

2017

# CIMoRE: Developing a framework for a critical infrastructure modeling and response environment

Julie Ann Rursch  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Computer Engineering Commons](#)

## Recommended Citation

Rursch, Julie Ann, "CIMoRE: Developing a framework for a critical infrastructure modeling and response environment" (2017).  
*Graduate Theses and Dissertations*. 16206.  
<https://lib.dr.iastate.edu/etd/16206>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

**CIMoRE: Developing a framework for a critical infrastructure modeling and response environment**

by

**Julie Ann Rursch**

A dissertation submitted to the graduate faculty in partial fulfillment  
of the requirements for the degree of

**DOCTOR OF PHILOSOPHY**

Major: Computer Engineering (Secure and Reliable Computing)

Program of Study Committee:  
Douglas W. Jacobson, Major Professor  
Manimaran Govindarasu  
Thomas Earl Daniels  
James H. Oliver  
Loren W. Zachary

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

Copyright © Julie Ann Rursch, 2017. All rights reserved.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS .....	iv
ABSTRACT .....	v
CHAPTER 1. INTRODUCTION .....	1
CHAPTER 2. DEFINITIONS.....	4
Critical Infrastructures .....	4
Dependencies .....	7
Interdependencies .....	9
CHAPTER 3. REVIEW OF LITERATURE .....	13
Modeling a Single Critical Infrastructure .....	13
Modeling Multiple Critical Infrastructures .....	16
Coupled Model .....	16
Integrated Model.....	18
CHAPTER 4. SCOPE OF WORK.....	20
CHAPTER 5. TRANSFORMING CRITICAL INFRASTRUCTURES INTO NETWORK REPRESENTATIONS .....	23
Acquire the Subsector Data .....	25
Highway Data From IDOT .....	29
Network, Network Segments, and Devices .....	31
Bandwidth and Loss .....	32
ICN Data.....	33
Network, Network Segments, and Devices .....	37
Bandwidth and Loss .....	37
Electricity Transmission Lines .....	38
Network, Network Segments, and Devices .....	38
Bandwidth and Loss .....	39
CHAPTER 6. ISEAGE MODIFICATIONS and CONFIGURATION CHANGES .	40
How ISEAGE Works .....	40
Allowing for Different Types of Traffic in ISEAGE .....	43
Defining New Types of Routers in ISEFLOW.....	50
Normal Router .....	51
Edge Node Router.....	53
Connector Node Router.....	55
Junction Node Router .....	57
Writing a CIMoRE Configuration File for ISEFLOW Version 1.1 .....	58
Backplane Definition .....	60
Global Link Definitions .....	61
Board Definitions .....	62
Connections Section .....	63
Router Section.....	66

How to Handle Latency / Loss in the Network.....	74
Generating Traffic for CIMoRE.....	83
Introducing Disruptive Events .....	87
CHAPTER 7. RUNNING CIMORE .....	92
Initial Startup.....	92
Steady State .....	92
Introduction of Disruptive Events .....	93
Failure.....	94
Recovery.....	94
CHAPTER 8. CONCLUSIONS AND FUTURE WORK .....	96
REFERENCES.....	103
APPENDIX. OTHER CRITICAL INFRASTRUCTURES .....	106
“Lifeline” Sectors - Yes.....	106
“Manufacturing-like” Sectors - Maybe .....	108
Other – No .....	110

## ACKNOWLEDGMENTS

I would like to thank my committee chair, Doug Jacobson, and my committee members, Manimaran Govindarasu, Thomas Earl Daniels, James H. Oliver, and Loren W. Zachary, for their guidance and support. I would also like to thank James A. Davis for being a substitute for my final oral examination.

Additionally, I want to thank my husband, Brian Brace, for being supportive of my efforts to earn this degree and for putting up with our “married but living in separate states” status for so long.

## ABSTRACT

Activities for individuals, organizations, and government agencies to plan for, protect from, and respond to cases of emergency or attack generally focus on paper and pencil planning sessions that don't include computer simulated information or decision data. Modeling critical infrastructures and cyber physical systems has become a growing research area, as well as a common theme in training activities for cyber security practitioners and first responders over the past decade. One approach to modeling multiple critical infrastructures is to model all critical infrastructures in a single environment by converting them into a single standard protocol and implementing them in a single testbed.

This dissertation provides the road map of how the Critical Infrastructure Modeling and Response Environment (CIMoRE) could be developed to allow all critical infrastructure subsectors to be modeled in a single TCP/IP testbed. The Internet Scale Event and Attack Generation Environment (ISEAGE) is the testbed that was used as the backbone of this framework.

This dissertation addresses three main problems with using a unified TCP/IP testbed for modeling. First, the physical world critical infrastructure subsectors must be turned into network representations of themselves. This includes transforming the characteristics of their traffic into TCP/IP traffic and node data, as well as representing interdependencies between the critical infrastructure subsectors. Second,

the ISEAGE testbed, its operational software ISEFLOW, and the ISEFLOW configuration file needed to be modified to allow for critical infrastructure subsector modeling. Additionally, the concept of network delay had to be added to ISEAGE. And, third, concept of traffic generation had to be added to ISEAGE to allow modeling of increases and decreases of traffic volumes for critical infrastructure subsectors. Along with traffic generation is the need to introduce events that simulate real world disruptions that could stem from that traffic generation.

## CHAPTER 1. INTRODUCTION

In the past decade the modeling of critical infrastructures and cyber physical systems has become a growing research area, as well as a common theme in training activities for cyber security practitioners and first responders. On a national front the government is worried about the actions of nation states or hackactivists against critical infrastructures including services such as water treatment systems and power grid, physical structures such as bridges and buildings, and communication systems such as data networks, control systems, and information systems. On a state level Iowa has the Air National Guard charged with a homeland security cyber operations and protection mission and the State of Iowa using a testbed to model its computer and information systems for cyber security testing and protection. Even current events such as natural disasters like hurricanes Harvey, Irma, and Maria focus attention on planning activities to anticipate outages of critical services, predict damages, recover systems gracefully, and rebuild what is necessary.

Training for individuals, organizations, and government agencies to plan for, protect from, and respond to cases of emergency or attack can be a daunting task. Most training is completed as table top exercises with multiple state, federal, and local organizations sending participants. In these exercises it is hard to replicate the volume of data to be analyzed and with which decisions need to be made, sometimes in a matter of minutes, that a real event generates. In some exercises the paper and



pencil activities are coupled with computer simulation information output from multiple, disparate computer systems that each has its own critical infrastructure as its focus. The computer simulated information currently is a disjointed approach in modeling and, generally, are single sector specific. During the table top exercises, as well as during a real event, creating a response requires a large amount of human effort and decision-making which may or may not correctly estimate how the interdependent, and sometimes conflicting, failing systems impact each other. A poor decision can lead to cascading events further degrading the critical infrastructure or impacting a different one.

One approach to solving the problem of multiple computer modeling systems each focused on its own critical infrastructure is to model all critical infrastructures in a single environment by converting them into a single standard protocol and implementing them in a single testbed. Fortunately, at Iowa State University, there is access to a flexible and highly configurable Internet testbed that was developed for cyber security research. The Internet Scale Event and Attack Generation Environment (ISEAGE – pronounced ice age) provides a scalable and resilient testbed in which to create the Critical Infrastructure Modeling and Response Environment (CIMoRE – pronounced “see more”) tool. This dissertation provides the framework needed to modify the ISEAGE environment for the modeling of multiple critical infrastructures at the

same time using a single Transmission Control Protocol / Internet Protocol (TCP/IP) testbed.

This dissertation is organized into 5 sections. Chapter 1 includes the brief overview presented above. Chapter 2 provides the definitions used when discussing critical infrastructures. Chapter 3 outlines previous literature on critical infrastructure modeling. Chapter 4 provides the scope of work. Chapter 5 shows how a physical critical infrastructure is turned into a network representation. Chapter 6 enumerates the changes in ISEAGE that must occur before CIMoRE can be fully functional. Chapter 7 shows how CIMoRE functions. Chapter 8 contains the conclusions and future work.

## CHAPTER 2. DEFINITIONS

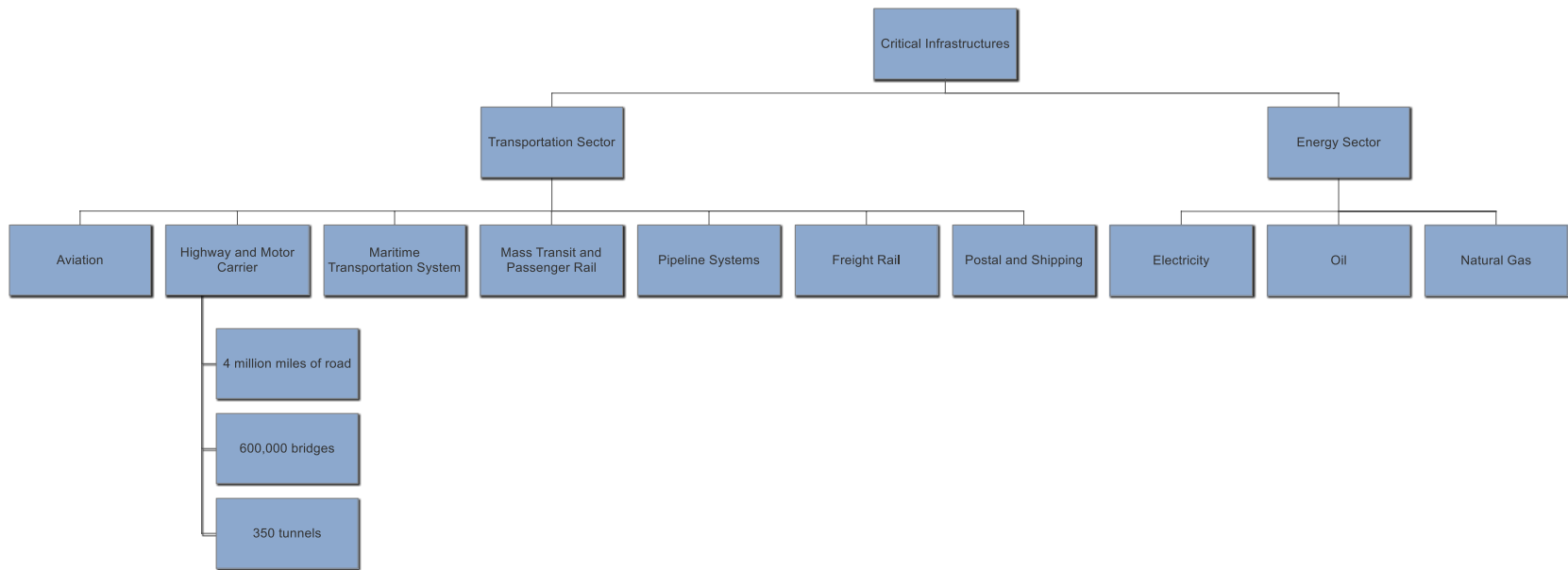
Before reviewing the literature, converting critical infrastructure systems to network nodes, enumerating the modifications needed in ISEAGE, and depicting how CIMoRE operates, it is important to provide definitions of the terms and concepts that will be used repeatedly in this dissertation.

### Critical Infrastructures

This dissertation uses the federal government's definition of critical infrastructures. Found in the U.S. Patriot Act critical infrastructures are defined as "systems and assets, whether physical or virtual, so vital to the United States that the incapacity or destruction of such systems and assets would have a debilitating impact on security, national economic security, national public health or safety, or any combination of those matters" [1]. The National Infrastructure Protection Plan (NIPP), updated in 2013, categorized these systems and assets into 16 critical infrastructure sectors that provide services vital to make everyday life occur for the nation and for its people. These are complex and highly interdependent systems, networks, and physical assets that are essential to physical well-being, as well as to a way of life. The 16 categories are chemical; commercial facilities; communications; critical manufacturing; dams; defense industrial base; emergency services; energy; financial services; food and agriculture; government facilities; healthcare and public

health; information technology; nuclear reactors, materials, and waste; transportation systems; and water and wastewater systems [2].

Each of the 16 sectors is very complex with many subsectors. The sectors and subsectors can have varying complexities and relationships between and among them. And, the subsectors may be owned in a more regional manner by multiple companies, wholly by a single company, or owned and managed by a government entity. Fig. 1. shows transportation and energy sectors. In the transportation sector alone there are seven subsectors (or modes of movement) including aviation; highway and motor carrier; maritime transportation; mass transit and passenger rail; pipeline systems; freight rail; and postal and shipping. And, within the highway and motor carrier infrastructure subsector there are more than 4 million miles of roads, 600,000 bridges and 350 tunnels. Using those road are vehicles including automobiles, motorcycles, trucks carrying hazardous materials, other commercial freight vehicles, motorcoaches, and school buses [3]. Likewise, the energy sector has three subsectors: electricity, oil, and natural gas [4].



**Fig. 1. Example of sectors and subsectors using transportation and energy**

## Dependencies

Dependencies in a sector or subsector are generally well-defined and known by the owners of the infrastructure. A dependency is a one directional relationship between the two subsectors. It can be thought of as a link between the two subsectors where the functioning (or non-functioning) of the subsector affects the ability of the second subsector to provide its normal services.

Fig. 2 shows the singular relationship found in a dependency. For example, approximately 48% of electricity generated in the nation is created by burning coal. The majority of the coal arrives at the generation plants by freight trains. Therefore, there is a dependency in the electricity subsector on freight trains to deliver coal on time to keep the generators working [5]. These are one dimensional relationships which do not truly manifest themselves this simply in the real world. In reality the relationships among and between the sectors and subsectors are much more complicated.

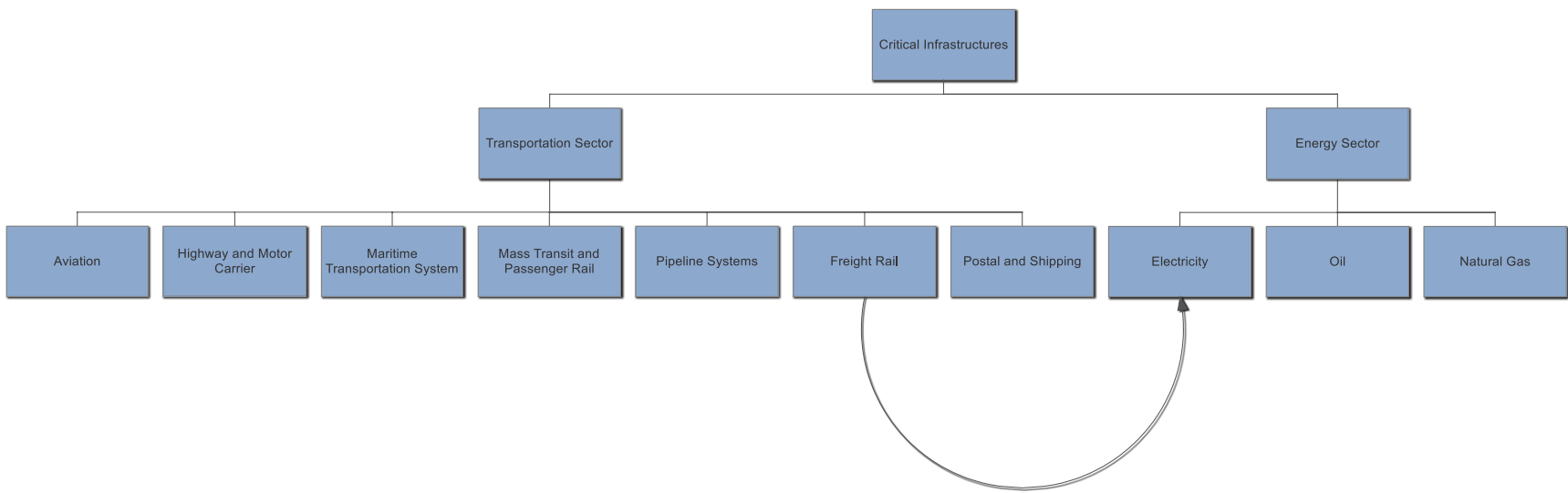


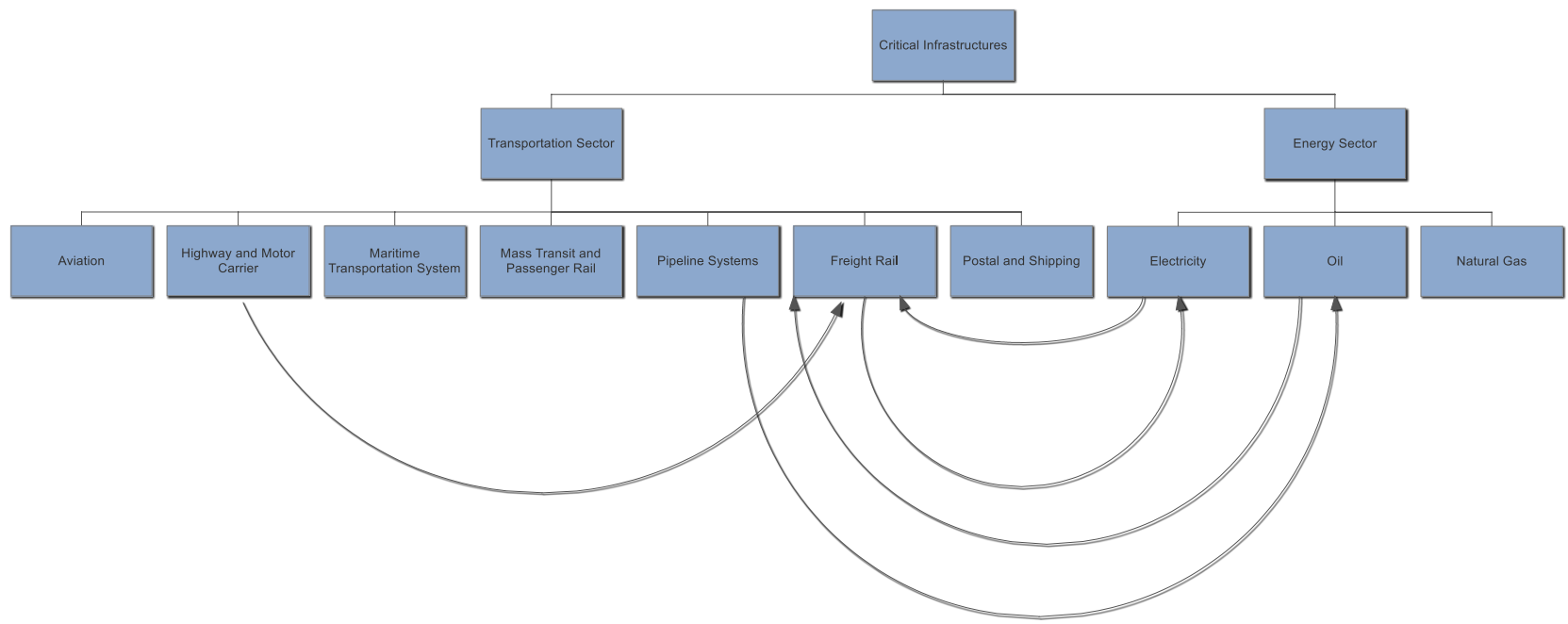
Fig. 2. Dependency of electricity on delivery of coal by freight trains

## Interdependencies

Identifying and organizing the interdependencies between and among critical infrastructures is a much larger problem and harder to quantify than the simplistic dependency described above. “An interdependency is a bidirectional relationship between infrastructures through which the state of each infrastructure is influenced by or correlated to the state of the other” [6].

