sites, videos, photos, and other Internet content [164].

Increasingly, then, Facebook is a content portal for Internet users, and this content is explicitly represented in Facebook's internal representation of the social network and content to which its members refer, as links formed between the sharer and the shared content. This graph of relationships is growing rapidly. The parts that represent content stored and/or referenced by site users grew in 2011 by about 30 billion links per month, 2.5 new pieces of content per day, and 2.7 billion user referrals or links to entities per day, as reported by Bhardwaj [164], and possibly close to 5 billion links per day in 2015, if that number includes some intersection of 1) the 24 billion photos and videos per month, and the 500 million photo-uploads and 300 million video-uploads per day that Bachar reported [16], and 2) the 4 billion video downloads per day from Facebook that Griffith

reported [165].

All of the new content generation "within" Facebook as well as all of the linking/referencing activity—which draws external content into Facebook as well—requires that the compute nodes actively update the Facebook graph that represents all the relationships and store both the new links and any new content related to them. These represent additions to several facets or services at Facebook, among them being: 1) the content stored by Facebook, either as direct stored content such as a user photo, or a location reference to content stored elsewhere, such as a URL; 2) the Facebook graph representing all entities and relationships; 3) the indexes Facebook uses to find content later, such as Facebook's *Unicorn* index, which is updated in parallel to keep it current with the changes, within seconds [166].

Despite the truly remarkable growth of the datasphere explicitly represented within the Facebook graph, Andreyev reported that the main traffic generated by all of this is not incoming, or north/south, network traffic, but machine-to-machine, or east/west, traffic within the data center [167]. Andreyev stated that machine-to-machine traffic volume has been doubling more than once per year within Facebook's data centers, and that furthermore this is following an exponential growth curve, meaning that the *rate* of doubling is growing exponentially [167]. Scaling the data center network for exponential traffic growth is a challenge, and Andreyev reported that Facebook's prior solutions, such as allocating more network ports to accommodate traffic between compute clusters, creates problems, in that it reduces the allowable cluster sizes. This reduces the opportunities to fit large processes within a cluster, again increasing between-cluster network traffic. [167].

To counter this growth, and to simplify the scaling-up and data-rate upgrades that will be necessary to grow with their data center network traffic, Andreyev and Bachar reported that Facebook recently reengineered their data center network, the physical and logical structure by which compute nodes communicate. In the process Facebook entered the network switch market, designing their own Ethernet-based network switches and racks, and simultaneously with it their own data

center network architecture to interconnect all the compute nodes used to run and maintain Facebook site and services [167, 168].

One beneficial aspect of Facebook's approach for the research community is that Facebook has already made public, or announced their intention to open-source, the details of both their data center network architecture and their network hardware in sufficient detail that others can examine and adapt their ideas, and can manufacture the switch hardware [168].

6.2.2 Energy Efficiency of Data Centers

Another important facet of the global ICT, namely the rising energy use of commercial data centers such as Yahoo and Google, has also received a great deal of scrutiny in the past ten years that has resulted in considerable reported energy-efficiency gains. Until quite recently, data centers were conspicuous consumers of energy within the ICT, and their rate of increase in consumption caused public concern. A report on data center energy consumption by Koomey estimated a 100% increase in their energy consumption from 2000–2005 [169], leading to awareness in academic, government, and business circles; public outcry on this topic formed after an exposé by The New York Times reported that the productive use of energy within data centers was only 6–12% of their total energy use [170].

This negative press caused the commercial giants either to reveal or improve the energy productivity of their data centers. Smart systems engineering and computer- and facilities engineering have made astonishing gains in energy productivity of data centers, with Facebook reporting that its most efficient data centers currently run at a mere 1.09 times the energy cost of running their servers alone, and Google reporting even lower, at 1.06 [170]. While the field lags behind these forerunners, averaging a power utilization efficiency (PUE) factor of 2.5 [170], the lessons learned from these successes and others will allow the best practices to be translated to other data centers [170]. In a more recent study, Koomey found that the rate of increase in energy consumption

has slowed. Commercial data centers are now estimated to consume about 1.3% of the total energy used globally [5].

A number of large commercial data centers currently claim to operate very efficiently by this PUE measure, at a ratio of around 1.1:1 of physical operations to computer power usage. Power utilization efficiency or PUE has been defined rather curiously, however, similarly to that of power plants, such as nuclear power plants, or coal power plants, despite the important differences. Such PUE numbers for the physical plant as those quoted above are at best a proxy of energy efficiency and are not the same as the energy efficiency of the data center. From the point of view that motivated the current work, such numbers are a rather poor representation of energy efficiency.

PUE makes sense for power plants. Power plants are assessed a power utilization efficiency from how much electricity or power the plant expends in order to obtain a unit of useful work; the work portion of the ratio is the power the plant adds to the general power grid. PUE for power plants is a true cost:benefit ratio. While one might view PUE as an overhead calculation, and thereby by extension construct a similar measure for data centers, this is not the main point of PUE.

Data center PUE is defined as (total power in to the data center) over (power drawn by the IT portion of the data center). This ratio fails to address the question of how many units of useful work were performed: running the computers is not the same as doing useful work. This is really more of an overhead calculation, and is useful to the facilities and operations teams designing, building, and operating data centers, as a facility operations overhead number.

Within the larger context of society, however, in the debate over energy consumption, data center PUE doesn't really help the discussion, because it doesn't address the useful work—the benefit to society—of the resources consumed; it just itemizes where those resources went, like a budget would. It's more of a cost:cost ratio, with which data center facilities operations energy budgets have been audited and substantially trimmed. It is time now to look deeper into the second term of

the ratio, the power drawn by the IT portion of the data center.

The true output of the data centers is the useful work performed per Watt within the datasphere, engaged in providing information and services, and data centers would more ideally be measured by this, within a true cost:benefit ratio. For this we need to define the benefit; reporting PUE as if it stood for the metric society truly wants is dangerous, because it might actually penalize an efficient solution in the IT portion of a datacenter because this worsens the PUE.

In particular, PUE does not properly highlight efficiency of the IT portion of the datacenter. It does not relate the *benefit* of the computer power usage, says nothing about what those computers are doing, and whether their time is efficiently occupied, or whether hardware resources are efficiently idled or shut off when not in use. This appears to be an important problem area, with some reports of data centers being so network-latency-bound, or suffering enough resource stranding and fragmentation, that as much as 74% their resources are frequently unoccupied [171].

The current work is intended to partially bridge the gap between insufficiently insightful definitions of PUE and the remaining question of the truer energy efficiency of the data center, for which Bianzino has begun a taxonomy of the energy efficiency measures that have been proposed [172]; here, the focus is on the architecture and its use of physical resources or infrastructure within data centers. The larger goal is to eventually answer the question of what is produced in exchange for the power consumed by the data center IT resources. In this chapter, the question is approached by examining how well the input characteristics match the networked distributed computing environment of a recently designed data center architecture.

6.3 Facebook Data Center Fabric Architecture

Facebook redesigned its data center architecture and deployed its first data center using the new design in 2014. The name given to the redesigned architecture is the data center fabric (FDC).

Farrington and Andreyev first described the new architecture that uses small, commodity switches arranged as a five-stage folded Clos, or fat-tree network [173, 174].

Fat-tree networks fall within a general set of more richly interconnected switch architectures featuring higher bisection bandwidth, for cost-effective high-port-count, high-bandwidth networking. Higher bisection bandwidth from more meshlike connectivity has been proposed as a better alternative to the traditional 2N data center network. The folded Clos network topology is the focus of various research efforts in meshlike topologies, by Guo, Al-Fares, Greenberg, Farrington, Mysore and others [159, 175–178].

The complete smallest unit of the FDC is diagrammed in Figure 18. The network features a five-stage Folded Clos tree layout connecting all data center server racks to one another. The network is shown in red and fuchsia. At the bottom of the diagram in black, the data center racks containing networked compute nodes are shown, organized into *pods*; only two pods are shown, for clarity. One data center hall consists of a 48x48 set of server racks, with each pod a line of 48 racks, or a 1x48 plane of the total floor; or a 96x96 set of server racks, with each pod a 1x96 plane of the total floor. There are four redundant such network connections between pods; only one is shown here for clarity, consisting of all the red switch nodes and lines. There are four network hops between servers in separate pods, and two hops to communicate within a pod via top-of-rack (TOR) switches.

The unit in Figure 18 is repeated for greater bisection bandwidth and throughput, although, based on need, edge /gateway and backbone nodes can be shared across subnetworks. Backbone nodes are conceived of as being employed in some multiple of subnetworks. For example, at Facebook's first FDC data center, the network has initially been supplied with 12 backbone nodes total shared among the 48 pod networks [167]. The network *architecture* is perfectly balanced, but the network *traffic* need not be.

In regards to the construction of the data center facility, Facebook designed the cabling layouts and floor layouts as well for simplicity and scalability. In regards to the hardware used, Facebook designed its own switches and has announced its intention to release the manufacturing information files for constructing these boxes. Bachar recently described the hardware designs of the single- and multi-function switch boxes, called the Wedge and 6-Pack [168]. Rather than rely on high-end expensive switches with more ports or bandwidth to increase network capacity, the FDC limits the fan-in and fan-out of the intermediary switches of the fabric to 48 or 96, and instead adds capacity by adding redundant pathways. These are added by repeating the basic structure of the network portion of Figure 18. The present work contributes by exploring specifically the *energy efficiency* of the FDC design elements, given what is known about traffic distribution for types of Internet traffic.

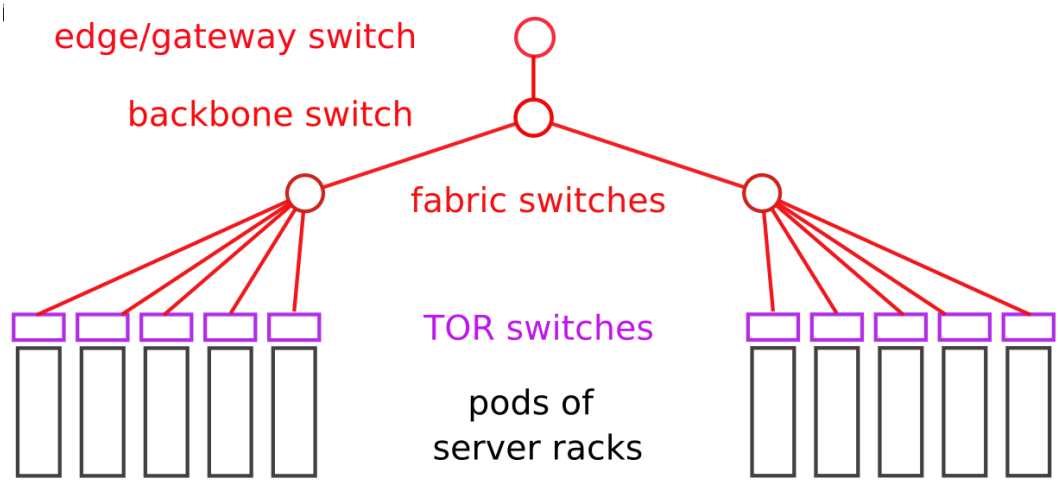


Figure 18: Network Diagram. Architectural diagram of the Facebook data center fabric (FDC) network that Facebook is using for new data centers. The first data center using this design was brought online in 2014. Shown is the basic element of the network used to join any two *pods*, or collections of compute nodes. Adapted from Andreyev [167].

6.4 Related Work

Energy efficiency research within the general area of large-scale distributed systems, including many aspects of data centers, were recently surveyed by Orgerie [179]. However, Heller states that most data center energy efficiency efforts have focused on servers and cooling—either improving the physical components or optimizing the software [160]. Yet both Heller and Abts argue that the data center network's energy share automatically rises as the server and cooling shares fall [156, 160]. Abts demonstrated that a flattened butterfly network topology is inherently more energy-efficient than the folded Clos topology upon which FDC is based. They showed an 85%, six-fold reduction in energy use for typical data center workloads, when using a butterfly network combined with link dynamic range, which was achieved by periodically reconfiguring data rates at multi-rate links to consume less power while still meeting their near-term bandwidth needs [156].

Taking a different approach—of pooling traffic for greater opportunity for idling switches and ports—Heller demonstrated a network power manager that saves up to 50% of network energy by explicitly minimizing energy when dynamically choosing which links and switches are active for a particular traffic load [160]. Wang also demonstrated a 50% energy savings by taking a traffic-engineering-oriented approach, assigning virtual machines to servers to minimize traffic or to make traffic more convenient for traffic engineering and then shaping traffic to reduce the number of active switches [180].

For both Abts and Heller, and for a portion of the results Wang achieved, the energy savings is derived from taking advantage of the concept of energy-proportionality at the network, by lowering the data rates of links, or by powering off unused links and switches and optimizing flows to turn off the maximum number of unused links or switches. With energy-proportional approaches, the pocket of energy saved is from the difference between current load and peak capacity of the network. Kilper cautions that scaling energy use continuously with traffic load may not always be

possible, for those equipment for which the benefit is seen only in an *off* or *idle* state, depending on the speed at which the various network equipment involved can switch between states (*on, idle, off*) [15].

This chapter examines the fit of the architecture of the folded-Clos topology of the FDC to typical item access patterns that hold for items within the Internet, as well as for those same items as they are reflected within Facebook's own internal representations of the Internet-of-interest to their users, such as users' Facebook pages, with links to popular content, and the explicit representation Facebook keeps of all linked objects and people in the form of a large graph, along with the associated search indices. The observed distribution of commonality in URL access requests for Internet traffic has been heavily studied over the past twenty years, for which some of the earliest results and summaries come from Glassman [153] and Breslau [154]. These and more recent studies describe a Zipf-like distribution of access frequency by rank or *popularity* in which requests from a fixed pool of network users for the i^{th} most popular element of the access set occur with a probability $1/i^{\alpha}$.

