Development of Person-Person Network and Interacting PTTS in EpiSimdemics

Gaurav Mishra

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

Master of Science
in
Computer Science and Applications

Madhav V.Marathe, Chair
Keith R.Bisset
Anil Kumar S.Vullikanti

May 02, 2014
Blacksburg, Virginia

Keywords: EpiSimdemics, PTTS, Interacting Networks

Development of Person-Person Network and Interacting PTTS in EpiSimdemics

Gaurav Mishra

(ABSTRACT)

Communications over social media, telephone, email, text etc have emerged as an integral part of modern society and they are popularly used for the expression of anger, anxiety, fear, agitation and opinion by the people. People's social interaction tend to increase dramatically during periods of epidemics, protest and calamities. Therefore, above mentioned communication channels plays an important role in the spread of infectious phenomenon, like rumors, fads and effects. These infectious phenomena alters people's behavior during disease epidemic [1][2].

Social contact networks and epidemics co-evolve [1][2]. The spread of a disease influences people's behavior which in turn changes their social contact network, thereby altering the disease spread itself. As a result, there is a need for modeling the spread of these infectious phenomena that lead to changes in behavior. Their propagation among population primarily depends on the social contact network. The nature of social contagion spread is very similar to the spread of any infectious disease as they are contagious in nature. To spread contagious disease requires direct exposure to an infectious agent, whereas social contagions can be spread using various communications media like social networking forums, phones, emails and tweets.

EpiSimdemics is an individual-based modeling environment. It uses a people-location bipartite graph as the underlying network [3]. In its current form, EpiSimdemics requires two people to interact at a location to model simulations. Thus, it cannot simulate the spread

of social contagions that do not necessarily require the meeting of two agents at a location.

We enhance EpiSimdemics by incorporating Person-Person network, which can model communications between people that are not contact based such as communications over email, phone, text and tweet. This Person-Person network is used to model effects (social contagion) which induce behavioral changes in population and thus impacting the disease spread. The disease spread is modeled on Person-Location network. This leads to the scenario of two interacting networks: Person-Person network modeling social contagion and Person-Location modeling disease. Theoretically, there can be multiple such networks modeling various interacting phenomena.

We demonstrate the usefulness of this network by modeling and simulating two interacting PTTSs (probabilistic timed transition systems). To model disease epidemics, we have defined Disease Model and to model effects (social contagion), we have defined Fear Model. We show how these models influence each other by performing simulations on EpiSimdemics with interacting Disease and Fear Model. Therefore a model that does not include the affect adaptations on disease epidemics and vice-versa, fails to reflect the actual behavior of a society during disease epidemic spread. The addition of Person-Person network to EpiSimdemics will allow for a better understanding of the affect adaptions, which can include behavior changes in society during an epidemic outbreak. This would lead to effective interventions and help to better understand the dynamics of disease epidemic.

Acknowledgments

This work has been supported by DTRA Validation Grant HDTRA1-11-1-0016, NSF PetaApps Grant OCI-0904844, DTRA CNIMS Grant HDTRA1-07-C-0113, DTRA CNIMS Grant HDTRA1-11-D-0016 and NetSe Grant CCF-1011769.

I would like thank my advisory committee for their continuous support and encouragement throughout the master's program. Special thanks to my advisor Dr.Keith Bisset, who has been an inspiration and role model in this journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Disease epidemics pose a serious impact on the function and morale of society. They have a tendency to destabilize the society by influencing the population that forms the building blocks of society. The spread of smallpox, H1N1 virus and HIV are common examples of epidemics [5]. They are known to spread fear, anxiety, rumors and concern among the population which then impacts the activities of people. This affects the normal functions of society. Therefore, preventing and controlling such epidemics is a top concern for government agencies and public health officials.

The spread of these infectious diseases depends on numerous parameters. The most dominating among them are person's contact network, their immunity towards disease (prior vaccination) and disease characteristics (strength of the virus). Government officials are responsible for monitoring the spread of these diseases. They take preventive measures in the form of collective interventions such as, vaccinations [6]. People can adopt personal interventions. An example of such individual interventions include isolation by staying at home to curb the spread of a virus. But, the spread of disease largely depends on compliance to such interventions and compliance towards these interventions largely depends on the socio-economic status of the subjects. The collective and personal interventions can have a conflicting nature where one of them may belittle the effect of the other [7][1][6].

Epidemics is one specific example of a general class of phenomena involving the spread of contagions which include disease or another social phenomenon pertaining to society. Social contagions can be in forms like fear, rumors and anxiety etc. Addictions such as playing online games, smoking also fall into the category of social contagion [1][8][9][10].

For the transmission of infectious disease, people physically interact with their social groups, which leads to the spread of infection. Whereas for the spread of social contagions people do not need to meet each other in-person. They can be communicated via phone, email and text etc. Although, the mode of transmission of disease and effects that are contagion related are different, they do influence each other[2][1].

In order to model the spread of social contagions like effects of epidemics, we extended the EpiSimdemics simulation platform by adding Person-Person network. With this enhancement, EpiSimdemics now runs simulation on following two networks:

1. **Person-Location Network**- bipartite graph network modeling disease spread.
2. **Person-Person Network**- graph network modeling the spread of social contagions which would lead to affect adaptations related to epidemic.

Person-Person network can be generated from any communication media that are not contact based, such as communications over email, text, phone and social media. Population can have interactions in both Person-Person and Person-Location networks and both of these networks interact for the same person. Figure 5.6 depicts possible interaction in both of these networks in a population comprising of three people.

We present the progression of disease over Person-Location network using *Disease Model* and progression of the effects of disease using *Fear Model* on the Person-Person network. Each of these models is represented using enhanced finite state machine called PTTS (Probabilistic Timed Transition System). Their interaction is modeled as interacting PTTS in EpiSimdemics. Interacting PTTSs lead to following interacting networks: Person-Location modeling disease diffusion and Person-Person modeling effect (social contagion) diffusion. Theoretically, there can be multiple interacting networks modeling various interacting infectious phenomena. This enhancement allows individuals to change states in the Fear Model based on the state in the Disease Model and vice-versa.

We have modeled fear (social contagion) as an effect of disease epidemic in this work, which should be interpreted as ”concern.” But the framework is generic to include other type of effects. Fear about the disease induce behavior changes, which impacts the disease progression and vice-versa. Experimental results as a part of this work show that the relationship is inversely proportional. For example, a person who is infected with effects of disease epidemic like 'fear' of H1N1 has a lower probability of catching the virus because people undergo affect adaptation and starts responding to the spread by adopting behavioral changes such as, isolation [1][2][5]. It is a complex scenario to handle two dependent and different kinds of infectious phenomena. The nature of interaction is different in both cases; spread of disease involves temporal and spatial co-locality, while for spreading social contagions physical presence is not required.

This coupled model has the capability to simulate both kinds of interactions together. This model closely represents the society behavior, when it comes to epidemics and the effects of disease epidemic together [11][12].

Before social media evolved, people's interactions resulted from temporal-spatial co-locality that lead to the spread of infectious diseases. So, a model that comprehends location based interaction, used to be an accurate measure for disseminating the interventions. In today's world, social media is one of the most popular ways of expressing opinions. People transmit their fears, anger and rumors using social media. In recent times, twitter has been used as popular means of spreading protest and rumors in many countries. In such scenarios, to develop effective interventions against infectious disease and social contagion, policy makers need to consider a model that can simulate interaction between disease and its effects. This capability is achieved by building Person-Person network in EpiSimdemics.

Chapter 2

Related Work

Social interactions have a very broad impact on the functioning of society as a unified entity. People who are part of this society interact with each other. Such interactions lead to the exchange and transmission of social contagions which may be in form of emotion, fear, anxiety, anger and dissatisfaction to other individuals who may or may not be infected with that contagion. Such social contagions also affect the individual's thinking and behavior during the epidemic spread of a disease. Thus, the spread of disease influences the spread of social contagions, such as fear, rumor and anxiety about that disease and vice-versa[13][14].

Mathematical modeling and simulation of social contagion and diseases provide powerful tools to study how the spread of these infectious phenomena can be changed by underlying interventions [3][15].

Broadly speaking, there are two kinds of interaction models that are widely used for simulation purposes:

1. population based models
2. individual interaction based models

The population based model divides the population into clusters according to the contagion infection status or disease status (susceptible, infected etc) of the person and demographics.

This model assumes uniform mixing of population and deploys differential equations to study the spread of infectious phenomena in the population [8][3][15].

The individual based interaction model relies on the individual interaction between agents and takes into account such interaction to calculate infection transmission[8][3][16].

In society infection primarily depends on individual agents who interact and spread infection. Therefore, individual based interaction model closely represents the behavior of society, when it comes to disease or social contagion propagation.

The Kermack-McKendrick model [17], divides the population into clusters according to people's health state or other contagious parameters such as fear levels, and uses differential equations to study the spread of contagion in the population. Though their model is applied to the study of disease epidemics, it is also applicable to the spread of social contagions, where the population is compartmentalized. EpiSimdemics [3], EpiFast [18] and EpiSims are simulation engines built upon the simdemics framework [19][15]. EpiSimdemics is an advancement of EpiSims. Both EpiSims and EpiSimdemics use Person-Location network representation as their underlying graph network that includes detailed individual activities. It can be used to study very general disease models but requires significant computing resources[10][19].

Epifast, a disease modeling algorithm, uses a relatively simpler yet realistic disease model because its running time is many folds slower than EpiSimdemics. Experiments shows that EpiFast runs 10 times faster than EpiSimdemics. The underlying distributed memory system used in EpiFast enables it to run faster than the native EpiSimdemics model.

Both EpiFast [18] and EpiSimdemics [3] derive the motivation of epidemiological simulation from the work of Eubank et al [20]. The model described by Eubank et al [20] is among the first disease spread model to deploy individual based interaction model. Their model relies on empirical data for building an interaction based network. Their empirical data is borrowed from "TRANSIMS", which is a simulation engines used in transportation systems [20].

Longini et al [21] and Ferguson [22] developed a simulation engine built on shared memory for their inter-process communication. The design decision to include shared memory puts a constraint on population size. Their population is based on a very structured contact network, which is not the realistic representation of an actual population. They have deployed surrogate data in some components of the population, such as locations. Population data is massaged so that the deployment of such a model on parallel platforms is feasible but they are not the accurate representation of actual behavior of society during epidemics of a disease or any other contagion.

Parker's [23] simulation engine is highly scalable and is equipped to simulate the entire United States population. It also relies on underlying shared memory for communication purposes, therefore it is constrained by available memory. It uses highly structured population components, which introduces certain anomalies in results of simulation.

Percolation-based approach is also a popular approach in modeling infectious phenomena [24]. Though, percolation based approach has a simplistic design and is efficient in execution time, but it does not predict the state of population at the end of each day, which makes it harder to analyze the number of people in different infectious states during an epidemic. It gives the final disease spread size and number of people infected during epidemic. Therefore, it is not very suitable for administering interventions.

Substantial research has been done to understand the types of social interactions and dynamics of online networks [25]. There are possibly two types of social interaction: active and latent (passive). Huang et al [26] conducted research to study the interaction types in Renren, which is a social network in China, similar to Facebook and Twitter. They measured the visible interaction, which are active interactions, preference in Renren. Jiangjing et al [25] described measuring latent interaction in social networks. Latent interactions play an important metric when observing epidemic spread, as people can be influenced by others without even knowing them. Such influence can lead to an activity that can influence the epidemic spread.

Chapter 3

Background Work/EpiSimdemics

3.1 Computational Models

A society or community is said to be impacted by an epidemic when cases of acute illness or death are above normal expectancy. Some of these infectious diseases have claimed more lives than wars. Health organization across the globe realize the importance of tackling epidemics.

Computational epidemiology focuses on using computational models to analyze and simulate the spread of infectious phenomena. These computational models help to better understand the spread of infectious phenomenon temporally and spatially. Although it is not possible to predict the accurate number of deaths or illnesses with these models, they do give an overall behavior of society during an epidemic and trend of disease spread. These predictions can greatly help the policy makers understand the dynamics of disease and adopt adequate measures. Depending upon the dynamics of disease spread varied intervention measures can be enforced or suggested by public health officials. These interventions can be pharmaceutical in nature such as administering vaccination for a particular disease. In some cases, non-pharmaceutical interventions may be effective including the, isolation of infected people. Such isolation can be achieved by abstaining from public gatherings.

There are primarily two types of computational model based on the way they model interactions:

1. compartmental/partitioned: this model divides the population into partitions or compartments based on some attributes such as age.
2. individual interaction: this model considers individual interactions as an underlying network rather than some fixed partitioning of population.

3.2 EpiSimdemics

EpiSimdemics is a parallel simulation environment used to model disease progression in large social networks. It is built on individual interactions as an underlying computational model. Person-Location bipartite graph is the underlying network in EpiSimdemics.

Figure 3.1 is the bi-partite graph representing a social contact network. Each eclipse represents a person and the squares represent locations. The directed arrows from person to location represent locations people visited during the day at times marked on arrows. The progression of disease is controlled by two connected and independent processes: within host disease progression and between hosts disease progression. The within host disease progression is modeled by disease PTTS defined in fig. 5.2. Each individual in population will proceed through the set of states in PTTS. Each of these states have infectivity and susceptibility values, which are attributes of states and are used to compute infection for people in those states.

