PREDICTING NETWORK PERFORMANCE IN HETEROGENEOUS, MULTI-FIDELITY, SIMULATION NETWORKS

Christina Bouwens and Ron Matusof
CAE-Link Corporation
Binghamton, New York

ABSTRACT

Simulation networking is no longer new or novel. Heterogeneous, multi-fidelity networks have been successfully demonstrated using either proprietary protocols such as SIMNET, or Distributed Interactive Simulation (DIS) protocols. As the technology for simulation networking has matured, it has resolved some major issues. For example, we now have a standard for the exchange of information between networked simulations (IEEE-1278-1993). There has been very little work done toward prediction and accurate measurement of simulator network loading, and little significant work has been published concerning the implications of network loading toward the overall network fidelity and the successful transfer of training. Implicit in the underlying structure of the DIS is an assumption that network performance is purely an issue of applying appropriate technology to support a particular set of objectives. However, network loading imposes limitations upon these objectives and it is unclear what effect unexpected network performance has upon meeting a particular set of objectives.

This paper addresses the problem of predicting network loading in a heterogeneous, multi-fidelity simulation network. It discusses the issues associated with heterogeneous networks and multi-fidelity simulation. Using objective data obtained from a variety of networked exercises (both DIS and non-DIS) for context, this paper discusses the detailed issues involved in measuring network loading. Finally, it makes some recommendations for the future.

About the Authors

Christina Bouwens is a project engineer with CAE-Link Corporation, Binghamton, NY. She has over five years experience in simulator networking. She currently serves as systems engineer for several CAE-Link projects that use Distributed Interactive Simulation (DIS) and Aggregate Level Simulation Protocol (ALSP). Before joining CAE-Link, Ms Bouwens was the program engineer for the DIS standards program at the Institute for Simulation and Training. Ms. Bouwens has a Masters of Science in Mathematical Science from the University of Central Florida and a Bachelor of Science in Mathematics from Geneva College. She has published several papers on the subject of DIS and interoperability. Ms. Bouwens chairs the DIS working group for Communication Architecture and Security.

Ron Matusof is a staff engineer with CAE-Link Corporation, Binghamton, New York. He has over eleven years of experience in tactical, radar, and image simulation and in simulator networking. He has been an active participant in the DIS standards development process since the first DIS workshop in 1989, most notably in the area of fidelity definition. Mr. Matusof is currently the Principal Investigator for several CAE-Link research and development programs. He holds a Bachelor of Science in Electrical Engineering from the University of Pittsburgh. He has published several papers on the subjects of interoperability, mission rehearsal, image generation and cue correlation.

PREDICTING NETWORK PERFORMANCE IN HETEROGENEOUS, MULTI-FIDELITY, SIMULATION NETWORKS

Christina Bouwens and Ron Matusof
CAE-Link Corporation
Binghamton, New York

INTRODUCTION

Simulation networking is not a new concept and its use in large scale exercises has moved beyond proof-of-principle and into production. We now have a standard for the exchange of information between networked simulations (IEEE-1278-1993)[1], but the supporting infrastructure for implementation of a simulation network has not yet been completed. This leaves the designers of simulation networks with a myriad of questions concerning the implementation of a simulation network and little concrete methodology for predicting how the network will act under a variety of conditions.

There has been very little work published concerning prediction and accurate measurement of simulation network loading, and the implications of network loading toward network fidelity and successful transfer of training are not well understood.

Our overall objective is to develop methodologies for predicting simulation network performance and for determining its impact upon fidelity and transfer of training. This paper concerns itself only with the first part of the objective: prediction of network performance.

BACKGROUND

A simulation network is an arrangement which allows two or more simulations to communicate. A simulation network is a conceptual arrangement. It does not imply a particular type of communication media nor does it imply a set of communication protocols. These are implementations of a simulation network, and for a given network there are a large number of potential implementations.

Generally, simulation networks are governed by a network architecture. The architecture provides a set of design principles for the network implementation. Network implementations can be viewed from two perspectives -- the physical network and the virtual network. The physical network (Figure 1) describes the schematic and topological connection between network nodes, including the placement of nodes, the media through which nodes communicate, and the hardware/software which allow communication to occur. The virtual network (Figure 2) describes the logical interconnection between simulations, defined solely in terms of the flow of data and control.