The appropriate value for α has been hotly contested by researchers for decades, and it varies from one network traffic trace to another [154]. For simplicity, the shape parameter α is here arbitrarily assigned to several values within the range of prior studies, and we report the effect of α on the results obtained. If $\alpha = 1$, then the distribution follows a true Zipf Law where the second most popular item is requested half as often as the first, the third a third as often, the fourth a quarter as often, and so on. If $\alpha < 1$, the less popular items are requested more often than this; whereas if $\alpha > 1$, they are requested less frequently than this. Reported values range from $0.63 <= \alpha <= 1.75$ depending on the study and the pool of requesters as well as the set or universe of requested items considered.

Figure 19 shows the nature of this curve for access distribution. For the typical reported values

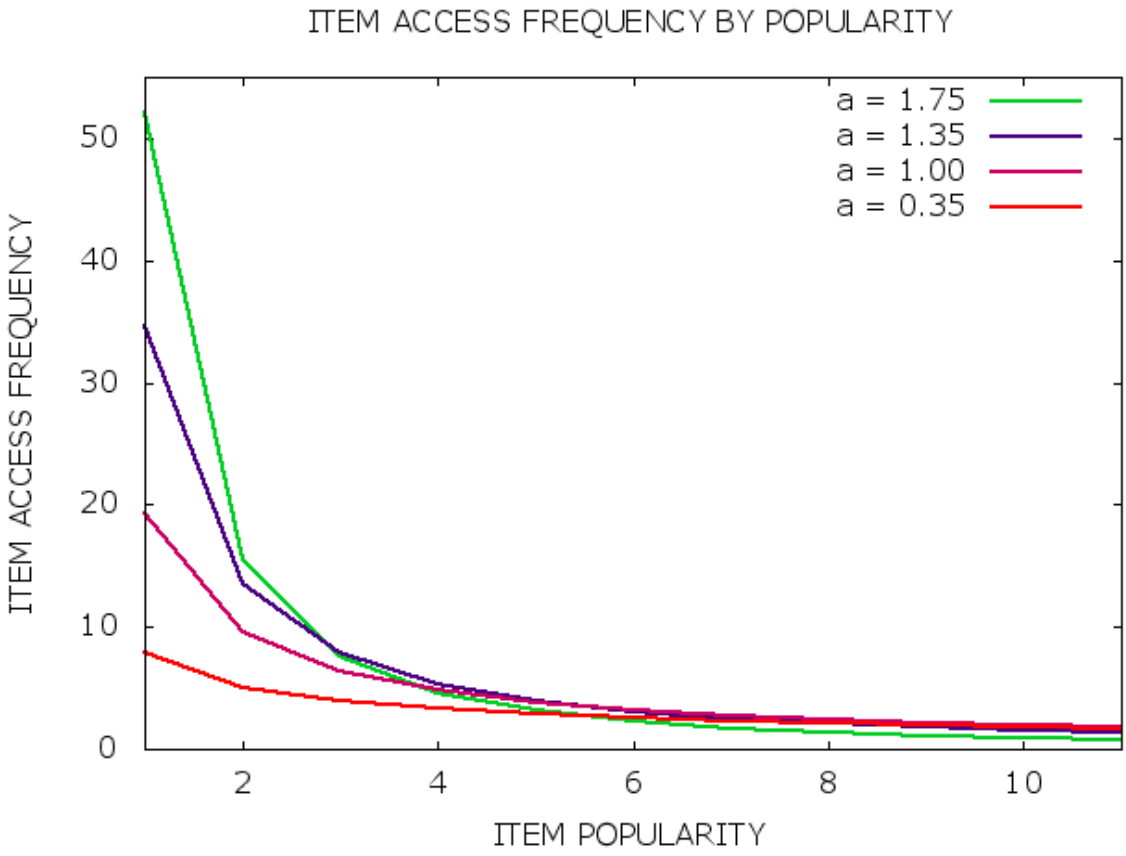


Figure 19: Zipf Distribution, Typical Shapes for Traffic. Typical reported distributions of content access requests on the Internet. Only the most popular items are shown (the top 10%); beyond 10%, the values continue to drop to zero. The essence is that, depending on the exact value for the shape parameter α characterizing the decay rate, between 34-90% of access activity is devoted to the top 10% most popular items in the universe of accessible objects, and overwhelmingly to the first few percent of those: 23-83% goes to the top 5% most popular items.

of α, the overwhelming majority of independent, separate access requests exhibit a remarkable similarity and tend to request a mere 6% of the items within the finite sets that were studied, such as catalogues.

A similar approach to assessing the traffic patterns within data centers with respect to their Zipf-like data store access patterns has precedent in Dong [157]; however, their aim was to use these traffic flows to determine and optimize the geographic locations of a set of data centers. Jin considered user-generated content like Facebook users' uploaded photos and videos that give a long tail to the popularity distribution, and related desirable number of replicas to popularity, within a multi-hub networked organization [181]; these, however, are inter- rather than intra- data center issues.

6.5 Energy Savings Approaches and Estimates

Figure 19 shows the commonly reported Zipf-like distribution curves for Internet content requests for several values of the shape parameter that have been reported in the literature. Within the domain of Facebook users, this phenomenon could take the form of posts on a person's page, or *likes*, which establish a relationship between the person and the content within Facebook's graph, or other persistent references or referrals of some kind. The same data store is shown both on the right and left of the figure, with the set of objects in the data store on the left establishing a *like* edge in blue to another object in the set, on the right, with a distribution given by the shape parameter α set to 1.35. The item on the right could be a popular news item recently published on a major media outlet, for instance. The links shown are established in the Facebook graph by updating the set of edges associated with both nodes. The darkened 76% of the nodes in the data store are all establishing links with popular items in the data store. In nearly any conceivable arrangement of the data store, this requires cross-pod traffic at full price in terms of network hops, for all user nodes and services not hosted on the same pod as the popular reference object data store.

The effect can be summed thus: the top 10% most popular URIs available on the Internet receive 34-90% of all access requests, and of them, the top 5% capture most of these requests, 23-85%. This phenomenon is referred to henceforth as the 90/10 split in access request targets: as much as 90% of accesses going to fewer than 10% of the targets. To our knowledge, the specific distribution for Facebook usage has not been established; there could very well be a longer, fatter tail associated with Facebook users' ability to promote content among their circle of friends, with access frequency conforming at the tail to group sizes. When viewed, however, within a sufficiently large sample size, all of this activity combined reflects the more general trends within Internet accesses such as those reported in large studies of Internet traffic. It is at this aggregate level–across many users–that the present work seeks to uncover efficiency gains that can be exploited.

What this suggests is that, if Facebook's entity graph grows by 30 billion shared links per month, then a very large share of these—ten to 27 billion of these—point to a mere 10% of nodes. For the entity relationship graph and other services provided by Facebook, this popularity distribution translates directly into a data store access pattern that significantly concentrates activity at a small number of items and locations. Across a large sample of simultaneous users of Facebook at any given moment, there will be an ever-changing set of popular items. However, the overall data store access pattern for such services will continue to reflect the Zipf-like distribution of content popularity, with the 10% most-popular items in the data store receiving 34-90% of all updates.

As a concrete visual illustration, Figure 20 illustrates the effect of popularity on network traffic with respect to the Facebook relationship graph. Within Facebook's relationship graph, there is an object in the underlying data store associated with each item such as a YouTube video. When a Facebook user posts a reference to that video, there is a corresponding update to two nodes within the graph, creating an edge between the node representing the user and the node representing the video. The important thing to note is that frequency of object updates to create new edges that

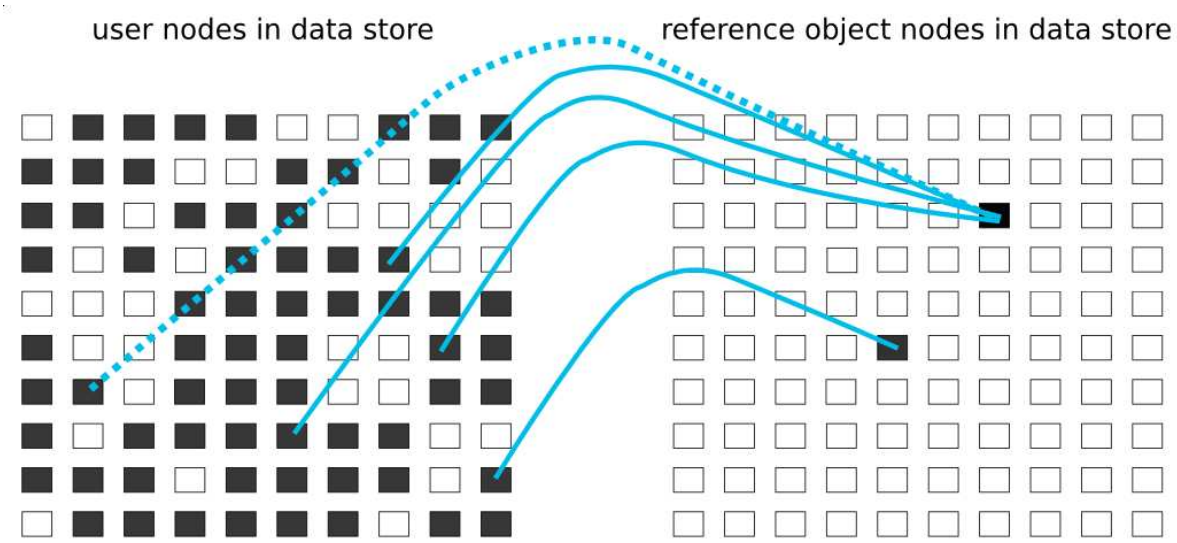


Figure 20: Data Store Access Pattern. A typical data store update scenario given the Internet access popularity distribution data, for which 10% of content, shown as a small number of items in the data store on the right, receive 34-90% of all access traffic during link updates. $\boldsymbol{\alpha = 1.35}$ shown, or 76% of items within the left set.

associate users with objects such as videos follows the popularity distribution. Thus with Zipf-like distributions of objects that are referenced, in general there are many objects in the data store (on the left) establishing a reference or link to a very concentrated set of objects in the data store (on the right). Thus, there is general activity representing the group on the left, but a marked concentration of traffic and updates onto a small number of data store locations representing the group on the right. This is true not only for the Facebook relationship graph but also on behalf of processes involved at compute nodes as part of several of the services that Facebook provides, such as the nodes responsible for maintaining the representation of this popular object within the Facebook graph, or adding links to it when a user links to this item on his/her Facebook page.

Figure 21 shows the correspondence between activity within the data store and activity within the data center network, showing the crux of the problem. It would be unreasonable for multiple reasons to store all data at one pod, and thus, highly popular data store items will not often be

updated by services running within the same pod. This ends up being true for 34-90% of the network traffic related to referrals such as posts and *likes* due to their uneven, Zipf-like distribution. The bulk of the traffic must therefore traverse the entire network hierarchy. To illustrate the path length, blue dashed active connections are shown; for clarity this is demonstrated for just one set of pods communicating to achieve the dashed-line data store update in Figure 20.

Particularly for the Figure 20 case of a 76-source, 34-target split in traffic from Zipf-like distributions, but also in general due to the provisioning of the data center for cost efficiency, updates to the data store objects on the right will not be limited to within-pod communication; thus, many of these updates will traverse the full network structure from one pod to another. Within-pod communication employs only the top of rack (TOR) switches for the racks directly involved and at most one of four fabric switches that connect TOR to TOR within the pod. Within-pod communication employs only two stages of the five-stage folded Clos, and therefore at most 3/5 the network resources used by pod-to-pod communication.

Given that *this update pattern is the common case, not the exception*, key to reducing the growth characteristics of machine-to-machine traffic is *decoupling network traffic from this relationship*. The goal here is to see if there are ways to reduce the potential *traffic imbalance*—particularly seeking ways to save energy and physical resources in the process. If the traffic can be better balanced, then it is not necessary to provision the data center to handle the packet rate and update rates required by the more resource-intensive right hand side of Figure 20.

