

A Resilient Network that Can Operate Under Duress: To Support Communication between Government Agencies during Crisis Situations

Sanjay Goel[†], Salvatore Belardo[‡], Laura Iwan[‡]
School of Business, University at Albany, State University of New York[‡]
{goel,s.belardo}@albany.edu,
NYS Office of Cyber Security & Critical Infrastructure[‡]
laura.iwan@cscic.ny.state.us

Abstract

The work in this paper is motivated by the weaknesses in communication networks that were observed among government agencies while responding to the emergency situation posed by the attack on the World Trade Center. The paper proposes a self-healing and self-managing architecture for supporting electronic communication between government agencies in crisis situations when the communication infrastructure is partially disabled. The architecture that we propose consists of independent services with standard interfaces and variable addresses. The services discover each other as required in real time by matching the standard interfaces. Disabled services are automatically pruned from the network and new services seamlessly replace the existing services at alternate network nodes. Complex operations can be performed using these services by integrating the services into existing workflows. The architecture allows for redundancy in the system as well as for requisitioning of additional services when the performance degrades due to a higher than normal load, which causes duress in the system. The paper presents the architecture for such a system as well as a model for simulating such a system under various scenarios of duress.

1. Introduction

The Internet and various computer networks have grown at a rapid pace over the last few years. They have become integrated into corporate and business culture to such an extent that businesses today would be unable to function without them. Day to day operations in companies rely on seamless integration of computers across the enterprise. In the public sector, state and federal governments, albeit slowly, have embraced networked computing in a large way. Critical government operations now depend on use of networked computers.

Some of the more visible examples of the use of computer networks in the government include the electronic tax return processing employed by the Internal Revenue Service (IRS), automated prescription processing in government hospitals, real-time criminal data access by law enforcement agencies for screening offenders, and financial processing for payroll, benefits and procurement by all the agencies.

The productivity of both government and the industry has improved significantly due to the seamless integration of data, and operations, however; the added reliance on networked computers has increased our vulnerability to disruptions of these networks. Most often organizations plan for small disruptions to their networks by having backups and alternate servers to recover and process data, however, most do not plan for large-scale (and long-term) disruptions of the network. The effects of disruption to the network were clearly demonstrated by the Denial-of-Service (DOS) attacks that were launched on high profile sites like Microsoft, Yahoo and AOL in 1999. In these attacks, the resources of the server were swamped with fake requests denying access to legitimate users. These attacks caused over 200 million dollars worth of lost revenue for Yahoo, Microsoft, and AOL in 1999 [1].

Computer Networks were as vulnerable a few years ago as they are today, however, our increased dependence on computer networks for performing critical functions makes security even more vital today. The vulnerability of our networks stems from a variety of sources including hacker attacks on the network, sabotage, destruction of physical networks, natural calamities and unexpected surges in the network traffic following an extraordinary event.

1.1 Threats Persisting Today

Hacking attacks have been a major source of disruptions in computer networks and loss of data

integrity. Hackers have clearly demonstrated that any computer system is vulnerable to attacks and that security features make hacking harder, but not impossible. This does not imply that network security is not useful because it does act as a strong deterrent, rendering a security breach less likely, however, effective mechanisms for recovery from network disruptions and loss of data integrity are essential for ensuring security of networks and information. Some of the commonly used techniques for network penetration include spoofing, session hijacking, buffer overflows and password based attacks. In addition, hackers can also launch virus and worm attacks on the networks without hacking into the network. *The Code Red, I Love You, and Nimda* are examples of viruses and worms [2] that have been released on the network resulting in serious financial and productivity loss for individuals and businesses. Physical destruction of networks is the next most likely cause of network disruption. It can be caused by a variety of sources like terrorist attacks, sabotage, floods, earthquakes and fires. If key routers or cables are destroyed the network can be disabled for long periods of time. This was clearly demonstrated in the World Trade Center attacks in September 2001 that caused the Metropolitan Area Network in New York city to become disabled when the 10 Gbits/s fiber optic ring passing under the WTC towers was severed in two places. This network was not only critical to the businesses connected to the ring within the metropolitan area, but also critical for routing traffic across the Atlantic. The destruction, coupled with a spike in network demand by users trying to access news and information from the Internet, resulted in a severe performance degradation of the network. Problems, such as these, are further exacerbated when critical emergency functions depend on the network communication itself.

Network communication can also be disrupted without any serious damage to the computing and network infrastructure. This was clearly demonstrated by the 1998 ice storm in upstate New York when extended power outages and inability of critical staff to manage operations in computer centers resulted in serious long-term disruption of network communication. State agencies need to be operational during serious crisis and thus need to plan for such contingencies.