The designers of simulation networks must be keenly aware of network performance. Network performance is the functional effectiveness of a network, and in the case of a simulation network, it is based on both the physical and virtual network implementations. The issues associated with measuring simulation network performance are derived from the network itself, the differences between individual simulation designs and differences in the level of realism across the network.

SIMULATION NETWORKING

Most simulation networks are a representation of a parallel processing methodology known as asynchronous data flow. Asynchronous data flow architectures are those in which the messages which flow between nodes provide control and synchronization of the system. In the data flow model, nodes are executed simultaneously, yet independently from each other. The output of the node depends only on the input and the function(s) that the node performs.

Simulation networks fit the asynchronous data flow paradigm in that each simulation is responsible for its own actions. Each simulation executes independently and simultaneously with other simulations on the network. All simulations are treated identically by the network because the functions of the simulation node depend only on the inputs provided by the network and the functions of the simulation.

Assessing the performance of network architectures is usually quite difficult due to the large number of parameters found in these systems. Because in asynchronous data flow architectures the nodes are functionally defined and are synchronized by the data

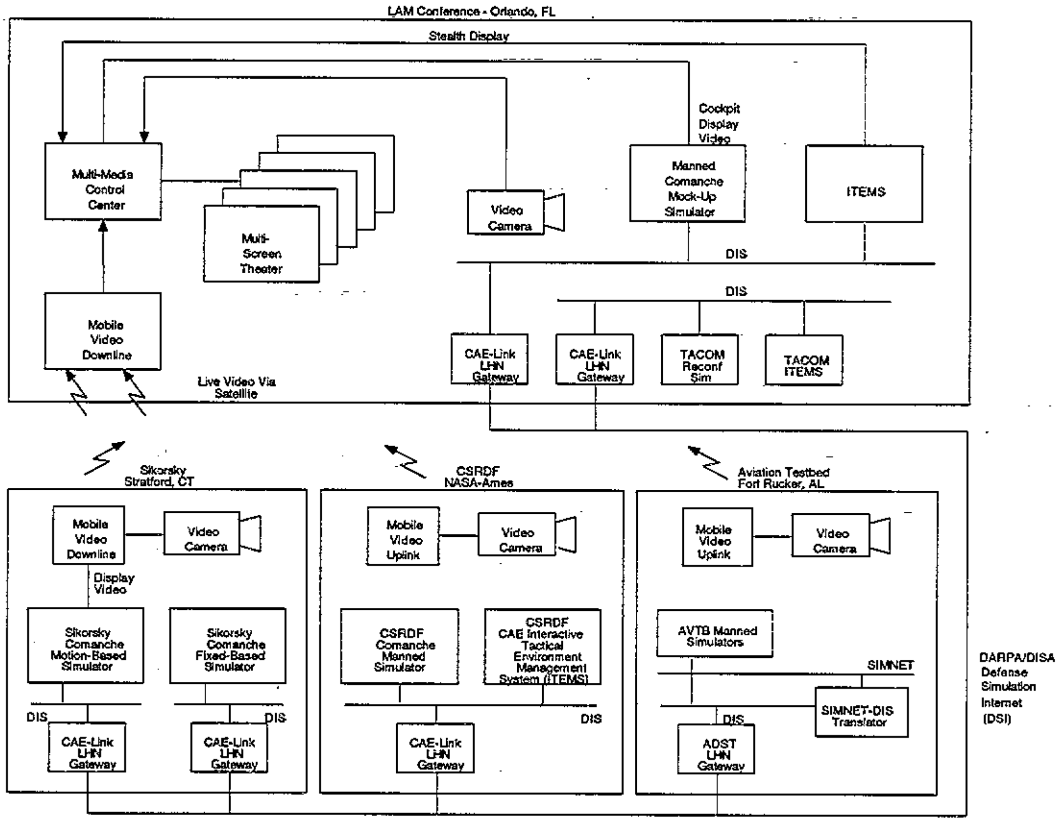


Figure 1
Physical Network

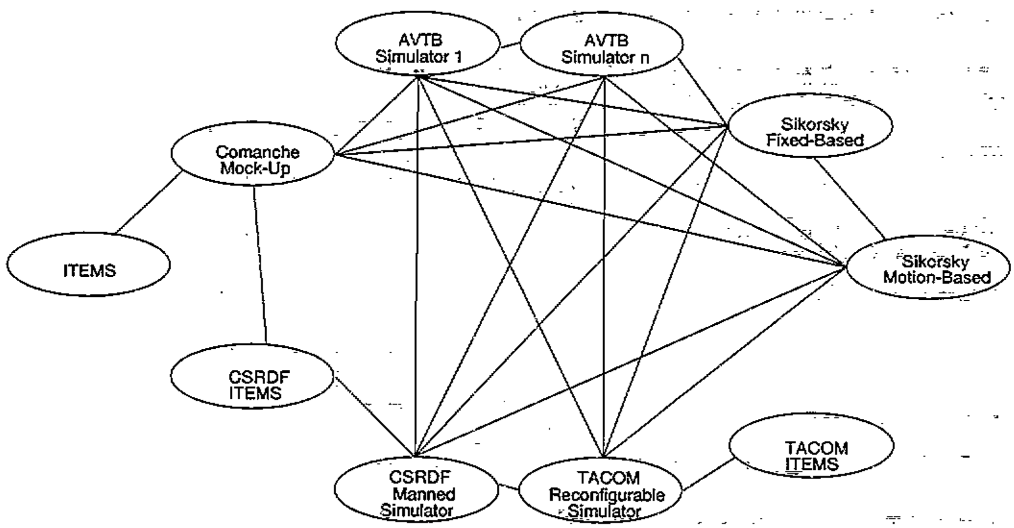


Figure 2
Virtual Network Example

that flow through them, meaningful prediction schemes can be developed which are independent of the characteristics of individual nodes. These schemes involve measurement and prediction of network loading based solely on the data flow. Schemes such as these can account for issues such as heterogeneous or multi-fidelity simulation.

HETEROGENEOUS SIMULATIONS

A simulation network is often characterized by the similarity between the individual simulations which comprise it. This characterization is associated with the physical network, and is usually divided into two domains: homogeneous and heterogeneous networks.

Homogeneous networks are composed of simulations which are essentially identical in design. Early distributed simulation networks, such as SIMNET, fall into this category. It is a relatively easy task to predict the network performance of homogeneous networks, since the interactions between two simulations on the network can be linearly extrapolated to almost any network size.