6.5.1 50% Energy Reduction from Within-pod Communication

We turn now to ways to improve on the energy efficiency of Zipf-like traffic flows over the FDC architecture. An existing architectural feature of the FDC that supports high-activity flows energy-efficiently is the pod. An important option to reduce energy consumption within the data center

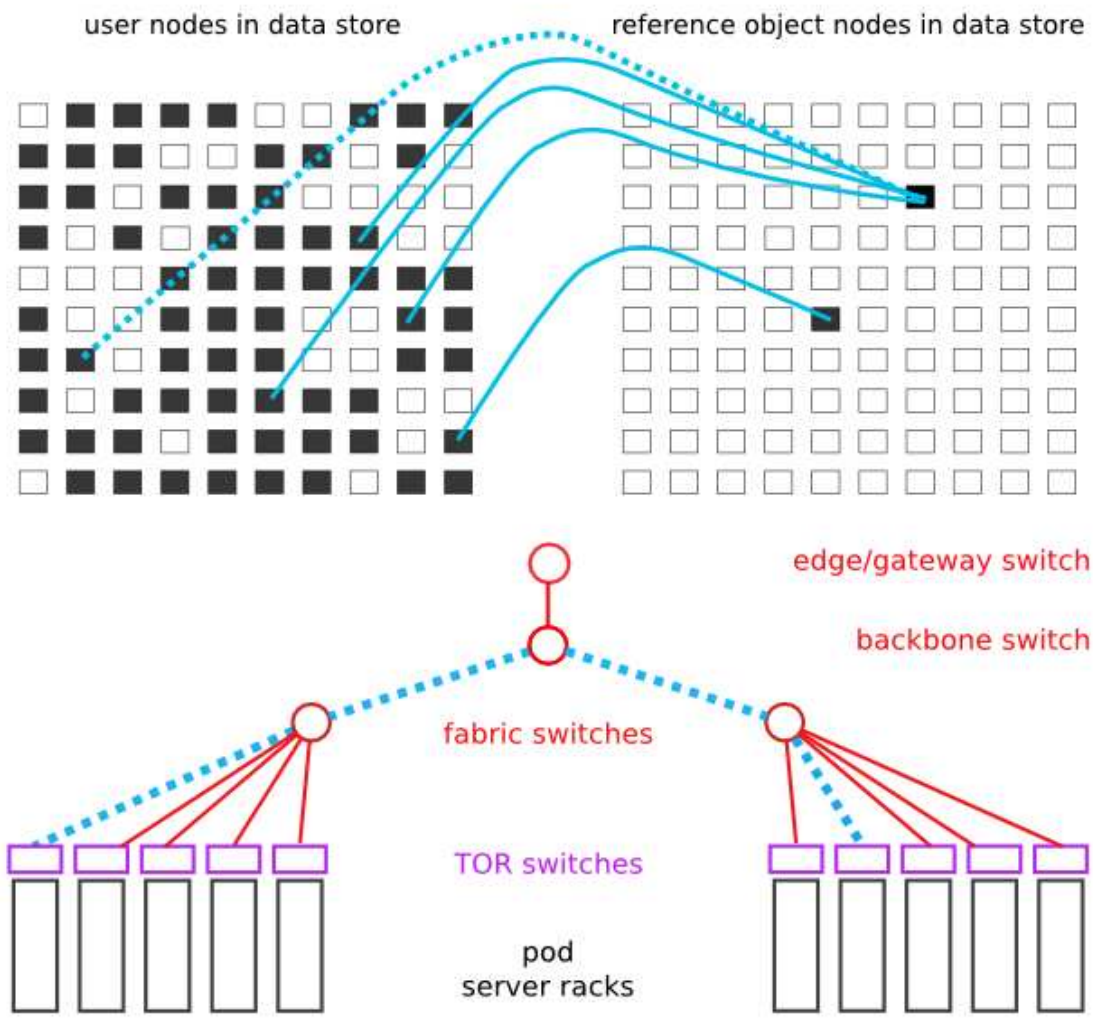


Figure 21: Illustration of the Problem. The crux of the problem: most updates generate nonlocal network traffic, for 34-90% of the network traffic related to referrals such as posts and *likes* due to their uneven, Zipf-like distribution. For clarity, this point is demonstrated with just one set of pods communicating to achieve the dashed-line data store update in Figure 20.

is to organize communication flows to utilize the resources within a single pod. This requires two hops for communication between racks instead of the four hops for flows between pods, saving half the communication energy and utilizing only $\mathbf{3/5}$ or 60% of the switch and link resources used by between-pod communication.

Organizing computation to occur within pods rather than between them is an optimization that Facebook surely already does for performance reasons; the above argues that they should also do so for energy reasons as well. Further complicating this is that not all processes fit on one pod, and that pod size is dictated in part by the number of connections or ports available on the network switches, on which there are finite limits. There is a fundamental architectural limit to the number of ports that the fabric and TOR switches can support; Facebook's design has a physical limit of 48 or 96 racks in a pod [167].

The average and worst-case k of the *k-neighborhood* of friends is a natural minimum sizing mechanism for provisioning within pods; however, many of the services run are only partially dependent upon this relationship to the Facebook graph. Data analytics that Facebook runs nightly might arrive at a better estimate. Andreyev stated that the data center also provides *virtual clusters* to Facebook programmers for the ease that the abstraction provides; these can contain more compute nodes than fit within a pod, or potentially nodes from multiple pods, and the provisioning is abstracted out of the programmers' control [167]. Considering energy efficiency, the mechanism for provisioning such clusters should group compute nodes onto the same cluster as much as possible.

Another potentially significant source of energy savings from arranging computation to fall within pods is to use the large body of information known about Facebook users to group Facebook users around the data store items that they are likely to refer to, with the goal of fitting as many of these users onto the same pod where the data store items reside. Graph updates linking content to Facebook users could then operate over these interest groups. The main benefit of this is that the compute clusters can be tasked based on a goal of clustering together similar access and refer-

ence patterns, so that updates tend to utilize local in-pod communication, which uses 40% of the network resources and 50% of the energy of between-pod communication. Facebook is already performing these analyses for the sake of target marketing; they could also provide an energy savings if the analyses were reflected back into how groups and the corresponding clusters were set up, in general, better-reflecting the connectivity of the underlying entity/relationship graph.

6.5.2 50% Energy Reduction by Promoting Popular Data Store Objects Up a Level in the Network

One potential change to the FDC architecture is to explicitly store and update the most popular 2-6% of data store objects at the backbone of the network. Updates to the most popular items would go out from all or many compute nodes down below the fabric, and be absorbed at the backbone, cutting traffic by half.

This is a direct approach to removing a large amount of traffic from the network that is due to popular items, and would have just that effect. However, it may or may not be difficult to adopt within Facebook. This may require a completely different abstraction from Facebook software programmers—or it might be handled just that way—as an abstraction, hiding at a lower level of the software stack *where* the specific elements of a graph- or data store-centric view are stored in the network. This may require an additional level of abstraction that inserts the location of the popular item in the network, and a mechanism at that level for promoting and demoting items onto the backbone and back onto their compute node as their popularity wanes. It may instead require Facebook hardware engineers to intercept and redirect packets from their intended compute node destination; or it may inevitably morph over time and effort to recreating a pod at the backbone, with subsequently little effect on energy efficiency. The simplest model perhaps is one of having an object cache at the backbone that is kept consistent with all other caches.

Facebook designed and built and completely controls its network hardware; these boxes contain a

micro-server as well as a lot of memory and storage [16]. Facebook could provision the backbone nodes with sufficient high-speed cache, memory and storage. There could be actual compute nodes residing within the backbone that are added to each cluster and hold the most popular items, or these could be virtual compute nodes or specialized representations within the backbone network boxes. When provisioning a virtual cluster as is done now, the portion of that cluster that houses the popular nodes could be satisfied by the backbone virtual or nonvirtual compute nodes.

This articulation at the backbone would remove the right-hand-side communication in Figure 21, thereby reducing by half a large proportion of traffic, using half as much energy for the 90/10 split of traffic related to popular nodes, or 17-45% less energy overall. A similar solution was proposed for serving popular content from data centers by Mandal, and they found a 15-18% overall reduction in the energy cost of delivery of IPTV to users across the network [88].

6.5.3 34-90% Energy Reduction through a Connected-Subgraphs Representation of Popular Nodes

Another interesting solution is to break a popular node into local representations of the node that are kept in each pod. Like the solutions above, this results in no less traffic to update the most popular data store objects; however, the traffic path is predominantly local within-pod communication at all pods, which costs 50% of the energy and 60% of the network resources. What else is interesting and potentially useful about this approach is that it evens out network traffic, eliminating the 90-10% traffic distribution at the upper network level entirely, because all pods contain the element they seek and are adding links to. This lends well to the FDC scaling model of homogeneity and would likely be an all-software solution to a physical networking problem.

The graph representation that may be most natural for this *compound object* is one of a hub-and-spoke subgraph G, with an extra designation for each node of the subgraph, and the corresponding property that certain operations such as searches for edge membership must be performed over

the entire subgraph. This could conceivably be achieved fairly simply in certain cases. Insofar as Facebook already uses its Unicorn index [166] to find content later, this existing mechanism could be used to find the current *network location* of content as well. The goal here, however, is not simply to find the content but to fracture it where convenient, and gather it up when necessary; this however could be done potentially through bookkeeping using a (network-cognizant) structure similar to Facebook's Unicorn index and representing the node entity as multiple lines within the structure for which a $\boldsymbol{union}$ step is explicitly performed before certain logic operations. For example, when checking for the presence of a particular edge in the subgraph $\boldsymbol{G}$ representing the popular node, the check is performed over the union of all nodes and edges within $\boldsymbol{G}$.

In the perfectly symmetric implementation, each pod contains a node of the subgraph $\boldsymbol{G}$, one of the 46 or 98 subnodes that represents the object. One node contains the hub of $\boldsymbol{G}$, and the union function that combines all separate aspects of the full object is performed whenever a complete view of this object is required; this draws each individual representation through the network. The full representation of the object could also be periodically cached at the backbone, a solution that may in practice be paired with the backbone articulation proposed in Section 6.5.2. The difficulties from a lower-level-software programmers' point of view are likely similar to those discussed in Section 6.5.2.

The energy savings is difficult to assess for this because it depends how often a consistent representation of the object is required, for which the independent representations must be gathered. Strictly adding links to an object does not require a consistent global representation, merely a consistent local one. For bracketing purposes, it is certainly not as high as the 34-90% reference frequency itself from the popularity distribution, because there is an extra step required to consolidate the separate representations when and where the full object's consistent data representation is required. But the important result is that the typical cadence $\boldsymbol{c}$ of this is as-needed rather than per-link to the entity, and the resultant network traffic is spurred by the explicit request to read or

write subgraph nodes of G that are not on the local pod.

6.5.4 RDMA Approach to Maintaining Popular Data Objects

Remote Direct Memory Access (RDMA) technology allows data to be written from one compute node directly into the memory of another compute node without running through the entire IP and TCP stack, bypassing many of the operating systems and network stacks that slow down transfers by using instead its own transport layer [182, 183]. A potential enhancement to the Facebook architecture is to implement data store reads and writes in RDMA using something such as QDR Infiniband, or to use RDMA over Converged Ethernet (RoCE), which uses the physical layer of the Ethernet stack rather than a native Infiniband network.

Infiniband is another meshlike network that is an alternative to the Gigabit Ethernet network used within FDC. This could replace the whole data center network, or could be treated as a completely separate network that provides a common data substrate to all the pods that the compute nodes can write to quickly. In this dedicated-network solution, the data center network is not utilized to perform the data store updates; they are on a separate read/write data plane of the data center.

Each compute node would need to contain an appropriate network interface card that lets it talk to the data store. With something like Micron's Hybrid Memory Cube (HMC), that has an array of 10GHz high-speed channels and the ability to write at 240 GBps, all data can be treated as local, because compute nodes can access it at near-memory-access time [184].

Alternatively, just a subset of nodes could be promoted to this specialized network, since it is only around 2-10% of the graph of all entities that qualifies as "popular" and deserving of special treatment. All compute nodes would then have equal RDMA access to the popular object- and entity nodes within the data store and a dedicated network to update those objects.

Like the earlier idea of backbone articulation in Section 6.5.2, the dedicated-network approach

uses specialized hardware and network to directly address the 90/10 split in traffic flows. Doing so adds complexity to the FDC architecture and its layout that would need to be weighed carefully against the benefits. From a Facebook programmer's perspective, memory-rate updates to remote memory locations could provide the most popular data objects "in memory," regardless of their network physical location.

With regards to going with an RDMA approach for the entire network, Facebook already indirectly achieves this by running the existing 90/10 split through its existing network; the key difference here is any energy savings from avoiding layers of the TCP/IP stack. Although there are some performance numbers available, it is not clear how much energy could be saved via direct Infiniband, or by RoCE. Trying something like RoCE as an experiment within a part of Facebook's highly redundant FDC would be an interesting experiment that would provide energy savings numbers. One reason for Facebook's choice of Ethernet is the economies of scale from its widespread adoption and improvement from its evolution to 40Gbps and 100Gbps [16]; thus, the RoCE approach where the underlying physical network remains Ethernet may be the most cost-effective way of providing what may be faster and less power-intensive updates to popular items within the data store.

6.6 Discussion

Open architectures like the one recently unveiled by Facebook allow a detailed assessment of the energy efficiency of commercial data centers. This chapter explored the fit of Zipf-like distributions typical of network traffic, to updates of user pages and the entity graph, for the new Facebook data center network architecture. We find that network resource consumption could be reduced by as much as 40-50% through several changes, either to the software, or to the data center design. Of these, employing a connected hub-and-spoke subgraph representation for each popular node, with each pod operating locally on its node of the subgraph, appears to hold the most energy savings potential.

Facebook engineers' recent redesign of Facebook's data center compute node interconnection networks addresses a number of concerns, such as scalability, and a desired reliance on inexpensive commodity hardware. Of these, one that is reported as a very significant motivation is the exponential growth in machine-to-machine traffic that Facebook engineers have witnessed over the past few years [167]. Exponential growth in machine-to-machine traffic was a primary concern in the designing of Facebook's new data center fabric architecture (FDC).

This chapter examined the order, or nonrandomness, within the traffic, the so-called 90/10 split, and whether and how the FDC design could capitalize on that order to save resources. The nonrandomness affects a large percentage of incoming traffic from referrals to web content, which follow a Zipf-like distribution, with popularity very high for a relatively small number of items and tapering off quickly. This chapter examines in particular the fit of Zipf-like distributions commonly reported for Internet traffic to the corresponding updates to the Facebook social graph of people and objects, from adding objects and linking to them. Within typical Zipf-like traffic flows, where as much as 90% of traffic refers to as little as 10% of the data store items or graph nodes, and as much as 83% to the top 5%; these updates dominate network traffic to reach a small set of compute nodes where the target data resides.