Between hosts disease transmission is determined by interaction between individuals. The equation (3.1) presents a compute infection function for calculating the probability of infection resulting from interaction between two individuals. When two individuals meet at a location then each of them have the possibility of infecting others based on their infectivities and susceptibilities. Transmissibility in equation (3.1) represents the characteristic of a disease which defines the probability of a susceptible person infected by infected person in one minute of disease exposure.

$$p = 1 - \exp(log(1 - transmissibility_{disease} * susceptibility_{infected} * infected_{infector})) * duration \tag{3.1}$$

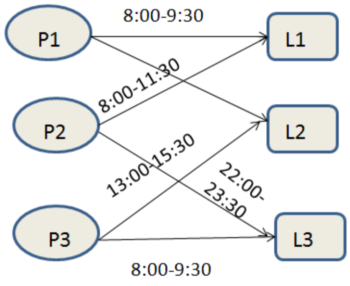


Figure 3.1: Person-Location Graph

EpiSimdemics achieves scalability by altering the number of compute nodes thereby adding hardware resources on a need basis. It allows the programmers to specify the number of compute nodes at run time. Each compute node is associated with a certain number of cores, which depends on underlying hardware. Each core of a compute node acts like processing element. Let each processing element be denoted by PE.

Assuming system has n processing element, then each of these PE can be denoted as $PE_1, PE_2, PE_3, ..., PE_n$.

Similarly, person and location will be divided into n groups.The round-robin allocation scheme is used to divide people and location into n groups.

Let $P_1, P_2, P_3, ..., P_n$ and $L_1, L_2, L_3, ..., L_n$ represent people and location groups respectively.

Person Manager PM_i is implemented for each P_i and Location Manager LM_i is implemented for each L_i.

Message Broker: It is responsible for message communication in EpiSimdemics. It connects PM_i and LM_i to their corresponding PE_i. Message broker does asynchronous message passing. It receives all incoming messages and outgoing messages. Depending on the location of the message it either passes it on to its target message broker or if the message is local, it is sent to its corresponding person manager or location manager. Algorithm 1 presents the parallel version of EpiSimdemics [3].

Algorithm 1 Parallel Version of EpiSimdemics [3]

```
  a. read person,locations and visits File
  b. for each day in simulation
  c. for each person $p_i$ in $P_i$
        c.1.compute set of visits for the person $p_i$ by reading visits file
        c.2.send the visits to its corresponding message broker $MB_i$
  end of for
  d. message broker $MB_i$ processes all the message,which are visits information of people.
  s1: synchronize
  e. for each location $l_j$ in $L_j$
        e.1. process each visits message to generate a simple discrete system,in which each
             visits is decomposed to an arrive or depart event.
  end of for
  f. message broker $MB_i$ process messages.
  s2: synchronize
  g. for each $j$ in $n$
        g.1. for each person $p_i$ in $P_j$
              g.1.2.update health state based on received outcomes.
  end of for
  s3: synchronize
end of for
```

3.3 Coupled Contagion Dynamics of Fear and Disease [4]

Epstein et al [4] have defined a coupled contagion model featuring disease and fear of disease spread. They have analyzed the behavior of people during an epidemic and described how fear can dramatically affect the spatio-temporal spread of disease.

They have defined the states an individual occupies during an epidemic. Each individual must be in one of following states:

1. Susceptible to disease and fear
2. Infected with disease only
3. Infected with fear only
4. Infected with both disease and fear
5. Isolated from population due to fear
6. Isolated from population due to fear and infected
7. Recovered from disease and immuned to fear

They have used the Kermack-McKendrick model to implement their system using a differential equation scheme. This assumes uniform mixing in the population. The compartments are the above mentioned seven states occupied by individuals in the population. They have also defined an agent-based computational model for the same. Their work shows the importance of behavioral patterns when it comes to disease spread, including flight as a behavioral response to disease [4].

Chapter 4

Person-Person Network Architecture

4.1 GDS Representation of Person-Person Network

Graph dynamical systems [27] depicts an abstract mathematical representation of complex systems. A complex system represents networks, where each node represents an entity with states and local functions that determine the results of interactions with its neighbor entities [28][29][27].

The examples of such complex systems include, socio-technical systems, power systems and the stock market.

The complex system resulting from interactions over Person-Person network can be described as GDS G_1 as $(G(V, E), F, W)$. The components of the GDS include $G(V, E)$ which is the underlying information flow contact network, where V denotes a set of people in the population and set of edges E represents their corresponding interaction. For each v_i in V, there is a transition function f_{v_i} and set of these functions is denoted by F. The transition in the state of vertex v_i between t to $t + 1$, is determined by f_{v_i}, which is a function of:

1. state of vertex v_i
2. set of interaction with its neighbor at time t

The function f_i for each vertex is stochastic in nature and is used to decide if there is a social contagion (fear) transmission between the host vertex and its neighbor. The component W over the set of all vertices $v_1, v_2, v_3, v_4, ., v_n$ represents the schedule that defines the order in which a state of vertices will be updated [29].

4.2 Person-Person Network Design

EpiSimdemics is an algorithm for simulating the spread of infectious disease over a large realistic social contact network. Person-Location, the underlying network for EpiSimdemics, is a bipartite graph consisting of Person as one set of nodes and Location as another set of nodes. The edge between person node p_1 to location node l_1, represents person p_1 visiting location l_1.

The underlying Person-Location network cannot simulate interactions that do not involve spatial co-locality of involved agents such communication over telephone, email and tweets. To model these kinds of interaction, adding a Person-Person network to EpiSimdemics was required. Essentially, Person-Person network captures various kinds of interactions, where people do not physically interact. Algorithm 2 presents Person-Person network design in EpiSimdemics.

The following are the three main functional modules added as a part of the Person-Person network in EpiSimdemics:

4.2.1 InputReader Module

InputReader Module: Simulation starts by reading all the input files, including persons, locations, home locations and visits files. The existing *InputReader* module in EpiSimdemics has subroutines for reading each of these data files. Person-Person network data is contained in socnet file, so additional functionality was added to *InputReader* module to read Person-Person network data from socnet file. This module creates a schedule vector for each person, which contains their corresponding interactions. Figure 4.1 depicts the functionality

Algorithm 2 P-P Network Design

```
(i)Read the person and socnet files (InputReader Module)
(ii)Generate schedule vector containing P-P records for each person
  for each day in simulation
    for each edge in P-P graph
      (iii)Send message to receiver along with infectivity and susceptibility of sender
      (iv)Compute fear infection for receiver and sender on receiver's
          cpu (between hosts fear propagation)
      (v)Handle infection invoked for receiver and sender based on their respective
          compute fear infection outcomes
    end of for
  end of for
```

of the *InputReader* module, which starts the simulation by reading the person and socnet file and storing each Person-Person communication record into a schedule vector for the communication sender.

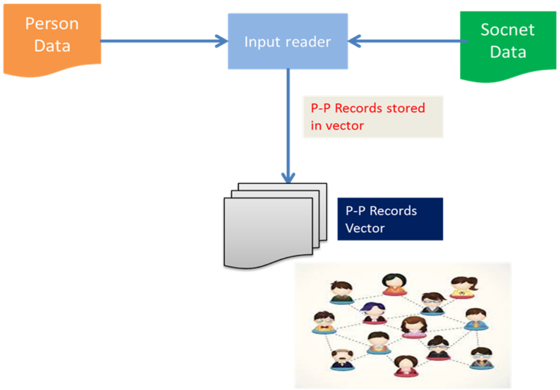


Figure 4.1: InputReader Module

4.2.2 SendInteractor Module

SendInteractor Module: This module reads the schedule vector for each person and send messages to the receiver of the communication along with the sender's fear characteristics. Figure 4.2 shows the *SendInteractor* module, reading records from the schedule vector and sending it to *ComputeInfection* Module.

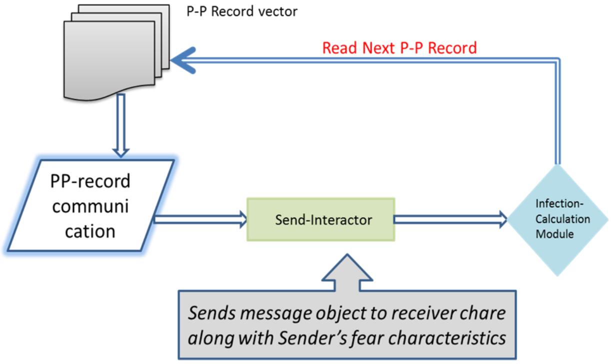


Figure 4.2: SendInteractor Module

4.2.3 ComputeInfection Module

ComputeInfection Module: This module is responsible for computing fear infection in sender and receiver. The computation happens in the receiver and based on the results of infection calculation, the receiver and sender handles the infection by undergoing a probabilistic state change. Since computation happens at the receiver, call to handle infection is local for the receiver whereas it is remote for the sender of the message. Figure. 4.3 shows the *computeInfection* module and depicts all the remote and local calls. The lines in red depict remote calls:

1. computeInfection call for sender
2. handle infection for sender

Lines in blue depict local calls which are:

1. computeInfection call for receiver
2. handle infection for receiver

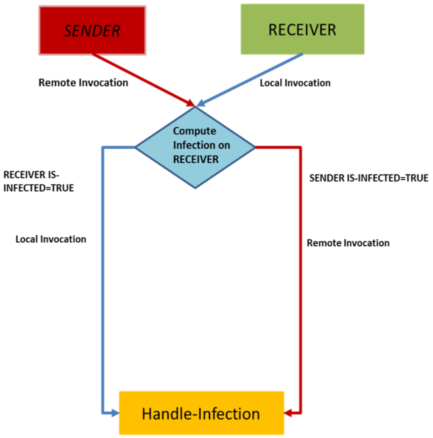


Figure 4.3: ComputeInfection Module

4.3 Input Data File

EpiSimdemics requires population information in the form of data files during run time in order to start simulations. The absolute path for the required input files must be provided in the configuration file. The following files are needed to run EpiSimdemics without a Person-Person network:

1. Person
2. Locations
3. Visits
4. Home Locations

The socnet file contains Person-Person network data.

Following is the format of the socnet file:

Table 4.1: Socnet File

PersonFrom	acttype	PersonTo	acttypeTo	duration
616509812	1	616509811	1	43799
616509813	1	616509811	1	43799
616509813	1	616509812	1	65699

- PersonFrom: Id of the person who will be sending the message.
- PersonTo: Id of the person who will be receiving the message.
- Duration: Total time for which communication took place.
- Acttype: Activity type for the person sending the message. Since, the Person-Person network does not involve location, this field would correspond to a home location, which is 1.
- ActtypeTo: Activity type for the person who is receiving the message.

4.4 Charm++

Charm++, a parallel programming language, developed by researchers at UIUC, provides a highly scalable framework. It provides strong parallel programming libraries in C++, which is particularly useful for scientific purposes. It is built on the principle of making parallel programming easier for novice programmers and thus, provides high-level abstraction to programmers so that they don't have to worry about parallelism at the instruction level [30][31].

Charm++ fragments the programs into a number of chares and they communicate among themselves by message passing interfaces. These chares are represented in the form of message-driven objects. So, whenever a sub-routine is invoked on an object, the internal Charm++ libraries forming the run time environment for programs sends out the message to the corresponding object. As Charm++ provides a distributed environment, objects can be distributed among various processors. Message handling is done asynchronously in chare objects [30].

The Charm++ libraries provide support for the array of chares and has the functionality to send messages to individual chares in the array. The individual chare can be accessed just like an element in an array by accessing it at a particular index. The Charm++ also provides broadcast support where all the chares receive the same message when it is broadcasted.

Charm++ libraries maps the individual chares to the processors and this process is transparent to the programmer. The programmers can logically map the chares to their processors based on a particular scheme.

The libraries can also provide a completely transparent assignment of these chares and in those cases, charm++ is responsible for load balancing, resource management and maintaining the continuity of service. The optimal resource management and avoiding denial of service are amongst the top priorities in Charm++ environment.

4.4.1 Program Structure In Charm++ Environment

The programs written using Charm++ comprise of many chares. These chares interact with each other by sending and receiving messages. During the program execution each chare object is associated with a state and entry methods. Whenever a chare object receives a message, its corresponding entry method is executed. The entry method can have any implementation that is suitable for chare functionality. This may include some file-processing or simply queuing all the messages. The entry method functionality depends primarily on the chare. The Charm++ run time libraries are responsible for routing the messages to their destination chare. This process is transparent from the programmers view.

4.4.2 Global Object Space and Main Chare

The execution of programs written in Charm++ starts with the main chare. Each chare is associated with a state. The global object space refers to the set of all chares in a program and chares in this global object space can request each other's attributes by sending and receiving messages through a remote procedural call.