Heterogeneous networks, on the other hand, are composed of simulations of different design. Different vendors producing simulations for identical specifications will generally implement the simulations in different ways. For each simulation the implementation will probably be fully compliant with the specification, yet will likely vary greatly from other implementations.

DIS supports the networking of heterogeneous simulations. Although two heterogeneous simulations may identically meet a simulation specification for a particular non-networked application they may produce significantly different results in a networked environment. Because of this, it is more difficult to predict the network performance of heterogeneous simulations than for homogeneous simulations.

MULTI-FIDELITY SIMULATIONS

Compounding the problem of heterogeneous simulations is another problem concerning multi-fidelity simulations. A multi-fidelity simulation is one that has varying levels of fidelity depending upon its application. Fidelity is a characteristic of the virtual network and, in this case, is described as the degree of similarity between a simulation and the real world[2].

In simulation networking, multi-fidelity networks can be constructed where the simulations on the network are not necessarily of identical fidelity. This may occur because simulations on the network are designed to different specifications, they are designed to the same specification but implemented differently, or identical implementations of a specification or interfaced to the network in different ways and therefore behave with different levels of fidelity in the network environment *(due to different filtering schemes, for example)*.

NETWORK PERFORMANCE

The designers of simulation networks will be expected to meet certain performance criteria for a particular simulation networking application. Unfortunately, the designer is left with almost no information as to how to predict network performance. Typical network performance criteria center around the physical constraints of the network, such as bandwidth and latency. While these are important criteria in determining overall network performance, they have little meaning without a corresponding set of virtual performance measures. There has been little research investigating the role of the virtual network in overall network performance.

To help determine the role of the virtual and physical network on overall performance, we reviewed data from five network exercises:

1. MULTISIM Experiments at Fort Rucker (1988):[3] This exercise involved the interconnection of four homogeneous, multi-fidelity devices via a proprietary (non-DIS) synchronous network transfer mechanism.