The inherent architectural problem appears to be the imperfect and divided packing of the graph onto the compute nodes, the network of which is arranged as the leaves of $\boldsymbol{n}$ duplicate hierarchical trees. Facebook's data update and storage plane design is flat: all compute/storage nodes have the same priority, and are a fixed identical distance from each other. While the new fabric network connectivity provides greater bisection bandwidth to benefit a wide variety of communication patterns, the traffic-efficiency for this particular pattern of between-pod communication is choked by the number of backbone switches and by demand for the top-of-rack (TOR) switches on the hot target pods. What this pattern benefits from isn't greater bisection width *per se*, because the source area is so much larger than the target area of the communication, it's shaped more like a funnel,

and could make use of an ability to focus more resources at the fabric and backbone switch layers.

Several remedies to this were proposed here and characterized in terms of their energy efficiency. These appear to provide 40-50% energy reductions for 34-90% of network traffic (14-45% total savings), and in one case potentially eliminate some substantial fraction of the 34-90% of machine-to-machine referral-based traffic altogether by partitioning the popular object across the network. These are rough estimates: what this means in practice depends on what fraction of incoming, outgoing, and machine-to-machine traffic is due to *likes* or referrals, and on obtaining the actual distributions for referral-based traffic seen within a Facebook data center.

This chapter also presents some objections to the common and problematic practice of citing facility overhead (power utilization efficiency, or PUE). We argue that simply tallying the power consumed by the IT equipment within the data center is a poor stand-in or proxy for the efficiency of the data center's design, or its fit to the work flows for which it was designed. A better metric of data center efficiency relates what is gained to the power consumed to gain it; what is gained is work done within the datasphere. The efficiency with which this is gained is here assessed based on other ways to perform the same work.

Chapter 7

Estimating the Impact on Energy Use

7.1 Introduction

Policymakers may want to know what the societal impact is of the choices investigated here, in terms of their effect on the energy grid and world power consumption. Saving energy within the global ICT requires both good ideas and wide application of them. The various strategies for conservation of bits within the network described in Chapter 3–6 have differing applicability within the ICT. Thus, the value of an idea is not solely from its energy-saving potential, but truly from its efficacy from an energy-saving standpoint, which is tantamount to energy savings multiplied by applicability—and ultimately not measured by potential applicability but by actual, wide usage. New ideas may be applied through the structuring of policy, or through the choices made by informed individuals and organizations that decide to use more energy-efficient approaches.

This chapter does not go into a bona fide assessment of applicability for each technique. What it does is gauge the size of the energy pool that each of these methods represents if applied fully, and makes an attempt to temper that result with a discussion of the current limitations of usage of each technique. This chapter estimates both the raw effect each has on energy consumption, and also the circumstances under which, or frequency with which, each can be intentionally applied.

7.2 Methods

7.3 Global-scale Energy

This section introduces and motivates another energy metric calculated for applications within this study. Components are intended to satisfy human activities; the effort involved in optimizing the *energy footprint* of a component has far less *energy impact* if the component is one that few people use. The macro scale effect of activity energy efficiency is also an important quantity to derive to inform decisions about what activities and activity types to work further on in order to achieve a larger macro scale effect. The interest here is generally not in precise numbers but in ballpark estimates, drawn or outlined in sufficient fidelity for uncovering orders of magnitude differences between activity types and between components used on a large-scale network, in order to identify high-energy-consumption activities and components.

Energy impact is used as a container for estimating this more global quantity, within which the focus might be the global usage energy associated with an activity; or the focus might be within a sub-population of the global population, such as European IPTV subscribers. For this it is useful to assess two quantities, the impact of the popularity of the general activity for which a particular component is one solution, and the impact of popularity of a particular component within the landscape of solutions that support the activity. The larger goal is a rank ordering of activity types in terms of their total energy consumption, and within each another rank ordering of components that support that activity, just as, for watching or reading news, within the market for video and audio content players, Windows media player on Samsung Galaxy tablets represents some percentage of the market.

To assess the energy consumed by all individuals engaged in an activity, such as watching short video clips, a bottom-up approach can often be used in which the individual effects from the

frequency of use of the particular salient activity details relevant to a component-based analysis (algorithm/hardware/software/firmware) are accumulated into an overall energy associated with the activity within the population of interest. The *population energy* P_e associated with an activity is given by

$$P_e \propto \sum_{i=1}^{m} i_e \tag{7.1}$$

where m is the size of the population of interest, such as the global television audience, and for each individual i, i_e is the total energy utilized over the timespan of interest by the individual. There are several ways to obtain that measure to varying degrees of fidelity; for the entropy reduction effect estimation presented here, it is estimated based on frequency of engagement in the activity and average length of time per engagement, or

$$i_e \propto ftubf_e \tag{7.2}$$

where f is the frequency with which the individual engages in the activity over some time period, t is the average length of time spent in each engagement, u is a refactoring over a common time unit used to compare different activity types with different time characteristics, b is the bit rate of input to the component [1], and f_e is the energy footprint in Joules/bit of the component. The quantities f, t, and b can be means or weighted distributions. The analysis used to arrive at a rough estimate of energy impact of entropy reduction for this study uses published information on television subscription rates and hours of television consumed per night as a convenient approximation of the larger, more general global demand for compressed content via the telecommunications net-

[1] More precisely, this is the achieved bit rate of the activity, not the component, due to a number of factors and represented here simply as the minimum of the bit rates from factors such as user-interface-limited input stream (such as in the case of a user reading a book online, the number of words per page times the number of pages the user reads divided by the duration of reading), or due to network connections limiting the input rate to the component (such as in the case of a perceptibly slow network connection) , or from the throughput of the component when supplied with input matched to its operating rate.

work; certainly, streaming video represents the most energy-intensive share of the global demand at homes or residences. The data, collected annually, provides averages for the frequency and length of engagement in 17 countries among the television-home-viewing population [139].

If data on frequency of use is not available for the activity or task, the population energy is instead derived in a top-down way from the total market for this activity factored by the approximate market share of the specific application, (e.g. Windows media player's share of the short video clips market). Given that the activity definition resolves down to the specific hardware/software/ firmware configuration utilized via the component model a specific hardware/software/firmware instance, this is further factored by the configuration's share of the market if known (e.g. Windows media player's share of the short video clips market on Samsung Galaxy tablets factored by the number of owners of Samsung Galaxy tablets among Verizon customers in North America). For this market share-based estimate of the total energy associated with a particular hardware/software, $\boldsymbol{P_e}$ can be defined as

$$\boldsymbol{P_e} \propto \boldsymbol{m}[\boldsymbol{c_1}]\boldsymbol{c_2} \tag{7.3}$$

for the population of interest, such as global, country, or age group, where $\boldsymbol{m}$ is the total number of individuals constituting the market for this activity type. In some cases it may be preferable to consider $\boldsymbol{m}$ a superclass, such as the total streaming video market, rather than a type class, in which case the following term $\boldsymbol{c_1}$ is used to further divide the superclass. $\boldsymbol{c_1}$ is the optional fraction of $\boldsymbol{m}$ that is occupied by this activity type, such as short video clips' share of the total streaming video market, and $\boldsymbol{c_2}$ is the fraction of either the total market or the activity class market—whichever quantity is derivable from the market information known—that is occupied by the specific hardware/software configuration represented in this activity, such as Windows media player run on Samsung Galaxy tablets.

Table 9 summarizes the quantities described above, their definitions, and their sources if outside

the present work.

quantity	symbol	definition	units	source
processing energy	p_e	$p_{e_{compression}} + p_{e_{decompression}}$	Joules/bit	[127, 128]
communication energy	c_e	$c_e \propto d, n$	Joules/bit	[8, 78, 138]
bitstream length	d	|bitstream|	bits	
compression ratio	CR	$d_{original} \div d_{compressed}$		
energy footprint	f_e	$f_e \propto \sum_{i=0}^{n}(p_{e_i} + c_{e_i})$ (4.1)	Joules/bit	
nodes	n	terminal or network node		
energy-efficient compression	T	$\frac{c_{e_{i-1}}}{CR} \leq p_{e_i}$ (4.5)	true/false	
energy impact	P_e	$P_e \propto \sum_{i=1}^{m} i_e$ (7.1)	Joules/bit	
population size	m	population of interest	people	[139]
individual energy	i_e	$i_e \propto ftubf_e$ (7.2)	Joules/bit	
frequency	f	frequency of activity engagement	per time unit	[139]
common time unit	u	refactoring for comparison	hours, minutes	[139]
bit rate	b	number of bits per engagement	bits/u	[8, 78]

Table 15: Defining Terms and Notation. Defining the quantities and symbols used for the energy analysis

7.4 Results

The energy impact was assessed in rough terms based on residential Internet use data available for 17 countries. Given the world Internet use population and the average number of hours of content streamed daily to households, a rough estimate of the subscriber population's daily bit consumption was constructed. The assumptions built into the model are summarized in Table 16. The subscriber population is the rough number of households globally engaged in watching television per night via the telecommunications network; the average hours of television-watching

Table 16: Energy Impact Parameters. Quantities and assumptions used to generate the *energy impact* for each compression technique

1.46 bn	subscriber population
3'41"	average hours of streaming content per day
500 Mb/sec	access rate
22.5 mo	months of output, 850 MW power plant [185]
	at 33% efficiency, 6% loss in transmission [186, 187]
900 g/KWH	CO_2 emissions for coal power [188]
400 g/KWH	CO_2 emissions for gas power [188]
5 g/KWH	CO_2 emissions for nuclear power [188]
131.76 MJ	energy per gallon of gasoline
19	gallons of gasoline per barrel of oil [189]

comes from a per annum study of 17 major countries including European countries, Japan, U.S.A. and China [139]; the access rate used is the typical observed access rate in 2013–2014 for US DSL subscribers, which corresponds well with the projected bit rates in the network analysis used to derive the energy footprint for each component [8].

The impact of choosing the most energy-efficient of the methods investigated is compared here to the impact of the least energy-efficient method; this reframes the problem within the space of global energy supply and energy production, and the carbon emissions that are at stake. The least energy-efficient method is the same standard by which all strategies were measured: full unicast transmission of full-length messages, $\boldsymbol{n}$ separate transmissions from sender to receivers.

Table 17: Energy Impact of Efficient Content Distribution. Before accounting for extra storage energy for replicates. The various energy metrics (power plant outputs and their equivalent in petroleum products) are reported based on the annual difference over the entire subscriber population.

10.0 KJ	household daily energy cost, best case
39.5 KJ	household daily energy cost, worst case
29.5 KJ	annual difference
15.7 TJ	annual difference, entire subscriber population
23 mo	months of output, 850 MW power plant, 40% loss in transmission
12.7 M tonnes	saved carbon emissions, coal power plant
5.6 M tonnes	saved carbon emissions, gas power plant
70 K tonnes	saved carbon emissions, nuclear power plant
327.2 K	equivalent gallons of gasoline
17,219	barrels of crude oil

The Energy Impact of Efficient Content Distribution The cumulative effect of the incremental difference across network devices—switches and routers–from suppressing redundancy among requests for the same content as discussed in Chapter 3 is presented in Table 17 in various ways. The best case represents the average savings fraction observed in Chapter 3 from the suppression of redundant transmissions by storing replicates at key positions within the network; however, actual savings will be lower, as the cost of such storage is not included in the assessment. The worst case is the energy cost of full, unicast transmission from the sender to each receiver, assessed for the Internet in general, using the numbers from Baliga and Tucker discussed in Chapter 4. Despite the complexity of different television programs being viewed by housholds, the effect as assessed on the lump sum, applying the fraction of savings equally to each, under the distributivity of multiplication.

These numbers are also presented in Table 17 using common energy metrics, to better represent

the meaning and impact of choices and policies for the ICT at a global scale. The annual difference between the best and worst case in Joules was translated to months of power supplied by a power plant; the power output of Three Mile Island (TMI) nuclear power plant near Harrisburg, Pennsylvania USA—the site of a major nuclear accident in the 1970s due to operator error, as shown in Table 16—is 850 MW, which was adjusted for the effective output, from a power loss of 6% in power line transmission to households [187], and a typical 33% efficiency of the plant in terms of how much energy it actually transfers to the power grid [186].

Using these assumptions, the energy impact of using efficient content distribution to satisfy the load on data networks from nightly television viewers results in the ability to shut down TMI nuclear power plant for almost two years (23 months)—or to shut down two similar power plants for one year—and still supply the same viewing audience. This saves 1629 MW of power within the energy grid, and it reduces greenhouse CO_2 emissions by 70,000–12.7M tonnes; these emissions estimates are based on industry estimates of carbon emissions for the range of typical power plants, with nuclear plants on the low end, and coal-fired plants on the high end of the range [188].

Another way of framing the choice is that the savings associated with exploiting redudancy among people's use of the Internet with more efficient content delivery represents the energy embodied in 17,219 barrels of oil used to produce 327,200 gallons of gasoline: enough to supply a small country for a day [190].

The Energy Impact of Entropy Reduction For entropy reduction discussed in Chapter 4, the best case is constructed using the nJ/bit cost of the entire processing and transmission chain of the most energy-efficient application in this analysis, bpzip2 at compression level 8, multiplied by the number of bits streamed (time period $\times$ access rate). This in effect applies the compression utility's achieved compression ratio to the bit stream. The worst case utilizes the nJ/bit cost of the

Table 18: Energy Impact of Entropy Reduction. The various energy metrics (power plant outputs and their equivalent in petroleum products) are reported based on the annual difference over the entire subscriber population.

