POSITIONING AND PERFORMANCE IN SIMULATED NETWORKS

Daniel Smith
Nevada State College
daniel.smith@nsc.edu

Paul Grey
Nevada State College
paul.grey@nsc.edu

ABSTRACT

In the last decade, simulation has been used with greater frequency to analyze company performance. Studies show that companies that incorporate network characteristics into their decision making processes produce better results over the competition. This study examines how simulations can enhance communication, trade and intercompany activities. Those, in turn, may result in better performance and higher profits. We develop a simulated network and investigate the network impact on the behavior of the companies and their outcome. The results show that positioning the enterprise in certain points within a network can improve its performance and increase its profits. Those findings validate previous studies on strategic networks.

INTRODUCTION

Performance analysis of simulated networks is rapidly gaining importance as networks increase in size and geographical extent (Ben-Zvi and Gordon, 2007; Smith and Goldman, 2009). The size of the networks and the inherent complexity of network protocols complicate this analysis (Ye and Kasemsarn, 2010). Analysis techniques such as queuing models have difficulty modeling dynamic behavior such as retransmission timeouts and congestion. In these cases, simulation offers a better method of studying computer networks, since one can simulate the details of actual protocols. Further, traces of execution under different environments provide insights into the dynamics of protocols that queuing analysis does not easily provide. Of course, there is considerable effort involved in building a simulator and substantial machine cost in running it, but this approach still is the most attractive (Hu and Pekin, 2010).

The immediate motivation for building such a system was to compare the performance of companies under the simulation and those of previous studies. While the simulator was built specifically for this study, simple modifications can greatly extend its power and generality.

This study presents a description of a simulator and the results of some simulations to illustrate its analytical power. The next section describes a user's view of the simulator. This simulation is a simulation server combined with a display client. The *server* carries out simulations, and connects through a socket to the client. The *client* maintains a 'simulation window' which reflects the current state of the simulation. A user sets up and controls the simulation through the display client. The client passes simulation parameters and control messages to the server; the server passes status information and results to the client. Such a partition has two advantages. A simulation can be run on a large compute server, and simulations running in parallel on several machines can be controlled from a single screen to the full variation of simulation windows (Xi and Yuan, 2010). We have taken advantage of both features - by running large simulations on a Macintosh, and by running up to six simulations in parallel on a cluster.

The remainder of the report is organized as follows: the next section presents an outline of the simulator from a user's point of view. Then, we provide details of the implementation of the simulation. Next, we sketch the transport protocols implemented. Then, we present some gateway scheduling algorithms. Following is the results of the simulation. We also provide performance evaluation of the simulator and analysis of the results. Finally, we conclude the study by drawing conclusions and suggesting directions for future work.

LITERATURE REVIEW

The first wireless networks were developed in the pre-industrial age (Bodevin and Suttikul, 2010). These systems transmitted information over line-of-sight distances (later extended by telescopes) using smoke signals, torch signaling, flashing mirrors, signal flares, and semaphore flags. An elaborate set of signal combinations was developed to convey complex messages with these rudimentary signals. Observation stations were built on hilltops and along roads to relay these messages over large distances. These early communication networks were replaced first by the telegraph network (invented by Samuel Morse in 1838) and later by the telephone. In 1895, twenty years after the telephone was invented, Marconi demonstrated the first radio transmission from the Isle of Wight to a tugboat 18 miles away, and radio communications was born. Radio technology advanced rapidly to enable transmissions over larger distances with better quality, less power, and smaller, cheaper devices, thereby enabling public and private radio communications, television, and wireless networking (Chang, 2010; Lei and Chong, 2009).

Early radio systems transmitted analog signals. Today most systems transmit digital signals composed of binary bits, where the bits are obtained directly from a data signal or by digitizing an analog voice or music signal. A digital system can transmit a continuous bit stream or it can group the bits into packets. The latter type of system is called a packet radio and is characterized by transmissions: the radio is idle except when it transmits a packet. The first packet network was developed at the University of Hawaii in 1971. This network enabled computer sites at seven campuses spread out over four islands to communicate with a central computer on Oahu via radio transmission. The network architecture used a star topology with the central computer at its hub. Any two computers could establish a bi-directional communications link between them by going through the central hub. Many studies investigated this type of networks and systems (e.g., Adams et al., 2010; Lamas and Martinez, 2010).