Fig. 3 illustrates the complexity of interdependencies. Above it was argued that electricity generation, so the electricity subsector, had a dependency on the transportation subsector of freight trains. However, freight trains rely upon diesel fuel to power their engines, thus the freight rail subsector relies on the energy subsector of oil. The transportation subsector of pipelines is the primary distribution system for oil across the nation to refineries where the diesel fuel is manufactured. The diesel fuel is then distributed by tractor and trailer to the rail yard for the trains to refuel. And it takes electricity to pump the diesel fuel into the train engines.





**Fig. 3. Interdependencies of transportation and energy sectors**

Two very similar taxonomies have been developed to organize and classify interdependencies by the types of relationships or influences they have on each other. [7] provided four general classifications to categorize interdependencies:

- Physical – dependence on another infrastructure for material inputs/outputs
- Cyber – dependence on information transmitted through a network
- Geographic – close spatial proximity
- Logical – does not fall into one of the other classifications

A second, a very similar classification, was developed by [8]. This second classification provides three of the four previously presented categories, albeit with different names, and expands the taxonomy by dividing the “other” bucket into two groups; policy and societal:

- Physical – a connection from one infrastructure to another in terms of supply or production
- Informational – a control mechanism between two infrastructures or their components
- Geospatial – relationship based upon physical closeness of the entities
- Policy/Procedural – a policy change in one infrastructure affects the second

- Societal – the consequence of an infrastructure having an impact on public opinion, public confidence, fear, or cultural issues

The example depicted in Fig. 3 illustrates the physical interdependency category with relationships between electricity, oil, freight rail, pipeline systems, and highway and motor carrier subsectors being shown.

### CHAPTER 3. REVIEW OF LITERATURE

Current critical infrastructure simulation software products available through government resources and private parties, as well as academic research, sift themselves into a dichotomy. Those that treat critical infrastructures as separate entities using different frameworks and metrics and those that attempt to model multiple critical infrastructures at the same time. While both approaches have proven useful in the past, the CIMoRE project attempts to improve upon the second category by building on a single framework, including their interdependencies in the modeling, and allowing the interjection of disruptive events during the modeling.

#### Modeling a Single Critical Infrastructure

The first group of modeling software makes no attempt to model multiple critical infrastructure sectors. It treats each infrastructure sector (transportation systems, energy, information technology, etc.) separately using differing frameworks and metrics. Treatment of critical infrastructures as standalone entities is not representative of the true world and provides little useful information in planning, protecting, and responding to events. They do serve a purpose, however, to discover the small details needed to be known to model an infrastructure before any more complex modeling can be taken.

This is the historical approach to studying critical infrastructures. Modeling one infrastructure, or one subsector, in its entirety. Each work

focuses singularly on one critical infrastructure subsector such as roads, bridges, telecommunications systems, cyber networks, power grids, rail systems, or water treatment facilities. Each critical infrastructure sector, and many times subsector, is treated as a system unto itself. Each uses its own, different, methodology to model, test and resolve the threats presented to it.

Examples of singularly focused works include systems that visualize road closures and alternate routes for traffic, but only look at surface transportation on primary roads [9]; transportation planning software programs that include multiple types of traffic (road, rail and public transportation), but only allow one type to be viewed at a time; or congestion relief models and rider/driver demand modeling softwares. This type of traffic simulation is used primarily for urban planning and transportation infrastructure planning [10]. An additional example of singularly focused software includes flood prediction programs that estimate the breaches of the levees and when levels will contaminate water supplies, but don't show damage to other methods of transportation or damage to buildings, lives and personal property [11]. Further, early work done on the power grid has been singularly focused [12, 13]. Each of these types of programs use a different format to model the critical infrastructure pieces and is compartmentalized with no data used beyond that sector or subsector. The different modeling techniques inhibit the

overlay of information and make interdependencies and relationships between and among the subsectors difficult to predict or measure.

While studying one infrastructure or subsector at a time simplifies the problem, it does not reflect the real world and does not provide a holistic approach for disaster preparedness. In addition to being too singularly focused on one sector, these works do not account for interdependencies between critical infrastructure sectors which represent the complexity of the physical world. While critical infrastructure subsectors can be owned and maintained by the government such as interstates, highways, and bridges, a majority of them are privately held companies such as power plants, oil refineries, telecommunications companies. Historically, each of these infrastructure subsector owners is concerned primarily with the continuity of their business and focus on their own well-defined area of control, thus, the past approach of studying a single infrastructure or subsector. However, the growing reliance upon network connectivity, as well as unforeseen secondary effects from the failure of another critical infrastructure, has these owners interested in studying interdependencies and modeling multiple critical infrastructures as well [8]. Therefore, more recent work attempts to study multiple critical infrastructures and account for their interdependencies. However, this field of study is still very much in its infancy.

## **Modeling Multiple Critical Infrastructures**

The second approach tries to model multiple infrastructures, however, within this group there are two methodological approaches to the problem of combining very different critical infrastructure sectors or subsectors. In reviewing models that are currently under development in the U.S., two different approaches were taken to allow multiple infrastructures to be included [8]. One coupled a series of simulations together; taking output from one model and providing it as input to another. The other approach is to try to find common underpinnings and base the modeling on those commonalities. The first approach, called the coupled model, tends to handle more specific details of the critical infrastructure subsectors it is modeling because each is modeled separately and then the output passed on. The second model, called an integrated model, generally works at a higher level of abstraction and doesn't handle the details of the subsector as well.

### **Coupled Model**

The first type of modeling is to couple simulations that model a single infrastructure sector. In this case, the output of one model may be fed into another model as input. In this way, multiple infrastructure sectors are included. This could be useful, but there is a lag in the creation of the input, as well as needed to make the different systems transparently pass information from one to the other. Any transformations of the output of

one system before being used as input to the next modeling system requires processing time and potentially human interaction.

There are no standards that exist for cross infrastructure modeling. In a coupled model, however, there are two widely accepted methods for exchanging information between simulations: the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS). HLA was developed to allow interoperability between a large number of simulations used by the Department of Defense. It allows multiple types of models and simulations to exchange data in real-time. It is an open source, general purpose simulation architecture specified as IEEE Standard 1516 [14]. HLA uses federations of different kinds of interactive members. These interactive members, called federates, communicate using services provided by the infrastructure. The publisher role allows a member to send information into the infrastructure and the subscriber role provides a way to receive the information. All is implemented in XML and all communication is real-time.

HLA has been more widely used to implement distributed critical infrastructures such as air traffic control and vessel traffic systems [15]; electric grid and its SCADA systems [16]; and telecommunications, railway and electric systems [17]. A similar publish-subscribe communication paradigm is called Quality of Service Descriptors that has been implemented in a grid architecture for the electrical grid to aid in information sharing among critical infrastructures [18].



The DIS framework is also real-time and allows distributed simulations. It is IEEE Standard 1278 [19] and was created to allow virtual simulations of warfare by interconnecting distributed computers/simulators. Both HLA and DIS are designed to pass data between models with synchronized timing. Other methods of exchanging data between simulations are XML and GIS shapes files [20].

### **Integrated Model**

The second type of modeling is to identify an underlying commonality on which several critical infrastructures can be modeled. While this approach provides an integrated model that has shown better results than those modeled as single sectors [8], few of these programs allow the import of measurements from ongoing, real-time events. Further, knowledge of interdependencies between the infrastructure sectors is relatively immature and there is no standard for multiple critical infrastructure sector evaluation [6]. It is within this concept of creating a single, comprehensive framework to integrate different critical infrastructures, their interdependencies, and real-time physical data is where the fully implemented CIMoRE tool provides an advantage.

When considering integrated models, there are not even accepted methods to follow. The most interesting previous integrated work found that used both multiple infrastructures and interdependencies was a model created by [21] that used an agent-based approach to model the levels of critical infrastructures and their interdependencies as a large graph. The

nodes in the graph are used to represent infrastructure components and the edges are the connections between the nodes. The nodes could be connected or disjoint. The edges may be directional, bi-directional, or both.

Because CIMoRE will model multiple critical subsectors at the same time in a single TCP/IP network, its development falls into the group of work focused on finding commonalities in subsectors. However, it differs from that work in definition of commonalities. Previous work focused on finding commonalities in the physical world in which the critical infrastructure subsectors exist. CIMoRE requires that the subsectors be transformed into networks. Because [21-25] opted to model critical infrastructures as nodes and their interdependencies as a set of connected graphs with traffic intensity and payload redistributions [22] , it logically follows that critical infrastructure subsectors can be expressed as an interconnected IP network topology. An interconnected IP network is just another way to draw multiple critical infrastructures and their interdependencies. However, CIMoRE goes beyond the previous work by using a functioning Internet testbed and an IP network topology, disruptive events can be easily introduced and nodes quickly reconfigured to accommodate for failing states due to multiple events. [21] readily admit in their graph model is a discrete event simulator because it cannot not accommodate multiple events or input data. CIMoRE built upon the ISEAGE testbed will be able to allow both when fully functional.

## CHAPTER 4. SCOPE OF WORK

To allow multiple critical infrastructures to be modeled in a single interconnected IP network testbed, all physical critical infrastructure subsectors and their corresponding real-time physical data must be transformed into IP network nodes, IP traffic and node data, and IP node relationships in this framework. Once the transformations are completed, the ISEAGE testbed needs to be modified to provide the ability to model and represent the subsector data and network representations. Finally, the ability to insert disruptive events into the model must be provided.

This dissertation serves as the foundation for a full implementation of CIMoRE. It provides the framework for converting subsector data into network data, as well as the changes needed to the ISEAGE environment to allow modeling of multiple subsectors at the same time using a single testbed. The three primary problems it addresses are

1. Turning the physical world critical infrastructure subsectors into network representations of themselves. This includes transforming the characteristics of their traffic into TCP/IP traffic and node data. It also includes the representations of relationships or interdependencies between critical infrastructures in networking terms. Further, it means determining what normal traffic looks like in those critical infrastructure subsectors and what an abnormally high level of traffic would look like.

2. Modifying ISEAGE, its operational software ISEFLOW, and the ISEFLOW configuration file to handle critical infrastructure modeling. Differentiating between types of traffic and introducing the concept of latency are all concerns in using the ISEAGE testbed for modeling critical infrastructure subsectors.

3. Providing for a way to increase and decrease traffic on the routers in the critical infrastructure subsectors, as well as introducing events that simulate real world disruptions. These disruptions could be increased use of subsector resources, node failure due to resource stress in the subsector, initial disruption in the subsector such as an attack. Additionally, the concept of recovery from a failed state is addressed.

The scope of CIMoRE's development in this dissertation is to create the framework for three of the 16 critical infrastructure sectors:

transportation, information technology, and energy. And, to more narrowly define the scope of work, three subsectors, one within each of the critical infrastructure sectors, were selected: the highway system, the communications network, and the power distribution system. To further reduce the amount of scope of work, the State of Iowa was selected as the geographic area to cover. Specially, the two major Interstates that cut across the State of Iowa, I-80 and I-35, are used for the highway system work. The Iowa Communications Network (ICN) is used to represent the communications network subsector, while the electric grid in the State of

lowa is used for the electricity subsector. To make the configurations readable for this dissertation, one physical “network” of each type will be depicted in the examples. Finally, only the geospatial interdependencies, where nodes are in close physical proximity, are modeled.

The work in this dissertation is the first vertical step in the development of the fully functioning CIMoRE tool. The development of the IP network translations for the three subsectors (highways, communications network, and electricity) selected for this project, the changes the ISEAGE testbed, and the ability to introduce disruptive events lay the foundation for future work to include all 16 of critical infrastructure sectors and all five types of interdependencies.

## CHAPTER 5. TRANSFORMING CRITICAL INFRASTRUCTURES INTO NETWORK REPRESENTATIONS

Placing the critical infrastructure subsector's attributes into a networking framework meant the most basic aspects of a network need to be enumerated. The most basic elements of a network include

- the connection from one device to another,
- the action the device can take such as routing or switching traffic,
- the bandwidth or amount of traffic the connection can accommodate,
- the number of packets dropped or the loss on the network,
- and the protocols allowed on the network.

While those elements seem rudimentary, there are a multitude of questions associated with trying to think about or talk about a physical critical infrastructure subsector in these networking terms. Questions arise such as what is a connection? What piece of the subsector is considered a device? Is the subsector node (device) a decision point? Can the path change at that node or does it simply exist as a point where traffic passes through it? In other words, is it a router or a switch? Additional questions would include determining how to convert the traffic or capacity of the critical infrastructure to a bandwidth and what is an acceptable loss on that capacity?

Table 1 shows the data that needed to be collected about each critical infrastructure subsector selected for this dissertation. Each critical infrastructure subsector is treated as its own network. The protocol that is being carried on the network path is identified based upon the subsector that is being transformed. The points along the network segment where critical decisions are made will be treated as routers. Otherwise, if there are points that need to be listed but no decisions are made, they will be treated as switches. The total capacity that a specific critical subsector path can accommodate, or produce, is considered the bandwidth. This has an average bandwidth, as well as a maximum value. The maximum value represents when the path is operating at full capacity. Additionally, the reduction in capacity or production is transformed into loss on the network. To try to find how each of these pieces of data could be built from cooperator files and eventually design the database from which the CIMoRE configuration file could be built, it was necessary to see what types of data existed and think about how those conversions could be made. Therefore, it was necessary to acquire the subsector data in its native form or from a willing cooperator to determine what was known and recorded on each critical infrastructure subsector.

**Table 1. Data needed to build a network for each subsector**

Sector	Transportation	Information Technology	Energy
Subsector	Highway system	Communications Network	Electricity
Network			
Network Segment			
Latitude/Longitude			
Protocol			
Device Type (Router/Switch)			
Average Bandwidth			
Maximum Bandwidth			
Loss			

### Acquire the Subsector Data

In meetings held with the state and national homeland security staff, as well as the Iowa Department of Transportation staff and the Iowa Communications Network staff, all parties expressed interest in and verbal support of the CIMoRE project. However, after repeatedly requesting data for use in the project and repeatedly being told the data would come, the data never would arrive. It became apparent that real data, especially data about critical infrastructures in a time when the nation, businesses, and public agencies have heightened security concern, was going to be difficult to acquire. This was compounded by the fact that critical infrastructure subsector data is held by many private companies and some, if not all, data is proprietary. And, once the data is received from a private company, it takes some level of expertise in the area to find the



connecting links and the interdependencies. While government agencies and private industry would willingly discuss the CIMoRE project, and appeared on the surface eager to help, it became apparent that by asking for access to the data and the long lag times that they were hesitant to give the full database that contained the pieces of information or put any people hours toward pulling the data.

Therefore, as a proof of concept, I had to find the native database and extract the useful fields in the case of the highway subsector. In the case of the Iowa Communications Network (ICN) and the power grid, paper maps were either acquired or purchased and representative data was created by hand. This was an unanticipated step of developing CIMoRE and took significant time to understand the data points, determine the physical layouts, and develop the actual physical drawing before any network transformations could be done. Gaining access may continue in the future to be a stumbling block for the development of a fully functional CIMoRE tool.

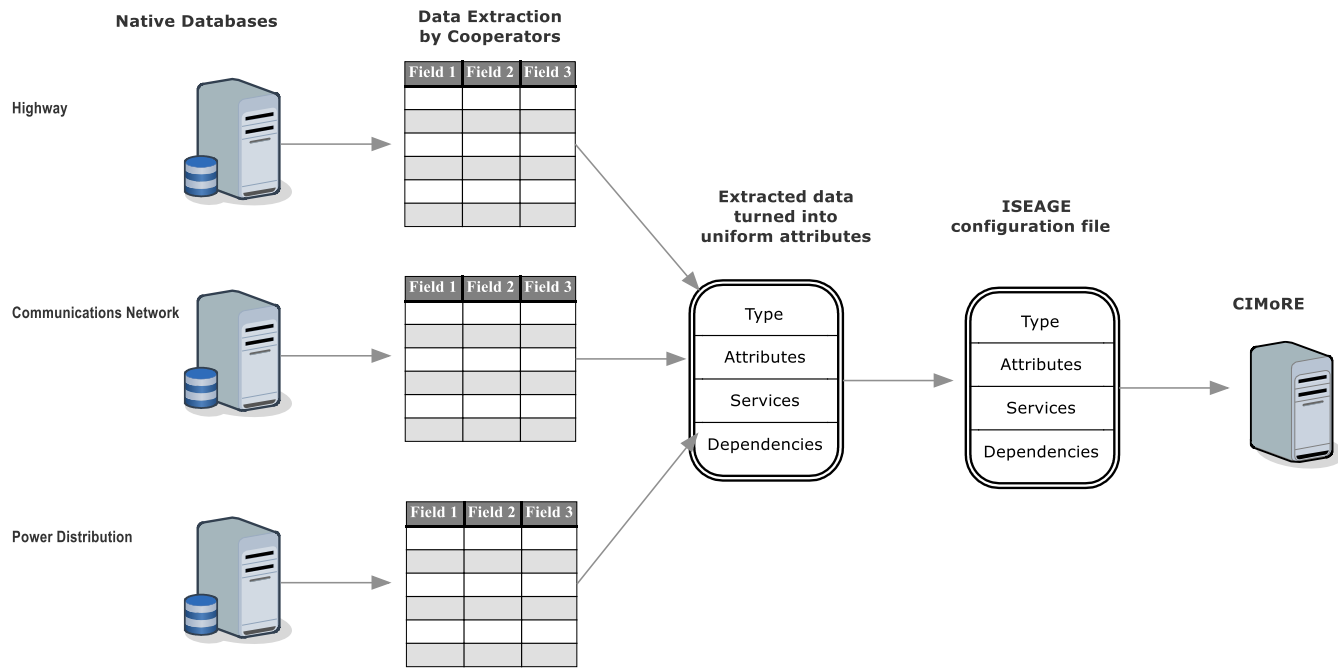


Fig. 4. CIMoRE network creation process

Fig. 4 shows the process that should have occurred in the creation of the ISEAGE configuration files for CIMoRE to operate. A cooperater from a government agency or a private company selects the necessary data points from the native database. The fields to be selected and included in the data export will be a joint effort between the CIMoRE developer and the subsector specialists. Their data would be in a variety of their own databases and stored in a variety of formats. These data would be output in a flat file format. A program would then read the data in the flat file and create the standard intermediate file that contains the uniform attributes. In addition to standardizing the attributes, the program that creates the intermediate file will add fields to record the subsector type and source of the data.