10.6 KJ	household daily energy cost, best case
39.5 KJ	household daily energy cost, worst case
28.8 KJ	annual difference
15.4 TJ	annual difference, entire subscriber population
22.5 mo	months of output, 850 MW power plant, 40% loss in transmission
12.4 M tonnes	saved carbon emissions, coal power plant
5.5 M tonnes	saved carbon emissions, gas power plant
69 K tonnes	saved carbon emissions, nuclear power plant
320.3 K	equivalent gallons of gasoline
16,858	barrels of crude oil

least energy-efficient approach found in this analysis, which is no compression at all.

In practice, the actual bitstream that viewers or subscribers receive is already compressed; however, here that is irrelevant, because the goal is not to assess the change to current energy use but rather to assess the contribution of compression to total energy in large-scale networks. Comparing the best case to using no compression reveals its contribution.

The energy impact of this particular form of entropy reduction represents the ability to shut down TMI nuclear power plant for almost two years (22.5 months), or shut down 1.9 similar plants for one year, and still supply the same subscriber population, saving 1600 MW of power within the energy grid.

In terms of carbon footprint of this activity at a global scale, using the 2013 analysis, the annual CO_2 emissions savings from this reduction in power plant production reduces greenhouse CO_2 emissions by 69K-12.4M tonnes, using industry estimates of carbon emissions for various types of

power plants to produce the equivalent power of TMI.

Another way of framing this choice of what application to use at content sources—whether they be devices or comparable processors within network servers housed at data centers—is that the savings associated with compressing content and messages before sending them represents the energy embodied in almost 17,000 barrels of oil, and that is when each transmission is assumed to have a unique compression cost associated with it; for popular content such as most media, the energy savings is the equivalent energy of an additional 243 barrels of oil annually, or another week and a half of operation of a power plant, and with it the avoidance of another 1K–190K metric tons of carbon emissions.

This analysis was done using the 2013 energy footprint data; for 2015, when network per-bit energy is 100% lower, but partially occluded by greater consumption, as average access rate is 30% higher, keeping the population of devices fixed, the the result is a savings of 14.5 months at 850 MW–still representing roughly a year of the energy supplied to households from a power plant such as TMI. However, a steady increase in demand—from higher bandwidth to homes, from exponential growth in traffic, and from added devices and added households—is the major force motivating this research, and will likely keep the potential energy savings closer to the 2013 analysis.

The Energy Impact of Multicast Transmission The cumulative effect of delivering nightly television via multicast is presented in two tables, Table 19 and Table 20, using two different estimates of multicast's energy effect that were discussed in Chapter 5.

The best case represents the average savings fraction from delivery of content via multicast transmission; the worst case is the relative, full energy cost of unicast delivery. Despite the complexity of different television programs being viewed by housholds, the effect as assessed on the lump

Table 19: Energy Impact of Multicast. Energy impact of multicast within the autonomous system networks examined here. The various energy metrics (power plant outputs and their equivalent in petroleum products) are reported based on the annual difference over the entire subscriber population.

0.2 KJ	household daily energy cost, best case
39.5 KJ	household daily energy cost, worst case
39.3 KJ	annual difference
21 TJ	annual difference, entire subscriber population
30.7 mo	months of output, 850 MW power plant, 40% loss in transmission
16.9 M tonnes	saved carbon emissions, coal power plant
7.5 M tonnes	saved carbon emissions, gas power plant
94 K tonnes	saved carbon emissions, nuclear power plant
436.2 K	equivalent gallons of gasoline
22,960	barrels of crude oil

sum, applying the fraction of savings equally to each television program, under the Distributive Property of multiplication.

These numbers are also presented in Table 17 using common energy metrics, to better represent the meaning and impact of choices and policies for the ICT at a global scale. The annual difference between the best and worst case in Joules was translated to months of power supplied by a power plant and its production of carbon emissions; also, it was translated to gasoline and barrels of oil used to produce the gasoline.

Using the assumptions in Table 16, the energy impact of multicast transmission of nightly television results in the ability to shut down TMI nuclear power plant for 2-2.5 years (25–31 months)—or to shut down four to five similar power plants for one year—and still supply the same viewing audience. This saves 1800-2175 MW of power within the energy grid, and it reduces greenhouse

Table 20: Energy Impact of Multicast II. This formulate uses Chuang and Sirbu estimate for the Internet in general [150]. The various energy metrics (power plant outputs and their equivalent in petroleum products) are reported based on the annual difference over the entire subscriber population.

6.9 KJ	household daily energy cost, best case
39.5 KJ	household daily energy cost, worst case
32.5 KJ	annual difference
17.4 TJ	annual difference, entire subscriber population
25.4 mo	months of output, 850 MW power plant, 40% loss in transmission
14 M tonnes	saved carbon emissions, coal power plant
6.2 M tonnes	saved carbon emissions, gas power plant
78 K tonnes	saved carbon emissions, nuclear power plant
361.4 K	equivalent gallons of gasoline
19,022	barrels of crude oil

CO_2 emissions by 78K–16.9M tonnes; these emissions estimates are based on the spread between the two tables and industry estimates of carbon emissions for the range of typical power plants, with nuclear plants on the low end, and coal-fired plants on the high end of the range [188].

Another way of framing the choice is that the savings associated with exploiting redudancy among people's use of the Internet with more efficient content delivery represents the energy embodied in between 19,022–22,960 barrels of oil used to produce 361.4–436.2K gallons of gasoline: enough to supply a small country for a day [190].

7.5 Discussion

For the common activity of nightly television watching, the cumulative energy impact of a global policy of using a method with a smaller energy footprint was significant, liberating for other uses the annual output of two to five power plants, and providing the energy supply equivalent to a small

country's oil use for a day, at no change in the quality of service experienced by viewers. All of the methods examined here, entropy reduction, multicast delivery, and efficient content distribution before accounting for storage cost, provide between 73–100% of this effect on the global energy supply. Chapter 8.3 outlines how these effects can be maximized.

Chapter 8

Conclusion

According to the U.S. President's Council of Advisors on Science and Technology, the need to address a worldwide energy shortage is one of the major adjustments required of this era [191]. As the number of devices connected through the global telecommunications network continues to grow, less energy-intensive practices and components would accomodate this growth more gracefully; otherwise, growth in the ICT must be met outside the ICT, with decreases in energy use in other aspects of socio-economic systems.

While power optimizations for individual devices have always been a design consideration for some classes of devices—notoriously, handheld mobile devices, global or *net* conservation of energy across the separate concerns that combine to form the global telecommunications network and the ICT-at-large is rarely addressed; as Kilper points out, the metrics for network design, for instance, have been operational complexity and cost [15]. The history of the global ICT and its separation of concerns cause a number of impediments to this, having the effect of compartmentalizing the larger system into separate markets, separating the stakeholders into separate economic, political, social, and academic affiliations and organizations, and separating study into distinct boundaries within separate fields of expertise. This work considers the larger system, and examines the competing tendencies among two beneficial and often-used methods of improving network

performance; here, the most relevant performance metric is conserving the path cost of transmissions in terms of number of hops, either by replicating the content closer to recipients, or by sending bulk messages to all recipients that are split into individual messages that travel unique paths only at the final hops of delivery to the recipient. Conserving hops conserves the number of bits that are transmitted through the network, and saves energy as a result.

In an attempt to illustrate entropy reduction's role in energy consumption on large data networks, one might compare the entropy of two different processing chains on the same hardware, such as two scenarios for delivery of home movies using two video compression techniques where one more computationally intensive technique results in a more-compressed video stream. What are the net energy effects of trading the increase in computational energy and latency of the more aggressive compression method for the transmission bandwidth reduction due to data reduction, or greater entropy? The static aspects of this question are directly addressed in this dissertation, for the more generalized notion of entropy reduction of content delivered to home residences through the application of compression techniques to the data streams. The energy used at the leaf nodes is directly compared to the energy used within the network by various compression schemes. How might these changes play out at full network scale with millions of users, changing for instance the network traffic load and the energy grid draw and device battery draw from greater computational energy use on the network periphery? The problem changes when one examines the energy profiles of an ensemble of users.

8.1 Primary Contributions

The primary contributions of this work are in finding support within several specific areas for the hypothesis that *the minimum-energy solution is not the sum of local minimum-energy solutions*;

First:

> *Extra energy spent at leaf nodes saves much more energy within the network.*
>
> *local:* higher leaf node power use for processing
>
> *global:* much lower node energy use system-wide for transmission

Second, for a sample of networks from the range of network types within the ICT:

> *Redundancy among requests provides a large pool of energy within the network.*
>
> *local:* one message from source to sink, $\boldsymbol{n}$ singleton messages
>
> *global:* of the link transmissions associated with these:
>
> - about 74.6% are redundant, on average
> - about 99% are redundant, for simultaneous requests
> - 34-90% are redundant for the data center case examined

Of these, the first was previously stated, by Barr and Asanović in 2003 [125, 126]; the contribution here is updating the ratio of transmission energy to processing energy a decade later, where it has dropped from 1000:1 in 2003 down to 76:1 in 2013, and potentially down to 10:1 within the next five years, with the projection from Baliga [78] that transmission may again be another order of magnitude lower in the next 5-10 years. Barr and Asanović pointed out the need to re-calculate these numbers periodically as computer and networking hardware change. This finding is important to the research community in the continuing dialog about where in the global ICT to pursue energy efficiency gains, within which this finding is a valuable contribution, as it stands in contradiction with the earlier result by Kothiyal that Kilper takes as evidence that compression may be worthwhile only for popular items [15, 131].

8.2 Major Findings

An Energy-Efficient Storage Strategy

There appears to be an enormous conservation potential from capitalizing on the redundancy among user requests within networks. Looking also at where to store popular content based on the redundancy found within traffic logs, as was done in Chapter 3, and based on the redudancy found within traffic distributions, as was done for data center networks in Chapter 6, gleaned information useful for devising more efficient global- and network-specific strategies for the storing of content as it flows through networks. This pattern has a distinct pattern as well within the specific Internet networks investigated: typically 30-60% of the energy within the network associated with redundancy could be captured by storage of content items at four to ten nodes, when examined using the benefit of perfect hindsight, as an indicator of what could potentially be achieved; this is a small handful of locations, representing less than 1% of the network in many cases.

Comparative Analysis

Table 21 provides comparative analysis of the effect of all aspects of the ICT examined here, and their effect within a small sample of networks from the three functional categories of networks on the Internet: *Tier 1*, *Transit* and *Stub* networks. The data center network improvements discussed in Chapter 6 fall under the general category of efficient content distribution, with the one possible exception of the proposal to use RDMA network access technology: but that again, is efficient content distribution, only abstracted.

If we were to characterize *how* each of these approaches saves energy over the predominant method, in a nutshell, it would be the following: 1) distributing content efficiently over networks cuts the number of transmissions while incurring storage costs; 2) multicast cuts the number of transmissions without incurring any additional storage costs; 3) entropy reduction cuts transmis-

Table 21: Comparative Assessment of All Covered Methods. A side-by-side comparison of the techniques assessed: unicast, efficient content distribution (minus storage cost), multicast, and entropy reduction. All savings reported are savings over the cost of unicast transmission, and for a pool of 100,000 requests for a particular item.

		before-storage				**entropy**	
Network	**unicast cost (J)**	**ECD cost (J)**	**savings (%)**	**multicast cost (J)**	**savings (%)**	**reduction cost (J)**	**savings (%)**
AS 1239	7,057,081	1,191,532	83.1	54,735	99.2	1,721,239	75.6
AS 224	4,597,382	1,058,126	77.0	35,185	99.2	1,121,313	75.6
AS 3549	6,025,916	722,217	88.0	30,223	99.5	1,469,736	75.6
AS 59	1,247,162	352,008	71.8	34,616	97.2	304,186	75.6
AS 2914	6,533,060	747,885	88.6	24,360	99.6	1,593,429	75.6
AS 3356	5,442,846	712,274	86.9	10,335	99.8	1,327,524	75.6
AS 3292	2,524,117	708,866	71.9	3,561	99.9	615,638	75.6
AS 109	6,626,477	717,788	89.2	3,419	99.9	1,616,214	75.6
AS 680	2,674,362	2,362,111	11.7	2,259	99.9	652,283	75.6
AS 3356.1	6,495,726	775,457	88.1	1,811	99.97	1,584,323	75.6
AS 680.1	1,991,771	707,211	64.5	193	99.99	485,798	75.6

sion length.

8.3 Policies Applying These Results

This work can be used to construct a policy for achieving the maximum energy savings out of the conservation strategies investigated, bearing in mind that the limitations discussed earlier apply also to the extent and predictive power of this set of policies.

Data Entropy Reduction: Always Use Use data processing that achieves entropy reduction on the data, before transmitting it, because the processing cost is repaid on the first hop, and the remaining hops are pure benefit. Additionally, for content that may be compressed once and transmitted to many receivers, the benefit is multiplied by further free hops: the decompression penalty is low compared to the transmission gain. This policy is as-of-yet based on limited results for the OMAP processors common in mobile devices running compression utilities; a more extensive energy audit of applications run on mobile devices will permit thresholding specific data processing based on the expected lifetime usage of the data that is produced—thresholding based on number of expected downloads or uses, for example, or the expected longevity of the data. It is better to compress and send, then decompress and use, data that is exchanged as part of a distributed computing process that is distributed over the Internet or ICT at large, such as cloud computing, or peripheral-user-accessed services and applications within datacenters.