The network incorporated the first set of protocols for channel access and routing in packet radio systems, and principles underlying these protocols are still in use today. Activities promoted by DARPA, peaked in the mid 1980's, but the resulting networks fell far short of expectations in terms of speed and performance. Packet radio networks today are mostly used by commercial providers of wide-area wireless data services. These services, first introduced in the early 1990's, enable wireless data access (including email, file transfer, and web browsing) at fairly low speeds, on the order of 20 Kbps. The market for these data services has not grown significantly.

In the 1970's Ethernet technology steered companies away from radio-based networking. Ethernet's 10 Mbps data rate far exceeded anything available using radio. In 1985 the Federal Communications Commission (FCC) enabled the commercial development of wireless LANs by authorizing the public use of the Industrial, Scientific, and Medical (ISM) frequency bands for wireless LAN products. The ISM band was very attractive to wireless LAN vendors since they did not need to obtain an FCC license to operate in this band. However, the wireless LAN systems could not interfere with the primary ISM band users, which forced them to use a low power profile and an inefficient signaling scheme. Moreover, the interference from primary users within this frequency band was quite high. As a result these initial systems had very poor performance in terms of data rates and coverage (Ding and Chen, 2009; Hu and Zeng, 2010).

More sophisticated systems were developed in the past 30 years. One of the more popular fields of systems today is healthcare, where systems are used to promote health in the general population and assist healthcare professionals (see, for example, Becker, 2010; Walsh, 2010). Those systems served as the basis for numerous studies on performance and company positioning. One of the main applications is this field is data mining (see, for example, Chong 2010; Sun and Chen, 2010; Woo and Hu, 2009; Patel and Chang, 2010). However, other applications may also be found in the information systems studies (e.g., Ben-Zvi, 2009; Chen and Lin, 2009; Lei, 2009).

THE NETWORK MODEL

The simulation we use simulates a packet switched, store and forward network similar to existing wide area networks such as the Xerox corporate net and the DARPA Internet. The network layer is datagram oriented, and packets can be lost or delivered out of sequence. The transport layer, which is modeled on the Transmission Control Protocol, provides reliable, sequenced packets using standard techniques of flow control, timeout estimation and packet retransmission. The network itself consists of a set of sources that execute a transport protocol, gateways that route packets and schedule them on their outgoing lines, and sinks that absorb packets sent to them, returning an acknowledgment for each packet received.

Our analysis assumes that the bottleneck in a network is always the limited bandwidth of transmission line. Hence all sources, gateways and sinks are infinitely fast. Note that the simulation sources attempt to model many of the details of flow control in the transport layer in contrast to some previous work. Sources are classified according to the workload they present to the simulator and the version of the transport layer protocol they employ. We chose to model three types of workloads. Workload models contain large file transfer. That is, whenever flow control permits, a source with workload (or source) immediately sends a maximal sized packet. A telnet source sends minimal sized packets with an exponential interspace the modeling the workload of a telnet (remote login) connection. An ill behaved source continuously sends maximally sized packets, limited only by the network's bandwidth.

In addition, we also provide details on the user interface: The display client runs on a Sun 3 or Sun 4 workstation, and presents a mouse based interface to a user. The network is represented as a graph, and a display manager maintains the current status of the network in the simulation window. A number of control panels allow simulation parameters (such as the speeds of the communication lines) to be set interactively by mouse clicks, or by typing a value.

Network graphs are drawn interactively using the mouse, and the result is sent to the simulator. A node in the graph represents a source, gateway or sink, as specified by its 'node function', which is set for each node by clicking on the appropriate menu. As an example, to draw the graph, the mouse is used to create the six nodes (by clicks at the appropriate locations) and the edges between the nodes. Additional menus prompt for the protocol to be run at each node and the bandwidth/delay characteristics of each line.