2. Project Desert STAARS (1991):[4] This exercise involved the development of a heterogeneous, multi-fidelity network of virtual and constructive simulations interconnected via a proprietary (non-DIS) synchronous transfer method.

3. I/ITSEC Demonstration 1 (1992):[5] This exercise was the first large-scale public demonstration of DIS, involving 18 manned and unmanned simulations, 22 listen only devices, and 1 live device. The network was multi-fidelity and heterogeneous and communicated using DIS 1.0 protocols.

4. I/ITSEC Demonstration 2 (1993):[6 7] These exercises were a large-scale DIS exercise involving an increased number of manned and unmanned simulations, listen only devices, and live devices. Also included in this data is a SIMNET data stream from the Wright Flyer simulation of the DoD Dependent School demonstration. Again the network was multi-fidelity and heterogeneous and this time communicated using a slightly modified DIS 2.0.3 protocol.

5. CELLNET (1994): This was a small-scale exercise connecting a heterogeneous. multi-fidelity network of virtual and constructive simulations. These simulations were interconnected via DIS protocols implemented as an application layer transported via a synchronous transfer method.

Our goal in selecting these exercises was to allow us to study the effect on network performance as the construct of the network varies. We studied both homogeneous and heterogeneous networks for both DIS and non-DIS applications. Network transfer schemes varied and included both synchronous and asynchronous methods. In all cases, the networks were multi-fidelity.

We must point out that all five of these exercises were experimental applications of simulation networking and that there is no conclusive evidence for the validity of our observations. However, there were some very interesting trends which we observed.

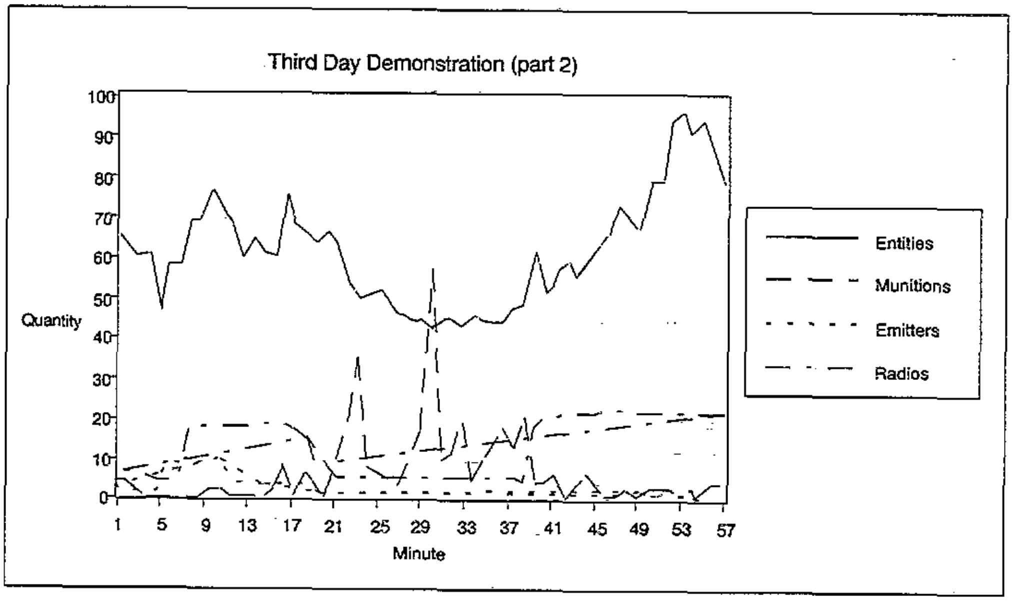
When evaluating network performance, the goal is to define how changes in either the virtual or physical network affects task performance The performance of the network is limited by the characteristics of both the physical and virtual network and by the mapping between the two. Bandwidth, latency, and throughput appeared to have the most pronounced impact on the performance of the physical network. Data synchronization and the interrelationship between network state updates (such as the issuance of PDU's) appeared to have the greatest impact on the performance of the virtual network. After reviewing our sample simulation networks, we noted the following trends:

Bandwidth: Bandwidth appeared to have no effect on network performance. Spare bandwidth ranged between 42% in the DIS flooding experiment during the 1993 I/ITSEC to 92% in the MULTISIM experiments. However, each of these exercises involved a small number of simulated entities. By definition, virtual networks have unlimited bandwidth. Therefore the design of a virtual network may be constrained by the bandwidth limits of the physical network. As a result, the designer of simulation networks must conciously determine if the virtual network can be appropriately mapped within the physical bandwidth limitations. Obviously, the mapping problem will get worse as the size of the virtual network (that is, the number of entities) grows.