Replace Unicast with Multicast as Much as Possible Use multicast over unicast wherever possible—and incentivize its use proactively—since its transmission cost in the sample of real networks was less than 1% of the cost of unicast transmission, and appears to cost between half and 9% the cost of unicast in the Internet as a whole. If the cost of multicast is 1% of that of unicast, as was the case for the networks examined here, there is great benefit to switching over to

multicast networks and the intentional practice of more of a *broadcast mindset* for requests for all popular content.

Content Replication: Devise a Network-specific Strategy This work indicated a small number of nodes, less than 1%, at which the majority of the gain from content replication occured. The specific pattern of number and location of these varied from network to network. The recommendation is therefore that each network be similarly analyzed by those managing it and knowledgeable about its structure and usage patterns to find the best replication sites for content traveling from the set of edge nodes and interior source-node sites within the network, and based on the specific distribution methods that will be used over that network—commonly unicast but perhaps increasingly other methods. Based on this analysis, we recommend that network managers construct a policy based on the replication strategy that will be used, such as router caches, that limits caching to those nodes that fall above an energy benefit threshold, and that limits cache size and caching to sufficiently popular items that the gain in reduced number of transmissions outweighs the cost of replication and storage at the node and within the network. The uncertainty associated with this can be reduced by offering a service to analyze networks, and offering network subnet templates for which this analysis has already been performed: this will take a lot of the effort (or guesswork) out of the issue.

8.4 Discussion of Limitations

The conclusions as to the energy dividend resulting from entropy reduction are not extensible to other hardware and are based on too small a hardware sample to be representative of what is experienced in reality. They serve more as a ballpark measure of the effect. Profiling the energy use of more devices is sorely needed to establish the relative cost of processing versus transmission for specific devices and applications.

The same is true of the wide range of router and switch models used to build the ICT infrastructure: this work relies on a very small number of samples for which detailed energy profiles have been constructed. As underscored in detail by Vishwanath, and as put to good use here in looking at incremental effects of traffic under one scenario versus another, those detailed profiles demonstrated the insufficiency of using wall power draw or vendor specs to assess the byte-level incremental effects not of having the device simply powered up [98].

For data center networks, this work utilized the publicly available information on the typical distribution of Internet traffic by popularity. To more accurately assess the energy efficiency of the Facebook data center network, one needs to know the specific distributions observed within Facebook's traffic.

For assessing content distribution, the work traced the effects of a single content item in the network; between-source effects were not considered, with the problem limited to a single item of content traveling from its original, single source within a network to its destinations. In fact, this work says nothing about the dynamics of network activity. By analysing traffic in hindsight, the present work rendered content distribution over networks as a static problem; and, each transmission was counted as if it were associated with a unique request [1]. Thus, this work ignores dynamic effects of such things as competition for loaded physical resources within the network. It assumes perfect productivity of the work done by the network on behalf of a user, when in fact there may be retransmissions required to fully satisfy a user request. Nonproductive work done in the network on behalf of a user includes transmission failures, dropped packets, buffer overruns at intermediate and destination nodes, cache competition, and node failure.

This analysis captured one important aspect of the ways in which many people's use of the ICT interact with each others' network usage, but not the important aspects of competition for resources.

[1] This is necessary in part because of the lack of knowledge present, from the lack of explicit representation of the destination device and end user in the network dataset from Mérindol and Marchetta

The dynamics are complex; not only will there be nonproductive work due to competition, but, on a loaded network, there will also likely be a multiplier benefit to the traffic-clearing effects of the methods investigated here: significant reduction in traffic from using multicast, and also from efficiently replicating content, and also from reducing the number of packets associated with a request via entropy reduction. Moreover, the unreliability that end users experience—particularly at home and other bandwidth-constrained environments—could have a silver lining in terms of energy conservation if traffic control techniques increasingly synchronized retries and exploited multicast to do so.

8.5 Broader Impact

This work underscores the important point that energy savings is potentially much larger from taking a whole-system approach to ICT and telecommunications networks, and this is possible only by organizations working across customary boundaries and industries.

The chief benefit to this work is in dispelling ignorance about the energy consequences of existing techniques in use today for communicating information; also, the new techniques further developed here may prove to significantly increase energy savings. An important follow-on goal for this work is dissemination: disseminating this information to decisionmakers and energy conservation advocates is needed to fully realize its benefit.

The findings from this work are widely applicable to policy formation for communities' ICT and telecommunications networks, as well as to inform future decisions within individual organizations, private and public. This work in general supports fact-based or *science-based* policy, and it fills a significant gap in the literature, for similarly cross-cutting research is rare across the many concerns that organically form the global web of telephony and computer networks.

The general tendencies, specific optimizations, and even the simple tweaks discovered by this

work could potentially become part of real systems in use in the future, paying dividends in energy savings year after year as part of the infrastructure.

Knowing, for instance, that a particular application run on mobile phones increases their energy use by 2% and reduces network energy use by 74% is valuable information that can be used to make more informed choices, not only by government and industry but also by communities and consumers. The public, knowing that multicast delivery mechanisms could greatly reduce energy use, might opt to switch *en masse* to applications that synchronously download daily media bundles. By working together in this way, individuals could drastically reduce the resource intensity of a common and popular activity—capitalizing on its very commonality—leveraging goodwill for the greater public good.

Examining energy consumption from the holistic standpoint taken here is important, because the energy savings opportunities are potentially much larger than any separate entity can achieve within only its realm of influence. While the shared concern and transparency required to enact some of the recommendations that arise from this work may be difficult to achieve—perhaps politically or practically, or from a need for competitive advantage among companies through secrecy—analysis of the energy consequences of existing techniques, and ranking various approaches in terms of their energy consumption is a necessary step towards the kind of informed decisions by each of these separate entities that could, collectively, significantly reduce global energy consumption for this arena that represents 8–10% of global energy use—and therefore 8–10% of annual energy production.

Another impact this work has is lending evidence and qualifying conditions as to whether and when Netflix, Verizon, and other companies with a large customer or subscriber base should choose entropy reduction, multicast or CDNs if they (or if the communities they serve) want to emphasize energy conservation.

From a policy standpoint, knowing not only that one content distribution method is better, but also *how much better*, supports advocacy for changes to policies, best practices, and standards at multiple levels: the level of individuals and their social networks of influence, of communities and countries, of industries, and even of global organizations.

8.6 Future Work

A larger effort to profile the energy use of popular applications on user devices is needed to extend the entropy reduction analysis to the common case. Further work is also needed to profile the actual energy use on loaded network hardware, as was formulated and begun by Vishwanath [98]; this would permit us to extend the analysis to cover common cases, and thereby increase the utility of the efficient content delivery work. Delving into the murk of lossy entropy reduction techniques may allow a lower-energy solution in the entropy reduction work of an acceptable quality to users at a fraction of the energy cost.

The method of network analysis developed and used here is amenable to composition of networks, permitting the tracing of content through multiple regions of the ICT; this would permit the explicit modeling of access networks, for example, and the full path from a server in a core network through several different autonomous systems to its ultimate destination. This would entail the incorporation of autonomous system interconnections and/or a description of the Internet Exchange Points between them. As no attempt was made to optimize the algorithm for exhaustive global pruned search for the best replacement or trim-node, a more efficient and perhaps less accurate method such as a technique based on the symbolic replacements that occur in compression schemes to *compress* the traffic logs might be used as a less accurate but much faster approach. Techniques from gene sequence alignment may also be useful here.

Covering more of the path from source to sink as discussed above would permit a more energy-

efficient placement of content replication sites. It would also allow a more accurate assessment of the energy benefit of multicast as well, coming at the problem of estimating multicast cost via induction–a different method from that used by Chuang and Sirbu.

In the short term, further work is needed to explore the idea of demand-side, incentive-based multicast group formation and establish its energy conservation potential. Incorporation of a specific policy and energy models for storage mechanisms within networks would allow a full assessment of the energy usage of real networks under a regime of unicast with efficient content storage. The findings here on the small number of sources required for efficient content distribution from within networks unicast transmissions and shortest-path routing would be interesting to extend to some of the models of content distribution within future Internet proposals, such as ICNs, where a different routing scheme is used. A broader assessment of real networks and real traffic would provide more evidence and perhaps counterexamples that would deepen the understanding of the effect of topology on efficient content distribution.

The main goal of the work presented here is to provide a better understanding of the energy consequences of content source or site duplication, content positioning, multicast, entropy reduction, and other means of modifying network behavior. The longer-term research goal is within-network adjustments—either structural or procedural decisions when to invoke a duplication ”move” and when to invoke a flow-bundling ”move” such as multicast—within an operating network on the fly based on necessary but perhaps insufficient local information not only about network activity and mobile participants and a changing set of nodes, but also about current traffic, offline nodes and unavailable links. For duplication and multicast moves, the goal is to derive a rule of thumb as to when and to what degree multicast is appropriate and when and to what degree content distribution is appropriate in terms of maximizing energy efficiency, and to what degree they cooperate in saving net energy.

To date, the decision has been largely gated by either *a priori* system information or by local

information. Yet the processing capabilities within the network itself have continued to improve; what stimuli trigger changes are usable in the network now, and what will be possible to compute and detect in two years, and in four years, and what could this mean for increasingly-less-local energy optimization—based perhaps on increasing amounts of information from the ability to run more event detection, and based on more information available at several different scopes or scales at once at nodes?

This could take the form of a computation-bandwidth-aware strategy for deciding whether to use multicast or content replication or some other method. While it is possible to run an integer linear program somewhere on the network that uses global knowledge about the network itself, this eventually devolves into an issue of scaling the information bandwidth to the observer to the system under observation: something needs to pass the *optimizer* information about the network, the topology, and all the hardware, and to update it as things change, and this approach will not scale well to very large networks. And, on the other end of the spectrum, at current network-processor control techniques deployed at nodes, what about something less local than a simple heuristic such as are now commonly used for caching? Given that the size of the network both can consider is limited: and, as we have shown here, that there is a benefit to be had by enlarging the view as much as possible—a global energy optimization is not the sum of local energy optimizations—how can we achieve the hard problem of observing as large a portion of the global ICT as possible in a computationally tractable and—more imminently problematic—a bandwidth-tractable way? These questions are the foundations of future work in this area.

Bibliography

[1] M. Peterson, "ICT at 10% of global electricity consumption?" http://www.vertatique.com/ict-10-global-energy-consumption, September 2013, accessed May 27, 2014.

[2] J. L. Antonio Capone, "Symposium on green networking and computing," http://www.josip-lorincz.com/Portals/0/2014_CfP_Green%20net_lorincz_capone.pdf, accessed 23 April 2014.

[3] G. Fettweis and E. Zimmermann, "ICT energy consumption-trends and challenges," in *Proceedings of the 11th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, no. 4, 2008, p. 6.

[4] R. Tucker, "Energy consumption in telecommunications," in *Optical Interconnects Conference, 2012 IEEE*. IEEE, 2012, pp. 1–2.

[5] J. G. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Tech. Rep., "2011".

[6] M. Peterson, "What is attached to our global ICT infrastructure?" http://vertatique.com/what-attached-our-global-ict-infrastructure, April 2014, accessed May 27, 2014.

[7] D. C. Kilper, G. Atkinson, S. K. Korotky, S. Goyal, P. Vetter, D. Suvakovic, and O. Blume, "Power trends in communication networks," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, no. 2, pp. 275–284, 2011.

[8] R. Tucker, "Green optical communications Part II: Energy limitations in networks," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, no. 2, pp. 261–274, 2011.

[9] R. Kurzweil, "IT growth and global change: a conversation with Ray Kurzweil," http://www.mckinsey.com/insights/business_technology/it_growth_and_global_change_a_conversation_with_ray_kurzweil, January 2011, accessed June 9, 2014.

[10] C. Shannon, "A mathematical theory of communication," AT & T Bell Labs, Tech. Rep., 1948.

[11] J. Aaltonen, J. Karvo, and S. Aalto, "Multicasting vs. unicasting in mobile communication systems," in *Proceedings of the 5th ACM international workshop on Wireless mobile multimedia*. ACM, 2002, pp. 104–108.

[12] R. Knutson, "AT&T deal pressures dish," http://online.wsj.com/news/articles/SB10001424052702304422704579570443151651808, May 2014, accessed 19 May 2014.

[13] J. Brodkin, "BitTorrent: Netflix should defeat ISPs by switching to peer-to-peer," http://arstechnica.com/information-technology/2014/04/bittorrent-netflix-should-defeat-isps-by-switching-to-peer-to-peer/, April 2014, accessed 26 Apr 2014.

[14] E. Schiattarella and C. Minkenberg, "Fair integrated scheduling of unicast and multicast traffic in an input-queued switch," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 1. IEEE, 2006, pp. 287–292.

[15] D. Kilper, K. Guan, K. Hinton, and R. Ayre, "Energy challenges in current and future optical transmission networks," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1168–1187, 2012.

[16] Y. Bachar, "ANCS 2015 keynote address," https://www.youtube.com/watch?v=uiiLMtO9nW8, May 2015.

[17] L. Durbeck, "Graph measures of network content-delivery energy," in *Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on*. IEEE, 2015, pp. 209–210.

[18] A. Passarella, "A survey on content-centric technologies for the current internet: CDN and P2P solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, 2012.

[19] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y.-D. Lin, P. Mähönen, M. Maier, R. Molva, S. Uhlig *et al.*, "Research challenges towards the future internet," *Computer Communications*, vol. 34, no. 18, pp. 2115–2134, 2011.