The simulation's parameters are set from the appropriate panel. The entire graph and associated simulation parameters can be sent to the server by clicking on a send selection or saved to a file and reloaded later.

The display client allows for graphical monitoring of a selected node's variables (such as the current window size) as the simulation proceeds. The nodes to be monitored are selected by clicking on their icon. For example, one might monitor the window size for the protocol running on node 1. This is analogous to attaching an oscilloscope to a signal line in a hardware circuit, and as useful. The simulation provides for automatic report generation. A library of routines for table generation collects relevant statistics (such as packets sent and received by each node), tabulates them, and prints them out at predefined intervals.

It is interesting to note that entire simulations can be set up from the display using only the mouse and menus. This visual programming style, analogous to that proposed in many other simulations. The availability of a graphical display to monitor the simulation in real time is useful, and is an example of scientific visualization.

SIMULATING THE NETWORK

The simulation is built upon a simulation test bed package. We now present an outline of the simulation, enumerating and describing the additions we made. The simulation provides (a) a display client and (b) a simulation library. The display client manages a simulation window, where the simulated network and associated parameters are initially input, and subsequently updated as the simulation proceeds. The library includes a threads package, primitives for inter-thread communication, and a blocking socket library for communication between the simulation server and the display client.

The network is specified by the set of nodes and their interconnecting edges. Each network node is assigned a function, the 'node function'. After the set of nodes, their node functions and their interconnections are specified, the simulation runs each node's function in parallel in a single common address space. The simulation is thus written as a set of node functions, one for each kind of source, gateway and sink. Writing such node functions is difficult. Not only do we have to solve standard synchronization problems, but also, since threads are pre-emptible, all node functions have to be made reentrant.

Communication between the server and the display occurs over a blocking socket. The network, described formally in a 'graph language' structure, is converted to a bit stream and exchanged between the display and the server. We wanted to leave the graph language unchanged to maintain compatibility with other NEST simulators, and so added an extra socket between the display and the server to carry the specific simulation parameters. We implemented the socket using the blocking socket library. An extra panel was added to the simulation window to hold the simulation parameters.

We now describe some libraries that we added. The Tables Library creates tables of statistics, adds entries to them and prints them neatly in a report. A large Buffer Management Library supports a variety of buffer allocation schemes at the routers. Buffer data structures and associated procedures were strictly encapsulated to assure correctness of buffer management. As the simulation progresses, values of selected variables can be written onto a file. A separate process reads these values and creates graphs (using the UNIX graph utility) that are then displayed on a Tektronix emulation window (tool kit). Finally, a number of tracing options were built into the simulation. These allow simultaneous tracing of one of many network functions, such as routing, buffer management and source protocols. This proved to be a valuable debugging tool, as well as a way of checking a protocol implementation's correctness.

Generic sources model features common to many of today's transport protocols such as TCP and XNS. They represent the minimal common functionality necessary for a reliable, connection-oriented transport layer built on an unreliable, connectionless network layer. Generic sources implement sliding window flow control and retransmission timeout. We note that there is a packet sequence number 10 at a gateway due to congestion. Window flow control allows packets with sequence numbers 10-14 to be outstanding. Assume that the sink receives packets 11-14. However, these packets are out of sequence and the sink does not acknowledge them. When packet 10 times out, the source retransmits it and the sink acknowledges packets 10-14. At this point, the source sends a burst of 5 packets, which further aggravates the congested gateway. Thus, the packet loss due to congestion triggers a burst of packets from a generic source which sustains the condition.

Public-switched telephone network to the mobile telephone switching office (MTSO) in that mobile's home city. When a mobile unit in the home city turns the handset on, that signal is relayed by the local base station to the MTSO. The MTSO authenticates the ID number of the mobile and then registers that user in its home location database. After registration, any calls addressed to that user are sent to him by the MTSO via one of its base stations.