Throughput: Throughput is the data capacity of a network. The throughput for the MULTISIM and Desert STAARS networks was almost constant over all applications, while the other networks exhibited "spikes" of activity of up to 35 kilobytes/second. These spikes appear to be related to the activity of entities on the network, and become significantly larger as the simulation workload increases. In the virtual network, data throughput is unlimited and spikes of activity pose no significant problem. However, the throughput of the physical network is constrained and these spikes affect the overall capabilities of the simulation network. In the data from the DIS exercises, the spikes increase in size and frequency when emission or radio PDUs are issued. Interestingly, there appears to be no correlation between the issuance of munitions PDUs and activity spikes (Figure 3).

Latency: The network designer, while concerned with the actual network latency, is more concerned with the effective latency of the network. Effective latency is the delay measured between an action initiated in one simulation and the action's representation by another simulation. It includes the latency of the physical network hardware as well as some additional delays introduced by the implementation of the network[8]. These delays include network transfer delay, network protocol delay, network transmission delay, network filtering delay, and network encryption delay.

We have limited empirical data concerning most latencies of the networks we studied. However, there is fairly good data concerning network transfer delay (the amount of time it takes to physically move data from a simulator to a network node). This delay ranged from 16 milliseconds to 200 milliseconds in the simulations for which it was measured. The total network transfer delay for a given interaction is the sum of the network transfer delay at the sending and receiving nodes. This means that in our sample networks, a maximum network transfer delay of 400 ms could occur.



	Entities	Munitions	Emitters	Radios
Averages	59.55932	6	2.101695	12.64407
Maximum	93	56	9	22
Minimum	41	0	0	4

Start of file: Wed Dec 1 20:00:00 1993
End of file: Wed Dec 1 20:59:59 1993
Total number of DIS PDUs 220717

Figure 3
Activity Spires vs. Issue of Munition PDU
(Courtesy Dr. Sandra Cheung[7])

There is insufficient data from the exercises that we reviewed to determine the effect of protocol delay (the delay introduced to a data stream due to the choice of network protocol) on overall network performance. However, subjective comparisons between the Desert STAARS and CELLNET networks (two networks which were different only in that one used a proprietary protocol while the other used DIS), revealed no discernible changes due to differences in protocol. Similar subjective observations[9] have been made between SIMNET and DIS applications.

Our limited data from the DIS exercises indicates that queuing delay (the delay which occurs as messages queue to be processed by a network node) had an insignificant impact upon network performance, provided that sufficient buffering exists at all network nodes. Networks appear to "deadlock" (message traffic ceases even though physical network is still active) when network receiving buffers overflow. More research is required before meaningful conclusions can be drawn.

Network filtering delay (the delay introduced by processing of asynchronous network state data updates) appears to become more pronounced as the allowable deviation between dead reckoned and actual entity state decreases. This is contrary to what we had expected, since with higher dead reckoning tolerances, more smoothing is required. Our theory is that a major cause of the filter delay is related to the rate at which entity state information is available from the network. When smaller tolerances are used, the dead reckoned position is corrected (and subsequently passed on the network) more often.[10] This implies that, on the average, there are more entity state PDUs to read. Since information is read from the network in a serial manner, an increase in the number of entity state PDU's implies that more time will be required to read and filter this information.

For example, the coordinate conversion processing measured at one node of the 1993 I/ITSEC demonstration was found to range from 30 µsec/entity when converting from geocentric to geodetic coordinates to 70 μsec/entity for the reverse transform.[11] There is, therefore, a 30 μsec/entity penalty to pay for each entity state PDU received. The more entity state PDUs are issued, the greater the filter delay. Interestingly, the coordinate conversion processing times were found to vary based on the desired accuracy of the conversion (a 0.004 foot error required four iterations of the conversion).