1.2 Why do These Threats Need to be Taken Seriously?

When computer networks were conceived, the nodes of the network were in close vicinity of each other and were insulated from the outside network. Also, the Internet at its inception was used as a means of accessing information, exchanging email messages and transferring data asynchronously. Computer networks today are

geographically distributed across the world, and private enterprise as well as governments depend on their network for sharing data, as well as conducting business transactions via synchronous operations among multiple entities scattered across the enterprise. Business-to-Consumer (B2C) commerce, which most people believe is the primary commercial leverage of the Internet, is a miniscule portion of the Internet based trade. The primary driver has become business-to-business (B2B) commerce. *Jupiter Media Metrix* [3] report claims that online retail sales (B2C) will reach \$104 billion in 2005 and \$130 billion by 2006, up from \$34 billion in 2001. According to a *Gartner Group* report [4], B2B commerce will grow from \$919 billion dollars in 2002 to 8.3 trillion dollars in year 2005. Statistics on growth of business-to-government and inter-government communication via the Internet are not yet available, but the potential for growth is tremendous. This can be clearly demonstrated in the case of Department of Motor Vehicles (DMV). In New York State, DMV services such as licensing, registration and transfers can be done online. Millions of customers each year access these services. Even if only a part of the transaction is done online the revenue generated in the DMV itself will exceed 50 million dollars. Procurement worth trillions of dollars is done by different government agencies. It is estimated that over the next 5 years electronic transactions in the government sector can reach over one trillion dollars.

Government agencies are already beginning to use computer networks to share data and resources across agencies and to provide web-based services to their customers. There is not only a potential for efficiency improvement, but also for enabling services that are not possible today. For instance, law enforcement agencies, hospitals, and weather departments need to share data in order to respond to natural disasters such as earthquakes and tornados. Similarly, the Federal Bureau of Investigation (FBI), Internal Revenue Service (IRS), and Homeland Security Office may need to share data to track the flow of money intended for terrorist activities. For the efficient functioning of the government and to generate a coordinated response to disasters, a smooth unfettered interaction between the different government agencies is essential. Numerous efforts are underway to coordinate the networking of different government agencies. The Federal Bridge Certification Authority [5] is running a project to allow easy sharing of secure data between different federal agencies in the U.S. and law enforcement agencies in other countries. New York State has built a high-speed fiber optic link to connect all state agencies through out the entire state via the Internet.

The paper proposes a distributed architecture that can address the security and reliability problems that

government agencies face today. The paper discusses the threats posed to government agencies, describes a security architecture for defending against those threats and presents some planned simulations to evaluate the performance of the new architecture. The paper is organized as follows: Section 2 discusses the security threats posed to the electronic communication among government agencies and the network requirements during crisis situations. Section 3 describes peer-to-peer systems and the service-based architecture that forms the basis of the proposed architecture and reviews some literature in the area. Section 3 describes the network currently used by New York State agencies. Section 4 describes the proposed architecture for supporting critical services on the NY state network. Section 5 describes the simulations designed to enable evaluation of the effects of disruptions to the New York State network. The paper is concluded by a brief discussion of the observations made in this work and plans for future work.

2. Threats posed to Government Networks

The last few years have brought unprecedented threats to all government institutions as well as the critical national infrastructure. Faced with a potential increase in threat levels that can cause increased load and diminished capacity of networks, State governments and the Federal government must accept the challenge of finding ways to ensure that the network operations and data integrity are maintained under such duress. One of the primary concerns of these governments is a breakdown in communications across different agencies. The increased interdependence of our information systems makes this a greater possibility than ever before. Our vulnerability in the area of critical infrastructure and services was exposed by the terrorist attacks on September 11, 2001. These attacks led to a large-scale disruption of the network and resulted in the failure of critical operations. One consequence of the network breakdown was that twenty-six psychiatric hospitals in New York City in which all the operations were automated (i.e., prescription filling, patient records availability, billing and monitoring of patients) were unable to function properly for months after the attack. These hospitals were networked together and relied on centralized software and operations. When they lost their network connection, there was total chaos; They had no contingency plan or procedures to handle prescriptions manually, nor were they able to activate a backup system that could have gotten the hospitals operational in a short time period.

This breakdown also led to fraudulent financial activity by opportunistic criminals, as was evident from the Municipal Credit Union case. During the confusion that followed the attacks on the WTC, enterprising

criminals exploited the lack of checks and balances in financial transactions stealing more than \$15 million from the Municipal Credit Union. Following the attacks on the Municipal Credit Union, communications with automatic teller machines were disabled and the credit union lost its link to the New York Cash Exchange (NYCE). As a result there was no way of verifying account balances of the ATM users. The credit union, however, allowed NYCE to continue dispensing cash to the ATM machines without verification of account balances in order to maintain trust in the financial institutions. Investigations are still undergoing in an effort to identify perpetrators of the fraud.

Another example of problems that result when the network linking interdependent critical data is compromised, is, the SQL slammer worm attack. The unintended consequence of this attack was a denial-of-service caused by network congestion that resulted from the proliferation of the worm on the network. The congestion interrupted the normal functioning of several airlines because of online ticketing and check-in problems, shut down 911 service in a community outside Seattle Washington (they were using voice over IP) and rendered unusable, ATM services for a large bank (the backend databases could not be accessed due to the congestion).