If a mobile is roaming in a different city then by turning the handset on the mobile registers with the MTSO in the visiting city. Specifically, the mobile's signal is picked up by a local base station in the visiting city, which relays the signal to the visiting city's MTSO. The visiting city's MTSO then sends a message to the MTSO in the mobile's home city requesting user authentication and call forwarding for that user. The MTSO in the mobile's home city authenticates the mobile's ID number, adds the location of the visiting city's MTSO to its home location database entry for the visiting mobile, and sends a confirmation message to the visiting city's MTSO. The visiting mobile is then registered in the visitor location database of the visiting city's MTSO. After this process is complete, when a call for a visiting mobile arrives at that mobile's home city, the home city MTSO sets up a circuit-switched connection with the visiting city's MTSO along which the call is routed. This method of call routing is somewhat inefficient, since a call must travel from its origin to the home city's MTSO and then be rerouted to the visiting city. The MTSO also coordinates handoffs between base stations by detecting when a mobile signal is becoming weak at its current base station and finding the neighboring base station with the best connection to that mobile. Location management and routing on the Internet is handled by the Mobile IP protocol. The protocol does not support real-time handoff of a mobile between different networks: it is designed mainly for stationary users that occasionally move their computer from one network to another.

An end-to-end connection in a wireless network is composed of one or more wireless and wired links, with at least one wireless link. These different links have widely varying data rates and delays. Moreover, user mobility causes one or more of these links to change over time. These characteristics make it difficult to insure reliability of the end-to-end network connection. Protocols like TCP that are designed for wired networks do not work well in wireless networks. These protocols assume that packet losses are caused by congestion, and they react by throttling the source.

On wireless networks most packets losses are due to poor link quality and intermittent connectivity. Using the congestion control mechanisms of TCP to correct for these problems can cause large and variable end-to-end delays and low network throughput. In addition, wireless channels have low data rates and high BERs, and the random characteristics of these channels make it difficult to guarantee or even predict end-to-end data rates, delay statistics, or packet loss probabilities.

Performance metrics such as data rates, end-to-end latency, and likelihood of packet loss are usually referred to as a connection's Quality of Service (QoS). The QoS requirements for a connection are based on the kind of data being transported over that connection. For example, voice has a high tolerance to packet loss but a low tolerance for delay, whereas data has the opposite requirements. The inherent impairments and random variations of the wireless channel make it difficult to provide anything other than best effort service in wireless networks. This difficulty is the main challenge in supporting high-speed real-time applications like video teleconferencing over these networks. One possible approach to compensate for the lack of QOS guarantees is to adapt at the application layer to the variable QOS offered by the network.

NETWORK ARCHITECTURE

The three main types of network architectures are a star (central hub) topology, an ad-hoc or any other structure, and a hierarchical or tree structure. These are illustrated in several studies. Hierarchical network architectures are usually only used for wireless networks spanning a range of coverage regions. We note that the lowest level of the hierarchy consists of indoor systems with small coverage areas, the next level of the hierarchy consists of cellular systems covering a city, followed by systems with regional and then global coverage. Since the coverage regions define a natural hierarchy of the overall network a hierarchical network architecture along with protocols for routing and identifying user locations are well-suited to this type of system.

In a peer-to-peer architecture the nodes self-configure into an integrated network using distributed control, and the connection between any two nodes in the network consists of one or more communication links. In a star architecture, communication flows from network nodes to a central hub over one set of channels, and from the hub to the nodes over a separate set of channels. The choice of a peer-to-peer or star network architecture depends on many factors. Peer-to-peer architectures require no existing infrastructure, are easily reconfigurable, and have no single points of failure.

Peer-to-peer architectures can use multiple hops for the end-to-end link, which has the advantage of extending the network range, and the disadvantage that if one of the hops fails, the entire end-to-end link is lost. This disadvantage is mitigated by the fact that each node may have connections to many other nodes, so there may be multiple ways to form an end-to-end connection with any other user. These advantages make peer-to-peer architectures the architecture of choice in military systems.

Since star architectures have only one hop between a network node and the central hub, they tend to be more predictable and reliable, however if that connection is weak then there is no alternative connection. A big advantage of star architectures is that they can use centralized control functions at the hub for channel estimation, access, routing, and resource allocation. This centralized control usually results in a more efficient and reliable network, and for this reason many commercial wireless networks use the star architecture. Common examples of wireless systems with several types of architectures include packet radio networks and some wireless LANs. The star architecture is employed in cellular and paging systems.