A second program would read the intermediate file and create the ISEAGE configuration file which the CIMoRE tool will use. The second program could also perform the calculation on the physical proximity of the different subsector nodes to each other to programmatically account for geospatial interdependences in the configuration files. Developing the algorithm to determine the level of the physical proximity of the nodes needed for a geospatial interdependency to be created is a unique research problem of its own. It will not be formally addressed in this dissertation other than to say that the latitude and longitude of the nodes will be used to place the nodes onto a map of the state. A grid will then be drawn over the state and any devices falling within the same cell of the

grid are considered to have geospatial interdependency. In this dissertation, the geospatial interdependences are created manually rather than by using a computer generated analysis.

While the process above is what I originally strove for, without a willing cooperator, the data used in this dissertation was gathered as described below.

### **Highway Data From IDOT**

The Iowa Department of Transportation (IDOT) has a database system called Geographic Information Management System (GIMS) which reports statewide and county-by-county data on all roads in the state. These database files are shared on a web site and publicly accessible [26]. I downloaded and extracted the zipped files for 2010 which was the most currently available data at the time of download, as well as the metadata files which described each field contained the database. There was a total of 14 zipped files that comprised the complete dataset. Each of the zipped files extracted into four files with the following extensions: dbf, shp, prj, shx.

In reading through the metadata about each of the files, it was determined that the files needed to be loaded into a mysql database so that the unique identifier for each road segment could be used to join tables and review data about road segments. A separate server running FreeBSD, apache, mysql, and php was then setup on an older computer to allow for the import of the dbf file into mysql. While an import program

was written in php due to my familiarity and the assumption that would speed the import process along, the number of records in the dbf files significantly increased the load time. Each file contained just under 330,000 rows and a minimum of 50 columns. All 14 dbf files were loaded into mysql.

In examining the data imported from the 14 dbf files and using the metadata information, a sequencing of the road information had to be determined. The roads were sequenced by county in ascending order based upon where the road starts in the county and either south to north or west to east. Therefore, to get a complete route of the road, the counties also had to be accessed in order as they move south to north or west to east across the state. While I originally determined the ordering by logical reasoning, it was confirmed by contacting the IDOT staff and receiving a confirmation email of the correct ordering.

While the ordering of the road segments problem was overcome, unfortunately, in the examination of the database fields, it was determined that the GIS coordinates for each segment were not contained in the dbf file. The GIMS program uses the shp file to provide GIS information. Therefore, to put the GIS coordinates into the extracted file to simulate what would ultimately come from the cooperator in the real world application of CIMoRE, I opened the shape file from within ArcGIS and then extract the coordinates using the ArcToolbox's conversion tool and saving the layer to a kml file. This text file was then used to import the xml

values for the latitude and longitude for each road segment, as well as its unique identifier, into the existing mysql database.

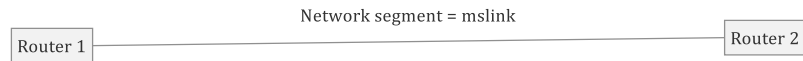
Once the road data existed in a single mysql database it could be examined to determine which fields would allow me to transform the road system into a network. Through reading the metadata information, running many queries of the database, and examining the extracted fields I began to get an overall view of the road system in Iowa. This work confirmed the initial suspicion that there would not be exact one to one mappings of data from the road database to network features needed for the ISEAGE configuration file. Therefore, multiple pieces of information would need to be used to construct each of the characteristics listed in Table 1.

### ***Network, Network Segments, and Devices***

Each road is identified by a unique road number. The data field is named state route and this can be used to represent the individual networks. By definition, a network segment is a connection between two devices. In the case of the road system, the roads are already divided into segments in the database. Each of these segments is identified by a field call mslink. However, the segments do not always have an on or off ramp or any other logical connector to another road which could be construed as a device. So, not only will the numbered pieces of the road be used, but also there has to be an interchange which will represent a router in the

network transposition. In the database, interchanges are marked, but there is also a piece of data called the access control which when combined with the interchange field means that there is an on and off ramp. This on and off ramp is a decision point for travelers. The analogy can be made that the on and off ramp decision point for a traveler is comparable to a router being a decision point for where a packet travels. This is shown in Fig. 5.

Network = state route



Router = interchange + access control mark

Order of the routers = state route number + county sequence number + route sequence number

**Fig. 5. Road database information used to create network**

Another piece of data that will be used for geospatially locating the devices is the latitude and longitude. Because of the manipulations completed to load the data files for the roads into a single mysql database each of the interchanges that act as routers can be located on a map of the state of Iowa. Finally, to get the network segments to be connected to each other in an ordered fashion, the state route number, the county sequence number and the route sequence number is used to build the connected network.

### ***Bandwidth and Loss***

For the road system, there is a report of the average daily traffic which is used to approximate average bandwidth. However, there is not a

maximum traffic field or data available. For the sake of proof of concept, a 25% overage on the average daily traffic was set as the upper limit of traffic or the maximum bandwidth in network terms. When a cooperator provides this information in the future, the cooperator's expertise would be used to estimate this upper limit.

In the case of the road system, there is not a data field reported in the IDOT database that can be translated into acceptable loss. However, lane closures and road reconstruction would roughly translate into loss on a network. Therefore, for proof of concept again, a 25% decrease in the calculated maximum traffic (or maximum bandwidth) was used to represent a single lane closure in one direction on a section of the Interstate. This would then translate to a 25% loss of packets on the network. Obviously, four lanes of closure on the Interstate, which can be experienced during an extremely severe vehicle accident, is translated as a 100% loss of packets on the network.

Although the mysql tables were populated in the CIMoRE database for all roads in Iowa, for this dissertation, the roads of interest will be the Interstates 80 and 35 only.

### **ICN Data**

The Iowa Communications Network (ICN) was selected as the communications subsector to be used in this dissertation. It is a state-owned agency that was created by legislation in the Iowa statehouse in 1989. By 1994 it had 104 endpoints. The ICN today has an estimated



8661 miles of fiber – 3400 miles owned by the state and 5261 miles leased. The ICN provides video, data, and voice for 1500 authorized users throughout the State of Iowa, including K-12 schools, higher education, hospitals, state and federal government, National Guard armories, and libraries [27, 28]. In this dissertation, I'm not looking at the video conferencing aspect of the ICN, but the ICN acting as an Internet Service Provider (ISP).

As with the IDOT data, I was unable to find a cooperator within the ICN organization to provide the data about its the physical layout and characteristics, so the data had to be approximated. Unfortunately, there was not an electronic format enumerating where the fiber runs in the state, what cities are connected, or the types of connections. There are multiple static maps available on various web sites, however, these did not give any detailed information such as the actual path the fiber took or the towns it passed through. A listing of the rooms where a person or organization can book an ICN room was the most specific electronic information available. However, I was fortunate that a copy of two paper maps from 2003 were in a faculty member's possession. While these are old, and probably outdated, they contained enough physical and logical information on them, that the proof of concept database could be constructed. From this constructed database a flat file was generated. Again, this step is to approximate what a cooperator would be providing to the CIMoRE project in the future. Fortunately, after working with the IDOT road data, I had

some ideas of the types of data fields that would be needed to create using these two maps.

The first map provided rough information about the network topology. It identified the five fiber loops that comprise the ICN for the state. It also showed the two levels of end nodes and the specific towns in which they were located. The first group of end nodes that were implemented in the building of the ICN were at the 15 community college districts in Iowa and they were identified on the map by the community college number. Additionally, the three regents universities were identified at this first tier. The second tier of end nodes were implemented at later date and were identified by town name only. Additionally, the number of fiber pairs were identified on the map, as well as the distance from fiber connection to fiber connection. Since the largest number of fiber pairs were at Camp Dodge (Johnston), IA, it was assumed that this was the center of the network and that all five of the network loops in the state would terminate back at this location.

However, the first map did not provide any information about the devices connecting the end nodes through the fiber, nor the geographic path these fiber pairs took. The second map provided approximate geographical information about the location of the fiber, primarily by identifying the cities through which the fiber passed on a map of the state of Iowa. Additionally, the cities in which switches and repeaters were located were shown on the map.

Using these two maps, the data that a cooperator to the CIMoRE project would provide was approximated. It was stored in an Excel spreadsheet which was uploaded into the CIMoRE mysql database. Because the original concept of the ICN was to connect all of the county seats with fiber and then create a star network off that central county location to the high school, the two maps were used to create the data points for the five network loops. Each of the network loops was numbered. Then, working between the two maps, the city names were used to identify the connections of each network loop. Fig. 6 shows how the networks were constructed.

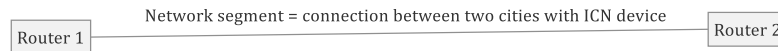
After the manual creation of the network, the latitude and longitude coordinates of the towns still need to be identified. By saving the Excel file to a comma separated table Google Refine could be used to retrieve the latitude and longitude of each city [29] in a programmatic way. This file was then uploaded into the CIMoRE mysql database to approximate what would be received from the cooperator.

Because the data was specifically constructed for this project and because this is already a communications network, less conversion work had to be done than with the highway information.

### ***Network, Network Segments, and Devices***

In the case of the ICN, I was already working with a network topology so there isn't a conversion process that is needed. It was more a process of trying to understand how the ICN is constructed which did not

Network = fiber optic loop



Router = city identified as having an ICN device

Order of the routers = order cities with ICN device on a map all starting from Camp Dodge

**Fig. 6. ICN information used to create network**

prove easy as no databases or cooperators were found. The fiber between the towns represented a network segment. Each town has a device in it and those are treated as routers. And, while the ICN can carry synchronous video, as well as data, for the purposes of this dissertation, the only traffic considered was data traffic.

### ***Bandwidth and Loss***

There is no indication on the maps what bandwidth is currently available on the fiber loops that connect the state. The assumption was made that it is single mode fiber and running at 40 Gbps. However, that is undocumented at this time. In a real set of data from a cooperator this value would be provided. So, for proof of concept purposes 40 Gbps is used as the maximum bandwidth. The assumption was made that 75% of that bandwidth is used on an average day or 30 Gbps rate.

### **Electricity Transmission Lines**

Again, there was no electronic data to be had, so a paper map was purchased from the Iowa Utilities Board for \$12 and a layout was created in an Excel spreadsheet using the data from the paper map. Again, because a comma separated file was created, Google Refine was used to add the latitudes and longitudes. The process mirrored the process undertaken for the ICN network. The Excel spreadsheet was loaded into the CIMoRE mysql database to approximate what would be received from the cooperator.

### ***Network, Network Segments, and Devices***

The transmission line type in kV and the line number was used to identify the network. Each city that the line passed through was recorded as a router and the segments were numbered from the generating station out to the final distribution point. If a generating station or a transmission station was in the city it was noted and stored in the mysql database. However, generating stations and transmission stations are not used in this dissertation. The data was just included for future use. Fig. 7 shows the creation of the electricity subsector network.

Network = transmission line type + line number



Router = city with transmission line through it

Order of the routers = order cities starting from the generating station

**Fig. 7. Electricity line information used to create network**

### ***Bandwidth and Loss***

Unlike the other two subsectors, electricity will not have a percentage loss in CIMoRE. It is either 100% loss meaning the power is out or 0% loss which means the lines are functioning normally.

## CHAPTER 6. ISEAGE MODIFICATIONS AND CONFIGURATION CHANGES

The Internet Scale Event and Attack Generation Environment (ISEAGE) was developed as a cyber security research testbed by the Information Assurance Center (IAC) at Iowa State University under funding from the Department of Justice (DoJ). It provides a routable Internet that has an air gap proxy server through which only three protocols are allowed: http, https, and ftp. This provides a safe environment for networking, cyber security experimentation, and penetration testing while preventing that traffic from entering the production network.

### How ISEAGE Works

The core of the ISEAGE testbed is a routable IP network which supports traffic to and from the virtual networks it hosts. It provides IP address space for each virtual network and allows the running of a network as if it were actually sitting on the Internet. The networks use public IP addresses that are “borrowed” from ranges used in the Internet, but, because of the air gap proxy, none of these public IP addresses ever escape into real Internet traffic. Because of ISEAGE’s internal programming called ISEFLOW, ISEAGE allows traffic to appear as if it is routed through the Internet, although all traffic is contained within the ISEAGE environment.

ISEAGE is unlike conventional testbeds where each router is represented by either a real router or a software router running on a computer. ISEFLOW, the internal programming, supports the concept of an internal cloud network where the cloud represents a cluster of routers. If one of the computers performs a traceroute to another network, it would see a number of hops between itself and the other network as if there were real routers between it and the other network. The TTL field in the IP header would also indicate the traffic traversed multiple routers. Again, this allows traffic to look as if it was traveling from a local network through the Internet to another destination local network. However, the traffic is contained in the testbed server.

ISEAGE can be deployed in a single server or over multiple pieces of hardware. When it is installed over a number of servers it has been named an ISERink. ISERinks have been used extensively in the classroom environment (Iowa State's CprE 230, 231, 431, 530 and 532, Oklahoma State's 4523 and 4233), in cyber defense competitions for high schools, 2-year and 4-year institutions, and in cyber security training for National Guard units in Iowa, Alaska, and Minnesota [30, 31]. The use of an ISERink has been casually called creating a "playground" where individuals can build, work, and test systems.

The following description and image in Fig. 8 is described in the terms of students working in a classroom environment. However, the



setup is the same whether the users are students, guardsmen, or professionals using the ISERink.

Currently, VMware ESXi servers are used to implement the ISERink. As the brown line in Fig. 8 depicts, students enter the management network from their residence hall or apartment via the vCenter server. The vCenter allows them access to the three ESXi servers where they can build and configure their networks or run penetration testing. In Fig. 8 the student servers are labeled Blue 1, Blue 2, and Blue 3. The servers that students build and configure can only communicate through the ISEAGE network. There are currently 45 public IP ranges that have been “borrowed” from the Internet with 15 assigned to each of three internal boards. These boards will be described in more details, but for now they are represented in Fig. 8 as the blue ISEAGE logo. For example, Board 1 is connected to Blue 1 server and the students assigned to that server build their networks or test their equipment using those IP ranges.

Traffic generated by these machines, noted by the blue connecting lines in Fig. 8, is routed internally by ISEFLOW. If the traffic is intended for an IP in the ISEAGE network and it is on the same board, ISEFLOW moves the traffic through its internal routers on that board to allow the traffic to look like it moves through multiple hops before returning back to an end node network. The end node routers are called an outside network in ISEFLOW terms. If the traffic is intended for one of the other

IPs in ISEAGE, ISEFLOW routes through the Backplane to the corresponding board again using its internal routers to demonstrate multiple hops. If the traffic is sent to an IP that is outside of the ISEAGE environment (noted by a gray line in Fig. 8), ISEFLOW sends the traffic through its internal routers to Keyhole 1. If it is http, https, or ftp traffic, then Keyhole1 performs a network address translation on the request and sends it out through Keyhole 2 to the Internet. Traffic using other protocols is dropped.

In a smaller deployment or for development work, ISEAGE and the systems being built for testing are housed in the same ESXi server. However, the testbed functions in the same manner as was described above. It was the smaller, single server environment that was used for CIMoRE development.

### **Allowing for Different Types of Traffic in ISEAGE**

Normally ISEAGE is configured with two or three hops between the backplane that connects multiple edge routers. Fig. 9 below shows the internal setup of ISEAGE which is depicted as the ISEAGE logo in Fig. 8 above. It is this routing that makes traffic look like it takes multiple hops in the ISEAGE network.