2.1 Network Needs to Support Critical Applications

In order to minimize the impact of various network breakdowns, networks need to be resilient, scalable and quickly reconfigurable. Unfortunately, existing networks are not robust and require high levels of maintenance. A catastrophic large-scale failure can take weeks and even months to recover from. As the software applications on the network have become more distributed and complex, the effort involved in supporting the applications on the network has increased significantly. This has resulted in decreased reliability of the applications and exposed a large number of security vulnerabilities. The work on making the network and computing infrastructure more secure and reliable has been piecemeal at best; patches are added repeatedly to an inherently unreliable system as new vulnerabilities are encountered creating a patchwork of bloated software bursting at its seams. Given our increased dependence on the availability of networked computing (and the Internet), coupled with their vulnerability and the increasing costs of supporting the infrastructure, it is time to rethink the architecture of the network.

An intelligent, self-configuring, self-healing, and self-managing architecture is required such that each node is autonomous, and capable of coordinating activities with

other nodes, without the need for a central coordinator. We take inspiration from nature, which has provided numerous instances of intelligent behavior in which groups of simple organisms function collectively and exhibit behavior that is much more complex than would otherwise be possible with individual effort. A striking example is that of the social ants [6] who live in colonies and collectively address the daily-needs of finding food, building homes, responding to external threats, spreading alarms, etc. Such behavior in social insects has been modeled using the theory of self-organization (SO) that describes how macroscopic behavior can be exhibited by microscopic interactions at the molecular level in materials and chemical reactions. The remarkable feature of the social ant behavior is that the different members of the colony have a specific role and are able to communicate and coordinate their activities precisely. Any ant in the colony can be replaced seamlessly with another ant without affecting the overall organization or performance of the colony. This is a distributed system where agents engaged in rather simple behavior cooperate and coordinate their activities to generate very complex behavior. In this case, the end result is considerably greater than the summation of the individual efforts. While the behavior of the system remains the same, the constitution of the entities changes dramatically. Similarly a robust computer network can be created in which agents across network nodes communicate with each other and coordinate activities, creating behavior patterns that can accomplish complex tasks.

To achieve such a network we propose using a service-based peer-to-peer (P2P) architecture. The proposed architecture consists of autonomous agents, each of which exhibits a different behavior and can form complementary federations with other agents to perform predefined tasks. Each agent in the architecture has the ability to perform simple specialized tasks, however, multiple agents can also collaborate with each other to collectively perform complex tasks. Self-healing and self-managing behavior can be intrinsically modeled in such a distributed peer-to-peer architecture.

In this architecture all operations are defined as services on the network. Each service on the network can be transparently substituted with another service without any loss of functionality. Services advertise themselves by broadcasting on the network and registering with any open registry. The services discover other services on the network by searching the registries. There are no fixed IP-addresses and no fixed associations; if a service breaks down it is removed from the registry and eliminated from the peer network. Similarly, if a new service is created it gets registered and enters the peer network. The architecture is based on an extensive body of literature on

peer-to-peer systems that is described in detail in the next section.

3. Peer-to-Peer Systems

A peer-to-peer system [7] is a network of computers and devices in which each node is equivalent to every other node on the network. A node may operate as a client, a server, or a router according to the task to be performed. P2P systems usually operate outside the domain name system (DNS) and use an instant-messaging protocol for communication. The instant messaging protocol establishes direct communication between two or more peers rather than rely on a central server. Additionally, peer-to-peer architectures contain lookup registries, which facilitates registration and discovery of peers. On the basis of the placement and usage of these registries, three classes of P2P architectures have evolved: centralized; structured and decentralized; and unstructured, and decentralized [8]. In the centralized architecture, a peer passes a request to a registry to locate the pertinent nodes. In the structured, decentralized architecture, the architecture positions registries at specified locations on the network. In an unstructured, decentralized architecture, registries are absent from the network: each peer maintains a list of neighboring nodes that may satisfy the request. In this arrangement, peers pass the request to a limited set of nodes that attempt to satisfy the request; if the request cannot be fulfilled, each node propagates the request to another set of nodes. This process continues until the request is satisfied, or the predefined number of propagations is exceeded. As the architecture moves from more centralized to more distributed privacy, scalability and resilience to attacks improve while complexity and search efficiency worsen. The propagation (discovery) mechanism needs to be evaluated for the specific needs of problem being investigated.

Robustness in P2P systems is not the result of any individual machine working more robustly but rather is due to the collective behavior of all the machines. In fact, an individual machine on average may be less robust than a typical server but collectively these machines exhibit greater robustness. This robustness stems from the ability to seamlessly replace one component of the system with another without having to reconfigure the system and from being able to add capacity to the network without disrupting its current operations.