Figure 4.4 represents an example of the user's view of chare communication. There are two chare objects represented in the figure. The chare Sender has two entry methods:

```
void entry-A()
void entry-C(msg)
```

Similarly, chare Receiver has two entry methods:

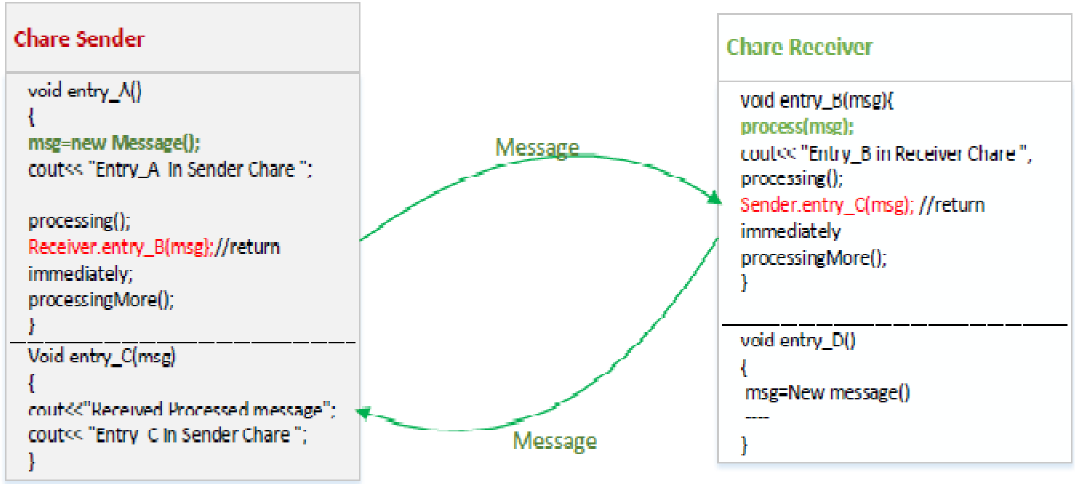
```
void entry-B(msg)
void entry-D()
```

Sender chare executes the entry method in *Receiver* chare by invoking its entry method-*entry-B(msg)*. Through, this remote procedural invocation chare *Sender* is able to send its message to the *Receiver* chare.

This would cause the execution of *entry-B(msg)* in *Receiver* with 'msg' as an argument which was sent from the *Sender* chare. This message passing is asynchronous in Charm++ which implies line following the method call in chare *Sender* is immediately executed and a call to *entry-A(msg)* returns instantly.

Similarly, the *Receiver* chare executes *entry-C(msg)* in chare *Sender* by sending 'msg' argument which was processed by chare *Receiver*. Therefore, the *Receiver* chare is able to send its processed message to the *Sender* chare.

This kind of message passing is also referred to as remote method invocation (rmi) because the calling chare is executing the member method of a different chare. These method calls are asynchronous in nature and program execution continues normally after these calls.



```
void entry_A()
{
msg=new Message();
cout<< "Entry_A In Sender Chare ";

processing();
Receiver.entry_B(msg);//return
immediately;
processingMore();
}
------------------------------------
Void entry_C(msg)
{
cout<<"Received Processed message";
cout<< "Entry_C In Sender Chare ";
}
```

```
void entry_B(msg){
process(msg);
cout<< "Entry_B in Receiver Chare ",
processing();
Sender.entry_C(msg); //return
immediately
processingMore();
}

------------------------------------
void entry_D()
{
 msg=New message()
 ----
}
```

Figure 4.4: User View Of Chare Communication

4.4.3 Message Passing in Person-Person network

The message passing in Person-Person network is achieved through inter-chare communication. The person manager corresponding to *PersonFrom* sends message to person manager for *PersonTo*. The message comprises of records in the socnet file and sender fear characteristics.

4.5 Logical Processor Assignment To Population

Each person is assigned to one of the logical processors, PersonManager, during initialization phases. The number of processors used for an experiment is configurable. The choice for the number of processors primarily depends on the population size as scalability is achieved by adding hardware resources. During simulation, increasing the number of compute nodes would allocate more hardware resources. Each person is assigned to a person manager based on the following scheme:

$$PersonManager = PersonId\%NumofProcessors \tag{4.1}$$

So, modulo operator is used to assign each agent to their corresponding person manager which also represents a person's logical CPU assignment. The person manager information for these agents is used by charm++ run time libraries for sending messages between them.

4.6 Instance of Inter-Chare Communication in EpiSimdemics

The following snippet of a method invocation is an example of inter-chare communication in EpiSimdemics:

```
gPmanArray[TargetCPU].TargetFunction(args)
```

1. **gPmanArray**: Array of person managers. Each CPU is mapped to a person manager. Each individual is assigned to a particular person manager. Array-indexing can be used to access different person managers and internally person managers can send message to receiver of communication.

2. **TargetCPU**: Index of person manager in person manager array that contains the person, where the *SampleFunction* needs to be executed.

4.7 Supporting Classes and data structures

The following header and class files are required to construct *PersonPersonData* objects and each such *PersonPersonData* object represents a row in socnet file, which is an instance of communication between two agents:

- **PersonPerson.h**
- **PersonPerson.C**
- **PersonPersonData.h**
- **PersonPersonData.C**

Each *PersonPersonData* message comprises of a sender (Person), receiver (PersonTo) and attributes of message (duration of communication). Each such record is pushed into a vector called,*fSchedulesPerson* for each person in EpiSimdemics.

4.8 Simulation Configuration

The essential parameters for running the simulation are configurable at simulation time. EpiSimdemics requires an absolute path to all the input data files in-order to start the simulation. Some important parameters that can be set in the configuration file includes:

- Random seed.
- Days- total number of simulation days.
- FSMs- details about the PTTS which would allow within host disease and fear transmission.

- Scenario file- the file containing triggers and interventions.
- Input Files Path - the configuration file contains the path for all the input files needed for simulation run.

  So, all the simulation specific properties are set in configuration file.

4.9 Running EpiSimdemics with Person-Person Network only:

In order to run simulations only on the Person-Person network in EpiSimdemics, the underlying Person-Location network needs to be disabled. To disable the Person-Location network, visits and locations files need to be disabled. As a result, EpiSimdemics does not read locations and visits because the underlying network does not require the data. Therefore, an empty file without any data needs to be provided when we want to run EpiSimdemics on a Person-Person network only. However, EpiSimdemics would throw a file not found exception, if these files are not provided in the configuration file at the run time.

4.10 EpSimdemics with Person-Person and Person-Location Network:

EpiSimdemics by default would incorporate Person-Location and Person-Person networks. The simulation would proceed by first reading the Person-Location network and then the Person-Person network. In this case, the configuration file needs to have pertinent location, visits and socnet files. The output files won't depict any of the states corresponding to the Fear Model unless, the scenario file is modified to set attributes of the Fear Model.

Chapter 5

Disease Model and Fear Model

5.1 Finite State Machine

Finite state machine (fsm) is a machine abstraction comprising of multiple states and an intervention/action is defined for the transition between two states of fsm. The fsm has a current state and can transition between states if corresponding action is triggered.

5.2 Disease PTTS

EpiSimdemics uses a variant of finite state machine named probabilistic timed transition system (PTTS) to represent the within-host disease progression. The PTTS representation of the Disease Model is presented in figure 5.1. Ovals represent disease states, while lines represent the transition between states, labeled with the transition probabilities. The line type represents the kind of action applied for that transition. The states contain a label and the dwell time within the state.

Both the Disease Model and Fear Model are represented using PTTS. The manifestation file in EpiSimdemics stages each of these PTTS.

Figure 5.2 shows a snippet of the manifest file for disease PTTS. Each of the transitions

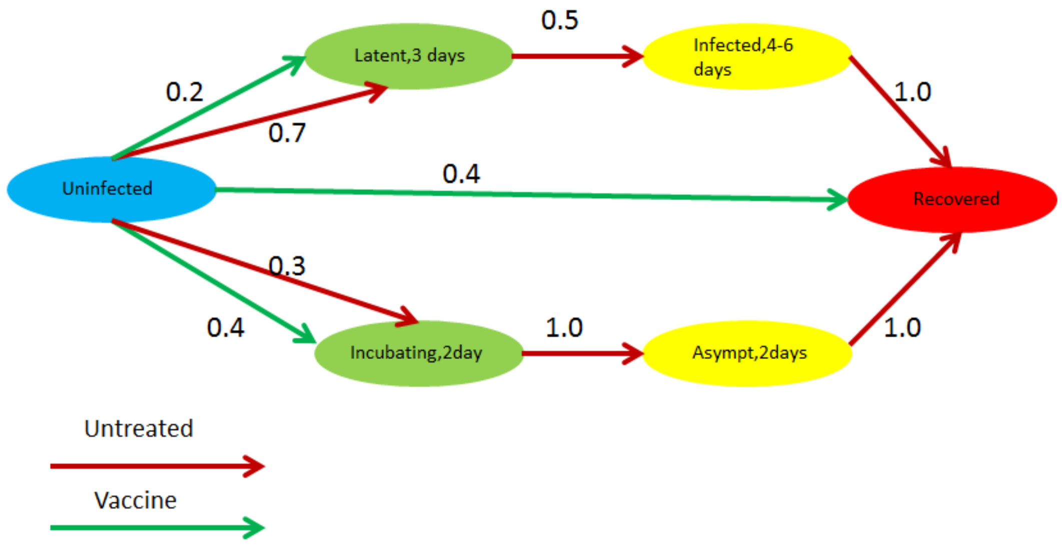
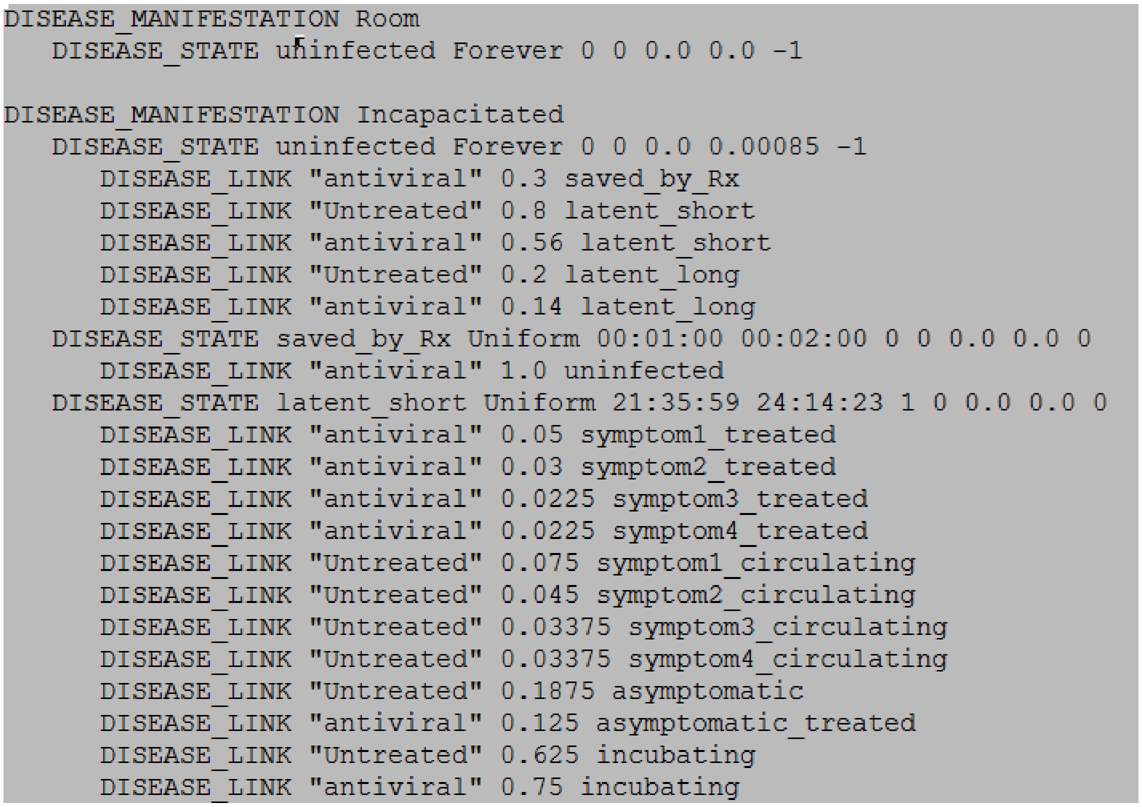


Figure 5.1: Disease Model

depicted in figure 5.1 is represented by a disease link in manifest file. Each DISEASE-STATE includes the following information:

- **StateName**
- **Dwell Time Distribution**
- **Infectivity**
- **Susceptibility**
- **Infection State(-1 for uninfected,0 for infected)**

The DISEASE-LINK represents a transition to different states for various transitions defined in the Disease Model. The sum of transition probabilities between states for a particular transition type, is equal to one.



```
DISEASE_MANIFESTATION Room
   DISEASE_STATE uninfected Forever 0 0 0.0 0.0 -1

DISEASE_MANIFESTATION Incapacitated
   DISEASE_STATE uninfected Forever 0 0 0.0 0.00085 -1
      DISEASE_LINK "antiviral" 0.3 saved_by_Rx
      DISEASE_LINK "Untreated" 0.8 latent_short
      DISEASE_LINK "antiviral" 0.56 latent_short
      DISEASE_LINK "Untreated" 0.2 latent_long
      DISEASE_LINK "antiviral" 0.14 latent_long
   DISEASE_STATE saved_by_Rx Uniform 00:01:00 00:02:00 0 0 0.0 0.0 0
      DISEASE_LINK "antiviral" 1.0 uninfected
   DISEASE_STATE latent_short Uniform 21:35:59 24:14:23 1 0 0.0 0.0 0
      DISEASE_LINK "antiviral" 0.05 symptom1_treated
      DISEASE_LINK "antiviral" 0.03 symptom2_treated
      DISEASE_LINK "antiviral" 0.0225 symptom3_treated
      DISEASE_LINK "antiviral" 0.0225 symptom4_treated
      DISEASE_LINK "Untreated" 0.075 symptom1_circulating
      DISEASE_LINK "Untreated" 0.045 symptom2_circulating
      DISEASE_LINK "Untreated" 0.03375 symptom3_circulating
      DISEASE_LINK "Untreated" 0.03375 symptom4_circulating
      DISEASE_LINK "Untreated" 0.1875 asymptomatic
      DISEASE_LINK "antiviral" 0.125 asymptomatic_treated
      DISEASE_LINK "Untreated" 0.625 incubating
      DISEASE_LINK "antiviral" 0.75 incubating
```

Figure 5.2: Disease Manifestation

5.3 Fear PTTS

The Fear Model is represented in the form of fear PTTS, defined in figure 5.3.