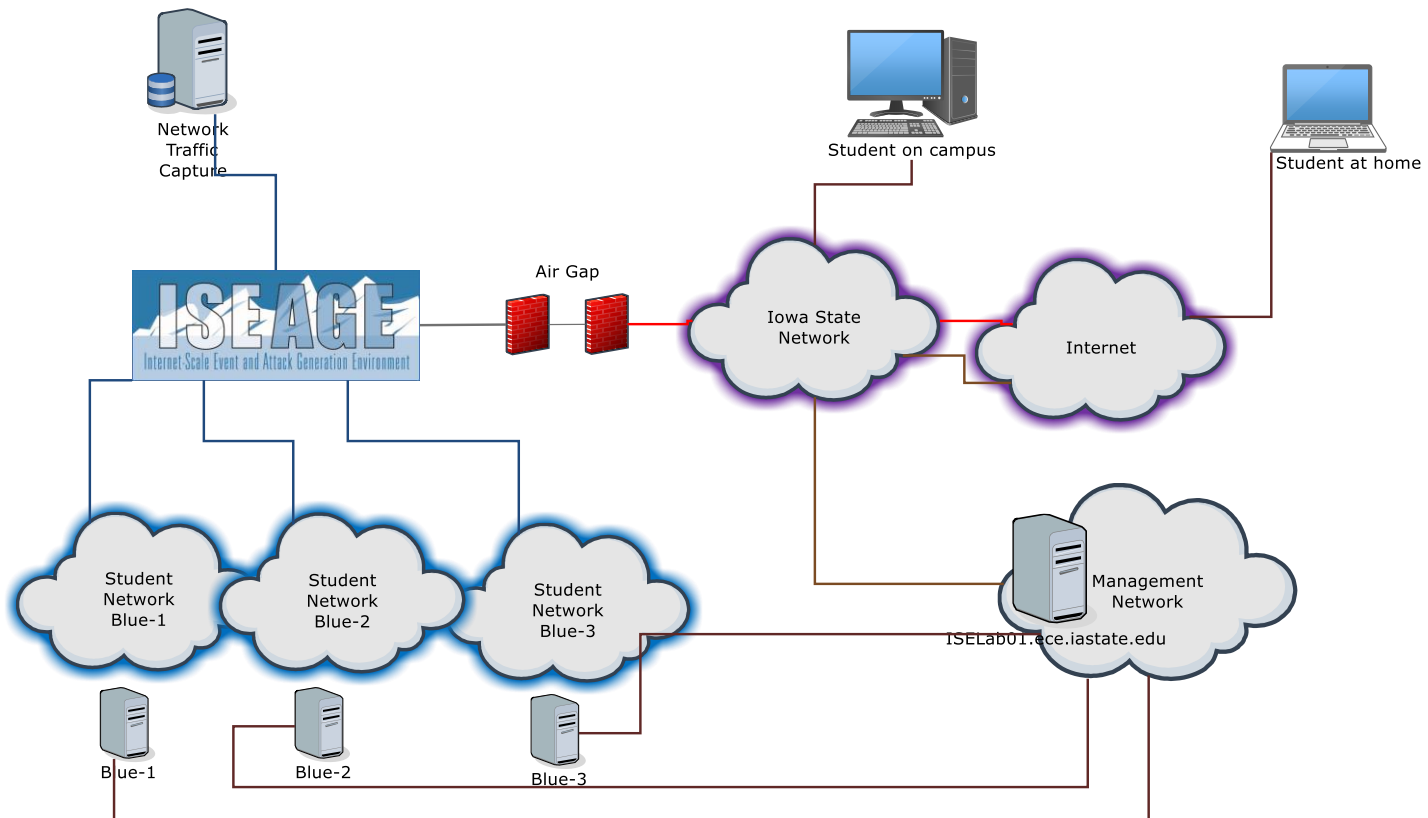


Fig. 8. Typical ISERink setup

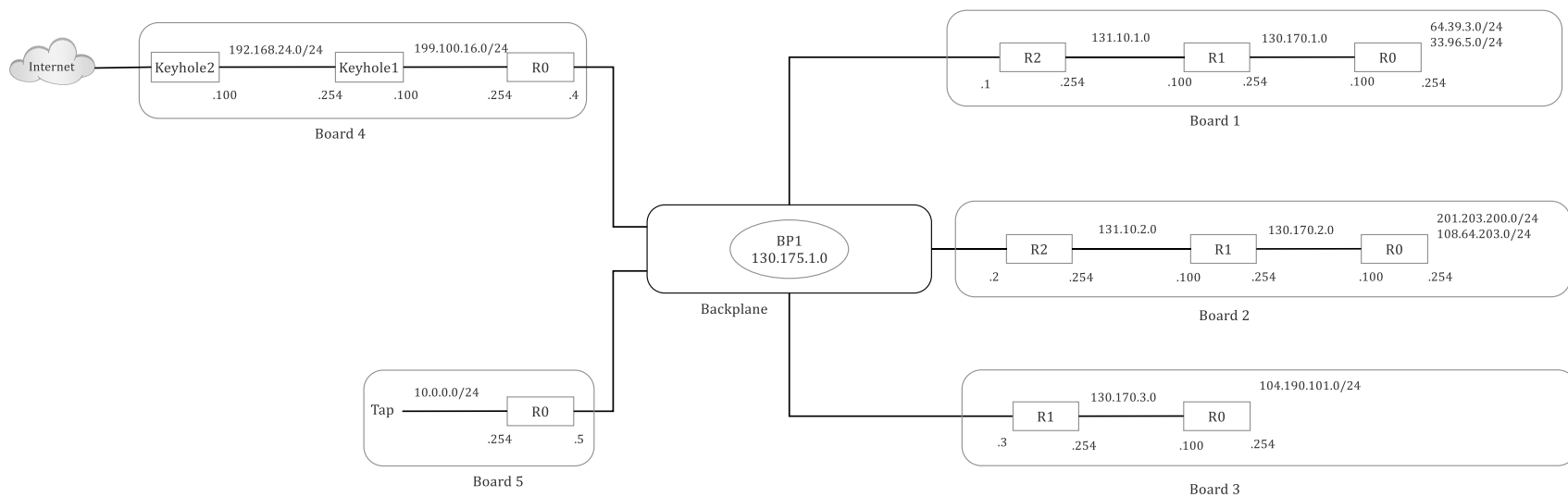


Fig. 9. Typical ISEAGE internal networking

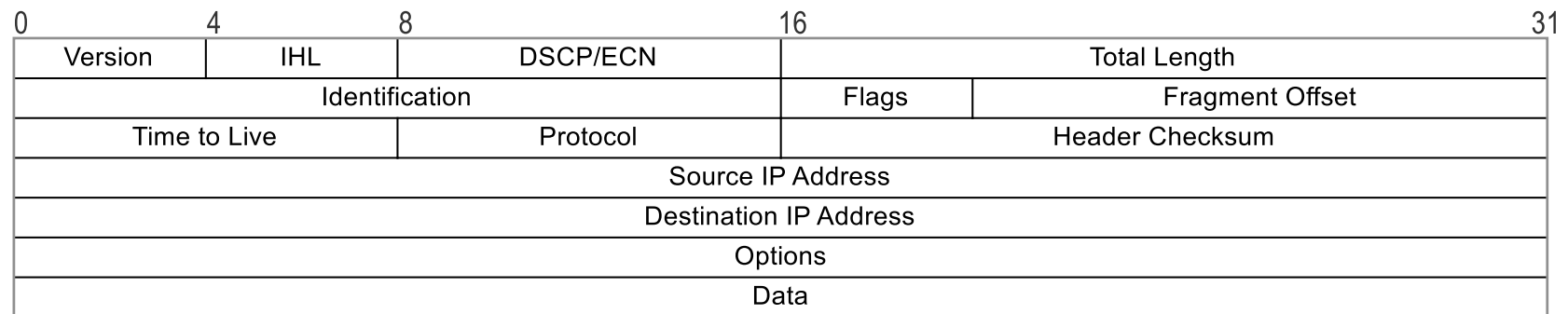
The ISEFLOW software performs the routing functions in ISEAGE by reading configuration files to build the routers and route tables at startup time. ISEFLOW then handles the network traffic as it moves from end node router to end node router inside the ISEAGE testbed. While ISEAGE works very well for routing TCP/IP traffic, several modifications need to be made in ISEFLOW to allow network traffic to model critical infrastructure subsector networks.

While all traffic moving in the CIMoRE model is TCP/IP traffic, there needed to be a way to differentiate which subsector the traffic represents and how to treat it. ISEFLOW does not differentiate between types of traffic at the IP layer. ISEFLOW was programmed to recognize the old Type of Service (ToS) field in IPv4. The ToS field was created to provide preferences to different packets depending upon the setting in the field. Historically this field was not used and all 8 bits were set to 0. However, as services such as Voice of IP (VoIP) which require a minimum bit rate and maximum latency to operate have evolved and become commonplace, the prioritization of certain types of packets has increased in importance. In 1998, the ToS field was changed to be the Differentiated Services field (DS field) by definition in RFC 2474 [32] and implemented in RFC 2475 [33]. It has since been updated in 2002 by RFC 3260 [34]. The use of the DS field allows a network administrator to define different classes of traffic such that each class can be treated differently. Each class is given a relative priority based upon the total bandwidth of the

connection. This effectively achieves different levels of Quality of Service (QoS) for each class of traffic.

Although traffic in a Differentiated Services (DiffServ) domain (a network that is using Differentiated Services) can be classified by fields other than, or in addition to, the DS field (for example, source address or destination address), the common practice is to have routers make traffic classifications based upon information contained in the DS field and then assign some relative priority of access to network resources on the egress of the router.

According to the standard the DS field contains two pieces of information; a 6-bit Differentiated Service Code Point (DSCP) and an Explicit Congestion Notification (ECN) code in the least significant 2 bits (See Fig. 10). This means there can be 64 different traffic classes defined with DSCP values each with its own differential forwarding treatment. The forwarding treatments of the packets are called Per Hop Behaviors (PHB). The value of the DSCP can be thought of as an index to the PHB action table. Each router can do different per hop behaviors based upon its mapping table. That means each router can do conditioning (add a DSCP) or can change the DSCP to PHB mapping.



**Fig. 10. IPv4 packet layout**

While the DS field was designed to be used for QoS purposes, the standard allows for locally customizable mappings of DSCPs and PHBs. There is no requirement to use standard mappings in a local network. Therefore, I have usurped the DS field to mark traffic for different types of critical infrastructure subsectors (See Table 2). Each router's route table which is configured through ISEFLOW would be set in the configuration file with the specific DSCP(s) allowed on the connection. If all traffic is allowed on the connection, all ones are set (63) in the router's configuration. This check would occur on egress of the interface because there could be multiple DSCPs entering a router, but potentially only one DSCP allowed to exit on each interface.

**Table 2. DS Code Point to PHB Mapping**

DSCP Value Binary	DSCP Value Hex	DSCP Value Decimal	PHB
111111	0x3F	63	All traffic allowed
000001	0x01	1	Road traffic
000010	0x02	2	ICN traffic
000011	0x03	3	electricity traffic

Additionally, the ECN 2-bit values can be used to identify a router that is in a healthy state, failing state, failed state, or recovering state (See Table 3). The routers could start in any of the four states which will be discussed in Chapter 7.



**Table 3. ECN Value and Interpretation**

ECN Value Binary	ECN Interpretation
00	Failed – No traffic allowed
01	Failing – Going down
10	Recovering – Returning back to normal
11	Healthy – Full use of bandwidth

All routers must be able to interpret DSCPs and apply PHBs.

Therefore, any traffic introduced into ISEAGE will need to have the DS field coded. Additionally, any physical routers or physical equipment such as SCADA devices that are connected into the ISEAGE environment will need to either have the same DSCPs to PHB mappings or have their traffic run through a conditioner that inserts the DS field information.

### **Defining New Types of Routers in ISEFLOW**

A network diagram of the internal network required to model three critical infrastructure subsectors is shown in Fig. 11. This network includes highway traffic, ICN traffic, and electricity traffic. It was drawn using both the old and the new router types defined in ISEFLOW for CIMoRE. The overall picture is shown here to give the reader perspective before the new types of routers are discussed. The four types of routers used in ISEFLOW will be described below.

The connections to Boards 4 and 5 were omitted in Fig. 11 to save space. However, Boards 4 and 5 will still be used in CIMoRE the same way they are used in a traditional ISEAGE implementation shown in Fig. 9. Fig. 11 also depicts significantly fewer routers than what the full dataset

would require (and what I recorded in the mysql database). They were reduced so that the picture was understandable.

### **Normal Router**

The first type of router to be discussed is a router that acts like a physical Cisco router. It has multiple ingress and egress interfaces and follows normal routing protocols. This router type already exists in ISEFLOW and is just called a router. For the purposes of this dissertation it will be labeled a normal router. As shown in Fig. 11 a majority of the routers in a CIMoRE configuration would be normal routers. They are the routers that pass a single type of critical infrastructure traffic and represent one location in the topology. In Fig. 11 they are gray for the roads, light blue for the electric grid, and green for the ICN.

Fig. 12 shows the detail of two normal routers on Board 1 labeled R3 and R2. These represent the interchanges on I-35 for Clear Lake (R3) and Northwood (R2) which would be normal routers. The traffic would represent the vehicular traffic moving between the two interchanges on the interstate highway. Clear Lake would have two interfaces; one that connects to Northwood and the other connecting to Ames which is not pictured. Northwood only has one interface and that is connected to Clear Lake. Eventually, the Northwood router could have a second interface that would be connected to an interchange in Minnesota or a system for

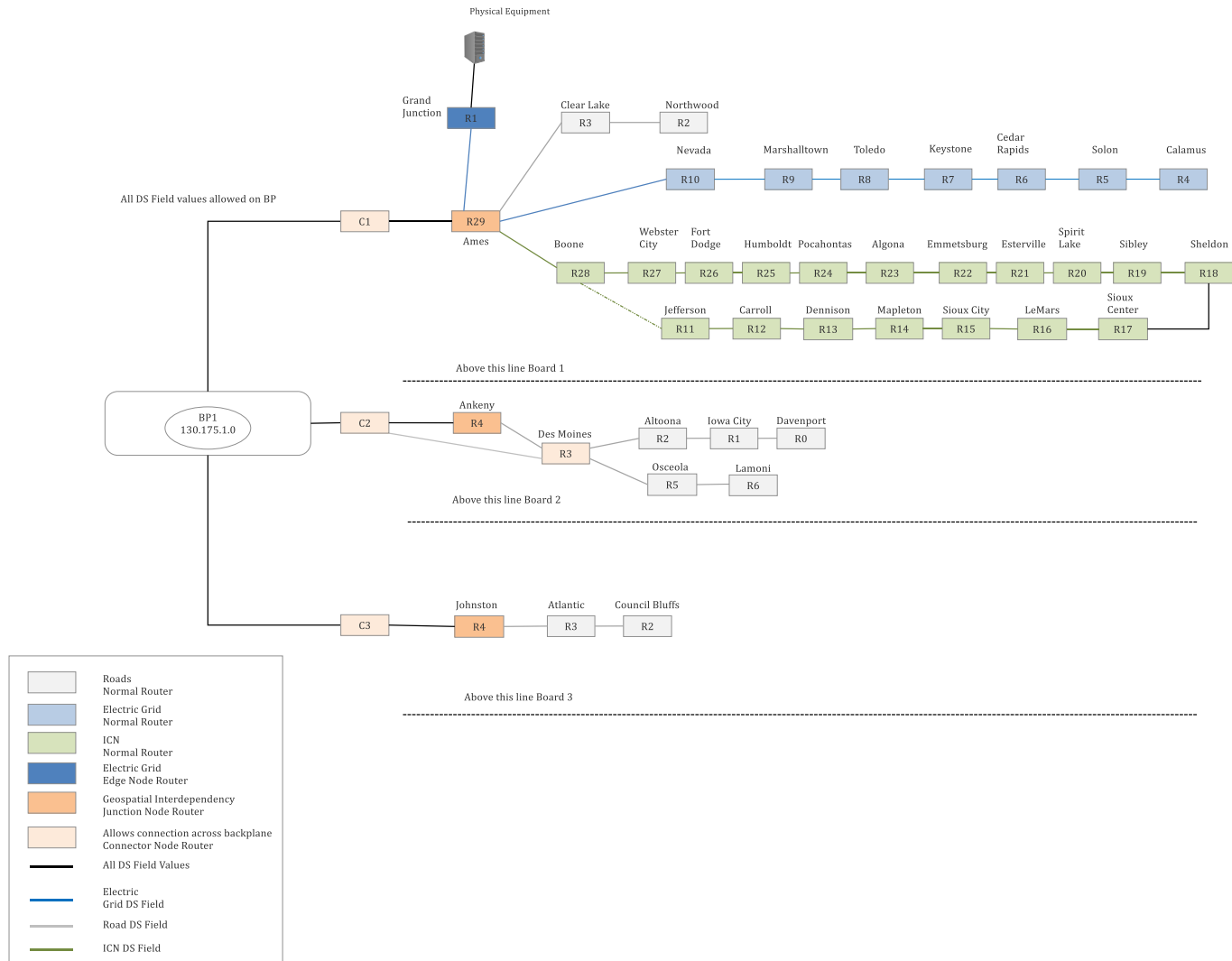
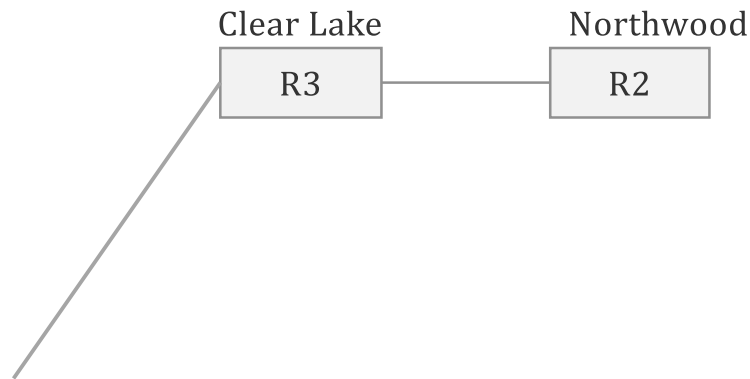


Fig. 11. Network diagram of three critical infrastructure subsectors

modeling Minnesota's roads, but that is outside the scope of this dissertation. Both routers would only allow traffic with a DSCP value of 1.



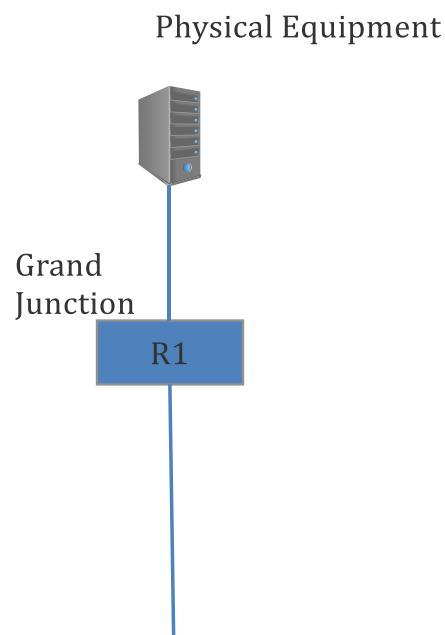
**Fig. 12. Normal routers used in the highway subsector**

### **Edge Node Router**

The second type of router that already exists in ISEFLOW is the outside router. These routers have an interface that is assigned multiple IP address ranges which allows multiple networks to be connected at these points. In a deployment of an ISERink for a classroom, CDC, or training it is this type of router that allows the 15 IP address ranges for the student networks to be connected to one board. While this router is called an outside router in ISEFLOW, that causes confusion for someone not familiar with the inner workings of ISEAGE. The naïve assumption is that it is “outside” of ISEAGE. Therefore, for the purposes of this dissertation the name of this type of router is changed to edge node router to more clearly define its function.

Fig. 13 shows the use of an edge node router in critical infrastructure subsector modeling. In this case, instead of providing an IP

for students to build their network, the IP range would be allowed so that physical equipment such as a supervisory and control data acquisition (SCADA) system could be connected into the CIMoRE. The figure shows a server depicting a SCADA system that would be passing traffic about the electric grid connected to an end node router. The DSCP value would be 3 on this edge node router.



**Fig. 13. Edge node router used in electricity network**

Instead of connecting physical equipment into CIMoRE as shown in Fig. 13, an edge node router could be used to connect another software program that models critical infrastructure subsectors. If the subsector modeled in software has an output or input interdependency with critical infrastructure subsectors being modeled in CIMoRE, the edge node router would allow that information to be fed into the testbed.

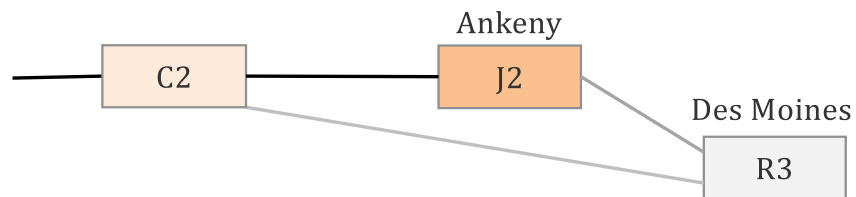
Additionally, an edge node router could be used to connect CIMoRE into another network testbed that is modeling multiple critical infrastructure subsectors. The second testbed could provide interdependency data for the critical infrastructure subsectors being modeling inside of CIMoRE through this type of router.

A cautionary tale to using an edge node router to connect to physical equipment, critical infrastructure subsector modeling software, or another testbed is that any physical connections into CIMoRE would need to be conditioned; adding the DSCP to match our network under test. Of course, ISEFLOW would also have to be configured to know about the external systems.

### **Connector Node Router**

In addition to the currently existing normal and edge node routers, two new types of routers have been added to ISEFLOW. The first is a connector node router. The normal router definition in ISEFLOW allows for multiple ingress and egress ports, but it does not allow for multiple connections to the backplane to allow traffic to cross from one board to another. As ISEFLOW currently works any traffic that needs to cross from one board to another must pass through a single normal router connected to the backplane. This router is included in the path and hop count the packet takes. In the case of modeling critical infrastructure subsectors there needs to be a way for multiple paths through the backplane without requiring the hop be included in the total count.