[20] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 75–86, 2011.

[21] N. Choi, K. Guan, D. C. Kilper, and G. Atkinson, "In-network caching effect on optimal energy consumption in content-centric networking," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2889–2894.

[22] J. Li, B. Liu, and H. Wu, "Energy-efficient in-network caching for content-centric networking," *Communications Letters*, vol. 17, no. 4, pp. 797–800, 2013.

[23] S. Imai, K. Leibnitz, and M. Murata, "Energy efficient content locations for in-network caching," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*. IEEE, 2012, pp. 554–559.

[24] J. Llorca, A. M. Tulino, K. Guan, J. Esteban, M. Varvello, N. Choi, and D. C. Kilper, "Dynamic in-network caching for energy efficient content delivery," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 245–249.

[25] K. Guan, G. Atkinson, D. C. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *Communications Workshops (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[26] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Peer-to-Peer Systems*. Springer, 2002, pp. 53–65.

[27] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, 2013.

[28] N. Zhang, T. Levä, and H. Hämmäinen, "Value networks and two-sided markets of internet content delivery," *Telecommunications Policy*, 2013.

[29] W. Wu and J. Lui, "Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1492–1503, 2012.

[30] S. Tewari and L. Kleinrock, "Analytical model for bittorrent-based live video streaming," in *Proc. IEEE NIME Workshop*, 2007.

[31] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 2, pp. 566–579, 2013.

[32] S. Tewari and L. Kleinrock, "Analysis of search and replication in unstructured peer-to-peer networks," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1. ACM, 2005, pp. 404–405.

[33] B. M. Waxman, "Routing of multipoint connections," *Selected Areas in Communications, IEEE Journal on*, vol. 6, no. 9, pp. 1617–1622, 1988.

[34] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[35] E. Gilbert and H. Pollak, "Steiner minimal trees," *SIAM Journal on Applied Mathematics*, vol. 16, no. 1, pp. 1–29, 1968.

[36] R. Karp, "Reducibility among combinatorial problems. complexity of computer computations,(re miller and jm thatcher, eds.), 85–103," Plenum Press, 1972.

[37] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A quantitative comparison of graph-based models for internet topology," *IEEE/ACM Transactions on Networking (TON)*, vol. 5, no. 6, pp. 770–783, 1997.

[38] M. B. Doar, "A better model for generating test networks," in *Global Telecommunications Conference, 1996. GLOBECOM'96.'Communications: The Key to Global Prosperity*. IEEE, 1996, pp. 86–93.

[39] K. L. Calvert, M. B. Doar, and E. W. Zegura, "Modeling internet topology," *Communications Magazine, IEEE*, vol. 35, no. 6, pp. 160–163, 1997.

[40] M. Thomas and E. W. Zegura, "Generation and analysis of random graphs to model internetworks," https://smartech.gatech.edu/bitstream/handle/1853/6735/GIT-CC-94-46.pdf, 1994.

[41] ——, "Georgia tech internetwork topology models (gt-itm) tarball," http://www.cc.gatech.edu/fac/Ellen.Zegura/gt-itm/gt-itm.tar.gz, 1996.

[42] J. M. Carlson and J. Doyle, "Highly optimized tolerance: A mechanism for power laws in designed systems," *Physical Review E*, vol. 60, no. 2, p. 1412, 1999.

[43] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou, "Heuristically optimized trade-offs: A new paradigm for power laws in the internet," in *Automata, languages and programming*. Springer, 2002, pp. 110–122.

[44] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 3–14, 2004.

[45] B. Gendron, T. G. Crainic, and A. Frangioni, *Multicommodity capacitated network design*. Springer, 1999.

[46] M. Grötschel, C. L. Monma, and M. Stoer, "Design of survivable networks," *Handbooks in operations research and management science*, vol. 7, pp. 617–672, 1995.

[47] D. Alderson, W. Willinger, L. Li, and J. Doyle, "An optimization-based approach to modeling internet topology," in *Telecommunications Planning: Innovations in Pricing, Network Design and Management*. Springer US, 2006, pp. 101–136.

[48] D. Alderson, H. Chang, M. Roughan, S. Uhlig, and W. Willinger, "The many facets of internet topology and traffic," *Networks and Heterogeneous Media*, vol. 1, no. 4, pp. 569–600, 2006.

[49] R. Bowden, M. Roughan, and N. Bean, "Cold: Pop-level network topology synthesis," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*. ACM, 2014, pp. 173–184.

[50] E. Parsonage, H. X. Nguyen, R. Bowden, S. Knight, N. Falkner, and M. Roughan, "Generalized graph products for network design and analysis," in *Network Protocols (ICNP), 2011 19th IEEE International Conference on*. IEEE, 2011, pp. 79–88.

[51] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1765–1775, 2011.

[52] J.-J. Pansiot and D. Grad, "On routes and multicast trees in the internet," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, 1998.

[53] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with Rocketfuel," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 133–145.

[54] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore, "On the bias of traceroute sampling: or, power-law degree distributions in regular graphs," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. ACM, 2005, pp. 694–703.

[55] Y. Shavitt and E. Shir, "Dimes: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 5, pp. 71–74, 2005.

[56] B. Donnet and T. Friedman, "Internet topology discovery: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 9, no. 4, pp. 56–69, 2007.

[57] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4. ACM, 1999, pp. 251–262.

[58] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degree-based vs. structural," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 147–159.

[59] A. Medina, I. Matta, and J. Byers, "Brite: A flexible generator of internet topologies," Boston, MA, USA, Tech. Rep., 2000.

[60] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*. IEEE, 2001, pp. 346–353.

[61] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*. Acm, 2000, pp. 171–180.

[62] C. R. Palmer and J. G. Steffan, "Generating network topologies that obey power laws," in *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE*, vol. 1. IEEE, 2000, pp. 434–438.

[63] S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology," *Proceedings of the National Academy of Sciences*, vol. 99, no. 21, pp. 13 382–13 386, 2002.

[64] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[65] H. Burch and B. Cheswick, "Mapping the internet," *Computer*, vol. 32, no. 4, pp. 97–98, 1999.

[66] R. Govindan and H. Tangmunarunkit, "Heuristics for internet map discovery," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2000, pp. 1371–1380.

[67] K. Claffy, T. Monk, and D. McRobb, "Internet Tomography," *Nature*, Jan 1999. [Online]. Available: \url{http://www.nature.com/nature/webmatters/tomog/tomog.html}

[68] T. C. A. for Internet Data Analysis, "Internet topology at router- and AS-levels, and the dual router + AS Internet topology generator," http://www.caida.org/research/topology/generator/, Nov 2013, accessed 04 Mar 2014.

[69] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@ home: an experiment in public-resource computing," *Communications of the ACM*, vol. 45, no. 11, pp. 56–61, 2002.

[70] D. Dolev, O. Mokryn, and Y. Shavitt, "On multicast trees: structure and size estimation," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 3, pp. 557–567, 2006.

[71] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 lessons from 10 years of measuring and modeling the internet's autonomous systems," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1810–1821, 2011.

[72] B. Augustin, B. Krishnamurthy, and W. Willinger, "Ixps: mapped?" in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009, pp. 336–349.

[73] J.-J. Pansiot, P. Mérindol, B. Donnet, and O. Bonaventure, "Extracting intra-domain topology from mrinfo probing," in *Passive and Active Measurement*. Springer, 2010, pp. 81–90.

[74] P. Marchetta, V. Persico, and A. Pescapé, "Pythia: yet another active probing technique for alias resolution," in *Proceedings of the Ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 229–234.

[75] P. Mérindol, B. Donnet, J.-J. Pansiot, M. Luckie, and Y. Hyun, "Merlin: Measure the router level of the internet," in *Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on*. IEEE, 2011, pp. 1–8.

[76] P. Marchetta, P. Mérindol, B. Donnet, A. Pescapé, and J. Pansiot, "Topology discovery at the router level: a new hybrid tool targeting ISP networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1776–1787, 2011.

[77] J. Pansiot, P. Mérindol, B. Donnet, A. Pescapé, and P. Marchetta, "The merlin topology discovery project," http://svnet.u-strasbg.fr/merlin/Download/Download, August 2011.

[78] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy consumption in optical IP networks," *Journal of Lightwave Technology*, vol. 27, no. 13, pp. 2391–2403, 2009.

[79] U. Lee, I. Rimac, D. Kilper, and V. Hilt, "Toward energy-efficient content dissemination," *Network, IEEE*, vol. 25, no. 2, pp. 14–19, 2011.

[80] T. Hasegawa, Y. Nakai, K. Ohsugi, J. Takemasa, Y. Koizumi, and I. Psaras, "Empirically modeling how a multicore software icn router and an icn network consume power," in *Proceedings of the 1st international conference on Information-centric networking*. ACM, 2014, pp. 157–166.

[81] J. Baliga, R. W. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.

[82] K. Guan, D. C. Kilper, and G. Atkinson, "Evaluating the energy benefit of dynamic optical bypass for content delivery," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 313–318.

[83] N. Osman, T. El-Gorashi, J. M. Elmirghani *et al.*, "Reduction of energy consumption of video-on-demand services using cache size optimization," in *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*. IEEE, 2011, pp. 1–5.

[84] C. Jayasundara, A. Nirmalathas, E. Wong, and C. A. Chan, "Energy efficient content distribution for vod services," in *Optical Fiber Communication Conference*. Optical Society of America, 2011, p. OWR3.

[85] N. T. Spring and D. Wetherall, "A protocol-independent technique for eliminating redundant network traffic," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 87–95, 2000.

[86] A. Anand, C. Muthukrishnan, A. Akella, and R. Ramjee, "Redundancy in network traffic: findings and implications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 37–48, 2009.

[87] J. Araujo, F. Giroire, Y. Liu, R. Modrzejewski, and J. Moulierac, "Energy efficient content distribution," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4233–4238.

[88] U. Mandal, C. Lange, A. Gladisch, P. Chowdhury, and B. Mukherjee, "Energy-efficient content distribution over telecom network infrastructure," in *2011 13th International Conference on Transparent Optical Networks*, 2011.

[89] T. Bektaş, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Computers & Operations Research*, vol. 35, no. 12, pp. 3860–3884, 2008.

[90] R. Modrzejewski, L. Chiaraviglio, I. Tahiri, F. Giroire, E. Le Rouzic, E. Bonetto, F. Musumeci, R. Gonzalez, and C. Guerrero, "Energy efficient content distribution in an ISP network," in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 2859–2865.

[91] D. H. Lorenz, A. Orda, D. Raz, and Y. Shavitt, "How good can ip routing be," *DIMACS Rep*, vol. 17, 2001.

[92] For a number of autonomous systems, the Merlin xml files did not produce a full single connected graph; rather, they produced a series of subgraphs.

[93] C. for Applied Internet Data Analysis, "AS-rank:AS ranking," http://as-rank.caida.org/, July 2015, accessed 23 Sept 2015.

[94] CISCO, "Fundamentals of digital video," http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Video/pktvideoaag.html, 2008.

[95] R. Sinha, C. Papadopoulos, and J. Heidemann, "Internet packet size distributions: Some observations," *USC/Information Sciences Institute, Tech. Rep. ISI-TR-2007-643*, 2007.

[96] C. for Applied Internet Data Analysis, "Packet size distribution comparison between internet links in 1998 and 2008," https://www.caida.org/research/traffic-analysis/pkt_size_distribution/graphs.xml, June 2010, accessed 11 Sept 2015.

[97] D. Murray and T. Koziniec, "The state of enterprise network traffic in 2012," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*. IEEE, 2012, pp. 179–184.

[98] A. Vishwanath, K. Hinton, R. Ayre, and R. Tucker, "Modeling energy consumption in high-capacity routers and switches," *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 8, pp. 1524–1532, 2014.

[99] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *NETWORKING 2009*. Springer, 2009, pp. 795–808.

[100] J. Baliga, R. Ayre, K. Hinton, and R. S. Tucker, "Energy consumption in wired and wireless access networks," *Communications Magazine, IEEE*, vol. 49, no. 6, pp. 70–77, 2011.

[101] R. S. Tucker, "Green optical communicationspart i: Energy limitations in transport," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, no. 2, pp. 245–260, 2011.

[102] This is not to be confused with per packet energy over the network: this is savings here, not spending. More specifically, it is the pooling effect of redundancy, factored per packet, of the specific content, due to redundant transmissions from multiple requests for the content. This is used in part to normalize the results across various content sizes, which is a more meaningful metric in some forms of the analysis.

[103] This is not to be confused with per bit energy over the network: here this is savings per bit, not cost per bit. More specifically, it is a pooling effect, factored per bit, of the specific content, due to redundant transmissions.

[104] This is closer to the traditional sense of per-packet e; however, it is energy savings, not energy cost; it also applies just to packets related to this content item, rather than any packet flowing over the node, and it is the total energy saved in the network rather than energy saved at a particular node. It is per-packet-e divided by the number of requests for the content item.

[105] X. Ma and K. Harfoush, "Traffic concentration for a green internet," in *High Capacity Optical Networks and Enabling Technologies (HONET), 2012 9th International Conference on*. IEEE, 2012, pp. 142–146.

[106] L. Durbeck and P. Athanas, "A global perspective on energy conservation in large data networks," 2014, IEEE PATMOS International Workshop on Power And Timing Modeling, Optimization and Simulation Sept 29 - Oct 1, Palma de Mallorca, Spain.

[107] C. Zhang and T. Chen, "A survey on image-based rendering–representation, sampling and compression," *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, 2004.