The four states represented in fear PTTS are:

1. Lowfear- lowest fear level about the disease. Possible transitions in this state are to stay in Lowfear state and to transition forward to Mediumfear state.
2. Mediumfear- higher level of fear than Lowfear about the disease. Possible transitions include moving back to Lowfear state, remain in Mediumfear state or move forward to Highfear state, as a result of state change.
3. Highfear- fear level higher than Mediumfear with three possible transitions in this state: remaining in Highfear, transition forward to VeryHighfear and transition back

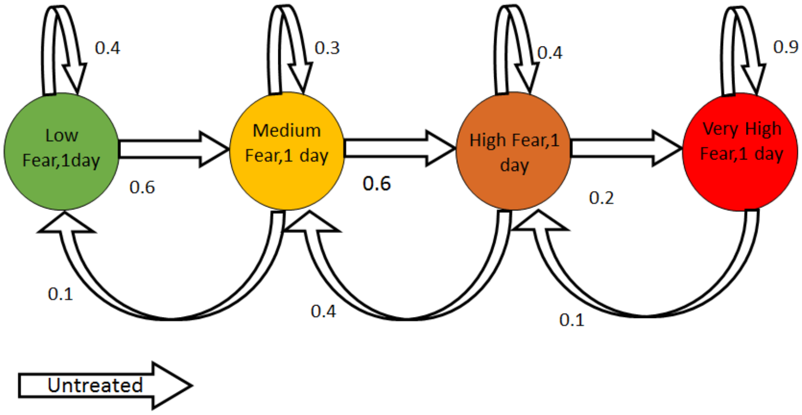


Figure 5.3: Fear PTTS

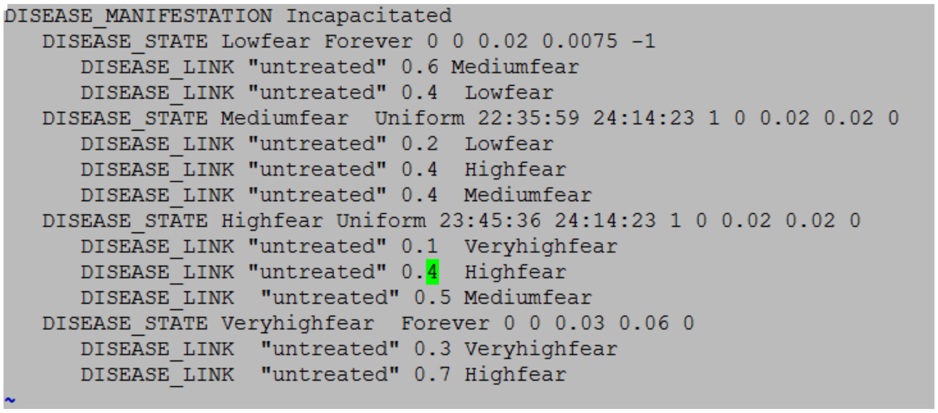
to Mediumfear.

4. VeryHighfear- highest fear level with two transitions from this state. The transitions are to remain in VeryHighfear or transition back to Highfear state, as a result of state change.

Each transition is represented by an arrow and marked with a probability value of that transition. A person can only be in one of these states and an entire population will be in one of these states. Figure 5.4 shows the manifest file for fear PTTS. As stated above there are four states in this file. Each state is associated with the state name, dwell time distribution, infectivity value, susceptibility value and infection state. The DISEASE-LINK depicts the transitions to different states and each link is associated with a probability and action type. The action in this manifestation file is ”untreated”.

5.4 Fear Dynamics

Fear transmission is controlled by two connected and independent processes, within host fear evolution and between hosts fear propagation. The following sections present these processes.



```
DISEASE_MANIFESTATION Incapacitated
   DISEASE_STATE Lowfear Forever 0 0 0.02 0.0075 -1
      DISEASE_LINK "untreated" 0.6 Mediumfear
      DISEASE_LINK "untreated" 0.4  Lowfear
   DISEASE_STATE Mediumfear  Uniform 22:35:59 24:14:23 1 0 0.02 0.02 0
      DISEASE_LINK "untreated" 0.2  Lowfear
      DISEASE_LINK "untreated" 0.4  Highfear
      DISEASE_LINK "untreated" 0.4  Mediumfear
   DISEASE_STATE Highfear Uniform 23:45:36 24:14:23 1 0 0.02 0.02 0
      DISEASE_LINK "untreated" 0.1  Veryhighfear
      DISEASE_LINK "untreated" 0.4  Highfear
      DISEASE_LINK  "untreated" 0.5 Mediumfear
   DISEASE_STATE Veryhighfear  Forever 0 0 0.03 0.06 0
      DISEASE_LINK  "untreated" 0.3 Veryhighfear
      DISEASE_LINK  "untreated" 0.7 Highfear
~
```

Figure 5.4: Fear Manifestation

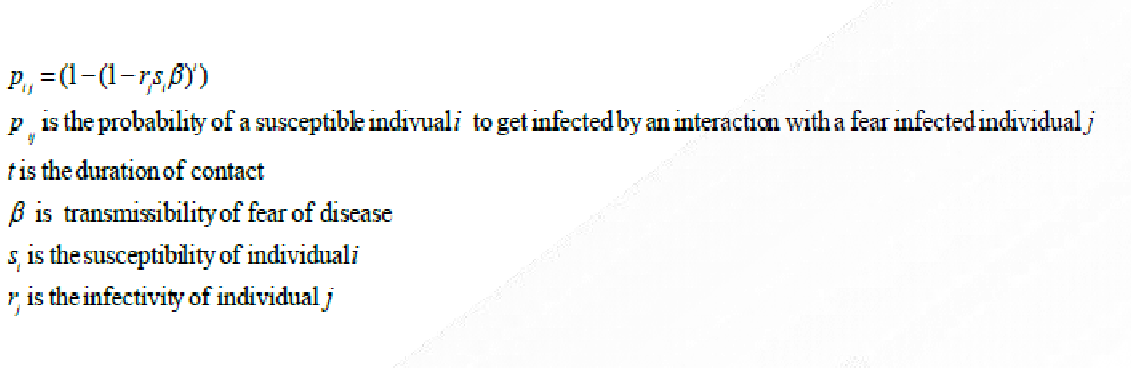
5.4.1 Within Host Fear evolution

Within host fear evolution defines the transition of person in fear PTTS, which are not due to handle infection. The dwell time of each state in fear PTTS defines the time a person will stay in that state and after that time elapses a person can probabilistic ally move to its neighbor states. For example, a person in *MediumFear* who is not interacting with any other person will remain in this state for one day. At the end of a day,a person in this state can choose to follow the three outgoing links from this state and may move to a new state.

5.4.2 Between Hosts Fear Transmission

Between-hosts fear transmission specifies the effects of interactions between individuals i.e., how does an uninfected individual become infected after being exposed to the social contagion (fear) by an infected individual. For example, how does an uninfected person's fear level change, after interacting with an H1N1 fear infected individual via person-person communication media. Such transmissions of social contagion is probabilistic in nature and depend on the susceptivity of the receiver, infectivity of the sender and duration of their contact.

If the infectivity of the sender and susceptibility of receiver are both greater than zero, then there is a possibility of the receiver getting infected with fear. However, if the susceptibility of the sender and infectivity of the receiver are also both greater than zero, then the sender has a probability of getting infected through interaction, with receiver of the message. This is because transmission works on information flow contact which can transmit the contagion in either direction. Therefore, simulation considers the flow of infection in both directions to compute the results of interaction. The probability calculation is the function of the infectivity of an infected person, susceptibility of a suscepted person, duration of contact and transmissibility of fear, which is the characteristic of a contagion that defines its strength in terms of the probability of a susceptible person getting infected by an infected person during one minute of exposure contact [3].



$p_{i,j} = (1-(1-r_j s_i \beta)^t)$

p_{ij} is the probability of a susceptible indivual i to get infected by an interaction with a fear infected individual j

t is the duration of contact

β is transmissibility of fear of disease

s_i is the susceptibility of individual i

r_j is the infectivity of individual j

Figure 5.5: Compute Infection-Fear

Figure 5.5 presents a function that determines the influence of interaction between two people.

5.5 Interaction of Person-Person and Person-Location Network

With the induction of a Person-Person network, EpiSimdemics stages multiple interacting networks. The existent Person-Location network interacts with the additional Person-Person network. Each person has two PTTS's that determine their transitions during between hosts and within hosts contagion spread. Disease progression happens over a Person-Location network for two people whereas fear progression happens over a Person-Person network for two people. These two different contagions represented by their respective PTTS's interact for the same person in a population, but their interaction between two persons is not modeled in the current implementation.

Person-Location network is used for disease progressions over the Disease Model and fear progression is modeled by the Fear Model over a Person-Person network.

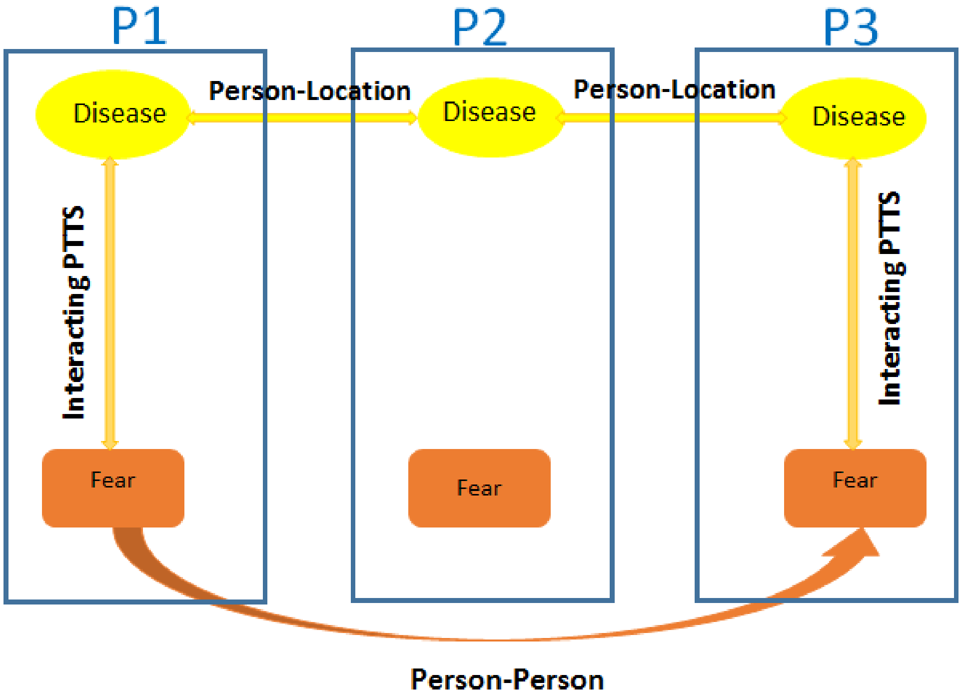


Figure 5.6: Interaction Over Person-Person and Person-Location Networks

Figure 5.6 represents all the possible interactions in EpiSimdemics with an interacting Disease and Fear Model for 3 people represented as $P1$,$P2$ and $P3$ in population. It shows the following interactions:

Over Person-Location Network: Yellow lines marked with "Person-Location" represent interactions over the Person-Location network. The two such interaction represented are $P1 - P2$ and $P2 - P3$. These interaction may lead to between host disease progression for interacting people.

Over Person-Person Network: The orange line marked with "Person-Person" represents an interaction over the Person-Person network graph. Only one such line is represented in figure 5.6 between $P1$ and $P3$.

Populations can interact over either or both of the above mentioned networks. Therefore, a person can interact independently over both of these networks. For example: $P1$ interacts with $P3$ over the Person-Person network. Whereas, $P1$ interacts with $P2$ over the Person-Location network.

Interaction between Disease and Fear Model: Lines marked with "Interacting PTTS" represent interaction between the Disease and Fear Model. Two such lines represented in figure 5.6 are between $P1 - P1$ and $P3 - P3$. These lines represent the interaction between two models for the same person. So, the Disease and Fear Model interact for the same person and this interaction may affect transitions in respective models.

Current implementation of the Person-Person network in EpiSimdemics does not support interactions between the Disease and Fear model of two individuals in a population. Therefore, there cannot be a line in figure 5.6 representing "Interacting PTTS" between two different people.