P2P systems have become popular in the context of music and file sharing [9]. Napster, Gnutella, Morpheus, and KAZAA have become icons in the movement of free sharing of music and other media files. These P2P systems are very limited in their scope; providing

primarily for the sharing of files across network nodes, which is only a small subset of applications of distributed computing. There is growing interest in using P2P systems for managing complex transactions in distributed environments where commands have to be executed in the object space of disparate computers. In these applications business processes are dissected into autonomous tasks that can be assigned to different peers on the network. These tasks are referred to as services in the literature. The services do not need to have a one-to-one correspondence with the network nodes and are mapped onto a virtual network of services overlaid on the network of computers. The key requirement for the services is standardization of the interfaces since the services are identified by the interfaces that they implement. Standardized interfaces allow transparent substitution of one service with another, thereby improving the robustness of the system and allowing a sharing of services between multiple processes.

This area of complex distributed transactions across the enterprise is being addressed by grid computing architectures [10] where protocols, services and tools that address some of the challenges in this environment have been created. In this architecture, communication goes directly from one system's enterprise object to another system's enterprise object. This allows software, data and other resources that are scattered across the enterprise to synchronously communicate with each other. The key requirements for implementation of a service-based distributed computing architecture are autonomy and standardization of services, real-time discovery of services, and communication between services at a peer level. In addition, the concept of leasing is used in some architectures to provide self-healing abilities to the system. The service-based architecture is described more elaborately in the next section.

3.1 Service Based Architecture

In a service-based distributed architecture all operations are defined as services on the network. Each service has a standardized well-known interface by which it is recognized. Multiple services of the same type can exist on the network and can be substituted for each other. This architecture allows all users to communicate in a standardized fashion. It abstracts away the resource specific details that would require an extensive software development effort to mask the lack of interoperability and compatibility between systems. These services form associations in real time to execute a specific sequence of tasks and dissolve these associations at the completion of the task only to form other associations to solve different sets of problems. Multiple services of the same type can exist on the system and can be substituted for one another.

These services can join a federation to perform a set of tasks. The selection of any given service depends on its availability and cost in case of fee-based services. New entities are constantly generated and existing entities periodically expire. If any service is disabled for some reason, another service can replace it seamlessly without affecting the overall behavior of the network.

In this system, unlike conventional systems where the IP-addresses of the different entities are known prior to communication, the entities required for the transaction are determined at the time of the transaction by searching over the network. This process called *discovery* is facilitated by using services with standardized interfaces that allow one service to be replaced by another service without affecting the behavior of the transaction. This standardization of the interfaces, rather than using fixed associations between interacting entities, provides the robustness to the network. If services are destroyed at any physical location at a time of crisis, the services can be restarted elsewhere without reconfiguring entities that require those services. The entities needing those services would automatically discover the new services instead of the old services from the network. A new service introduced on the network does not need to be added to the configuration files of the other services on the network. It automatically registers itself and makes itself eligible for discovery.

Resilience can be achieved by building redundancy into the network such that multiple services of the same type exist. The architecture uses the concept of a lease such that when a service registers it gets a fixed renewable lease. If the service is disabled, it is unable to renew its lease and thus leaves the network. When a request is made for a service, only active services are considered, thus making the system robust to failures. The architecture also introduces the concept of provisioning, whereby, if no service is available a new copy of the service is automatically generated on the network using a blueprint stored in the provisioning servers, which are scattered over the network. In addition, the service requests can be delayed until such time as a service becomes available. This architecture represents a paradigm shift from the existing client-server architecture and comes with its own set of issues that will be discussed in the next section.

3.2 Security and other Issues

P2P systems require a different perspective on security as compared to traditional client-server systems since the peers are characterized by a lack of name recognition and are only known by a standardized interface unlike client-server systems, which have a fixed

IP-address and an identity associated with that server. A centralized access control cannot be used in such a system as it introduces a central point of failure. Each service has to maintain its own access control using roles defined in the system or based on specific users which are registered with the service. Each service controls access to its resources, subject to constraints on what, when, where, and by whom a resource can be used. Such constraints can be defined in security policies that can change dynamically over time in terms of the resources involved, the nature of access permitted, and the participants to whom access is permitted. The access is not only applicable for users but also for other resources, such as applications requiring database access or computing resources. The access may also be delegated across a chain of users depending on the type of access allowed.

P2P systems are intrinsically more secure relative to client-server systems to attacks that depend on a single point of failure, such as, denial-of-service attacks and IP-spoofing attacks. Some questions, however, remain unanswered. For instance, in a distributed environment, would viruses and worms travel even faster than in traditional client-server systems; or could there be specific spoofing attacks launched specifically against such grid systems whereby rogue providers implement interfaces to perpetrate fraudulent services? Some of these issues will be investigated in our research using this architecture.

It has also been claimed that such systems bring simplicity to the configuration and management of systems. This fact is conceptually true, however, the configuration burden of such systems is quite large at the time the network is created. This stems from a lack of standardization among the various systems that inhibit interoperability and require extensive customization for different operating environments. In addition, P2P systems are still evolving and require constant upgrading and maintenance. In the long all of the issues mentioned above should be resolved as these systems mature.