A connector node router allows different types of critical infrastructure traffic to move from board to board without being counted in the number of hops from source to destination. In this example it will move from Board 2 to Board 3. Due to how ISEFLOW is written, a connector node router must be configured as a router, but it functions as a switch. Fig. 14 shows connector node router C2 which is connected to Board 2. It allows connectivity to the back plane for both the Ankeny router labeled J2 (a junction node router discussed below) and R3 a normal router depicting Des Moines interstate vehicle traffic. In this figure, the DSCP value to J2 is 63 which allows all types of traffic. On the connection to R3, only the DSCP value of 1 (highway traffic) is allowed.



**Fig. 14. Connector node router used in the highway network**

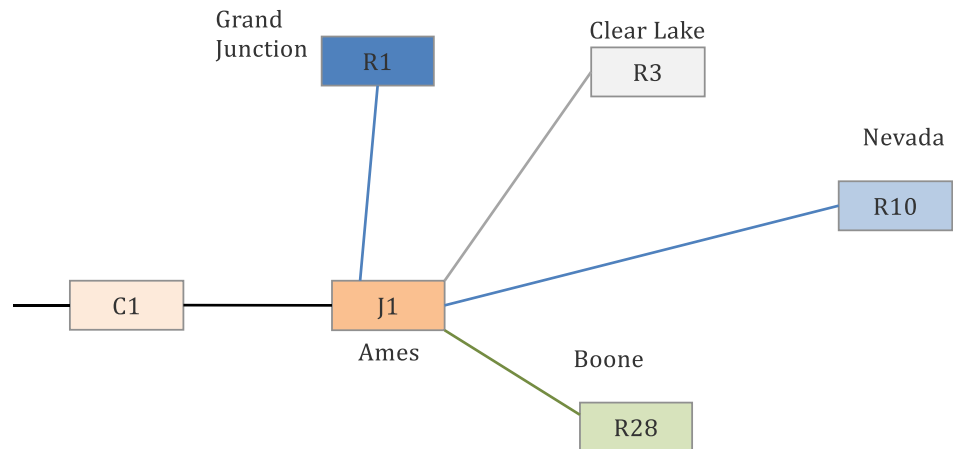
The Des Moines router labeled R3 on Board 2 represents interchange 137 where I-80 and I-35 merge. The traffic coming from the north arrives through the Ankeny interchange (greatly simplified from the multiple Ankeny interchanges so that a picture could be drawn). The road traffic coming from the west comes through Johnston interchange and then on to this location. So, while the router at 137 will have interfaces for I-80 and I-35, the road traffic cannot come through a single router. The

traffic from the north will come through Ankeny and will be directly attached as shown in Fig.14 because it is configured on the same board as Ankeny which is Board 2. However, Johnston is configured on Board 3 and the connection to it must be made through the connector node router. Again, the connector node routers will not be part of the hop count. They are only used to provide connectivity through the Backplane.

### **Junction Node Router**

The second new type of router defined in ISEFLOW is a junction node router. The junction node router allows an interdependency to be identified. In this dissertation only geospatial interdependencies are included, but the new junction node router definition allows multiple types of interdependencies to be included in CIMoRE. A junction node router attaches to a connector node router and has at least two different types of critical infrastructure subsector traffic on it. In this dissertation it depicts a physical location where two critical infrastructures reside in near proximity. In Fig. 15 Ames, J1, is a junction node. It has all three critical infrastructures located in near to each other. So, this junction node router will be able to receive all three DSCP values (63) on the interface connected to connector C1. It only allows electricity traffic (DSCP value of 3) on two interfaces to router R1 and R10. Only road traffic (DSCP value of 1) is allowed on the interface connected to R3. And only ICN traffic (DSCP value of 2) is allowed on the connection to R28.





**Fig. 15. Junction node router used to represent geospatial interdependency**

The junction nodes are created by using the latitude and longitude of the critical infrastructure subsector component to mark their locations on a map of Iowa. The map of Iowa then has a grid superimposed over top of it. Any critical infrastructure components that lie in the same grid cell are determined to be geospatially interdependent and are represented in CIMoRE as a junction node.

### **Writing a CIMoRE Configuration File for ISEFLOW Version 1.1**

Returning to the larger image, I added IP ranges and interface IPs to produce Fig. 16. This allowed me to write a full ISEFLOW configuration file to be used in CIMoRE. However, the full configuration file is more than 25 pages long, so it was not included in this dissertation or in an appendix. It is available upon request. As a sample of what a configuration file looks like, example code will be provided with figures to depict each type of router configured in CIMoRE.

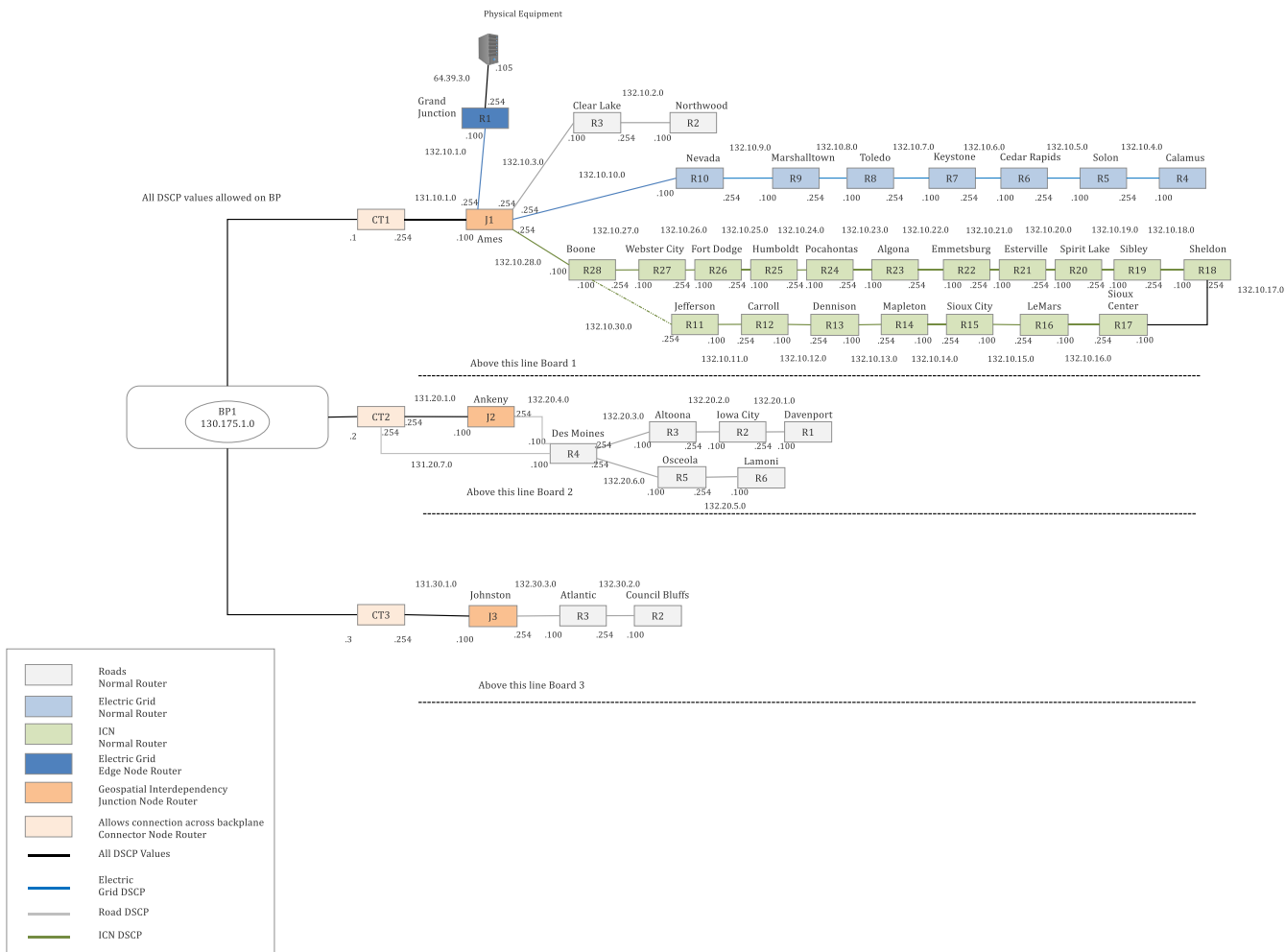


Fig. 16. Network diagram with IP addresses assigned

## Backplane Definition

The top of the configuration file for ISEFLOW constructs the Backplane which allows traffic to be passed from board to board. The global definitions for the Backplane for CIMoRE remain the same as for use in a normal ISEAGE configuration. As shown in Fig. 17 there are still five boards in the first backplane. Boards 1-3 are used in CIMoRE as they are used in any ISEAGE testbed setup; for machines or equipment that are under test. Board 4 connects the air gap proxy called Keyhole and Board 5 is the Tap board which allows for capturing all traffic that occurs in the testbed. There is also a second backplane created that is part of the packet captures that enables the two Keyhole servers to communicate with each other.

```

Globals= {
  BPnet=0,130.175.1.0,24
    board=1,2,1,130.175.1.1
    board=2,2,1,130.175.1.2
    board=3,1,1,130.175.1.3
    board=4,0,1,130.175.1.4
    board=5,0,1,130.175.1.5
  Pargs = {
    name=Primary Backplane
    #board=board, router, interface, IP
  /pargs
  BPnet=1,10.10.10.0,24
  Pargs = {
    name=KeyHole Backplane
  /pargs

```

Fig. 17. Backplane definitions in the ISEFLOW configuration file

## Global Link Definitions

The next section of the ISEFLOW configuration file defines the different types of links that can be used in CIMoRE. These global links allow the initial states for each of the critical infrastructure subsectors to be declared as well. In this case, each is defined as healthy and functioning with no loss. As shown in Fig. 18 there were three global links defined in this configuration file. One was created for each critical infrastructure subsector used.

Each of the global links contains information about three characteristics of that critical infrastructure subsector. The three characteristics used in this dissertation are maximum bandwidth, average bandwidth, and percentage of loss. These are values that are received from the cooperator when CIMoRE is in a fully functional state. For this proof of concept, these are the values I created and loaded into the mysql CIMoRE database. The number of parameters can be expanded in the global link definition if additional values are needed to describe a different critical infrastructure subsector or if CIMoRE needs to have additional features added to it in the future.

```

#These parameters are all the initial states for healthy infrastructures

GLINK=1,CIMORE
  Pams = {
    name=interstate
    #default maximum capacity of the road
    #can be changed information coming from database
    #number is in thousands
    maxbandwidth=10
    avebandwidth=5
    #reduction in lanes in percentage
    #50 means only half of average traffic volume
    loss=0

GLINK=2,CIMORE
  Pams = {
    name=icn
    #default maximum bandwidth of the link
    #can be changed information coming from database
    #number is in gbps
    maxbandwidth=10
    avebandwidth=1
    #reduction in bandwidth in percentage
    #50 means bandwidth has been reduced by 50%
    loss=0

GLINK=3,CIMORE
  Pams = {
    name=electric
    #type of transmission line
    #can be changed information coming from database
    #number is in kV
    maxbandwidth=161
    avebandwidth=161
    #reduction in bandwidth in is on or off
    #if 1, then the line is off. If 0, then line functioning normal
    loss=0

```

**Fig. 18. Global links to define defaults for each subsector**

### **Board Definitions**

The next section of the ISEFLOW configuration file defines each of the boards. While all three boards were fully defined based upon Fig. 16 in the 25-page configuration file, this configuration discussion will use

Board 1 because each of the types of routers used to configure CIMoRE are found on this board.

### ***Connections Section***

At the beginning of each board definition is the section about connecting the “physical” wiring for each router found on the board. These can be thought of as plugging the ethernet cables or fiber jumpers into the different routers to make the physical paths. In Fig. 19 the connections for all four types of routers are included.

The connections section starts with the edge node router. It is the simplest of the connections to make. The entry says that R1 on interface 0 is an outside router (O) which is ISEAGE speak for an edge node router. This special kind of router allows CIMoRE to connect physical equipment such as SCADA equipment or other modeling software and hardware into the testbed as described previously.

The next connection is a connector node router whose designation is a CT. The two letter abbreviation of CT is used rather than the single letter C because C is already used in ISEFLOW for the declaration of a cloud which represents Internet traffic for which there is no need to keep track of hops. The connector node router allows all traffic by using the global links of 1 for roads, 2 for the ICN, and 3 for electric. This means that all packets representing those critical infrastructure subsectors’ traffic will be allowed on the “wire.” Reading the line it says that connector node

1 (CT1) is wired to Backplane 0 on interface 1 and allows packets representing interstate traffic (DSCP value of 1), ICN traffic (DSCP value of 2) and electricity traffic (DSCP value of 3).

As discussed previously, the junction node router allows modeling of interdependencies and specifically in this dissertation the geospatial co-location of critical infrastructure subsectors. The designation is J for a junction node and in this case there are five interfaces that need to be defined. The first line of the junction node definition states that junction node router 1 on interface 0 (J1,0) is connected to connector node router CT1 on interface 0 and allows all three types of critical infrastructure packets to traverse the connection.

The second line of the junction node definition shows that J1 on interface 1 is directly connected to an edge router R1 on interface 1. This connection is configured as a global link that only allows electricity traffic and has the characteristics defined in the global link of being a 161 kV line that is operating in a healthy state. In other words, the power is on for that connection. The rest of the J1 interfaces are similarly configured in that they are specific to a type of critical infrastructure traffic and its characteristics based upon the global link attached.

```

board=1
connections={

#edge node router
#RRouter_num,Int => O
R1,0 => O

#connector node router
#all traffic types allowed
#RRouter_num,Int => backplane_num,Int,link_type(GLINK),[link_num]
CT1,1 => B0,1,1,2,3

#junction node router
#connects to one of four types
#edge node => R, normal router =>R, connector router => CT, junction node router => J
#RRouter_num,Int =>
edge_normal_connector_junction_num,Int,link_type(GLINK),[link_num]
J1,0 => CT1,0,1,2,3
J1,1 => R1,1,3
J1,2 => R3,1,1
J1,3 => R10,1,3
J1,4 => R28,1,2

#normal router
#RRouter_num,Int => router_num,Int,link_type(GLINK),[link_num]
R3,0 => R2,1,1
R10,0 => R9,1,3
R9,0 => R8,1,3
R8,0 => R7,1,3
R7,0 => R6,1,3
R6,0 => R5,1,3
R5,0 => R4,1,3
R28,0 => R27,1,2
R27,0 => R26,1,2
R26,0 => R25,1,2
R25,0 => R24,1,2
R24,0 => R23,1,2
R23,0 => R22,1,2
R22,0 => R21,1,2
R21,0 => R20,1,2
R20,0 => R19,1,2
R19,0 => R18,1,2
R18,0 => R17,1,2
R17,0 => R16,1,2
R16,0 => R15,1,2
R15,0 => R14,1,2
R14,0 => R13,1,2
R13,0 => R12,1,2
R12,0 => R11,1,2

```

**Fig. 19. Connections for Board 1**



The fourth router that is defined in the connections is the normal router. Many of the routers used in Fig. 16 were normal routers. The first line says R3 on interface 0 is directly connected to R2 on interface 1 and it has the characteristics of road traffic that is defined in global link 1. All entries for normal routers look similar with the exception of the type of global link applied.

### ***Router Section***

Each router on each board has its own definition that needs to be included in the configuration informatino. Fig. 20 is provides a network diagram to help illustrate each of the types of routers and their routing definitions discussed in this section. The first configuration entry in Fig. 21 is for connector node router 1 on board 1 (CT1). CT1 has two interfaces. Interface 1 is in the range of IPs for Backplane 0 which is 130.175.1.0/24. It is host .1 which represents Board 1. The DSCP value is also included. As discussed previously, 6 bits set in the DS field (DSCP value of 63) allows traffic representing all critical infrastructure subsectors in the model to be passed.

Interface 0 of CT1 is in the 131.10.1.0/24 network. The 131 represents one hop away from the Backplane and .10 means Board 1. Although not shown in Fig. 19, CT2 would have an interface 0 in the range of 131.20.1.0/24 which would represent one hop away from the Backplane (131) and on Board 2 (.20).

After the interfaces, the route tables for the connector node router need to be defined. For CT1, the route to the edge node router IP range (64.39.3.0/24), as well as routes to all normal routers must be added. In this case, the route going to those routers is to interface 0 on J1 (131.10.1.100). Additionally, the type of subsector traffic allowed on the route must be specified. For example, the route to the edge node router (R1) and the route to normal router (R10) only allow electricity traffic (DSCP value of 3).

There are two other routes that need to be defined for a node connected to the Backplane, in this case, CT1. There needs to be a path that connects to the air gap proxy Keyhole which is at 199.100.16.100. That traffic must exit to the Backplane and then go to Board 4 (130.17.1.4). Again, traffic representing all critical infrastructure subsectors in the model is allowed (DSCP value of 63). This would be for traffic (http, https, and ftp) that is requested to leave the network from Board 1.

And finally, the default route for traffic that is destined to move to another board inside the CIMoRE testbed is routed to Board 5 (130.175.1.5). Again, all traffic is allowed on this route which represented by the DSCP value of 63. Having the default route to Board 5 allows the capture of all traffic to any device that is connected to Board 5 and listening in promiscuous mode. Board 5 then routes the traffic back to the correct board on which the destination IP address is connected.