5.6 Interacting PTTS

The existing Person-Location network allows EpiSimdemics to model disease spread. With the addition of the Person-Person network, EpiSimdemics has the capability of modeling social (complex) contagion diffusion on an information flow contact network. To better understand the dynamics of disease epidemics, the Person-Person network helps gauge the influence of fear on the spread of disease and vice-versa. Disease spread is influenced by interactions over the Person-Person network. They tend to communicate over media like tweets and emails to spread sentiments, fear and anxiety about the disease. These interactions can result in awareness about the disease and in more critical scenarios this may take the shape of a rumor about the disease. The affect adaptations impact the spread of disease because they induce behavioral changes in the population and people start responding to the spread of disease by adopting preventive measures such as staying home from work, children staying home from school, avoiding leisure outings etc. Such an influence on disease spread by fear is modeled using interacting PTTS, which implies the state in one PTTS can affect the state in another PTTS by triggering transitions. The subsequent subsections present two examples of interacting PTTS and their influence on disease spread and fear.

5.6.1 Fear Elevation Intervention (Interacting PTTS)

In fear elevation intervention, an infected individual becomes more fearful by transitioning to a higher fear state. Such an interaction of Disease and Fear PTTS is depicted in Figure 5.7. The figure shows a person in state 'f' in Fear PTTS and state 'Infected' in Disease PTTS. The arrows depicts the probabilistic transition of a person's fear PTTS to higher fear levels. As per this intervention fear level increases probabilistic ally for a person if infected with a disease. Thus, the Disease Model causes a transition in the Fear Model of an individual.

This scenario closely represents society behavior during an epidemic break. People tend to be more fearful about a disease once they are infected. This shows how the state in one PTTS affects the transition in another PTTS.

5.6.2 Triggers and Interventions Definition For Fear Elevation

This section presents the semantics and syntaxes of fear elevation intervention, which is defined in EpiSimdemics.

The intervention defined below causes the transition from Lowfear to Mediumfear in the Fear Model.

```
intervention transLow
   transition fear:Incapacitated:Mediumfear
```

In the following trigger checks if a person's state in the Disease Model is Infected and their state in the Fear Model is Lowfear, a trigger executes *transLow* intervention, which causes probabilistic transition in Fear Model as per the *transLow* intervention.

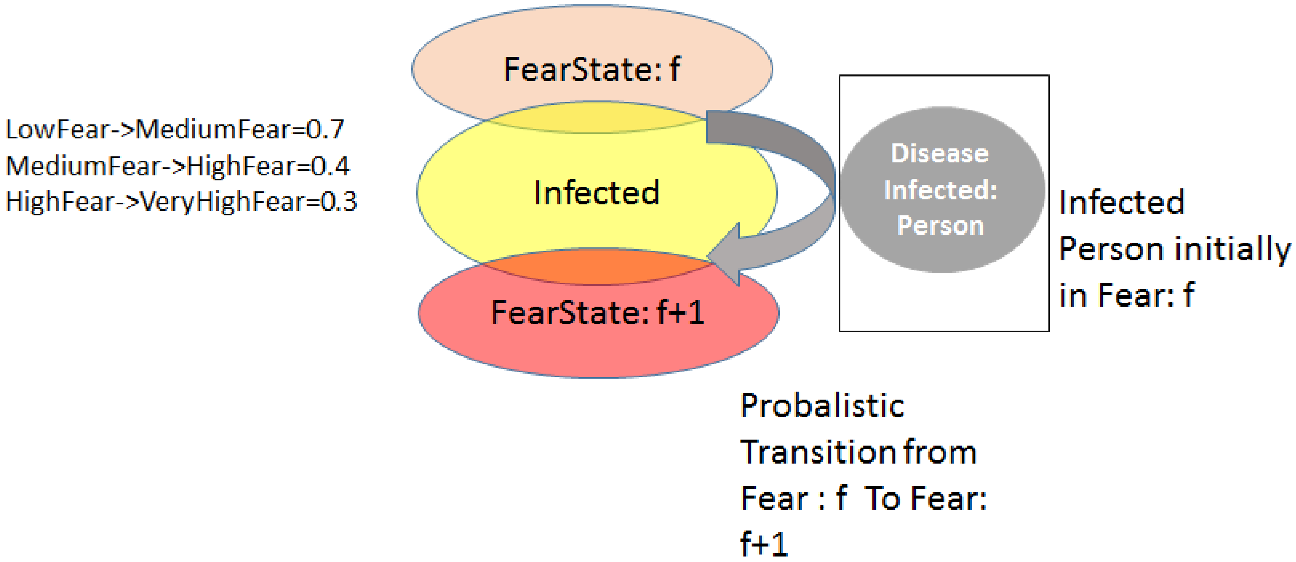


Figure 5.7: Fear Elevation Intervention

```
trigger disease.infected = 1 and fear.state ="fear:Incapacitated:Lowfear"
    apply transLow with prob=0.1
```

The following set of interventions and triggers describe transitions in the Fear Model based on the Infected state of person in the Disease Model.

```
intervention transLow
   transition  fear:Incapacitated:Mediumfear

intervention transMedium
   transition fear:Incapacitated:Highfear
```

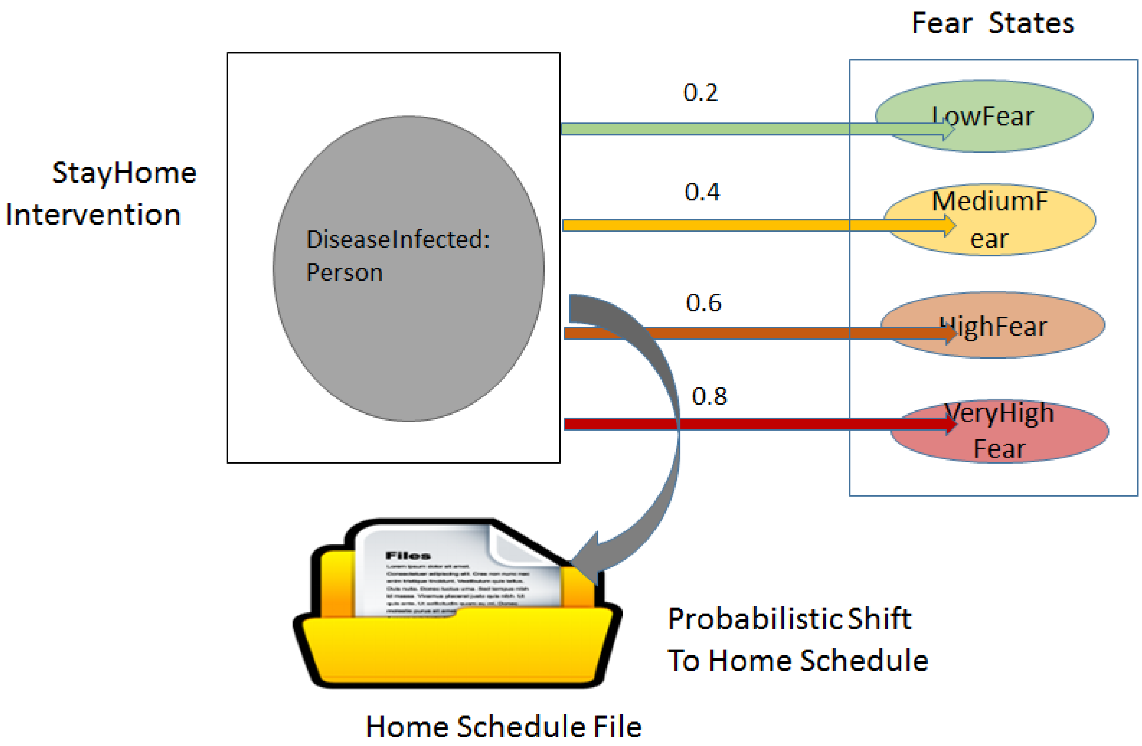
```
intervention transHigh
   transition fear:Incapacitated:Veryhighfear

trigger disease.infected = 1 and fear.state = "fear:Incapacitated:Lowfear"
   apply transLow with prob=0.9

trigger disease.infected = 1 and fear.state = "fear:Incapacitated:Mediumfear"
   apply transMedium with prob=0.9

trigger disease.infected = 1 and fear.state = "fear:Incapacitated:Highfear"
   apply transHigh with prob=0.9
```

5.6.3 StayHome Intervention (Interacting PTTS)

This subsection demonstrates interaction of the Disease and Fear Model in the form of a *StayHome intervention*, where fearful individuals remove themselves from the disease infected population through isolation. *StayHome intervention*, implies people will remain in their respective homes and will have interactions according to the Disease Model by interacting with their family members through a social contact network. In this scenario, based on the state in Fear Model individuals may choose to adopt a home schedule. People return to their normal schedule the next day. It does not directly change the person state in Disease PTTS, but it causes people to change their schedule. Thus, effectively changing the way disease get propagated.

People in more fearful states have a higher probability of staying home as they are very fearful about the spread of disease. This behavior in itself acts as a personal intervention because staying home prevents the spread of disease. Home schedule for each person is generated from their visits file by extracting visits corresponding to home locations.

Figure 5.8 presents the visual representation of *StayHome Intervention*, where a person executes StayHome intervention probabilistically based on different states in the Fear Model and an at home person can interact with their family members according to the Disease



begincenter

endcenter

Figure 5.8: StayHome Intervention

Model. In this example, the state in the Fear Model altered disease progression, as people started interaction with a different schedule, thus their usual interactions will differ. This will have a broad impact on the epidemic.

The addition of interacting PTTS gave EpiSimdemics more realistic, accurate and reliable prediction capabilities. This addition made EpiSimdemics more robust and well suited for a modern day epidemic prediction platform.

5.6.4 Triggers and Interventions For StayHome

The following section presents triggers when based on the state in Fear Model, probabilistically executes the *StayHome* intervention.

This intervention forces people to change their schedule to their home schedule. People return to their actual schedule the next day.

```
intervention stayhome
   schedule stayhome 1

trigger repeatable true
   unschedule 1

trigger repeatable fear.state ="fear:Incapacitated:Lowfear"
   apply stayhome with prob=0.1

trigger repeatable fear.state = "fear:Incapacitated:Mediumfear"
   apply stayhome with prob=0.2

trigger repeatable fear.state = "fear:Incapacitated:Highfear"
   apply stayhome with prob=0.4

trigger repeatable fear.state = "fear:Incapacitated:Veryhighfear"
   apply stayhome with prob=0.3
```

Chapter 6

Experiments and Results

EpiSimdemics produces comprehensive results at the end of each simulation phase. The major results of the experiment are generated in the form of a text file- **vars.txt**. This file contains the statistics about the results of the simulation and is continually updated as the simulation proceeds.

6.1 Disease Spread Output Header

Table 6.1 contains the header of *vars.txt* with a Person-Location network.

1. Day- simulation day number.
2. Cur-infected- number of people who are currently infected with the disease, i.e. daily infections.
3. Init-infected- number of people who were initially infected. It represents the number of people who were randomly infected on day 1.
4. Pop- represents the total population of the dataset which is being used for simulation. It is the total number of entries in Persons input file.

5. Total-infected- cumulative sum of daily infections till that day. It represents monotonically increasing value.

Table 6.1: Header of vars.txt (Output File for Disease Spread)

day	cur-infected	init-infected	pop	total-infected

These attributes help track the progression of disease during epidemics. It presents useful information like the number of infections or the number of recovery on a particular day, thus showing the trend in disease progression and time distribution when people start to recover, i.e when an epidemic starts to fade.

6.2 Fear Spread Output Header

The Table 6.2 presents the header of the output file for the Fear Model progression on a Person-Person network. The fear is used as an effect of an epidemic for the study and experimentation purposes.

Table 6.2: Header of vars.txt (Output File for Fear Spread)

day	highFear	lowFear	mediumFear	veryhighFear

1. **Day:** day number of the experiment.
2. **HighFear:** number of people in HighFear state in Fear Model on a that day.
3. **LowFear:** number of people in LowFear state in Fear Model on a that day.
4. **MediumFear:** number of people in MediumFear state in Fear Model on a that day.
5. **VeryHighFear:** number of people in VeryHighFear state in Fear Model on a that day.

Each person in the population occupies one of the states in the Fear Model on a particular day. Therefore, the sum of people in all states in the Fear Model on a given day equals the total population of the dataset.

Mathematically,

$$pop = people(highfear + lowFear + mediumFear + veryhighFear) \tag{6.1}$$

pop is the total population of the dataset, depicted in Figure 6.1

6.3 Combined Header (Disease and Fear) of Output

Figure 6.1 presents the snippet of the *vars.txt*, which is the output of simulation. Its header contains attributes of the Disease and Fear Model together because the underlying network in EpiSimdemics in this scenario contains both Person-Location and Person-Person networks.

The output shows the number for each attribute, which presents the number of people corresponding to that state.These different header fields and their names are configurable at the run time.

The following section presents the intervention and triggers which are responsible for capturing the values for the Disease Model.

6.4 Interventions and Set Attributes For Disease Model

The following intervention *infect-all* tracks the total number of infections up until a particular day in the *total-infected* header and the number of daily infections on a particular day in the *cur-infected* header.

```
intervention infect-all
   set cur-infected++
   set total-infected++
```

The following definition of intervention *not-infected-all* decrements daily infections by one.

```
intervention not-infect-all
   set cur-infected--
```

The following trigger checks the state of a person in the Disease Model and applies intervention *infect-all* if person is infected to account for infection. This call increments the value of *cur-infected* and *total-infected* by one.

```
trigger disease.infected = 1
   apply infect-all
```

The following trigger checks if the person in the Disease Model is in a *removed* state. The *removed* state implies a person has recovered from the disease.

```
trigger disease.state ="disease:Incapacitated:removed"
   apply not-infect-all
```

If a person is recovered from the disease, this trigger executes a *not-infect-all* intervention. This intervention decreases the value of *cur-infected* to account for recovery.

```
 intervention not-infect-all
    set cur-infected--
```

6.5 Interventions and Set Attributes For Fear Model

This section presents syntaxes and semantics of the command used to set attributes of the Fear Model.