Such a system will lead to an efficient, robust and resilient system that provides better use of resources. However, some organizational issues need to be addressed, such as, receptivity of the different State agencies to the sharing of information and legal issues regarding the use of data collected for one purpose and used for a different purpose, especially when sharing of sensitive information is involved. The impact of such systems on collaboration in state agencies in context of the NY State government is provided in the paper. A brief introduction to the networks linking the State agencies is provided in Section 4.

4. NY State Interagency Wide Area Network

Network communication is established across agencies to allow communication across geographically dispersed offices. Such intra-agency communication supports common applications at each remote office with the ability to process, store, and correlate the information centrally. In addition, agencies with sector (health, criminal justice, education, transportation) affiliations establish inter-agency communication in order to share data dependent applications. These network provided the cross weave for this spider web communication infrastructure. Each agency has some core applications controlled by a central agency. For instance, every agency needs to encumber and expend funds and maintain their payroll roster. These funds are disbursed by a central agency. To expediently and accurately process these transactions, a many to one inter-agency network is required. A metropolitan Intranet facilitated this inter-agency communication and sharing data.

As each agency attempted to resolve their individual communication needs relating to geographic dispersion, multiple leased lines were often drawn between the same cities. Figure 1 shows the network that evolved over time to support the enterprise computation needs of NY State agencies. This network consisting of a myriad of different sub-networks resulted in a complex system of security risks that needs to be managed. This level of complexity has presented challenges with respect to managing viruses, worms and software patches. In an environment consisting of disparate information systems, synchronous communication between applications distributed across the enterprise requires middleware that acts as a bridge between applications that support different protocols.

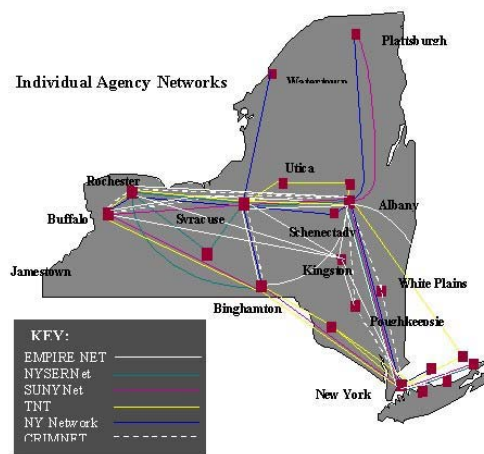


Figure 1: Existing NY State Network Architecture

To streamline the network architecture and to address the growing needs for bandwidth across the State, the agency constructed an enterprise digital backbone consisting of high capacity fiber optic cables operating at OC-48 speed with fourteen network access points (NAPs) as shown in Figure 2. Anticipating current and future needs, this enterprise network architecture also incorporates Lightweight Directory Access Protocol (LDAP) for enterprise wide identification, authentication and authorization.



Figure 2: New NY State Network Architecture

Communication between different State agencies has been tenuous and is usually exercised only to address a specific need. However, the potential for efficiency improvement from inter-agency communication is tremendous. Such intra-agency communication becomes essential for effective handling of serious crisis, such as, the ice storm of 1998 and the bombings of WTC in 2001. To facilitate electronic communication between the agencies, the first step is to lower the technical threshold for integrating information systems. The service based grid architecture proposed in the paper addresses both the improved security and streamlined inter-agency communication.

To investigate the feasibility of incorporating the new architecture in the State agencies, a simulation using an agent-based simulation package is being developed. The simulation is based on the Critical Infrastructure Response Information System (CIRIS), an application being which is described in the next section.

4.1 Critical Infrastructure Response Information System (CIRIS)

CIRIS is an application that allows public safety personnel and State officials to search for, locate and visualize information about critical assets and infrastructure components in NY State. The assets and

components are related to map locations. The system provides real-time response to information needs and will use secure Internet technology to deliver information to its customers. Customers will include select federal, state and local law enforcement, along with fire, medical, and civil defense government officials. CIRIS is a mission critical application that must be available for use by state and local officials in the advent of a natural or man made disaster. The CIRIS data will be made widely available to a large number of geographically dispersed users. The data centric model stores all of the information from many sources in a single repository. In CIRIS, data is imported from each data source into a central repository, converted to a single datum and projection, and cataloged and indexed according to feature types and attributes of the data. CIRIS, is intended to service the need for critical data in emergency situations and must, therefore, be available in all circumstances, at all times, with minimum system intervention. In addition, it requires built-in fault tolerance, redundancy, and universal availability. CIRIS is thus an excellent application for evaluating the proposed architecture; it inherently contains all the required attributes such as self-healing, resilience, and universal availability.

5. Proposed Architecture

The proposed architecture consists of autonomous services (agents) on the network the discover each other and communicate with each other on a P2P basis. In a distributed P2P system the agents can be broadly categorized into three categories: service providers, service requestors, and lookup services. A service provider represents a source of data or processing capacity on the network. Each provider resides on a single network node and can provide multiple services. The services may be database access, analysis, message dispatching etc. The main attributes of each service provider are processing capacity, failure rate, and queue capacity. Any transaction on the network can involve a single service or multiple services that run concurrently across multiple providers, or a choreographed sequence of operations involving multiple services with precedence relationships.