Mobility management consists of two related functions: location management and call routing. Location management is the process of identifying the physical location of user so that calls directed to that user can be routed to that location. Location management is also responsible for verifying the authenticity of users accessing the network. Routing consists of setting up a route through the network over which data directed to a particular user is sent, and dynamically reconfiguring the route as the user location changes. In cellular systems location management and routing is coordinated by the base stations or the central mobile telephone switching office (MTSO), whereas on the Internet these functions are handled by the Mobile Internetworking Routing Protocol (Mobile IP).

The location management and routing protocols in mobile IP and in cellular systems are somewhat different, but they both use local and remote data bases for user tracking, authentication, and call routing. In cellular systems location management and call routing are handled by the MTSO in each city. An MTSO is connected to all base stations in its city via high-speed communication links. The MTSO in each city maintains a home location database for local users and a visitor location database for visiting users. Calls directed to a particular mobile unit are routed through the

DISCUSSION AND CONCLUSIONS

In order to connect wired and wireless networks together they must share a common networking protocol such as TCP/IP and ATM. As we have seen TCP has problems operating over wireless links, mainly due to its use of congestion control in response to packet delays. However, wireless links can experience large and variable delays, sporadic error bursts, and intermittent connectivity due to handoffs. Large and variable link delays cause large oscillations in the TCP sending rate, resulting in large and variable end-to-end delays. Error bursts can result in unnecessary retransmissions by TCP, since these errors are usually corrected at the link layer, and also cause significant throughput degradation, since flow is reduced in response to every error burst.

The effect of intermittent connectivity on TCP is similar to that of error bursts, resulting in unnecessary retransmissions and throughput reduction. Various modifications to TCP have been proposed to address this issue, but none has emerged as a clear solution. For example, ATM provides QoS guarantees, which are required for some applications. But it is not clear that the QoS guarantees of ATM can be achieved in a wireless network.

Wireless communication systems are inherently less private than wired systems because the wireless link can be intercepted without any physical tap, and this interception cannot be detected by the transmitter or the receiver. This lack of link security also makes wireless networks more subject to usage fraud and activity monitoring than their wire counterparts. Opportunities for fraudulent attacks will increase as services like wireless banking and commerce become available. Thus, security technology is an important challenge. Security issues can be broken down into three categories: (1) network security; (2) radio link security; and (3) hardware security. Network security includes countermeasures to fraudulent access and monitoring of network activity, and end-to- end encryption. Radio link security entails preventing interception of the radio signal, ensuring privacy of user location information and, for military applications, anti-jam and low probability of interception and detection capabilities. Hardware security should prevent fraudulent use of the mobile terminal in the event of theft or loss, and user databases should also be secure against unauthorized access.

A desire for mobility coupled with the demand for voice, Internet, and multimedia services indicates a bright future for wireless networks. Digital cellular and paging systems have enjoyed enormous growth, but current products and services for wireless data have not lived up to expectations. This is due mainly to their high cost and poor performance. New standards and systems are emerging worldwide to address this performance and cost issues. These systems support a wide range of voice, data, and multimedia services for fixed and mobile users both indoors and out, in cities, rural areas, and remote regions.

There are many technical challenges to overcome in building high-performance wireless networks. The wireless channel is a difficult communications medium. Sophisticated techniques exist to compensate for many of the channel impairments, but these can entail significant cost and complexity. The spectrum must also be used extremely efficiently through advanced link layer, access, and cellular system design. Networking protocols to support roaming users and end-to-end QoS guarantees also pose a significant technical challenge. The limited size and battery life of mobile terminals impose significant complexity constraints, so complexity must be distributed throughout the network to compensate for this limitation. Finally, the unpredictable nature of the wireless channel requires adaptation across all levels of the wireless network design: the link layer, network layer, transport layer, and application layer. This requires interaction between these layers, which violates the traditional network design paradigm of designing each layer in the OSI model independently from the others. While this paradigm has worked well on wired networks, especially as wired technology has evolved to the high performance of today's networks, high-performance wireless networks will not possible without significant technical breakthroughs at all levels of the system design as well as an integrated and adaptable design for the overall network.