The following increments Lowfear attributes of Fear Model if a person is entering *LowFear* state in the Fear Model.

```
state on entry fear:Incapacitated:Lowfear true
      set lowFear++
```

The following decrements a Lowfear attribute of the Fear Model if a person is exiting a *LowFear* state in the Fear Model.

```
state on exit fear:Incapacitated:Lowfear true
      set lowFear--
```

The following increments Mediumfear attributes of the Fear Model if a person is entering a *MediumFear* state in the Fear Model.

```
state on entry fear:Incapacitated:Mediumfear true
      set mediumFear++
```

The following decrements Mediumfear attribute of the Fear Model if a person is exiting a *MediumFear* state in the Fear Model.

```
state on exit fear:Incapacitated:Mediumfear true
      set mediumFear--
```

The following increments Highfear attributes of the Fear Model if a person is entering a *HighFear* state in the Fear Model

```
state on entry fear:Incapacitated:Highfear true
      set highFear++
```

The following decrements Highfear attribute of the Fear Model if a person is exiting a *HighFear* state in the Fear Model.

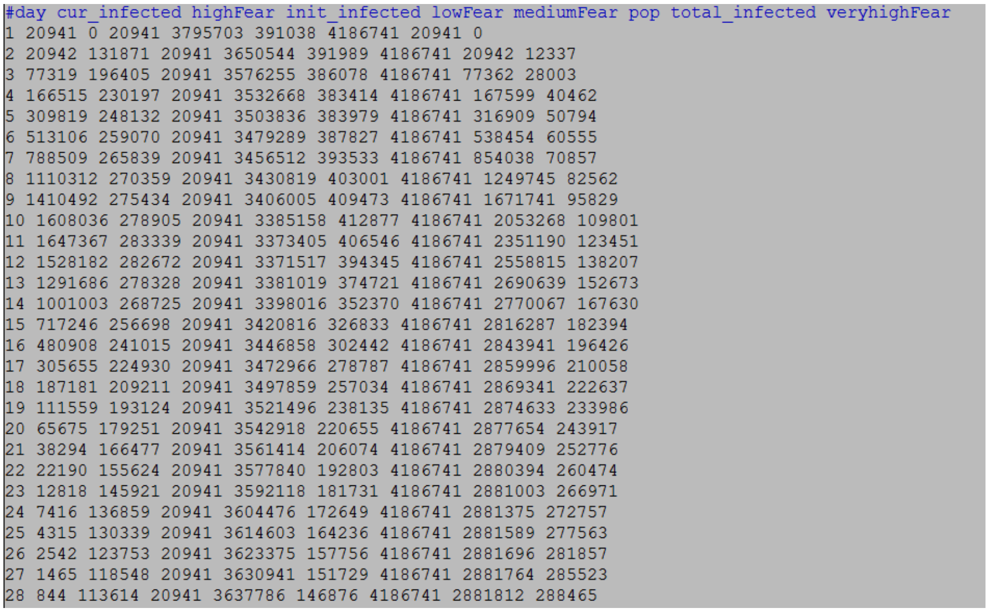
```
state on exit fear:Incapacitated:Highfear true
      set  highFear--
```

The following increments veryHighfear attributes of the Fear Model if a person is entering a *veryHighFear* state in the Fear Model

```
state on entry fear:Incapacitated:Veryhighfear true
      set veryhighFear++
```

The following decrements VeryHighfear attribute of the Fear Model if a person is exiting a *VeryHighFear* state in the Fear Model.

```
state on exit fear:Incapacitated:Veryhighfear true
      set veryhighFear-
```



```
#day cur_infected highFear init_infected lowFear mediumFear pop total_infected veryhighFear
1 20941 0 20941 3795703 391038 4186741 20941 0
2 20942 131871 20941 3650544 391989 4186741 20942 12337
3 77319 196405 20941 3576255 386078 4186741 77362 28003
4 166515 230197 20941 3532668 383414 4186741 167599 40462
5 309819 248132 20941 3503836 383979 4186741 316909 50794
6 513106 259070 20941 3479289 387827 4186741 538454 60555
7 788509 265839 20941 3456512 393533 4186741 854038 70857
8 1110312 270359 20941 3430819 403001 4186741 1249745 82562
9 1410492 275434 20941 3406005 409473 4186741 1671741 95829
10 1608036 278905 20941 3385158 412877 4186741 2053268 109801
11 1647367 283339 20941 3373405 406546 4186741 2351190 123451
12 1528182 282672 20941 3371517 394345 4186741 2558815 138207
13 1291686 278328 20941 3381019 374721 4186741 2690639 152673
14 1001003 268725 20941 3398016 352370 4186741 2770067 167630
15 717246 256698 20941 3420816 326833 4186741 2816287 182394
16 480908 241015 20941 3446858 302442 4186741 2843941 196426
17 305655 224930 20941 3472966 278787 4186741 2859996 210058
18 187181 209211 20941 3497859 257034 4186741 2869341 222637
19 111559 193124 20941 3521496 238135 4186741 2874633 233986
20 65675 179251 20941 3542918 220655 4186741 2877654 243917
21 38294 166477 20941 3561414 206074 4186741 2879409 252776
22 22190 155624 20941 3577840 192803 4186741 2880394 260474
23 12818 145921 20941 3592118 181731 4186741 2881003 266971
24 7416 136859 20941 3604476 172649 4186741 2881375 272757
25 4315 130339 20941 3614603 164236 4186741 2881589 277563
26 2542 123753 20941 3623375 157756 4186741 2881696 281857
27 1465 118548 20941 3630941 151729 4186741 2881764 285523
28 844 113614 20941 3637786 146876 4186741 2881812 288465
```

Figure 6.1: Output Of Combined Fear and Disease Model

Figure 6.1 shows the results for a particular simulation for 28 days on the Boston dataset. On day 1, random function infects a certain number of people, which is 20941 as reflected in *cur-infected* and *init-infected* headers. The *pop* header reflects the total population, which is the number of entries in Person input file. Figure 6.1 shows 41286741 as the total population for the Boston population dataset.

An entire population starts in a *LowFear* state in Fear Model. They start interacting and change states as they infect each other with fear and state changes occur because of within host fear evolution.

6.6 Experiments on Boston Population

Experiments were carried out to evaluate the correctness and usefulness of EpiSimdemics with a Person-Person network along with an underlying Person-Location network. EpiSimdemics is an interaction based simulation engine, therefore a highly dense urban population would be an idealistic choice for the study of added an Person-Person network and Fear Model.

Boston, being a metropolitan city presents a realistic population suited for an epidemics study. The density of interaction plays an important role in the spread of epidemic in society. An epidemic can spread rapidly in a highly dense environment thereby leaving a broader impact on health and society morale.

The synthetically prepared cellular network for Boston is used as data for the Person-Person network. Each such record has information about the callee, caller and duration of their talk. Communication over mobile devices between two people would form an instance of a calling network record. The person, location and visits data, which are input for Person-Location network are also synthetically generated. The complete data-set is generated in our lab Network Dynamics and Simulation Science Laboratory (NDSSL).

6.6.1 Experimental setup

Population is not infected with disease until the 60th day in the simulation. Till this point of time, fear follows its progression without being impacted by disease as there are no infections in the population. Population was randomly infected on the 60th day in the simulation.

6.6.2 Fear Progression Graphs

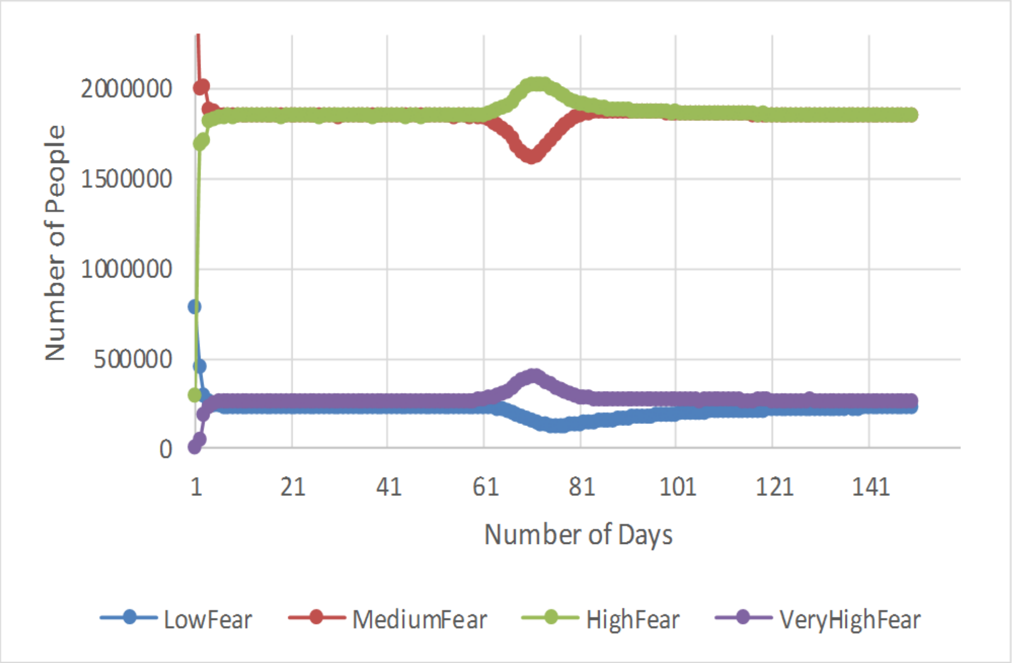


Figure 6.2: Fear Graph for Boston

Figure 6.2 represents the fear progression for the Boston population. The graph shown in the figure only contains fear progressions for the first 60 days when two PTTS's are non-interacting because there was no infection in population until the 60th day. So, fear follows its normal progression without being impacted by disease spread.

On the 60th day, people were randomly infected and disease progression began which impacted the fear progression. The graph shows peaks in HighFear and VeryHighFear states whereas dips are depicted in LowFear and MediumFear population states.

These changes are attributed to the interaction in two PTTSs. Fear Elevation intervention causes a probabilistic step up in population fear levels if they are infected. Therefore, we notice sudden changes in the graph. The fear progression again takes its normal course once people recover from the infection.

Figure 6.3 represents the fear graph for the Boston dataset where the Person-Person network was derived from its corresponding visits data. EpiSimdemics outputs the derived Person-Person network from visits data during simulation runs. The Person-Person network derived from visits data had more edges, which basically represents interactions among people. Table 7.3 presents the comparison between the size of these two networks in terms of number of the interactions. The levels of fear progression in both networks are similar, though they differ greatly in terms of simulation time.

6.6.3 Disease Progression

Figure 6.4 represents epi-curves of daily infections comparing scenarios when PTTS are non-interacting and interacting. Because of StayHome interventions interacting PTTS causes fewer daily infections because, depending on their fear states, people pickup their home schedule with a certain probability. This intervention isolates people, thereby decreasing daily infections.

Similarly, figure 6.5 represents the total infections. Since, the total infection cumulative occurred until a particular day is derived from daily infection, the nature of graph in figure 6.5 is similar to figure 6.4. Both cases depict lesser infection when disease and fear interact with each other. Intuitively, fear introduces awareness among people which forces them to

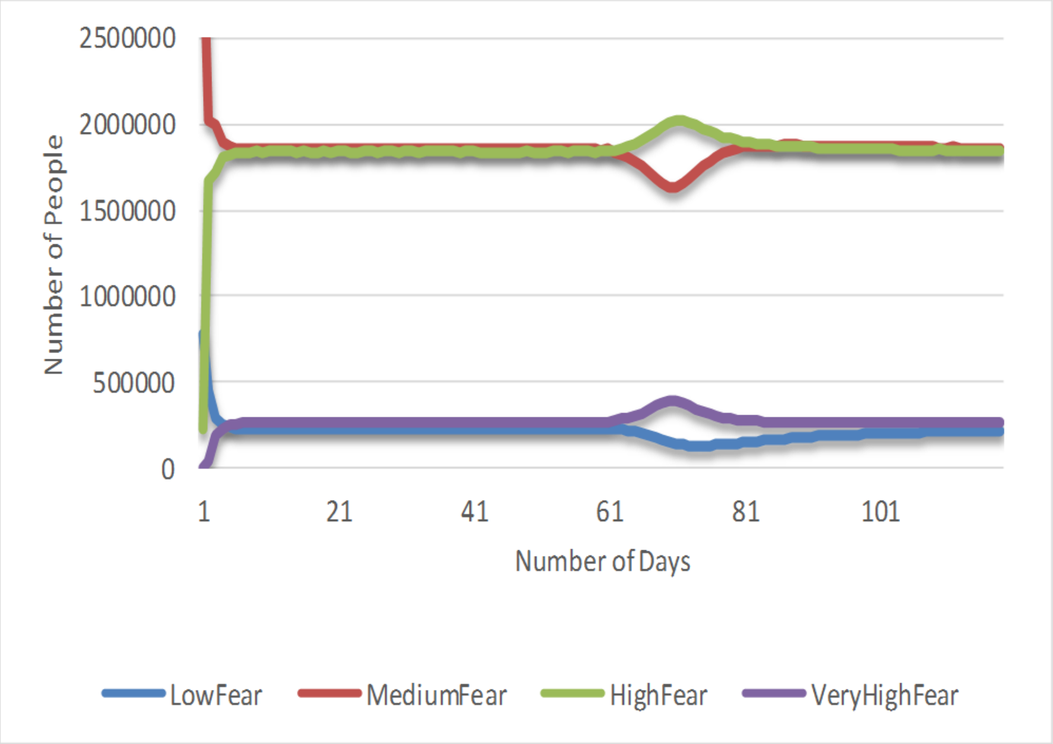


Figure 6.3: Fear Graph of Boston Person-Person Derived From Visits

be cautious and careful about the disease thereby minimizing their exposure towards such a disease. In that event a drastic decrease in infection is noticed.