In the most general form, a service request will be a directed acyclic graph in which the nodes represent the events and the connectivity represents the choreographed sequence of operations that are required to complete a transaction. Lookup services allow the services to register themselves and for service requestors to discover the services. The architecture uses the concept of leasing used in some p2p architectures, which forces the service provider to keep renewing its lease with the registry. This ensures that if the service is not operational, it is unable to

register and falls out of the network. The last major piece of the architecture is the provisioning server that allows applications not currently operational to get started on any available node on a request from a user.

A schematic of the network configuration and the components that reside on the network are presented in Figure 3. The components are: nodes (solid circle), peers (spiked circles), and provisioning servers (double walled circles). The nodes are the physical locations where network access is provided. The peers are the service providers, service users and lookup registries. The provisioning servers (shown in double lined circles) are backup servers containing a blueprint of all applications provided on the network. These can be used to start any service that is registered with them and is unavailable either because of failure or because its capacity is already saturated. Whenever a new application is created or an existing application is updated, a blue print of the application is created on the requisition server. When the user makes a request for a service, the first available service is returned to the user. If a service is requested and is unavailable, the request then passes to the provisioning server, which automatically starts the service on an available node on the network. If the backbone of the network (Figure 3) is severed in the middle, disabled services can be generated by the provisioning servers on either half of the network creating two independent networks that are mirror images of each other.

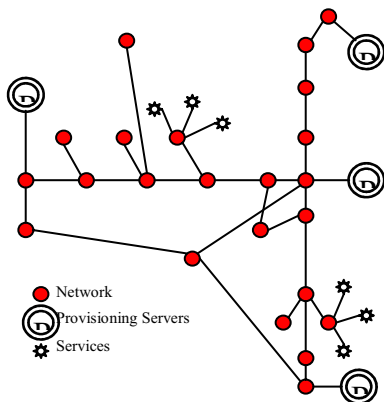


Figure 3: Schematic of the State Network configuration using a peer-to-peer architecture.

In order to manage the capacity better when the network is degraded, only the most critical services may be restored while other services stay dormant until a specific request is made. Such an arrangement where a service can exist anywhere on the network helps in the consolidation of server resources at strategic locations and reduction of the overall maintenance cost while leading to better resource utilization.

Each node on the network can contain multiple service users, service providers, and lookup registries. However, only relatively few nodes contain the provisioning servers. The provisioning servers are located at strategic locations on the network so as to minimize the risk of catastrophic failure of critical services in any part of the network. The new architecture can co-exist with the existing client-server based network operations without any adverse effect on performance. The new architecture allows multiple services on the network to be easily integrated and to work seamlessly together. A new node added to the network has access to the services anywhere on the network within access control constraints. The next section describes some of the planned simulations based on different disruption scenarios.

6.0 Simulations

Several simulations will be run on the network model shown in Figure 3. A baseline simulation will be conducted to model the behavior during normal activities. Different duress scenarios will be modeled in the simulation by altering the baseline topology as well as the capacity of the network. The simulations will consider dynamic networks where configuration of the network is continuously evolving as new services enter the network and existing services leave the network. The network topology can also change radically following catastrophic events, such as, failures of critical nodes or severing of specific communication channels. Also, network capacity may be severely degraded following DOS or virus attacks on the network. The network performance will be tested by randomly degrading the processing capacity at different nodes as well as by escalating the demand rate for the services. A comparison of the network performance under different scenarios relative to the base case will be provided for each simulation. Several metrics, such as availability, successful requests, wait-time, etc., are defined to measure different aspects of performance of the architecture. The simulation environment selected and the simulations planned are discussed in the next two sections.

6.1 Simulation Tools

The proposed architecture is a self-organized system that contains autonomous agents that can interact with other agents and make independent decisions while subscribing to a set of organization rules. There are several classes of modeling tools available, including general purpose discrete-event modeling tools, specialized network modeling tools and agent-based modeling tools. General purpose discrete-event based simulations tools (Vensim, Quicksim, and Anylogic) [11] are excellent for scheduling, and other discrete-event problems, however,

they do not contain the basic infrastructure to model self-organized systems efficiently. Network simulation tools (ns, REAL, Opnet, and Maise) [12] contain a rich set of network protocol libraries that are useful for investigating P2P/C-S systems. They are unsuitable in this case since our simulation is at a higher level than the network protocols. In addition, these tools do not have infrastructure for the self-organizing behavior either.

Agent-based simulation tools that are specially designed for modeling self-organizing behavior of natural systems are suitable because of the flexibility and ability to model self-organized systems. RePast [13] and Swarm [14] are two of the most widely used simulation software frameworks for agent-based simulation modeling in social insects. RePast is selected as the agent-based simulation package for this work. A typical RePast model contains a set of agents where each agent has specific attributes and exhibits behavior that characterizes it. Agents interact with one another and manage complex tasks without the intervention of an external entity. A RePast simulation program contains the logic for setting up and controlling the agents behaviors. RePast represents agents behaviors as events and actions. At setup time a RePast simulation program instantiates agents, creates seminal events and performs other initializing activities. During execution or processing, the simulation program schedules and processes all subsequent agents, behaviors, and outcomes. A RePast program can schedule the behaviors dynamically as well as stochastically.