We believe that a similar effect will happen for encryption and transmission delays, but there is insufficient data in the exercises that we studied to either support or disprove this contention.

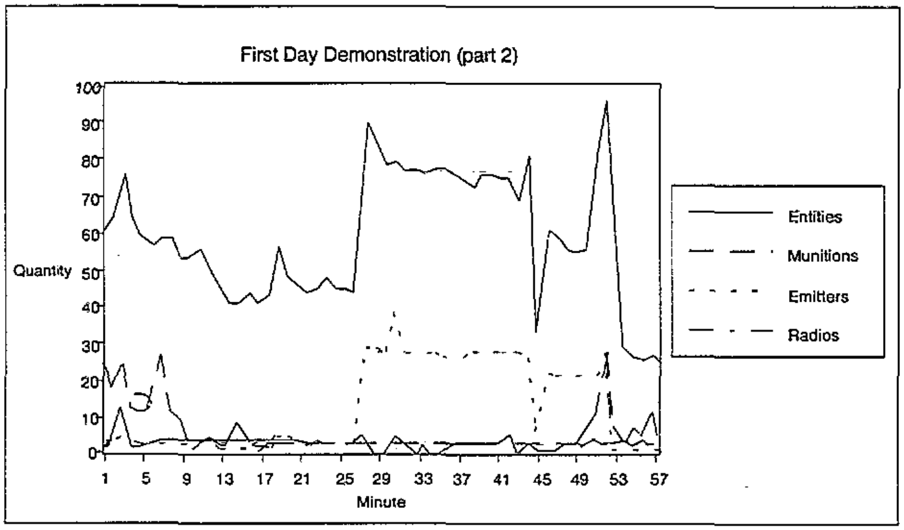
Delay dispersion: Unlike the cues in a single simulation, the latency of individual cues on the network has a component which is both random and unbounded. Therefore, not only is the delay of an individual cue important, but the variation in the length of the delay (an effect known as delay dispersion) is important as well. The variance in delay can cause a disordering of packets such that the sequencing of state information (player position and velocity, for example) is incorrect. We observed no dispersion in the MULTISIM and Desert STAARS synchronous networks and a very small amount of delay dispersion in the other networks. There is no evidence that packet disordering caused any network performance problems.

Data Synchronization: The simulation network represents an asynchronous data flow architecture. Asynchronous data flow architectures are, by their nature, synchronized by the flow of data between nodes. In this architecture, data is synchronized only at the network nodes. The simulations themselves are not synchronized by the data flow. Therefore, we often observe data synchronization problems in all asynchronous applications (including CELLNET, where DIS was applied as an asynchronous application layer over a synchronous transfer method). In synchronous virtual networks, data flow is, by definition, time coherent. When coherence was lost (due to loss of synchronization signals, for example), we observed catastrophic failure in that the system could not automatically resynchronize and erroneous data was produced. The skewing of data and the inability to resynchronize it caused several cases of "extrapolation induced oscillation." When this occurs, extrapolations based on erroneous data produce increasingly inaccurate results until the extrapolations themselves become unstable and the simulation becomes unusable.

Interrelationship of PDUs: Certain information fields, such as position and attitude, are repeated in several different PDU types. The assumption is that an antenna, which is generally offset from the center of a vehicle, may move out of a positional tolerance without the vehicle moving at all. For this case, we would need to perform a tolerance check on the position and attitude of the antenna, and issue new PDUs whenever the antenna goes out of tolerance. In the DIS applications that we studied, and in particular the 1993 I/ITSEC demonstration and the CELLNET exercise, we observed a stunning interrelationship between Entity State PDU generation, Emission PDU generation, and Radio Emission PDU generation. Increases in the generation of either Emission or Radio PDUs resulted in a two-fold increase in the generation of entity state PDUs (Figure 4). This was highly unexpected, but can be observed in all 1993 I/ITSEC DIS demonstrations and in recorded data from the CELLNET exercise. The implication of this trend is that the use of emission or radio PDUs may affect network performance in a disproportionate manner than other types of PDUs. We believe that more information must be gathered before this trend can be considered more than coincidental.