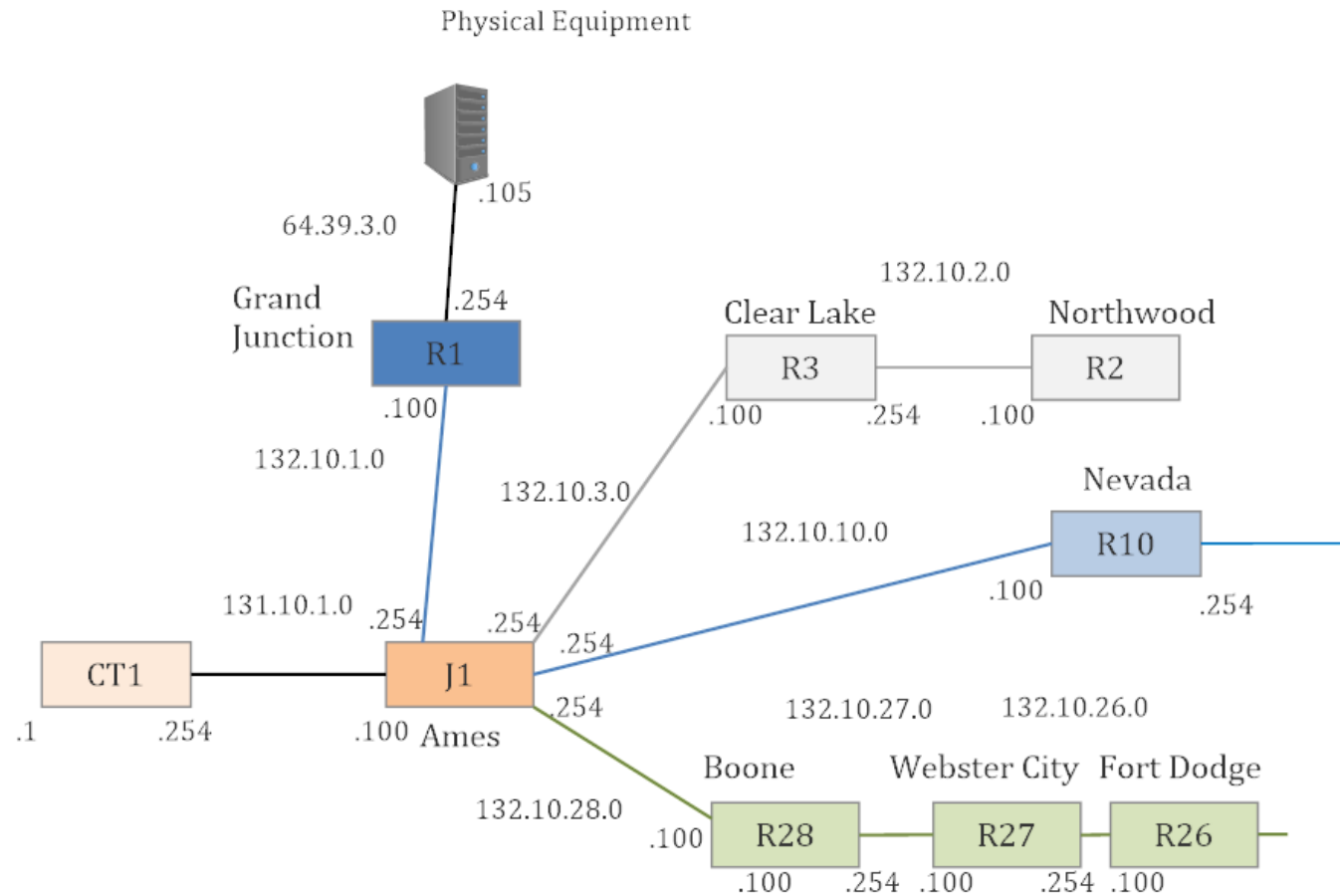


Fig. 20. Detail of network diagram with IP addresses for routes

```

#one connector node router
device=connector,1
#int,IP,mask,dsfield
if=0,131.10.1.254,24,63
if=1,130.175.1.1,24,63
#dest_ip,mask,next_ip,next_interface,dsfield
r_table=64.39.3.0,24,131.10.1.100,0,3
r_table=132.10.1.0,24,131.10.1.100,0,3
r_table=132.10.2.0,24,131.10.1.100,0,1
r_table=132.10.3.0,24,131.10.1.100,0,1
r_table=132.10.4.0,24,131.10.1.100,0,3
r_table=132.10.5.0,24,131.10.1.100,0,3
r_table=132.10.6.0,24,131.10.1.100,0,3
r_table=132.10.7.0,24,131.10.1.100,0,3
r_table=132.10.8.0,24,131.10.1.100,0,3
r_table=132.10.9.0,24,131.10.1.100,0,3
r_table=132.10.10.0,24,131.10.1.100,0,3
r_table=132.10.11.0,24,131.10.1.100,0,2
r_table=132.10.12.0,24,131.10.1.100,0,2
r_table=132.10.13.0,24,131.10.1.100,0,2
r_table=132.10.14.0,24,131.10.1.100,0,2
r_table=132.10.15.0,24,131.10.1.100,0,2
r_table=132.10.16.0,24,131.10.1.100,0,2
r_table=132.10.17.0,24,131.10.1.100,0,2
r_table=132.10.18.0,24,131.10.1.100,0,2
r_table=132.10.19.0,24,131.10.1.100,0,2
r_table=132.10.20.0,24,131.10.1.100,0,2
r_table=132.10.21.0,24,131.10.1.100,0,2
r_table=132.10.22.0,24,131.10.1.100,0,2
r_table=132.10.23.0,24,131.10.1.100,0,2
r_table=132.10.24.0,24,131.10.1.100,0,2
r_table=132.10.25.0,24,131.10.1.100,0,2
r_table=132.10.26.0,24,131.10.1.100,0,2
r_table=132.10.27.0,24,131.10.1.100,0,2
r_table=132.10.28.0,24,131.10.1.100,0,2
r_table=132.10.29.0,24,131.10.1.100,0,2
#proxy board 4
r_table=199.100.16.100,24,130.175.1.4,1,63
#default router board 5
r_table=0.0.0.0,24,130.175.1.5,1,63
Parms = {
name=b1ct1

```

Fig. 21. Connector node router CT1 definition

The routing table for an edge node router (R1) is unique as shown in Fig. 22. The edge node router can have multiple IP ranges connected to its “outside” interface. That interface is defined with `if_out` instead of the normal `if`. In the case of R1 there is only one IP range in Fig. 20 where SCADA equipment or other software or hardware modeling can be connected (64.39.3.0/24). The IP address given in the definition section is .254 which means R1 is the default route for all equipment physically added into CIMoRE so its traffic can move into the testbed. Notice also there is a media access control (MAC) address defined on this line. For each outside interface on an edge node router there needs to be a MAC address defined. This is generally done in some sequential manner if more than one IP range is added (multiple `if_out` s are defined.) Interface 1 on R1 connects to junction node router 1 (J1) and uses its interface 1 as the default route. Again, only electricity traffic is allowed on this interface (DSCP value of 3).

```

device=router,1
if_out=0,64.39.3.254,24,00:00:0c:31:01:aa,3
if=1,132.10.1.100,24,3
# Dest IP, mask, next IP, interface, dsfield
r_table=64.39.3.0,24,64.39.3.254,0,3
#default route J1
r_table=0.0.0.0,24,132.10.1.254,1,3
Parms = {
#abbreviation for outside router 1
name=b1or1

```

**Fig. 22. Edge node router R1 definition**

The junction node router J1 on Board 1 has five interfaces defined. Interface 0 connects back to the connector node router CT1 and has the IP of 131.10.1.100. It allows all types of traffic representing the critical infrastructure subsectors being modeled in CIMoRE (DSCP value of 63). Interfaces 1-4 allow only traffic from single critical infrastructure subsector to enter and leave the router on those interfaces.

The route tables for this junction node router are large. All routes going toward other routers under test in the testbed must have a route to them created. For example, all road traffic destined for the network 132.10.2.0/24 which connects R2 and R3 routes to 131.10.3.100 which is on R3 interface 1. Only road traffic (DSCP value of 1) is allowed on this route.

All electricity traffic (DSCP value of 3) for the 132.10.4.0/24 network which connects R4 and R5 routes to 132.10.10.100 which is R10 on interface 1. All other routes in the table are similarly configured.

The default route for all traffic not destined for an IP address on Board 1 is through the connector node router CT1 (131.10.1.254) which again allows all traffic on it.

```

device=junction,1
#int,IP,mask,dsfield
if=0,131.10.1.100,24,63 #ALL
if=1,132.10.1.254,24,3
if=2,132.10.3.254,24,1
if=3,132.10.10.254,24,3
if=4,132.10.28.254,24,2
#dest_ip,mask,next_ip,next_interface,dsfield
r_table=64.39.3.0,24,132.10.1.100,1,3
#directly connected 132.10.1.0
r_table=132.10.2.0,24,131.10.3.100,1,1
#directly connected 132.10.3.0
r_table=132.10.4.0,24,131.10.10.100,1,3
r_table=132.10.5.0,24,131.10.10.100,1,3
r_table=132.10.6.0,24,131.10.10.100,1,3
r_table=132.10.7.0,24,131.10.10.100,1,3
r_table=132.10.8.0,24,131.10.10.100,1,3
r_table=132.10.9.0,24,131.10.10.100,1,3
#directly connected 132.10.10.0
r_table=132.10.11.0,24,131.10.28.100,1,2
r_table=132.10.12.0,24,131.10.28.100,1,2
r_table=132.10.13.0,24,131.10.28.100,1,2
r_table=132.10.14.0,24,131.10.28.100,1,2
r_table=132.10.15.0,24,131.10.28.100,1,2
r_table=132.10.16.0,24,131.10.28.100,1,2
r_table=132.10.17.0,24,131.10.28.100,1,2
r_table=132.10.18.0,24,131.10.28.100,1,2
r_table=132.10.19.0,24,131.10.28.100,1,2
r_table=132.10.20.0,24,131.10.28.100,1,2
r_table=132.10.21.0,24,131.10.28.100,1,2
r_table=132.10.22.0,24,131.10.28.100,1,2
r_table=132.10.23.0,24,131.10.28.100,1,2
r_table=132.10.24.0,24,131.10.28.100,1,2
r_table=132.10.25.0,24,131.10.28.100,1,2
r_table=132.10.26.0,24,131.10.28.100,1,2
r_table=132.10.27.0,24,131.10.28.100,1,2
#directly connected 132.10.28.0
r_table=132.10.29.0,24,131.10.28.100,1,2
#default router CT1
r_table=0.0.0.0,24,131.10.1.254,0,63
Parms = {
name=b1j1

```

Fig. 23. Junction node router J1 definition

Fig. 24 defines a normal router, in this case R3, that has two interfaces. The first interface connects R3 to R2 and the second interface connects back to the junction node J1. There is no need for additional routes to be defined here because R3 is connected to the same network as R2 and there are no networks beyond R2 to which packets would be sent. Fig. 25 defines R10 which is a normal router with two interfaces on it. However, unlike R3, there are multiple networks it needs to access that are beyond R9 which is directly connected to it. Therefore, there are multiple routes to those networks as shown in the figure.

```
device=router,3
#int,IP,mask,dsfield
if=0,132.10.2.254,24,1
if=1,132.10.3.100,24,1
#no additional routes
#no proxy board here
#default route J1
r_table=0.0.0.0,24,132.10.3.254,2,1
Parms = {
name=b1r3
```

**Fig. 24. Normal router R3 definition**

```
device=router,10
#int,IP,mask,dsfield
if=0,132.10.9.254,24,3
if=1,132.10.10.100,24,3
#dest_ip,mask,next_ip,next_interface,link_type
r_table=132.10.4.0,24,132.10.9.100,1,3
r_table=132.10.5.0,24,132.10.9.100,1,3
r_table=132.10.6.0,24,132.10.9.100,1,3
r_table=132.10.7.0,24,132.10.9.100,1,3
r_table=132.10.8.0,24,132.10.9.100,1,3
#default router J1
r_table=0.0.0.0,24,132.10.10.254,3,3
Parms = {
name=b1r10
```

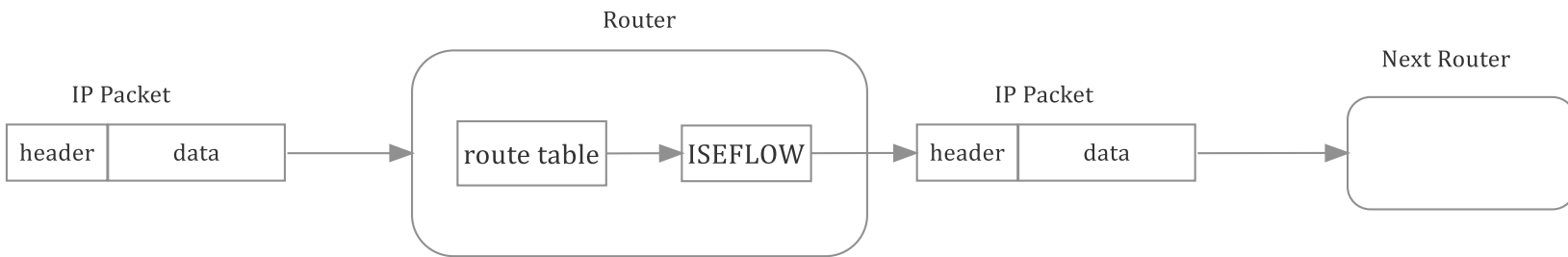
**Fig. 25. Normal router R10 definition**



## How to Handle Latency / Loss in the Network

The ISEAGE testbed and specifically the ISEFLOW routing program was not written to allow latency or packet delay. It was programmed so that a packet would be delivered through the routing tables to its final destination unless it was sent to a destination IP address outside of the testbed IP ranges on a port other than http (80), https (443), and ftp (21). If the packet was destined for the Internet on a port that was not allowed, the packet would be dropped with no response back to the client making the request. The concept of latency or delaying packets does not exist in ISEFLOW.

Because CIMoRE needs a way to allow router nodes to decline in health state to show a failing system or fail completely, ISEFLOW had to be modified to allow latency to be introduced. This benefits CIMoRE in the immediate term, but it also provides functionality to ISEAGE to expand its testing ability to more closely replicate true network traffic. To allow the delay to occur in the CIMoRE environment (and the ISEAGE testbed), a new queue was created to implement the delay. This queue was named the D\_QUEUE and is called from within the ISEFLOW program when the global link is defined as being a CIMoRE link or an ISEAGE link with delay.



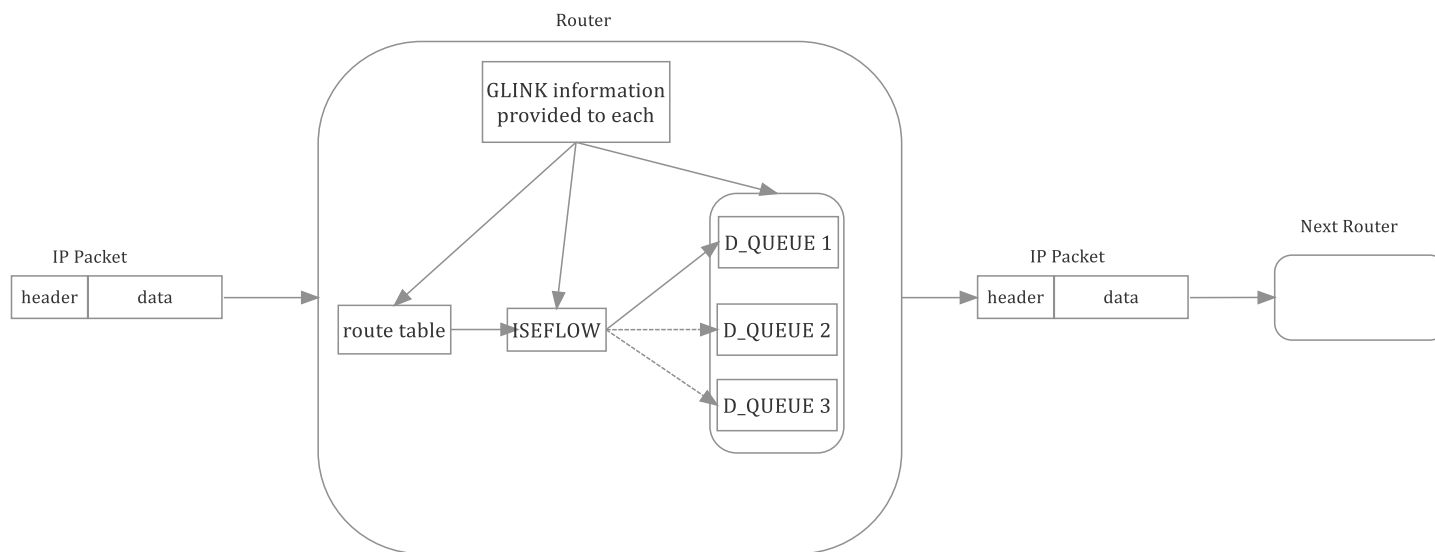
**Fig. 26. Traditional way ISEFLOW routes traffic**

The traditional way that traffic is routed in ISEAGE is depicted in Fig. 26. ISEFLOW uses the route table defined in the configuration file to setup the routes in each router. When a packet arrives on the ingress of a router, the header information is read and ISEFLOW looks at the route table for the router. Based upon the destination IP address, ISEFLOW determines which interface on the router the packet should be sent out to the next hop router or to its final destination router. Technically, since ISEFLOW is implemented in software, the router hops are a queue in which the packets are “moved” and TTL incremented with the packet being placed in for the final router to read. While the movement all happens within the queue, the packet appears to have moved through a physical network when it arrives at the final router or a destination computer.

Fig. 27 shows a high-level picture of how the addition of the D\_QUEUE has allowed the introduction of delay into the CIMoRE testbed. The IP packet still comes to the ingress of the router. However, the router now not only has route table information in it, but there is also information provided in the configuration file about what type of traffic is allowed on the interface and the connection between two routers. The global link (GLINK) information says what types of traffic are allowed on the connections and the DSCPs tell what types of traffic are allowed on the interfaces. ISEFLOW reads this information from the configuration file into

memory and also uses it in the route tables. For this dissertation the parameters provided via the GLINK parameters are maximum bandwidth, the average bandwidth, and the loss on the connection.

Also new are multiple D\_QUEUEs inside the router. One D\_QUEUE is established for each type of traffic the router will be passing. In the case of this dissertation there are three D\_QUEUEs. D\_QUEUE 1 processes traffic representing the interstate highway system. D\_QUEUE 2 handles traffic representing the ICN while D\_QUEUE 3 is present for traffic representing the electricity network. Fig. 19 depicts all three queues, but the connection from ISEFLOW to D\_QUEUE 2 and D\_QUEUE 3 is drawn as dotted lines to show that for this particular packet it is depicting interstate road traffic, but in other packets it could use the queue for ICN or electricity traffic.



**Fig. 27. D\_QUEUE logic allows delay in traffic for ISEFLOW**

The new processing logic inside the router is shown in Fig. 28. When an IP packet arrives on a router ingress, ISEFLOW reads the header for destination IP and DSCP. ISEFLOW determines whether the IP is destined for the current router or if it needs to be forwarded. If the packet needs to be forwarded, then it determines which interface it should go out. It then checks the global link information about the router to see what the type of traffic that is allowed out that interface. It then compares the global link information to the DSCP to see if the packet is for the correct type of critical infrastructure subsector. If they match, ISEFLOW then checks to see if the queue for that critical infrastructure subsector is accepting packets and what state the router is in. If it is failing or recovering the ECN value in the packet is changed to match. Then packet is added to the end of the queue for that critical infrastructure subsector type where the D\_QUEUE program will process. If D\_QUEUE has reported failure, then the packet is dropped.

At CIMoRE startup the D\_QUEUE reads the maximum bandwidth, the average bandwidths, and the loss from the GLINK values and stores the values. The D\_QUEUE then checks to see if there are any new values for traffic volume and loss to replace the values that are stored. The need for this will be explained in the sections below on traffic generation and disruptive events.