In this agent-based simulation, multiple agents having different attributes and diverse behavior coexist and work synergistically. They complement each other's role and collectively exhibit behavior that exceeds the ability of any individual agent. Each agent in the system has a predefined life cycle of, birth, life and death. The population of each type of agent is stochastically controlled based on the mean starting population, the birth rate and the death rate. All the events during the lifetime of the agent are also stochastically controlled based on the probability distribution function that characterizes the agent. The agents in the simulation correspond to the services, providers, provisioning servers, and registries. The services are comprised of the distributed databases that contain the assets of the organizations.

6.2 Simulations Planned

Distress in the network can occur by many different causes, such as physical destruction of a portion of the network, cyber-attack on the network, increased demand during emergency situations, and random downtime of network components such as servers and routers. Different simulations are planned based on these crisis

scenarios. The planned simulations cover: 1) physical damage to the network, 2) increased demand in crisis situations, and 3) hacking attacks that degrade the network performance. The following sections describe the three general simulation scenarios. In all the simulations, access to the data in the distributed CIRIS database would be used to investigate the performance metrics. Due to the highly sensitive nature of both the state's network and the data that will be employed in the actual simulations we are compelled to "sanitize" our discussions and results. The discussions in this paper and subsequent discussions in other forums will, however, provide ample evidence of the value of the P2P as a means of ensuring reliable uninterrupted communication.

6.2.1 Physical Damage to the Network

Natural disasters like earthquakes, ice storms and flooding as well as man-made disasters, like the World Trade Center bombings, can destroy or disable portions of the network. These disasters not only result in disruption of communication, but at the same time, create an increased need for communication. This increased need for communication stems from the need to coordinate the disaster relief activity as well as the need to support law enforcement agencies investigating criminal activity. The incident of the World Trade Center attacks is considered for simulation of network under duress.

In the aftermath of the WTC attacks, there was a need to coordinate the activities of the police, fire, and other relief agencies. Databases of police, fire and emergency workers, as well as employees of the companies housed in buildings, were required to match employee information with the DNA samples of bodies found on the site. In addition, FBI, the New York State Police, Airport Security across the country, as well as Immigration and Naturalization Service, needed to identify the terrorists involved in the attacks. At the time of the attack a key router underneath WTC buildings was destroyed leading to chaos among some state agencies that depended on the router for communication. The simulation attempts to model such events by removing specific links between critical nodes as shown in Figure 4.

6.2.2 Increased Demand During Crisis Situation

To investigate increased demand during crisis situations a case of water supply contamination with some lethal toxins, microbes, or bacteria is considered. Though hypothetical, this problem is under serious investigation by state agencies where different scenarios of impact of water supply contamination on the population are investigated. In such situations hundreds of people become sick, thereby necessitating coordination between

the hospitals, law enforcement agencies, Federal Drug Agency, US department of agriculture, and other agencies. To respond to such a crisis, the contaminant has to be identified so that correct drugs can be administered, hospital capacities have to be checked to direct patients, the water supply has to be cleaned, and the perpetrators of the crime have to be tracked to prevent recurrence of such incidents, all in a short time. At the same time there would be an increased need for information among citizens and the state crisis management workers (including police, fire, health care, engineers etc.) As shown in Figure 5 the demand will be scaled at specific nodes in response to specific sets of information and the response of the network to those requests will be measured. The information requested would be from the distributed CIRIS databases as described earlier in the paper. Some specific nodes will be disabled along with a scaling of the demand in order to estimate the robustness and sensitivity of the system to coordinated physical and cyber attacks.

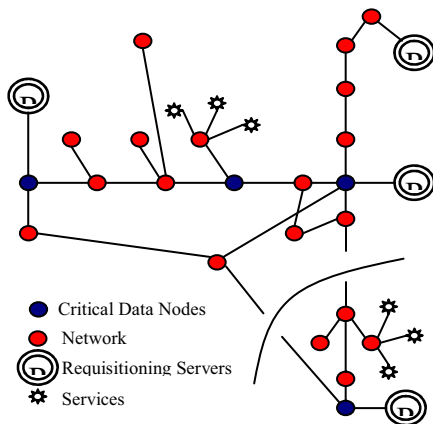


Figure 4: Network severed by destruction of links.

6.2.3 Degraded Performance during Hacking Attacks

Hacking attacks can lead to a compromise of sensitive information, disruption of communication, degradation of network performance, as well as destruction of data. The current work considers primarily the disruption of communication as well as degradation of network performance due to hacking attacks. There have been some studies on the impact of hacking attacks on generic P2P systems, however, significant research results on the impact of hacker attacks on grid systems are unavailable. There are two specific attacks that are being investigated in the simulation, that is, denial of service attacks and virus attacks.