REFERENCES

Adams, J., Lamas, J. and Walsh, J. (2010) "DSS: Measurement and Evaluation of an Experiment", Proceedings of *the Portland International Center for Management of Engineering Technology (PICMET) Conference*, Phuket, Thailand.

Becker, P. (2010) "Improving Healthcare: A Data Mining Approach" Proceeding of *the 16th Americas Conference on Information Systems (AMCIS)*, Lima Peru.

Ben-Zvi T., (2009) "Network Structure and Centrality: A Simulation Experiment", Proceeding of *the 30th International Conference on Information Systems (ICIS)*, Phoenix, Arizona.

Ben-Zvi T. and Gordon, G., (2007) "Corporate Positioning: A Business Game Perspective", *Developments in Business Simulation & Experiential Exercises*, Vol. 34, pp. 101-110.

Bodevin, S. and Suttikul, T. (2010) "Employing Emerging Technologies", Proceedings of *the Portland Interna-*

tional Center for Management of Engineering Technology (PICMET) Conference, Phuket, Thailand.

Chang, L. (2010) "Distance Information Technology", Proceeding of *the 16th Americas Conference on Information Systems (AMCIS)*, Lima Peru.

Chen, L. and Lin, C. (2009) "DSS Interaction: A Simulation Experiment", Proceedings of *the 8th pre-ICIS Workshop on HCI in MIS*, Phoenix, Arizona.

Chong, W. (2010) "Security and Information Technology", Proceeding of *the 16th Americas Conference on Information Systems (AMCIS)*, Lima Peru.

Ding, C. and Chen, L. (2009) "Production Reengineering and Risk", Proceedings of *the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Hong Kong, China.

Hu, L., and Pekin, O. (2010) "Social Behavior and Network Analysis" Proceeding of *the Pacific Asia Conference on Information Systems (PACIS)*, Taipei, Taiwan.

Hu, L., and Zeng, D. (2010) "IT and the Environment: An Application in Supply Chain Management", Proceeding of *the Pacific Asia Conference on Information Systems (PACIS)*, Taipei, Taiwan.

Lamas, J. and Martinez, P. (2010) "Knowledge Management: An Application", Proceedings of *the Portland International Center for Management of Engineering Technology (PICMET) Conference*, Phuket, Thailand.

Lei, L. (2009) "E-Learning in Engineering Education" Proceedings of *the International Conference on Advances in Computational Tools for Engineering Applications*, Lebanon, pp. 604-608.

Lei, L. and Chong, W. (2009) "Benefiting from Technology: A Game Approach", Proceedings of *the SIGED IAIM Conference*, Phoenix, Arizona.

Patel, B., and Chang, L. (2010) "Data Mining Clustering in Healthcare", Proceeding of *the Mediterranean Conference on Information Systems (MCIS)*, Tel-Aviv, Israel.

Smith, B. and Goldman, D. (2009) "An Experiment in Social Networks", Proceedings of *the Joint SIGDSS & TUN Users Group Pre-ICIS Workshop and Congress*, Phoenix, Arizona.

Sun, S. and Chen, L. (2010) "Does Distance Matter? An IT Application" Proceeding of *the Pacific Asia Conference on Information Systems (PACIS)*, Taipei, Taiwan.

Walsh, J. (2010) "Data Mining: Healthcare Application" Proceeding of *the 16th Americas Conference on Information Systems (AMCIS)*, Lima Peru.

Woo, X. and Hu, L. (2009) "On Data Mining and Decision Support", Proceedings of *the Joint SIGDSS & TUN Users Group Pre-ICIS Workshop and Congress*, Phoenix, Arizona.

Xi, L., and Yuan, J. (2010) "Simulating Networks: An Experiment", *Developments in Business Simulation & Experiential Exercises*, Vol. 37.

Ye, L. and Kasemsarn, B. (2010) "Analyzing Social Networks Characteristics", Proceedings of *the Portland International Center for Management of Engineering Technology (PICMET) Conference*, Phuket, Thailand.