The D\_QUEUE then checks to see if the state of the link is up, recovering, failing or down based upon the maximum bandwidth, traffic

volume, and percent loss. If the state is down, then the critical infrastructure subsector queue will not accept any new packets. ISEFLOW is told of the downed state and that no packets should be sent to the queue. If the state is recovering or failing, the D\_QUEUE also sends that information to ISEFLOW. Upon receipt, ISEFLOW will change the ECN value on the packets before sending the packets to D\_QUEUE.

If the state is up, the D\_QUEUE checks the queue to see if there is anything waiting in it. If there isn't anything in the queue, then the D\_QUEUE waits for 100 milliseconds then rechecks the queue. If there is something waiting in the queue, then D\_QUEUE checks to see if the available bandwidth value is greater than or equal to the size of the packet or packet fragment. This indicates there is bandwidth available to process the packet or fragment. D\_QUEUE removes the first packet or fragment out of the queue and moves all remaining packets or fragments ahead by one position. Then, the available bandwidth variable is reduced by the size of the packet or fragment. Next, the packet or fragment is sent to ISEFLOW and it will send the packet or fragment out the egress interface. Once the packet or fragment is handed to ISEFLOW, the variable storing the available bandwidth is increased by the size of the packet or fragment.

The D\_QUEUE continuously checks the queue to see if there are packets waiting to be processed and if there is enough bandwidth to send them back to ISEFLOW to move to the next router.

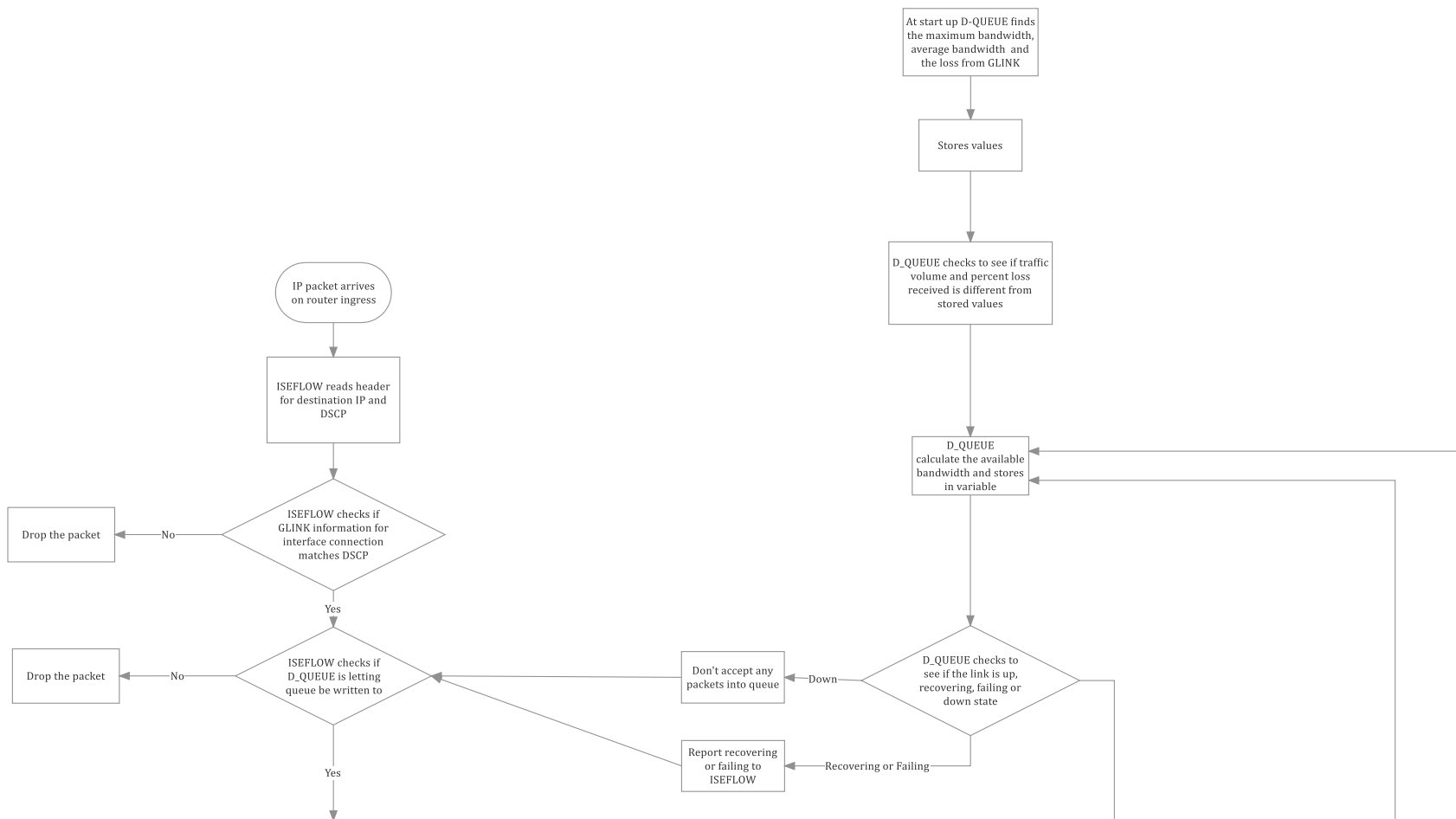


Fig. 28. Logic for processing packets in ISEFLOW with D\_QUEUE





Fig. 29. Logic for processing packets in ISEFLOW with D\_QUEUE, continued

## Generating Traffic for CIMoRE

At the present time there is neither a directly attached server nor a software process that creates a constant flow of traffic entering the ISEAGE testbed. Any traffic in the testbed is generated by the users of the testbed. In the use case of a classroom, the students either build or use servers and clients. Those machines create their own network traffic, but there is nothing introduced into the network that represents an average load on the system, outsider users, or malicious traffic.

The addition of a traffic generator is necessary for the operation of CIMoRE. CIMoRE needs a way to have routers receive increasing or decreasing amounts of traffic. The traffic generation could be implemented by adding another virtual machine to the ISEAGE testbed that generates multiple packets to represent the average traffic. However, a more efficient way to implement the traffic generator is in software that starts after ISEFLOW and D\_QUEUE are started. The traffic generator creates an IP traffic flow for each critical infrastructure subsector route defined in the configuration. ISEFLOW works at the IP layer, so I can manipulate the TCP and Application layers for my own purposes. The generation of traffic packet is discussed below. Traffic flows are created in both directions through the subsector networks when modeling critical infrastructure subsectors that have two-way traffic in the physical world. For example, roads and real network traffic have two-way traffic. In the case of electricity, the IP traffic stream would be one directional.

The traffic generator initially reads the average traffic for each critical infrastructure subsector and identifies the route for each critical infrastructure subsector path from the ISEFLOW configuration file. It generates the packet with the source IP address of the “beginning” router in the path and the destination IP address set as the “ending” router in the path. Since its software running on each board, it can directly put the generated traffic packets into the ISEFLOW queue.

Each traffic packet generated and sent to ISEFLOW will have a payload that provides the volume of traffic present on the critical infrastructure subsector network at that time. It also will have the DSCP value correctly set to represent the correct critical infrastructure subsector traffic. The ECN value will match the health state of the node. For example, in the startup state with no loss, the ECN value will be 3 which means the node is healthy.

When ISEFLOW receives the traffic generation packet, it reads the information stored in the payload for its own uses and also forwards those values to the D\_QUEUE. This is shown in Fig. 30. The D\_QUEUE then checks the values it has been sent with the average bandwidth and percent loss it has stored. If the values are different, D\_QUEUE updates the stored values. The D\_QUEUE then continues on as described earlier. Traffic volumes that are different from the average traffic and the startup percentage loss will come from the disruptive events generated in the web front-end and are discussed in the next section.



Fig. 30. Processing logic including traffic generator and disruptive event introduction

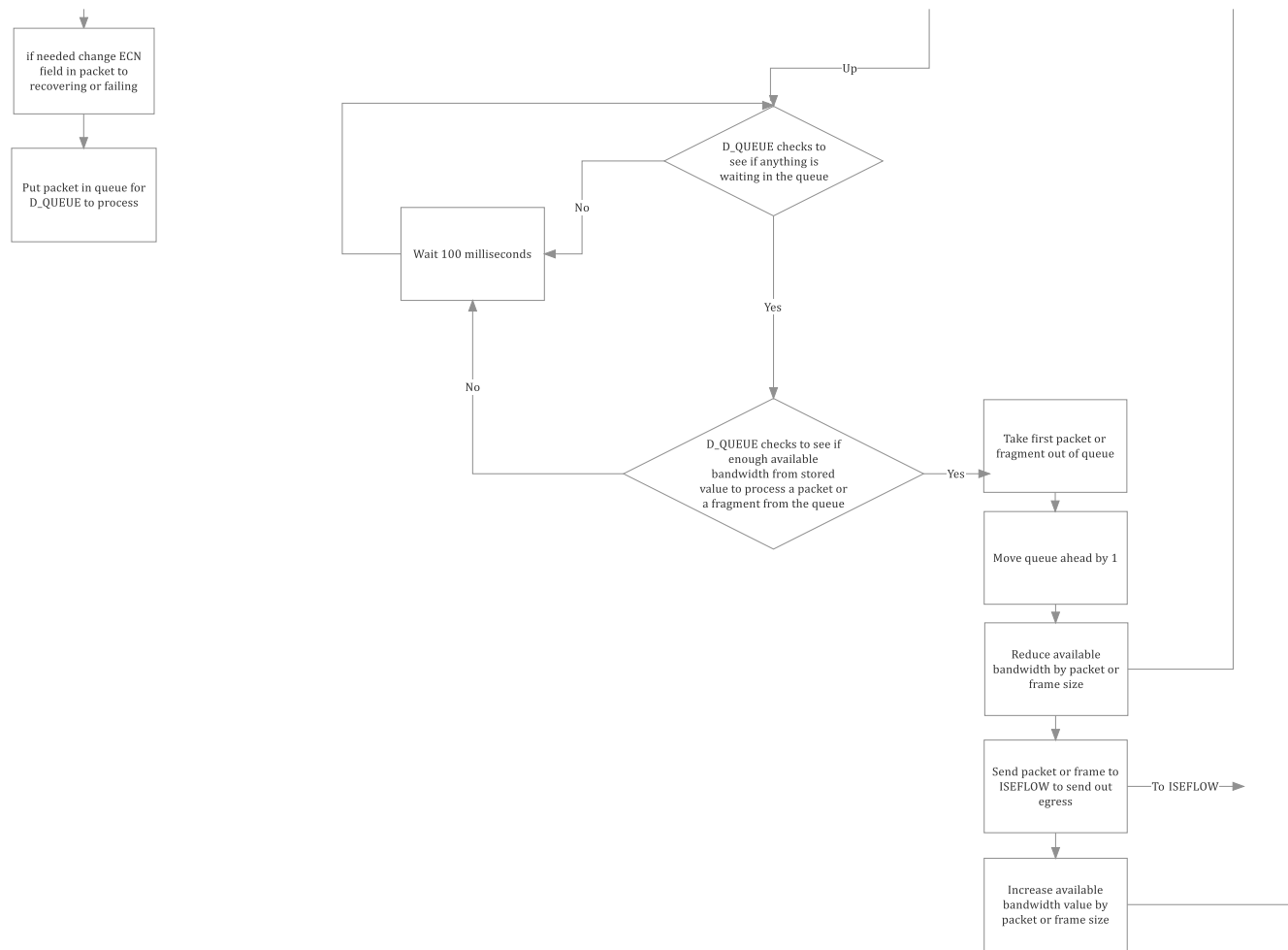


Fig. 30. Processing logic including traffic generator and disruptive event introduction, continued

## Introducing Disruptive Events

ISEAGE already has a web-based front-end to manage the restarting of the ISEFLOW on each board. This management tool lives on Snowflake and is controlled by the “developer” Scrat. Currently, the tool only displays the health of Boards 1-5, facilitates restarting the ISEFLOW on each board, allows a single board to be rebooted, or reboots all five boards. This is shown in Fig. 32 and 33.

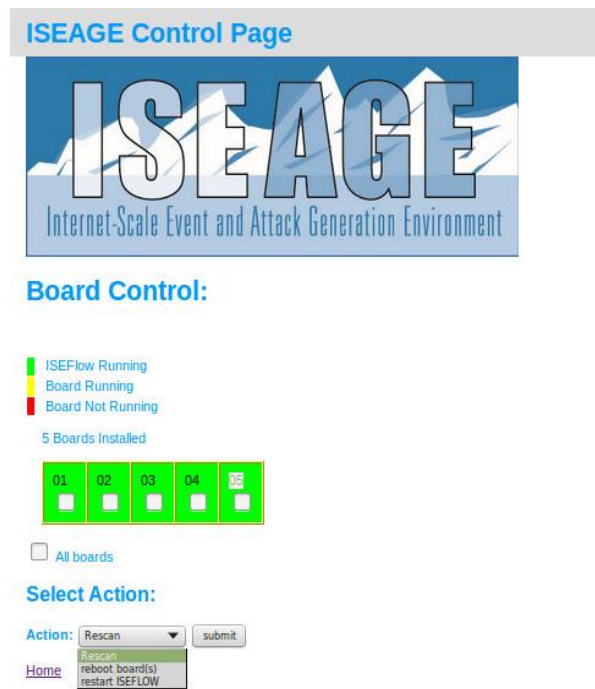



Fig. 31. Current web-based front-end to restart ISEFLOW and/or board(s)

**ISEFlow Status Page**



**ISEFlow Status:**

ISEPACK 1

Board	ISEFlow Status	Num routers IP Address	Outside Packets in	Outside Packets out	Inside packets in	Inside packets out
1	running	3	14727237	14697490	13901143	13907389
		64.39.3.254/24	00:2883750	00:1396075		
		33.96.5.254/24	01:323508	01:464281		
		190.100.60.254/24	02:726805	02:717730		
		104.190.100.254/24	03:1032398	03:1336742		
		61.14.60.254/24	04:650828	04:705367		
		2.104.163.254/24	05:628527	05:774585		
		64.5.53.254/24	06:705085	06:814808		
		52.135.80.254/24	07:1471241	07:732950		
		79.5.64.254/24	08:728134	08:845192		
		49.49.33.254/24	09:822697	09:1203862		
		42.49.30.254/24	10:780370	10:745542		
		73.54.41.254/24	11:936803	11:1065023		
		84.56.33.254/24	12:1452056	12:1672845		
		87.45.75.254/24	13:533449	13:712112		
		200.35.23.254/24	14:1051586	14:1510376		

**Fig. 32. Current web-based front-end to show the status of the end node routers**

The web-based front-end already uses the ISEFLOW configuration file to display health information about the end node routers (outside routers in ISEAGE terminology) on each of the boards and issues the command to restart ISEFLOW and the boards. It would be relatively simple to modify this tool to create disruptive events which would modify the values that ISEFLOW has for each router and then to issue the restart ISEFLOW and/or restart the board(s) command.

### Disruptive Event Page



### Disruptive Event:

Board	Health Status	Router Number	Interface	DSCP	Traffic Volume	Traffic Volume Reference	Percent Loss
1	Green	CT1	0	63	--	--	--
	Green		1	63	--	--	--
	Yellow	R1	0	3			
	Green		1	3			
	Green	D1	0	63	--	--	--
	Green		1	3			
	Green		2	1			
	Green		3	3			
	Green		4	2			
	Green	R3	0	1			
	Green		1	1			
	Green	R2	1	1			
	Green	R10	0	3			
	Orange		1	3			
	Red	R9	0	3			
			1	3			

Fig. 33. Prototype of disruptive event web tool

As shown in Fig. 34 the web tool will be modified to have a visual depiction of each interface on each router. The visualization will be simple. They will be colored cells. The green cell stands for healthy, the orange cell for failing and going down, the yellow cell for recovering from failure, and the red cell for completely failed. The web front-end also has two text boxes behind each interface of each router. The first text box will allow the entry of a numeric value for traffic volume. The average volume, as well as traffic volume thresholds for up, down, failing, and recovering will be listed so the user has an idea of how to change the traffic volume for the desired disruptive event. The second text box only



allows a numeric value between 0 and 100, inclusive. This is the percentage of packet loss.

Once the values are filled out in the web tool and the submit button is pressed, the web tool sends the new values for traffic volume and percent loss to the traffic generator as shown in Fig. 30. The traffic generator then writes those values to be used for all future packets until it receives a new change notice. Once it records these new values, it creates a special traffic generation packet using the new values and sends it to ISEFLOW.

ISEFLOW takes two actions. First, it forwards the packet to D\_QUEUE. It is at this point in the D\_QUEUE process where the value D\_QUEUE has stored for traffic volume and loss are compared to the new incoming values and these values are written for future use and also used to calculate available bandwidth.

Second, ISEFLOW modifies the running version of ISEFLOW to include the new values for the router interfaces in the traffic generation packet. ISEFLOW also recognizes any routers that are downstream of the affected router and changes their values in the running configuration as well.

While the traffic generator continues to put normal IP packets into the “beginning” router with the ECN value set at 3 meaning healthy, as soon as the packet arrives at a failing node, the packet is rewritten by ISEFLOW to include a 1 as the value in the ECN field so that anyone

observing the traffic will see the router beginning to fail. While these packets are not used in this dissertation, they could be used in the future for a visual map of the system to show the nodes failing to the anyone observing the modeling, but not in control of manipulating the disruptive event.

## CHAPTER 7. RUNNING CIMORE

There are five different states of CIMoRE. Each of them are defined below.

### Initial Startup

The initial startup state begins when ISEFLOW is started and it reads the configuration file. The network is built with all routers correctly identified and “physically” connected to the appropriate interfaces. The average and maximum bandwidth, as well as the percentage loss, has been set on each router interface. For this discussion, the loss in the initial startup state is 0. The different global links could be set with a loss at startup to simulate the critical infrastructure subsector nodes in a failed state at startup. However, starting with no loss allows me to discuss each state of CIMoRE. The D\_QUEUE is started after ISEFLOW is running and before the traffic generator. Once ISEFLOW and D\_QUEUE have been started, the routers are sitting idle waiting for traffic to arrive. Next, the traffic generator software is started and begins putting traffic information packets into the CIMoRE network. The traffic packets have average traffic volume and 0 percent loss.

### Steady State

Once the first traffic generation packet has been received by the “end” router for each critical infrastructure subsector in the defined path, CIMoRE is considered to be in a steady state. In steady state, normal IP

traffic which could be generated by end users or devices connected to end node routers can start sending their traffic. Also, the traffic generator continues to put traffic packets into each critical infrastructure subsector network. Again, those packets represent the average traffic volume and that they are operating as healthy nodes. This can be verified by looking at the web front-end which has a colored square indicating that each node is green. It also can be observed by doing a tcpdump and viewing the DS field for DSCP and ECN values.