Interaction between the Fear and Disease models cause a rise in the fear level of a population and decreases the over-all population infection. Therefore, spreading information through awareness (fear about the disease) can aid in controlling the epidemic of disease.

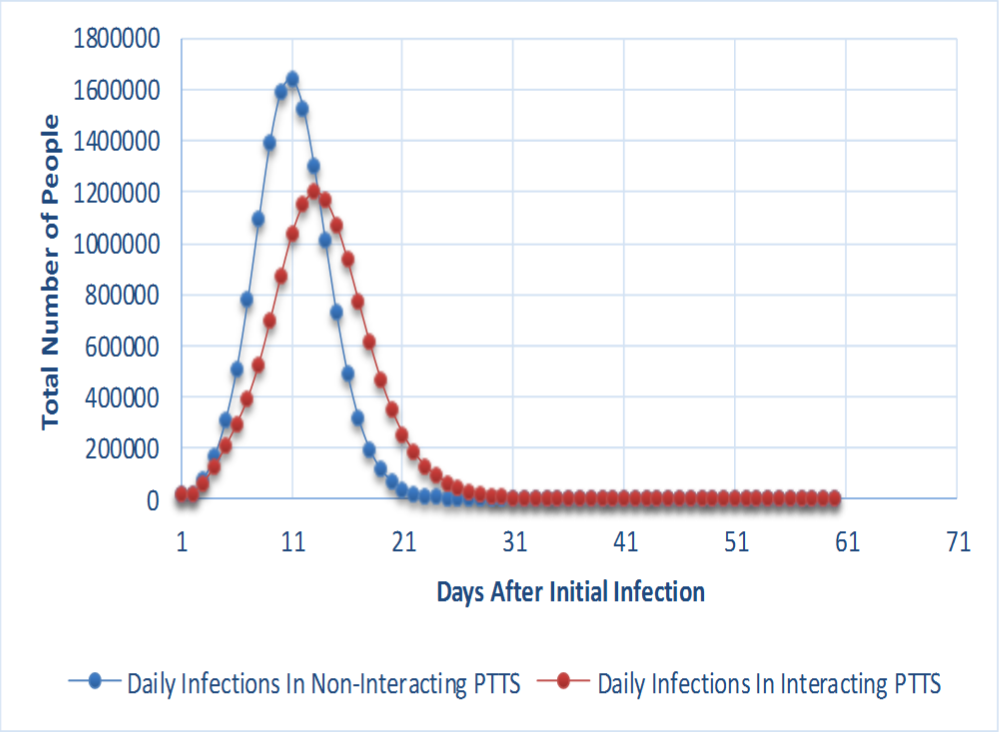


Figure 6.4: Currently (Daily) Infected for Boston

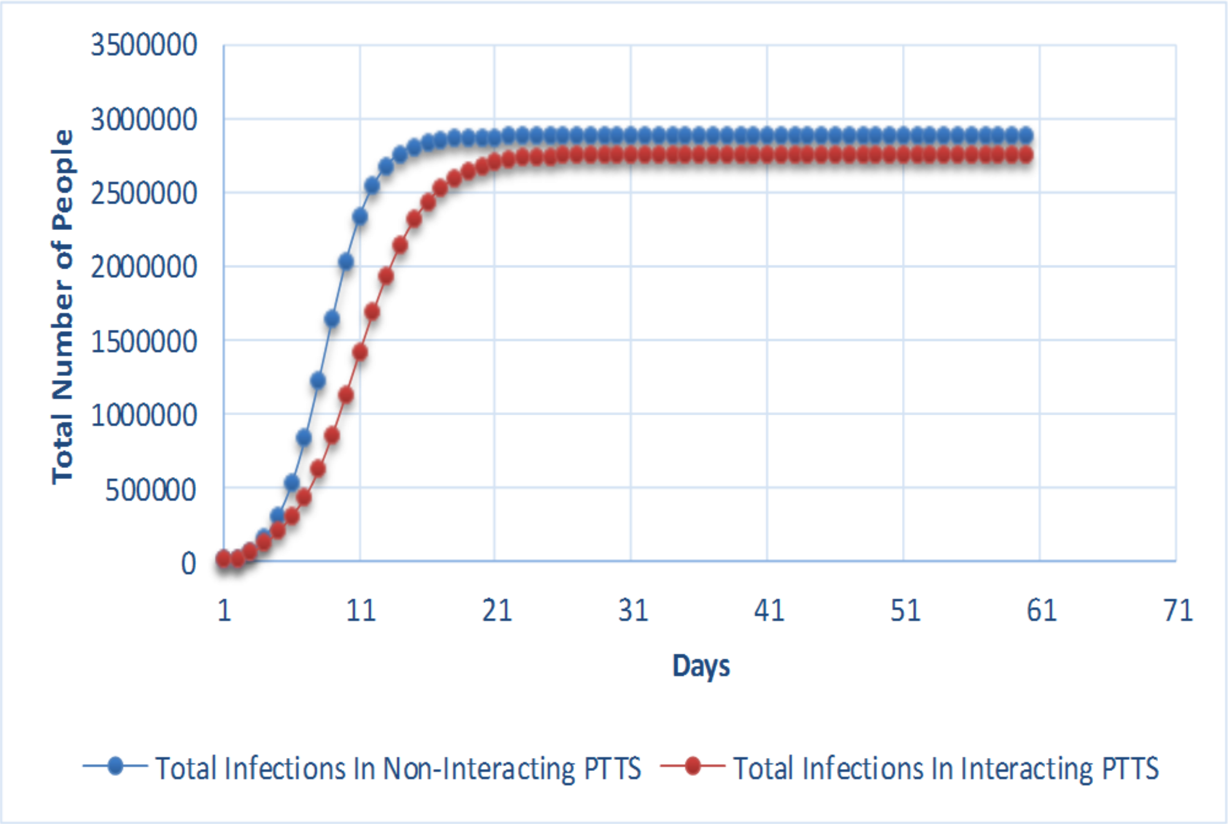


Figure 6.5: Total Infected for Boston

Chapter 7

Datasets, Execution Details and Time complexity

7.1 Time Complexity and computational analysis of Person-Person Network

The simulation time in EpiSimdemics is dominated by communication and computational loads. The communication comprises of inter-chare communication between two interacting people. During the start of each day in the simulation, each person's schedule is sent to their respective neighbors. The compute infection also contributes to the communication load as it uses a communication channel to send infect messages.

The computational aspects of simulation arises primarily from the compute infection phase. In this phase, probabilistically method invocation happens twice for each Person-Person schedule.

Let the total population of the dataset be n. Let the total number of interactions in Person-Person networks, which represents the number of edges in graph be m. Let the communication cost for one edge in Person-Person graph be c and the computation cost for processing each edge in Person-Person graph be p. A combined cost of communication and computa-

Table 7.1: Cluster Configuration

vendor-id	GenuineIntel
cpu family	6
model	44
model name	Intel(R) Xeon(R) CPU X5670@2.93GHz
stepping	2
cpu MHz	1600.000
cache size	12288 KB
cpu MHz	1600.000
cache size	12288 KB
cpu cores	6
RAM	4GB

tion for each edge will be $c + 2p$. Since there are a total of m edges in the graph, the total execution would be $m \times (c + 2p)$, which is O(m).

In the worst case scenario, which would be complete graph representing a Person-Person network, the number of edges in a Person-Person graph will be n^2. Therefore, the running time of algorithm in case of a Person-Person network being a complete graph, would be O(n^2).

7.2 Cluster Details

The result data was collected from simulations running on clusters. The number of compute nodes for these experiments are configurable, whereas the number of cores for each compute nodes depends on the hardware. The following is the hardware details of each core of a compute node:

The hardware where the experiments were performed has 12 cores per compute nodes. The charm++ creates 12 processes for each of those 12 cores per compute node.

7.3 Population Statistics

Table 7.2: Dataset Details

Networks	# of Persons	# of Daily Visits	Entries in Socnet	# of StayHome Schedules
Boston	4186746	23060075	1151186	8373478

Table 7.2 shows the sizes of various dataset tables for the Boston dataset.

7.4 Comparison of Simulation Time

Figure 7.1 depicts the comparison in simulation time between EpiSimdemics with a Person-Location network and EpiSimdemics with both Person-Person and Person-Location networks.

The Fear Model causes more invocations of compute infection than the Disease Model due to the inherent continuous fear circulation nature of the Fear Model. Invocation of compute infection is many fold higher a in Person-Person network than a Person-Location network. This adds to the simulation time because compute infection adds to computation costs and may also add communication cost, if infect messages needs to be communicated.

The message passing in a Person-Location network uses a complex message buffering approach, which buffers the outgoing messages to the same destination before sending them. This reduces the communication cost as a group of messages are sent together.

7.5 Performance Analysis For Varied Graph Density and Structure

Simulation were carried out on varied graph configurations to understand the simulation time with respect to the structure and density of graph. As the running time of simulations on a Person-Person network primarily depends on the number of edges in the graph, increasing

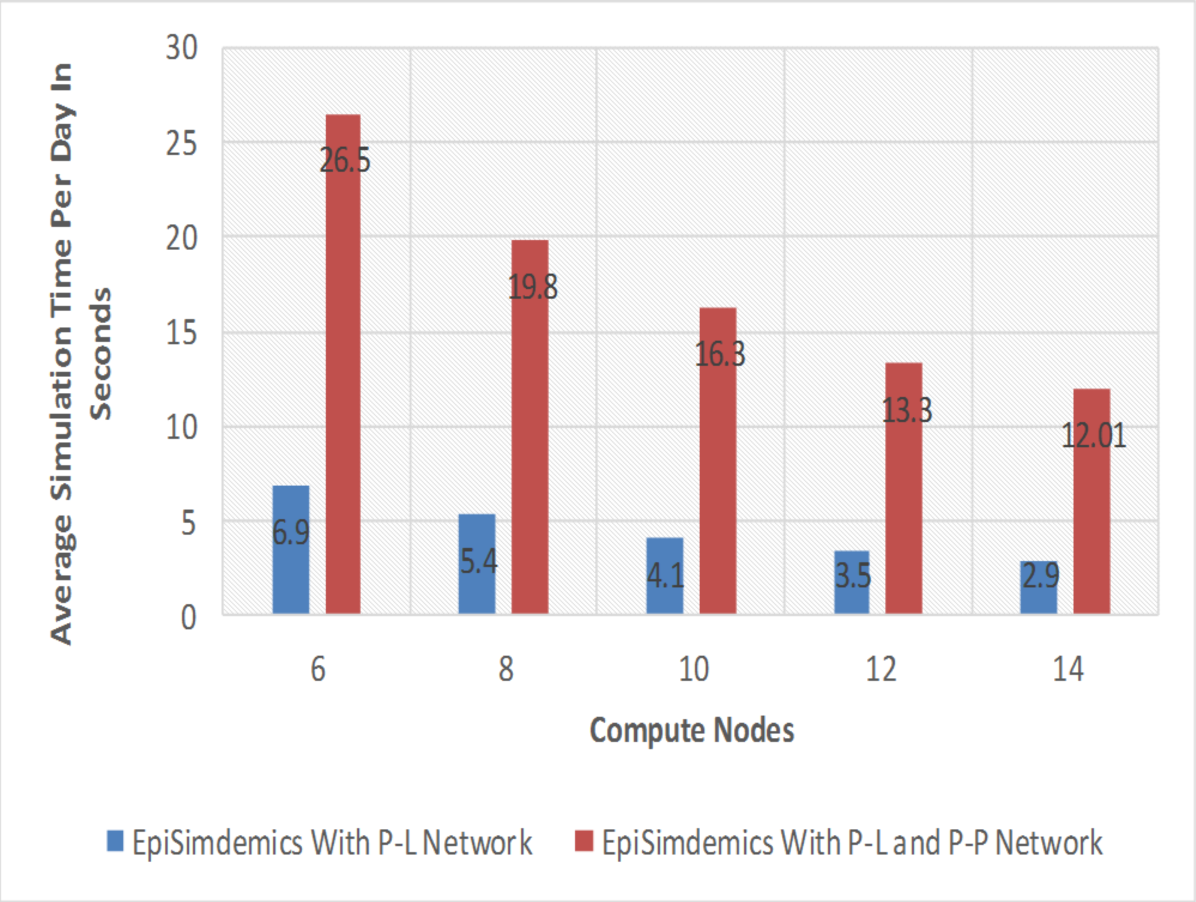


Figure 7.1: Comparison of Simulation Time of EpiSimdemics on Boston Population

the number of edges would increase the execution time of the algorithm. This is because edges represent interactions in a Person-Person graph and the addition of interactions are proportion addition to computation and communication costs. Figure 7.2 presents execution time details in following four setups:

1. **Person-Person Network Derived From Calling Network Data**: In this setup, the Person-Person Network file is derived from calling network data of Boston population.
2. **Person-Person Network Derived From EpiSimdemics Visits Data**: EpiSimdemics produces the Person-Person network from the visits data during simulation.
3. **Person-Person Network Derived From Randomized Calling Network Data**: This network is derived by randomizing the graph edges in a way to preserve the degree of each node, thereby maintaining the number of graph edges. Calling network data was randomized to generate this Person-Person Network.
4. **Person-Person Network Derived From Randomized EpiSimdemics Visits Data**: This network is derived by randomizing the Person-Person network derived from visits data,using each node degree and total edge preserving mechanism.