[108] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G. B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3dtva survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 11, pp. 1606–1621, 2007.

[109] B. Wohlberg and G. De Jager, "A review of the fractal image coding literature," *Image Processing, IEEE Transactions on*, vol. 8, no. 12, pp. 1716–1729, 1999.

[110] P.-c. Tseng, Y.-c. Chang, Y.-w. Huang, H.-c. Fang, C.-t. Huang, and L.-g. Chen, "Advances in hardware architectures for image and video coding-a survey," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 184–197, 2005.

[111] M. L. Hilton, B. D. Jawerth, and A. Sengupta, "Compressing still and moving images with wavelets," *Multimedia systems*, vol. 2, no. 5, pp. 218–227, 1994.

[112] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006, pp. 309–316.

[113] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 265–278.

[114] C. N. Taylor and S. Dey, "Adaptive image compression for wireless multimedia communication," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 6. IEEE, 2001, pp. 1925–1929.

[115] D.-G. Lee and S. Dey, "Adaptive and energy efficient wavelet image compression for mobile multimedia data services," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 2484–2490.

[116] H. Wu and A. A. Abouzeid, "Power aware image transmission in energy constrained wireless networks," in *Computers and communications, 2004. Proceedings. ISCC 2004. Ninth international symposium on*, vol. 1. IEEE, 2004, pp. 202–207.

[117] D.-U. Lee, H. Kim, M. Rahimi, D. Estrin, and J. D. Villasenor, "Energy-efficient image compression for resource-constrained platforms," *Image Processing, IEEE Transactions on*, vol. 18, no. 9, pp. 2100–2113, 2009.

[118] D.-U. Lee, H. Kim, S. Tu, M. Rahimi, D. Estrin, and J. D. Villasenor, "Energy-optimized image communication on resource-constrained sensor platforms," in *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*. IEEE, 2007, pp. 216–225.

[119] C. Poellabauer and K. Schwan, "Energy-aware media transcoding in wireless systems," in *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE, 2004, pp. 135–144.

[120] L. Wang and J. Manner, "Evaluation of data compression for energy-aware communication in mobile networks," in *Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC'09. International Conference on*. IEEE, 2009, pp. 69–76.

[121] R. Xu, Z. Li, C. Wang, and P. Ni, "Impact of data compression on energy consumption of wireless-networked handheld devices," in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*. IEEE, 2003, pp. 302–311.

[122] B. Welton, D. Kimpe, J. Cope, C. M. Patrick, K. Iskra, and R. Ross, "Improving i/o forwarding throughput with data compression," in *Cluster Computing (CLUSTER), 2011 IEEE International Conference on*. IEEE, 2011, pp. 438–445.

[123] R. Kothiyal, V. Tarasov, P. Sehgal, and E. Zadok, "Energy and performance evaluation of lossless file data compression on server systems," in *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*. ACM, 2009, p. 4.

[124] P. Hayinga, "Energy efficiency of error correction on wireless systems," in *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*. IEEE, 1999, pp. 616–620.

[125] K. C. Barr and K. Asanovic, "Energy aware lossless data compression," in *MobiSys*, ACM. ACM, 2003, pp. 231–244.

[126] K. C. Barr and K. Asanović, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems (TOCS)*, vol. 24, no. 3, pp. 250–291, 2006.

[127] A. Milenkovic, A. Dzhagaryan, and M. Burtscher, "Performance and energy consumption of lossless compression/decompression utilities on mobile computing platforms," in *Proceedings of the 2013 IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. IEEE Computer Society, 2013, pp. 254–263.

[128] A. A. Dzhagaryan, "Performance and energy efficiency of compression/decompression utilities: An experimental study in mobile and workstation computer platforms," Master's thesis, THE UNIVERSITY OF ALABAMA IN HUNTSVILLE, 2013, thesis.

[129] Source coding in which statistical redundancy is removed from the source signal without loss of any information contained within the signal.

[130] A. Dzhagaryan, A. Milenkovic, and M. Burtscher, "Energy efficiency of lossless data compression on a mobile device: An experimental evaluation," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 126–127.

[131] R. Kothiyal, V. Tarasov, P. Sehgal, and E. Zadok, "Energy and performance evaluation of lossless file data compression on server systems," in *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*. ACM, 2009, p. 4.

[132] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, MASSACHUSETTS INSTITUTE OF TECH, December 1984.

[133] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali *et al.*, "A taxonomy and survey on green data center networks," *Future Generation Computer Systems*, 2013.

[134] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 1, pp. 3–20, 2012.

[135] A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya *et al.*, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, no. 2, pp. 47–111, 2011.

[136] X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon, "A survey of green mobile networks: Opportunities and challenges," *Mobile Networks and Applications*, vol. 17, no. 1, pp. 4–20, 2012.

[137] G. Hardin, "The tragedy of the commons," *science*, vol. 162, no. 3859, pp. 1243–1248, 1968.

[138] D. Loguinov and H. Radha, "Large-scale experimental study of internet performance using video traffic," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 1, pp. 7–19, 2002.

[139] Ofcom, "The communications market report: Internet and web-based content," http://stakeholders.ofcom.org.uk/market-data-research/market-data/communications-market-reports/cmr13/internet-web/?pageNum=3#in-this-section, 2013, accessed 10 Mar 2014.

[140] M. Milosevic, A. Dzhagaryan, E. Jovanov, and A. Milenković, "An environment for automated power measurements on mobile computing platforms," in *Proceedings of the 51st ACM Southeast Conference*. ACM, 2013, p. 19.

[141] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: routing distances and fault resilience," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 395–406.

[142] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 2, pp. 85–110, 1990.

[143] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1607–1617.

[144] S. Meiling, T. Steinbach, T. C. Schmidt, and M. Wählisch, "A scalable communication infrastructure for smart grid applications using multicast over public networks," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 690–694.

[145] L. Valcarenghi and P. Castoldi, "Impact of unicast and multicast traffic on onu energy savings," in *Transparent Optical Networks (ICTON), 2012 14th International Conference on*. IEEE, 2012, pp. 1–5.

[146] R. Fantini, D. Sabella, and M. Caretti, "Energy efficiency in LTE-advanced networks with relay nodes," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, 2011, pp. 1–5.

[147] H. Burch and B. Cheswick, "Mapping the internet," *Computer*, vol. 32, no. 4, pp. 97–98, 1999.

[148] T. Stynes, "Circulation up at Journal, Times," http://online.wsj.com/news/articles/SB10001424127887324482504578454693739428314, May 2013, accessed 28 May 2014.

[149] L. Durbeck and P. Athanas, "Energy interactions between multicast and content distribution within data communication networks," 2015, IEEE ICIT International Conference on Industrial Technology special session on smart green systems, technologies and approaches, Seville, Spain 17 - 19 March, 2015.

[150] J. C.-I. Chuang and M. A. Sirbu, "Pricing multicast communication: A cost-based approach," *Telecommunication Systems*, vol. 17, no. 3, pp. 281–297, 2001.

[151] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of multicast trees: Comments on the chuang-sirbu scaling law," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4. ACM, 1999, pp. 41–51.

[152] P. Van Mieghem, G. Hooghiemstra, and R. Van Der Hofstad, "On the efficiency of multicast," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 6, pp. 719–732, 2001.

[153] S. Glassman, "A caching relay for the world wide web," *Computer Networks and ISDN Systems*, vol. 27, no. 2, pp. 165–173, 1994.

[154] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 126–134.

[155] L. Durbeck, J. G. Tront, and N. Macias, "Energy efficiency of zipf traffic distributions within facebook's data center fabric architecture," 2015, IEEE PATMOS International Workshop on Power And Timing Modeling, Optimization and Simulation Sept 1 - 4, Salvadore, Brazil.

[156] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3. ACM, 2010, pp. 338–347.

[157] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green ip over wdm networks with data centers," *Journal of Lightwave Technology*, vol. 29, no. 12, pp. 1861–1880, 2011.

[158] J. Arjona Aroca and A. Fernández Anta, "Bisection (band) width of product networks with application to data centers," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 3, pp. 570–580, 2014.

[159] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in *ACM SIGCOMM computer communication review*, vol. 39, no. 4. ACM, 2009, pp. 51–62.

[160] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks." in *NSDI*, vol. 10, 2010, pp. 249–264.

[161] Statista, "Comparison of unique u.s. visitors to facebook and google from april 2011 to january 2015 (in millions)," http://www.statista.com/statistics/268252/comparison-of-unique-us-visitors-to-facebook-and-google/, March 2015, accessed May 15, 2015.

[162] internetlivestats, "Facebook active users," http://www.internetlivestats.com/, May 2015, accessed May 20, 2015.

[163] E. Schonfeld, "Sharethis study: Facebook accounts for 38 percent of sharing traffic on the web," http://techcrunch.com/2011/06/06/sharethis-facebook-38-percent-traffic/, June 2011, accessed May 14, 2015.

[164] K. Bhardwaj, "50 facebook facts and figures," https://www.facebook.com/notes/kuldeep-bhardwaj/50-facebook-facts-and-figures/10150274471574235, August 2011, accessed May 17, 2015.

[165] E. Griffith, "How facebook's video-traffic explosion is shaking up the advertising world," http://fortune.com/2015/06/03/facebook-video-traffic/, June 2015.

[166] S. Sankar, S. Lassen, and M. Curtiss, "Under the hood: Building out the infrastructure for graph search," //www.facebook.com/notes/facebook-engineering/under-the-hood-building-out-the-infrastructure-for-graph-search/10151347573598920, March 2013, accessed May 18, 2015.

[167] A. Andreyev, "Introducing data center fabric, the next-generation facebook data center network," https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/#, November 2014, accessed May 15, 2015.

[168] Y. Bachar, "Introducing 6-pack, the first open hardware modular switch," https://www.youtube.com/watch?v=uiiLMtO9nW8, February 2015, accessed May 13, 2015.

[169] J. G. Koomey, "Estimating total power consumption by servers in the us and the world," http://hightech.lbl.gov/documents/DATA\$_CENTERS/svrpwrusecompletefinal.pdf, February 2007.

[170] A. S. Brown, "Keep it cool! inside the world's most efficient data center," The Bent of Tau Beta Pi, pp. 12–16, Spring 2014.

[171] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68–73, 2008.

[172] A. P. Bianzino, A. K. Raju, and D. Rossi, "Apples-to-apples: a framework analysis for energy-efficiency in networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, no. 3, pp. 81–85, 2011.

[173] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *Computers, IEEE Transactions on*, vol. 100, no. 10, pp. 892–901, 1985.

[174] N. Farrington and A. Andreyev, "Facebooks data center network architecture," in *IEEE Optical Interconnects Conf.* Citeseer, 2013.

[175] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.

[176] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63–74, 2008.

[177] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable fault-tolerant layer 2 data center network fabric." in *SIGCOMM*, vol. 9, 2009, pp. 39–50.

[178] N. Farrington, E. Rubow, and A. Vahdat, "Data center switch architecture in the age of merchant silicon," in *High Performance Interconnects, 2009. HOTI 2009. 17th IEEE Symposium on*. IEEE, 2009, pp. 93–102.

[179] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 47, 2014.

[180] L. Wang, F. Zhang, J. Arjona Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "Greendcn: A general framework for achieving energy efficiency in data center networks," *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 1, pp. 4–15, 2014.

[181] Y. Jin, Y. Wen, K. Guan, D. Kilper, and H. Xie, "Toward monetary cost effective content placement in cloud centric media network," in *Multimedia and Expo (ICME), 2013 IEEE International COnference On*. IEEE, 2013, pp. 1–6.

[182] J. O'Reilly, "Will rdma over ethernet eclipse infiniband?" http://www.networkcomputing.com/networking/will-rdma-over-ethernet-eclipse-infiniband/a/d-id/1316950, October 2014, accessed June 1, 2015.

[183] P. Grun, "Roce and infiniband: Which should i choose?" http://blog.infinibandta.org/2012/02/13/roce-and-infiniband-which-should-i-choose/.

[184] P. Computing, "Sb-800," http://picocomputing.com/products/backplanes/ex-800-blade-server/, June 2015, accessed June 6, 2015.

[185] Wikipedia, "Three mile island nuclear generating station," http://en.wikipedia.org/wiki/Three_Mile_Island_Nuclear_Generating_Station, accessed 12 Apr 2014.

[186] U. E. I. Administration, "What is the efficiency of different types of power plants?" http://www.eia.gov/tools/faqs/faq.cfm?id=107&t=3, May 2014, accessed 22 May 2014.

[187] ——, "Table 8.1. average operating heat rate for selected energy sources," http://www.eia.gov/electricity/annual/html/epa_08_01.html, May 2014, accessed 22 May 2014.

[188] N. E. Institute, "Life-cycle emissions analyses," http://www.nei.org/Issues-Policy/ProtectingtheEnvironment/Life-Cycle-Emissions-Analyses, accessed 15 Apr 2014.

[189] U. Energy Information Administration, "How many gallons of diesel fuel and gasoline are made from one barrel of oil?" http://www.eia.gov/tools/faqs/faq.cfm?id=327&t=9, accessed 11 Nov 2015.

[190] Wikipedia, "List of countries by oil consumption," https://en.wikipedia.org/wiki/List_of_countries_by_oil_consumption, accessed 13 Nov 2015.

[191] L. Sha, S. Gopalakrishnan, X. Liu, and Q. Wang, "Cyber-physical systems: A new frontier," in *Machine Learning in Cyber Trust*. Springer, 2009, pp. 3–13.