### **Introduction of Disruptive Events**

For right now the introduction of a disruptive event is a manual process using the web tool described above. Once the traffic volume and/or a loss percentage is entered and the submit button on the web front-end is pressed, the new values are sent to the traffic generator. The traffic generator then stores the new values and crafts a special traffic packet that includes the new traffic volume and percent loss values. The traffic packet is sent to ISEFLOW. ISEFLOW forwards the information to the D\_QUEUE where it is used to recalculate available bandwidth. ISEFLOW also modifies the running version of itself to include the new values for the router interfaces in the traffic generation packet. ISEFLOW also recognizes any routers that are downstream of the affected router and changes their values in the running configuration as well. The colored square on the web front-end will reflect any changes to the health state of the router interface.

In the future it will be possible for a disruptive event to be introduced automatically from the reading of a file that stores data about a series of disruptive events. This idea is further discussed in the future work section of this dissertation.

### **Failure**

There are two ways to achieve complete failure for a node. The first as described in the section on traffic generation. It would rely on introducing traffic generation packets that would reduce bandwidth on the route. If enough packets get delayed it can cause the D\_QUEUE in the router to stop accepting packets. Therefore, the node has failed because arriving new packets in ISEFLOW cannot be serviced.

The second way of failure would be selecting 100 percent loss in the traffic generator web front-end and pressing submit. This would have ISEFLOW reload the running configuration with the traffic volume equaling the maximum bandwidth. This is the kind of failure that would be introduced with the electricity subsector. The routers either are up or down in the electricity subsector. There is no percentage loss in an electricity subsector network.

### **Recovery**

As with failure, there are two ways to recovery for a router. The first would be if the D\_QUEUE begins to catch up with the IP packets being sent. Because it stopped accepting packets, it could then process all it has in the queue and begin to signal ISEFLOW that it can now send

packets again to it. This would mean that the traffic generator's web front-end would display the color of yellow showing a recovery mode. Also, the normal network traffic that moves beyond the recovering node would be marked with a 2 to show that the state is recovering, but still not operating at a healthy state. The D\_QUEUE would begin allowing normal network packets to be submitted to it again as well.

The second way to recover is by marking the router as fully recovered in the traffic generator web front-end and pressing submit. Once the submission is made and ISEFLOW is restarted the router would operate in a healthy state.

## CHAPTER 8. CONCLUSIONS AND FUTURE WORK

A fully functioning CIMoRE is a very large project that falls well outside the scope of a dissertation project. The first step in any very large project is the creation of a road map of how the project could be undertaken. That was the purpose of this dissertation. My contribution to the CIMoRE project is trying to determine if the project really is possible and then creating a framework in which development could occur. This dissertation recognized many problems with using the ISEAGE testbed and identified solutions for how to overcome the problems. I also outlined development that needs to be completed in ISEAGE to allow CIMoRE to operate. The implementation of many of the items I outlined could become their own Masters level implementation project and move CIMoRE toward full functionality.

Specifically, in this dissertation I demonstrated that critical infrastructures subsectors can be analyzed and pieces of those subsectors in the physical world can be used to create a network representation using TCP/IP networks. Additionally, there are physical characteristics of the subsectors in the real world that can be used as a proxy for IP traffic. Further, once enumerated, the interdependencies (or relationships) between critical infrastructure subsectors can be translated to network terms and modeled in the TCP/IP testbed.

All of these transformations would have been significantly easier if cooperators in each of the three critical infrastructure subsectors could

have been found to provide data and help identify key characteristics of each critical infrastructure subsector. However, my proof of concept data was a legitimate substitute.

While generating data for three critical infrastructure subsectors and determining how to create network nodes and traffic was thought provoking, the largest number of tasks and the most challenging was to determine all the modifications that needed to be made in ISEAGE to allow CIMoRE to operate. First, ISEAGE had to understand that in this dissertation the traffic in the testbed was comprised of three different types of service. One type of service would run for each critical infrastructure subsector in the test. That was accomplished by using the Differentiated Service field.

Next, the number and function of routers used in ISEFLOW, the internal programming of ISEAGE, need to be expanded. In addition to the two existing router types of normal router and edge node router, two more types of routers were defined: connector node router and junction node router. Connector node routers allow the implement of the functionality of a switch in ISEAGE while the junction node routers allow interdependencies to be model. Specifically, in this dissertation the geospatial interdependency was used.

Historically ISEAGE has only used two or one normal routers between the Backplane and the end node routers where the students work. Because of the sheer number of routers in the three critical



infrastructure subsectors and the introduction of two new types of routers, a full configuration file was created. To demonstrate how each of the types of routers were used detailed examples from the full configuration file were included in the dissertation.

Further, ISEAGE has never handled latency in a network. Packets were routed without any delay. However, for CIMoRE it was necessary to introduce delay into packet delivery. To handle delay, the logic for a new piece of software to be called from ISEFLOW was developed. This delay queue was named D\_QUEUE.

Traffic generation is also a missing feature in the current implementation of ISEAGE. The ability to increase or decrease the amount of traffic sent to a network path allows the D\_QUEUE to introduce delay into the network. The traffic generation also allows ISEFLOW to modify its running configuration so that packets are marked with ECN codes of failing or recovery, as well as routers can be marked as in total failure. The modification of an existing web tool that monitors and restarts ISEAGE routers and boards allows the introduction of disruptive events. Coupled with the traffic generator, the disruptive event can trigger a failure in routers based upon information sent by ISEFLOW and D\_QUEUE.

Finally, after the modifications to ISEAGE were outlined, the five running states of CIMoRE were explained. How to use each of the new pieces of ISEAGE, as well as how to introduce disruptive events, how to fail a node, and how to recover were included.

The framework presented in this dissertation is just the beginning of much more work for other graduate students, as well as myself, before a useful tool for critical infrastructure subsector modeling can be provided for disaster planners and table top exercises. The most immediate work that should be undertaken is the writing of all the modifications to ISEAGE that I outlined. Without those modifications, it does not matter how many cooperators and real data sets are collected for use. In tandem with the ISEAGE modifications, making contacts and circulating a white paper on CIMoRE's proof of concept would help in earning cooperator's trust and, hopefully, real data.

Another step in CIMoRE's development would be to add a fourth critical infrastructure subsector and to do the work to identify its physical components that could be turned into a network representation and network traffic. A logical next step critical infrastructure subsector might be drinking water systems or wastewater treatment systems. It would need to be determined if one or both of these function like the electricity subsector, the roads, or information technology subsector, or if they are unique and have their own values that need to be established.

While only the geospatial interdependencies were considered in this dissertation, it would seem wise to look at another of the interdependencies to see if they could be modeled using the junction node router and what that traffic would look like. They may fit seamlessly into

the existing framework, but there could be additional modifications needed in ISEAGE or to the logic of handling junction node router traffic.

All of the creation of files and data for this dissertation was generated by hand. One of the major drawbacks of implementing more than the small configuration used in this dissertation is the need for elaborate networking schemes and decisions about network connections. The setup and implementation of a CIMoRE model in ISEAGE needs to be automated. There needs to be a way that data file garnered from the cooperators could be read in and the determination of whether the node is a normal router, an edge node router, a junction node router, or a connector node router would be based upon the fields extracted from the cooperator's database.

Likewise, the generation of disruptive events is a manual process at this time. However, in the future a file with the values needed for the traffic generator could be loaded into ISEAGE and the web front-end tool would just provide secondary management of the routers and display of the health states. This would allow the replay of disruptive events from a file or a series of events to be played out over time. The reading of the data stream could even be a historical recording of an event or data coming from real-time from sensors in the physical world. CIMoRE could also include a recording of traffic feature so that exercises could be replayed and evaluated for the decisions made. And, CIMoRE could provide information about alternate path for rerouting traffic. As nodes fail,

CIMoRE could forecast alternate paths and provide a calculated best fit alternative.

The determination of the interdependencies, in this case geospatial, needs to be implemented in software. The longitude and latitude values are available, but the overlaying of the state map and the locating of the nodes in the same cell should be automated. The enumeration of other interdependencies should also be automated with software.

A very far out consideration is to make ISEAGE IPv6 compliant. This was not discussed in the dissertation because this is very far afield from the scope of this project. However, as the world continues to move toward IPv6 or at least a world that runs both IPv4 and IPv6, it would be prudent for ISEAGE have IPv6 be added as an option to be enabled when the testbed is being used.

One final consideration for future work is that when modeling more than three critical infrastructure subsector paths, many of the routers will no longer be normal routers. In fact, they may become junction node routers. And, those junction node routers may make be so interconnected that CIMoRE becomes a mesh network. The addition of using a mesh network may require a few additional modifications to the ISEAGE code that I have not considered for this dissertation, but may need to be revisited in the future.

Converting critical infrastructure subsectors to TCP/IP networks is a novel concept in its own right. However, not only is CIMoRE a novelty, but

when fully implemented it has the potential to be a useful modeling tool for critical infrastructures subsectors and their interdependencies. There is much development left to do to get to this point. I'm proud to say this dissertation laid the groundwork for such a tool.

## REFERENCES

- [1] (2001). *U.S.A. Patriot Act*. Available: <http://epic.org/privacy/terrorism/hr3162.html>
- [2] (2013). *National Infrastructure Protection Plan 2013: Partnering for Critical Infrastructure Security and Resilience*.
- [3] (2017, June 6). *Critical Infrastructure Sectors: Transportation Systems Sector*. Available: <https://www.dhs.gov/transportation-systems-sector>
- [4] (2017, June 6). *Critical Infrastructure Sectors: Energy Sector*. Available: <https://www.dhs.gov/energy-sector>
- [5] (2017, January 15). *Homeland Security Energy Sector Overview*. Available: <https://www.dhs.gov/energy-sector>
- [6] S. M. Rinaldi, "Modeling and Simulating Critical Infrastructures and Their Interdependencies," in *37th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, 2004.
- [7] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *Control Systems Magazine, IEEE*, vol. 21, no. 6, pp. 11-25, 2001.
- [8] US Department of Energy,. (2006). *INL/EXT-06-11464, Critical Infrastructure and Interdependency Modeling: A Survey of US and International Research*.
- [9] citygate GIS. (August 11, 2008). *E-MAPS Computer Aided Emergency Management and Planning*. Available: <http://www.citygategis.com/emaps.htm>
- [10] roadtraffic-technology.com. (August 11, 2008). *PTV - Software, Consulting and Research for Traffic and Transportation Planning*. Available: <http://www.roadtraffic-technology.com/contractors/it/ptv/>
- [11] EnvironmentalExpert.com. *floodFILL - GIS-based Modeling of Floods*. Available: [http://management.environmental-expert.com/STSE\\_resultteach\\_product.aspx?cid=3695&idprofile=262&idproduct=36324](http://management.environmental-expert.com/STSE_resultteach_product.aspx?cid=3695&idprofile=262&idproduct=36324)
- [12] R. Schainker, P. Miller, W. Dubbelday, P. Hirsch, and Z. Guorui, "Real-time dynamic security assessment: fast simulation and modeling applied to emergency outage security of the electric grid," *Power and Energy Magazine, IEEE*, vol. 4, no. 2, pp. 51-58, 2006.
- [13] I. G. Sloman and L. Benedicenti, "Displaying the state of an electric system: a preliminary study," in *Electrical and Computer Engineering, 2005. Canadian Conference on*, 2005, pp. 861-864.
- [14] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -- Federate Interface Specification," *IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000)*, pp. 1-378, August 18 2010.
- [15] M. Ficco, G. Avolio, L. Battaglias, and V. Manetti, "Hybrid Simulation of Distributed Large-Scale Critical Infrastructures," in *International Conference on Intelligent Networking and Collaborative Systems*, Salerno, Italy, 2014.

- [16] I. Eusgeld and C. Nan, "Creating a simulation environment for critical infrastructure interdependencies study," in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pp. 2104-2108.
- [17] E. Castorini, P. Palazzari, A. Tofani, and P. Servillo, "Ontological Framework to Model Critical Infrastructures and their Interdependencies," in *Complexity in Engineering, 2010. COMPENG '10.*, 2010, pp. 91-93.
- [18] T. Okathe, S. S. Heydari, V. Sood, and K. El-Khatib, "Unified multi-critical infrastructure communication architecture," in *2014 27th Biennial Symposium on Communications (QBSC)*, 2014, pp. 178-183.
- [19] "IEEE Standard for Distributed Interactive Simulation (DIS) -- Communication Services and Profiles," *IEEE Std 1278.2-2015 (Revision of IEEE Std 1278.2-1995)* pp. 1-42, November 6 2015.
- [20] D. D. Dudenhoeffer *et al.*, "Interdependency modeling and emergency response," presented at the Proceedings of the 2007 summer computer simulation conference, San Diego, California, 2007.
- [21] D. D. Dudenhoeffer, M. R. Permann, and M. Manic, "CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis," in *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, 2006, pp. 478-485.
- [22] C. Wang, L. Fang, and Y. Dai, "National Critical Infrastructure Modeling and Analysis Based on Complex System Theory," in *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*, 2011, pp. 832-836.
- [23] A. D. Giorgio and F. Liberati, "A Bayesian Network-Based Approach to the Critical Infrastructure Interdependencies Analysis," *IEEE Systems Journal*, vol. 6, no. 3, pp. 510-519, 2012.
- [24] S. Puuska, K. Kansanen, L. Rummukainen, and J. Vankka, "Modelling and real-time analysis of critical infrastructure using discrete event systems on graphs," in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2015, pp. 1-5.
- [25] K. E. Lever, M. Á, and K. Kifayat, "Evaluating Interdependencies and Cascading Failures Using Distributed Attack Graph Generation Methods for Critical Infrastructure Defence," in *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, 2015, pp. 47-52.
- [26] "GIMS Data," I. D. o. Transportation, Ed., ed, 2010.
- [27] (2010). *Video Sites, Backbone, and Leased Connections for the ICN*. Available:  
[http://icn.iowa.gov/sites/default/files/documents/ICNroomsbyTownNov2010\\_s.pdf](http://icn.iowa.gov/sites/default/files/documents/ICNroomsbyTownNov2010_s.pdf)
- [28] (June 25, 2013). *Iowa Communications Network Frequently Asked Questions*. Available: <http://www.icn.iowa.gov/about-icn/frequently-asked-questions>
- [29] (2013, August 13, 2013). *Openstreetmap*. Available:  
<http://nominatim.openstreetmap.org/>

- [30] J. A. Rursch and D. Jacobson, "This IS child's play - Creating a playground (computer network testbed) for high school students to learn, practice and compete in cyber defense competitions," in *Frontiers in Education*, Oklahoma City, OK, 2013.
- [31] J. A. Rursch and D. Jacobson, "When a testbed does more than testing: The Internet-Scale Event Attack and Generation Environment (ISEAGE) – providing learning and synthesizing experiences for cyber security students. ," in *Frontiers in Education*, Oklahoma City, OK, 2013.
- [32] "RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers," *Internet Engineering Task Force*, December 1998.
- [33] "RFC 2475 An Architecture for Differentiated Services " *Internet Engineering Task Force*, December 1998.
- [34] "RFC 3260 New Terminology and Clarifications for Diffserv," *Internet Engineering Task Force*, April 2002.



## APPENDIX. OTHER CRITICAL INFRASTRUCTURES

This appendix provides two tables to show which of the 16 critical infrastructure sectors could potentially be modeled in the future using CIMoRE. CIMoRE requires critical infrastructure subsectors to have characteristics that can be used to create a network. Because not all 16 of the critical infrastructure sectors have clearly delineated processes and procedures some cannot be depicted as networks. Therefore, the 16 critical infrastructures were categorized into one of three types; yes, maybe, and no.

### “Lifeline” Sectors - Yes

There are six critical infrastructure sectors in the U.S. that are categorized as “lifeline” sectors. These sectors focused on public health and safety issues. These critical infrastructure sectors include the three used in this dissertation: transportation systems, information technology, and energy. In addition, water and wastewater systems, communications and healthcare and public health are included in the “lifeline” critical sectors. Of these sectors, CIMoRE can model all but healthcare and public health. And, if looking closely at the specific subsector of electronic records and information movement in healthcare and public health, CIMoRE could fit all of the “lifeline” critical infrastructure sectors.

**Table 4. "Lifeline" critical infrastructure sectors – yes category**

Sector	Subsector	Network	Network Segments	Devices	Bandwidth	Loss
Communications	Data	Fiber loop	City	Router	Actual bandwidth	Percentage
Energy	Electricity	Transmission line	City	City	Line voltage	On/off
Information Technology	Identify management / Trust systems	Trust support systems	Trusted partners	Validation objects	Actual bandwidth	On/off
Transportation Systems	Highway systems	Roads	Link segment	Interchanges	Volume of traffic	Percentage
Water and Wastewater Systems	Drinking water treatment facility	Treatment process	Movement from raw water storage to distribution	Where the product changes	Volume of water treated per hour or volume of water distributed per hour	Percentage
Healthcare and Public Health	Electronic health records	Provider networks, pharmacy networks	Physical office locations	Medical records systems	Number of records or number of transactions	Percentage

### **“Manufacturing-like” Sectors - Maybe**

The second category of critical infrastructure sectors are sectors that start with a raw product, go through some number of processes to end with a final product. Although the chemical and critical manufacturing sectors easily fall into this category, the food and agriculture sector could also be considered “manufacturing-like” if I consider seeds, fertilizer, and chemicals as the input with a raw product as the harvested crop and the final product the cereal or food product created from the crop. The “manufacturing-like” sectors may be able to be modeled by CIMoRE if it models the movement from input to final product.

Similar to the flow of a raw product through a manufacturing process is the information sharing that occurs in critical infrastructure sectors such as the defense industrial base and the emergency services. The movement of information being shared among divisions or branches of each of these critical infrastructure sectors could potentially be modeled by CIMoRE with some more detailed knowledge of how messages flow.

**Table 5. "Manufacturing-like" critical infrastructures – maybe category**

Sector	Subsector	Network	Network Segments	Devices	Bandwidth	Loss
Chemical	Agricultural chemicals	Raw materials into final product	One form to another	Custody change, sale, refinement	Volume of raw materials, transported, capacity of manufacturing line	Percentage
Critical Manufacturing	Primary metals manufacturing	Same as above	Same as above	Same as above	Same as above	Same as above
Food and Agriculture	"farm to fork"	From seed to harvest to refinement	Same as above	Same as above	Same as above	Same as above
Defense Industrial Base	Information sharing among divisions	Division or department	One division to multiple divisions	Centralized communication for division	Bandwidth, people	Percentage
Emergency Services	Information sharing among services	Service or department	Firemen to police to medical	Centralized communication for service department	Bandwidth, people	Percentage

**Other – No**

The remaining critical infrastructures do not have characteristics that allow them to be modeled as a network. The remaining five are commercial facilities, dams, financial services, government facilities, and nuclear reactors, material, and waste.