Denial of service attacks attempt to overwhelm the server providing the service such that all its resources are tied up in fake requests. As a result, legitimate requests cannot be processed. In this simulation fake service requests will be generated for specific resources on the network. The requests will originate across multiple nodes on the network and will target one or more resources. Countermeasures for these attacks will be employed to estimate their impact on prevention of degradation of the system performance. The expectation is that other redundant services on the network will cover the slack or the provisioning server will start a new service.

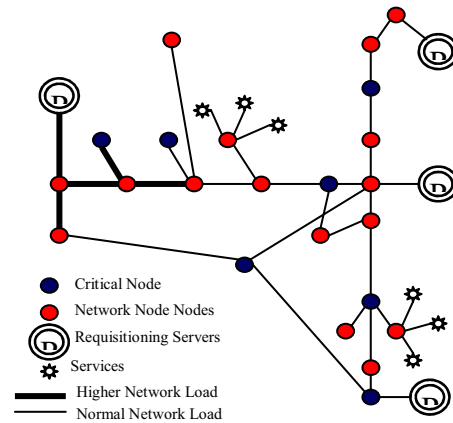


Figure 5: Increased network load during crisis

In the second simulation the network performance will be evaluated under virus (and worm) attacks. An infection and detection rate will be defined for the simulation, and the performance of the system will be observed over time. Requests will be infected with viruses (and worms) at random. These requests, in turn, can infect the services that they are being processed by the request. Once the service is infected, all the other requests that are processed by this service will also get infected, and these infected requests then pass the virus on to the other services that are required by the request. The infected programs can also get disinfected at random by the antivirus software running on the network. The virus first spreads uncontrolled, and gradually as the response kicks in the viruses in the system are destroyed. In this simulation a random graph model will be used to study the impact of virus spread on the network performance.

6.0 Conclusions & Future Work

A self-healing architecture, using Grid Computing concepts has been proposed to ensure uninterrupted communication among state agencies during crisis situations. This architecture fosters inter-agency communication in NY State and also leads to lower maintenance costs for the networked applications. This

architecture abstracts the functionality that the agencies need to expose to other agencies into a service with a standardized interface. This improves the security by limiting access to only the functionality that is provided in the interface and shielding the rest of the application from inadvertent security lapse. This service-based architecture uses the concept of real-time discovery of services using registries thereby making it resilient and robust to

failures. Such resilience is critical for any infrastructure that will be used for coordination during any crisis in the State that requires intervention of multiple agencies. The simulation for the self-healing resilient architecture proposed in the paper is currently being evaluated and will be discussed with all respect to security concerns in the upcoming conference and in subsequent presentations.

References

- [1] CSI/FBI 2000 Computer Crime and Security Survey. <http://www.pbs.org/wgbh/pages/frontline/shows/hackers/risks/csi-fbi2000.pdf> (Last Visited 09/26/2003)
- [2] J. Chirillo, "Hack Attack Encyclopedia: A Complete History of Hacks, Cracks, Phreaks, and Spies over Time". New York: Wiley Computer Publishing, 2001.
- [3] ITAA: B2C e-data: "Jupiter Projects that Online Retailing Will Continue to Grow, June 2001", <http://www.ita.org/isec/pubs/e20016-06.pdf>. (Last visited on 06/16/03).
- [4] ITAA: B2B e-Data: "Gartner Projects Worldwide Business-To-Business Internet Commerce to Reach \$8.5 Trillion In 2005", <http://www.ita.org/isec/pubs/e20013-06.pdf> (Last visited on 06/16/2003.)
- [5] Federal Bridge Certification Authority Website <http://csrc.nist.gov/pki/fbca> (Last visited 9/28/03).
- [6] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems (Santa Fe Institute Studies on the Sciences of Complexity)", *Oxford University Press*, 10/1/1999.
- [7] P. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, I. Zaihrayeu, "Data Management for Peer-to-Peer Computing: A Vision". In Proceedings of the Fifth International Workshop on the Web and Databases (WebDB), 2002.
- [8] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks," In Proceedings of the 16th annual ACM International Conference on Supercomputing (ICS), 2002.
- [9] A. Oram, "Peer-to-Peer: Harnessing the Power of Disruptive Technologies", O'Reilly, 2001.
- [10] I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of The Grid Enabling Scalable Virtual Organizations, *International Journal of Supercomputer Applications*, vol. 15(3), 2001.
- [11] J. J. Swain, "Simulation Reloaded: Sixth biennial survey of discrete-event simulation software tools that empower users to imagine new systems, and study and compare alternative designs", *OR/MS Today* 2001.
- [12] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, H. Yu, "Advances in Network Simulation", *IEEE Computer*, vol. 33, No. 5, p. 59-67, May 2000.
- [13] M. Daniel, "Integrating Simulation Technologies with Swarm," Agent Simulation: Applications, Models and Tools Conference, University of Chicago, Oct. 1999.
- [14] N. Collier, "*Repast: An extensible framework for agent simulation*", 2002. http://repast.sourceforge.net/docs/repast_intro_final.doc (Last Visited 9/30/03).