The essence of this experiment was to analyze the affect of graph density and structure on the simulation running time. Figure 7.2 depicts the average simulation time for 150 days, maximum average simulation time and minimum average simulation time for the above mentioned four Person-Person network graphs.

As depicted in figure 7.2, there is a subtle difference in simulation time between the Person-Person network derived from a calling network and the Person-Person network derived from EpiSimdemics visits data. The higher edge density of the Person-Person network derived from EpiSimdemics visits data is attributed to a higher simulation time.

There is not much difference between a given graph and its randomized version because, as stated above, randomization only changes the distribution by maintaining the degree sequence of each node. This ensures the same number of edges in the original graph and its randomized version.

Table 7.3 presents the number of edges in varied graph configurations. As shown in the table, the Person-Person network derived from EpiSimdemics visits data contains a substantially large number of edges as compared to the Person-Person network derived from calling network. This impacts the simulation running time. The randomized version of graphs contain the same number of edges, therefore there was not a noticeable change in simulation time.

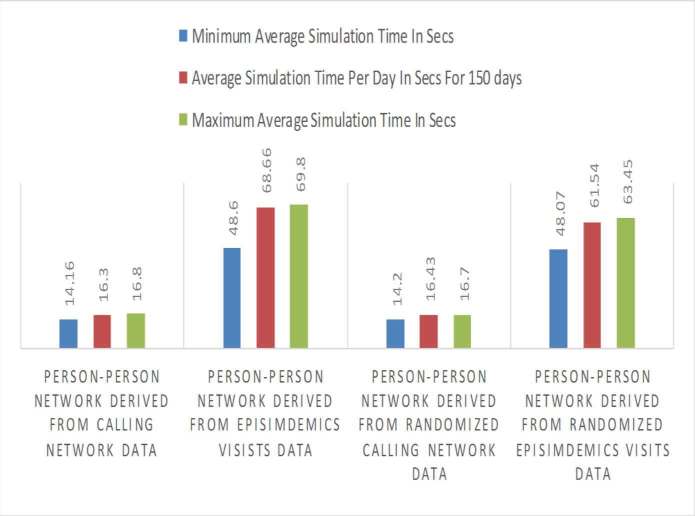


Figure 7.2: Performance Evaluation Against Varied Graph Density and Structure

Table 7.3: Number Edges in Person-Person Network

Person-Person Networks	# of Edges in P-P Networks
Derived From Calling Network	1151186
Derived From EpiSimdemics Visits Data	117191503

Chapter 8

Future Works

The following subsections identify possible future enhancements as an extension of this work.

8.1 Complex Contagions

The current implementation of the Person-Person network in EpiSimdemics, model interactions that do not necessarily require the physical presence of participating agents. We have demonstrated its usefulness by modeling interactions between the Fear Model represented by Fear PTTS and the Disease Model represented by Disease PTTS. This implementation considers individual interactions between agents to decide the outcome of interaction. A single interaction can change the states of interacting agents in Fear PTTS.

However, some of the social contagions such as fear fall into a category of complex contagions. Complex contagions refer to a class of contagions in a social network that requires multiple exposure depending on the threshold of the subject, before the subject is influenced by the contagion. Fear, rumors and innovation are examples of complex contagions. People normally have a greater trust level with friends and family than strangers. So, family and friends generally have a greater influencing power and thus can transmit fear more easily. It also depends on the kind of complex contagion and the characteristics of the infector.

To better understand the dynamics of complex contagions, a scheme should be devised where each person maintains the list of all its neighbors, their fear states and also its own fear threshold. The neighbors of a person are all individual with which it interacts directly. This enhancement will modify the way state changes occur in EpiSimdemics and would also add some memory overhead for maintaining the state of each neighbor for an entire population

The probability of a state change in an individual would depend on the state of its neighbor and on the fear threshold of that agent. We believe this addition to the existing Person-Person network infection computation would make it very suitable to model varied contagion types.

8.2 Vaccination and Rumor Injection

In this work, we have described interventions that raise people is fear levels or forces people to adopt preventive measures, such as staying at home. However, in real scenarios the public immediately start responding during an epidemic by adopting preventive measures like vaccination and closing of school and offices. So, a vaccination intervention would also decrease the fear levels in a population because they would then be immune to the disease. Such interventions impact disease and fear spread. Therefore, developing an intervention that also lowers the fear levels of a population based on preventive measures would give a more realistic view of an epidemic spread.

Similarly, rumor injection would increase fear levels in a population which would have major a impact on epidemics. Modeling this intervention in EpiSimdemics would make it well suited in the study of administering interventions.

8.3 Performance Enhancement

The simulation time of EpiSimdemics with Person-Person network can be improved by adopting efficient message buffering approaches and by reducing inter-chare communication [3].

Chapter 9

Conclusion

EpiSimdemics is a highly scalable parallel environment for modeling disease spread. It uses interaction based approach to model disease dynamics whereas many traditional models are based on differential equations and do not accurately represent the population interaction mix. Computational epidemiology that does not account for adaptations in individual behavior during epidemic may not represent the true dynamics of epidemic spread. People may not adapt drastically during non-lethal disease spread but, fatal diseases such as H1N1 and HIV tend to influence people's behavior in society [2]. People are concerned and fearful when a fatal epidemic outbreak occurs. Therefore, disease spread has a major psychological impact on society.

Health care agencies are always looking for a better mechanism to deal with epidemics and its effects which are social contagions related to an epidemic. These epidemic phenomena are detrimental to unified functioning society. Rumors destabilize the society and affect the governing, welfare and economy of the nation.

The addition of the Person-Person network as a part of this work has given EpiSimdemics the capability of modeling disease spread and its effects. This enhancement to EpiSimdemics extends its capability as a comprehensive simulation engine that models interaction of PTTS's, representing the Disease Model and Fear Model. Experimental results shows the fear effects in a population which alters the disease spread. The interaction of the Disease and Fear Model happens on an underlying interacting Person-Person and Person-Location networks. This interaction has been modeled using two interventions: *StayHome intervention* and *Fear Elevation intervention*. These interventions show the influence of the Disease and Fear Model on each other. Thus, modeled interacting PTTS gives accurate dynamics of disease and fear spread in a population. Such dynamics have a major impact on the way interventions should be administered in society.

Bibliography

[1] C.Barrett, K.Bisset, J.Chen, S.Eubank, B.Lewis, V.S.A.Kumar, M.Marathe, and H.Mortveit, "Effect of Public Policies and Individual Behavior on the Co-evolution of Social Networks and Infectious Disease Dynamics," *DIMACS*, 2007.

[2] SY.Del-Valle, H.Hethcote, JM.Hyman, and C.Castillo-Chavez, "Effects of Behavioral Changes in a Smallpox Attack Model," *Journal of Mathematical Biosciences*, Vol. 195, 2005, pp. 228–251.

[3] C.Barrett, K.Bisset, S.Eubank, X.Feng, and M.Marathe, "Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks," *SuperComputing*, 2008.

[4] J.Epstein, J.Parker, D.Cummings, and R.Hammond, "Coupled Contagion Dynamics of Fear and Disease: Mathematical and Computational Explorations," *SFI*, 2007.

[5] DS.Burke, JM.Epstein, DA.Cummings, JI.Parker, KC.Cline, RM.Singa, and S.Chakravarty, "Individual-based computational modeling of smallpox epidemic control strategies," *PubMed*, 2006.

[6] C.Barrett, K.Bisset, J.Leidig, A.Marathe, and M.Marathe, "Estimating the impact of public and private strategies for controlling an epidemic: A multi-agent approach," *IAAI*, 2009.

[7] C.Barrett, K.Bisset, J.Leidig, A.Marathe, and M.Marathe, "Economic and social impact of inuenza mitigation strategies by demographic class," *Elsevier*, 2010, pp. 19–31.

[8] C.Barrett, J.Smith, and S.Eubank, "If Smallpox Strikes Portland," *Scientific American*, Vol. 292, 2005, pp. 54–61.

[9] M.Marathe and V.S.A.Kumar, "Computational Epidemiology," *Communications of ACM*, 2013, pp. 88–96.

[10] S.Eubank, V.S.A.Kumar, M.Marathe, A.Srinivasan, and N.Wang, "Structure of social networks and their impacton epidemics," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 70, 2001, pp. 179–185.

[11] S.Funk, M.Salath, and V.Jansen, "Modelling the influence of human behaviour on the spread of infectious diseases: a review," *J. R. Soc. Interface 7*, 2010, pp. 1247–1256.

[12] J.Chen, A.Marathe, and M.Marathe, "Coevolution of Epidemics, Social Networks and Individual Behavior: A Case Study," March 2010.

[13] A.C.Auld, "Estimating Behavioral Response to the AIDS Epidemic," *Contributions to Economics Analysis and Policy*, Vol. 5, No. 1, 2006, pp. 1235–1235.

[14] C.Barrett, M.Marathe, and S.Eubank, "Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 70, 2001, pp. 179–185.

[15] S.Eubank, H.Guclu, V.S.A.Kumar, M.Marathe, A.Srinivasan, Z.Toroczkai, and N.Wang, "Modelling disease outbreaks in realistic urban social networks," *Nature*, Vol. 429, 2004, pp. 180–184.

[16] C.Barrett, S.Eubank, V.S.A.Kumar, and M.Marathe, "Understanding Large-Scale Social and Infrastructure Networks: A Simulation-Based Approach," *SIAM*, Vol. 37, No. 4, 2004.

[17] W.Kermack and A.McKendrick., "A contribution to the mathematical theory of epidemics," *Royal Society of London.Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 115, No. 772, 1927, pp. 700–721.

[18] K.Bisset, J.Chen, X.Feng, V.S.A.Kumar, and M.Marathe, "EpiFast: A Fast Algorithm for Large Scale Realistic Epidemic Simulations on Distributed Memory Systems," *ICS*, 2009.

[19] C.Barrett, S.Eubank, M.Marathe, H.Mortveit, and C.Reidys, "Science and Engineering of Large Scale Socio-Technical Simulations," *Western Simulation Conference*, 2002.

[20] S.Eubank, "Scalable, efficient epidemiological simulation," *ACM symposium on Applied computing*, 2002, pp. 139145.

[21] I.Longini and A.Nizam, "Containing pandemic influenza at the source," *Science*, Vol. 309, No. 5737, 2005, pp. 1083–1087.

[22] N.Ferguson and M.Keeling, "Planning for smallpox outbreaks," *Nature*, Vol. 425, No. 6959, 2003, pp. 681–685.

[23] J.Parker, "A flexible, large-scale, distributed agent based epidemic model," *Winter Simulation Conference*, 2007.

[24] M.Newman, "The structure and function of complex networks," *SIAM*, Vol. 45, No. 2, 2003, pp. 167–256.

[25] J.Jiang, C.Wilson, X.Wang, P.Huang, W.Sha, Y.Dai, and B.Y.Zhao, "Understanding Latent Interactions in Online Social Networks," *IMC*, 2010.

[26] L.Huang and Z.Xia, "Measuring User Prestige and Interaction Preference on Social Network Site," *ACIS-ICIS*, 2009.

[27] C.Barrett, H.Hunt-III, M.Marathe, S.Ravi, D.Rosenkrantz, R.Stearns, and P.Tosic, "Gardens of Eden and fixed points in sequential dynamical systems," *International Conference on Discrete ModelsCombinatorics, Computation and Geometry*, 2001.

[28] H.S.Mortveit, "Graph Dynamical Systems," *Discrete Models in Systems Biology Workshop*, 2008.

[29] C.Barrett, H.Hunt-III, D.Rosenkrantz, M.Marathe, R.Stearns, and S.Ravi, "Complexity of reachability problems for nite discrete dynamical systems," *Journal of Computer and System Services,Elsevier*, Vol. 72, 2006, pp. 1317–1345.

[30] L.V.Kale and A.Bhatele, *Parallel Science and Engineering Applications: The Charm++ Approach*, CRC Press, 2013.

[31] G.Zheng, L.V.Kale, E.Bohm, A.Bhatele, C.Mei, F.Gioachin, and R.Venkataraman, “Parallel Languages/Paradigms: Charm ++ - Parallel Objects,” `http://charm.cs.uiuc.edu/research/charm/`, 2014, [Online; accessed 15-May-2014].