Dead Reckoning Thresholds: Dead reckoning thresholds directly affect the amount of entity state traffic on a DIS network. It has been shown that network traffic can be reduced by up to eighty per cent by using a dead reckoning algorithm.[12] However, this reduction in network traffic was accomplished by allowing vehicle appearance to vary up to three degrees in rotation and up to ten per cent of the vehicle's dimensions in position before a state update is required. In all of the DIS applications that we studied, the threshold was always set to 1 meter and 3 degrees. Therefore, we have no data to determine the impact of varying the thresholds in these exercises.

Non-Simulation Network Traffic Non simulation traffic appeared to be a problem in the I/ITSEC demonstrations. It is reasonable to assume that simulation networks will not, in general, occur on pristine networks. Therefore, the non-simulation network traffic must be quantified prior to any prediction. Again, we were unable to quantify the effect of non-simulation traffic on the performance of any of the exercises that we studied.



Averages	53.79661	4.644068	11.38983	2.4068
Maximum	88	25	35	11
Minimum	22	0	1	1
StDev	16.29341	6.180511	11.08745	1.274669
Totals	151	73	40	15

Start of file: Tue Nov 30 11:00:00 1993
End of file: Tue Nov 30 11:59:59 1993
Total number of DIS PDUs 266055

Figure 4
Issue Rate of Emission vs. Entity State PDUs
(Courtesy Dr. Sandra Cheung[7])

MEASURING NETWORK PERFORMANCE

Network performance is a complicated combination of a number of factors. We have tried to define a reasonable measurement paradigm which allows the network designer to determine the feasibility of a network design prior to its implementation. Our methodology relies on determining the available bandwidth and effective latency of the network, and from these we determine the maximum number of entities that can be supported by the network given the exercise requirements that the network must support.

We have based part of this analysis on work presented at the 1993 I/ITSEC.[13] In this work, a four step program was outlined to estimate bandwidth requirements:

1. Document assumptions about minimum attributes of each entity class represented
2. Estimate the exercise bandwidth requirement to approximate actual PDU issue rates
3. Determine the number of entities (and tactical links) required for an exercise
4. Calculate exercise bandwidth based on these individual estimates.

One problem with this methodology is that it relies heavily on an exercise designer's ability to estimate exercise requirements. Additionally, this does not address the performance issues introduced by mapping the virtual network onto the physical network. Our hope was to develop a similar methodology which does not rely on the exercise designer's a priori knowledge of networking.

Our methodology involves determining a set of equations which can be used to determine the worst case latency and bandwidth of the physical network, applying knowledge of the intended exercise and the simulations involved to determine worst case PDU issue rate (a characteristic of the virtual network). We then combine these performance measurements to determine the maximum number of entities the network can support under worst case conditions. In other words, we map the virtual network onto physical network, and then bound the network performance by the physical constraints.

The latency of the system is heavily dependent upon applications and upon the transmission medium and protocol selected. For any individual path, the worst case latency is the sum of the individual delays in the system, namely the transfer delay (D_t), the protocol delay (D_p), the queuing delay (D_q), the filtering delay (D_f), the transmission delay (D_x), the encryption delay (D_e), and the worst case delay dispersion (D_{disp}). The first four factors are multiplied by two to account for delays at both the transmitting and receiving nodes (Figure 5).

$$L_{wc} = 2\,(D_t + Dp + D_q + D_f) + D_x + D_e + D_{disp}$$

The maximum bandwidth of any network is a known quantity. It is physically impossible for a network of simulations to exceed this bandwidth. The effective bandwidth available to a simulation node is determined by taking the maximum bandwidth and subtracting out the overhead due to protocol, and the bandwidth used for non-simulation network traffic.

$$BW_{eff} = BW_{max} - BW_{overhead} - BW_{other}$$

Bandwidth may affect the message delay time at network nodes in cases where variable intensity traffic exists (such as most DIS exercises). In these cases, the network may be modeled as a Poisson message data stream, and the effects of limiting bandwidth on the network queues can be predicted in a relatively straightforward manner.[14] One can then predict if a exercise will meet a particular network requirement. For the networks we reviewed, bandwidth had no effect on latency (or vice versa), since the worst case loading still had 42 per cent spare capacity.

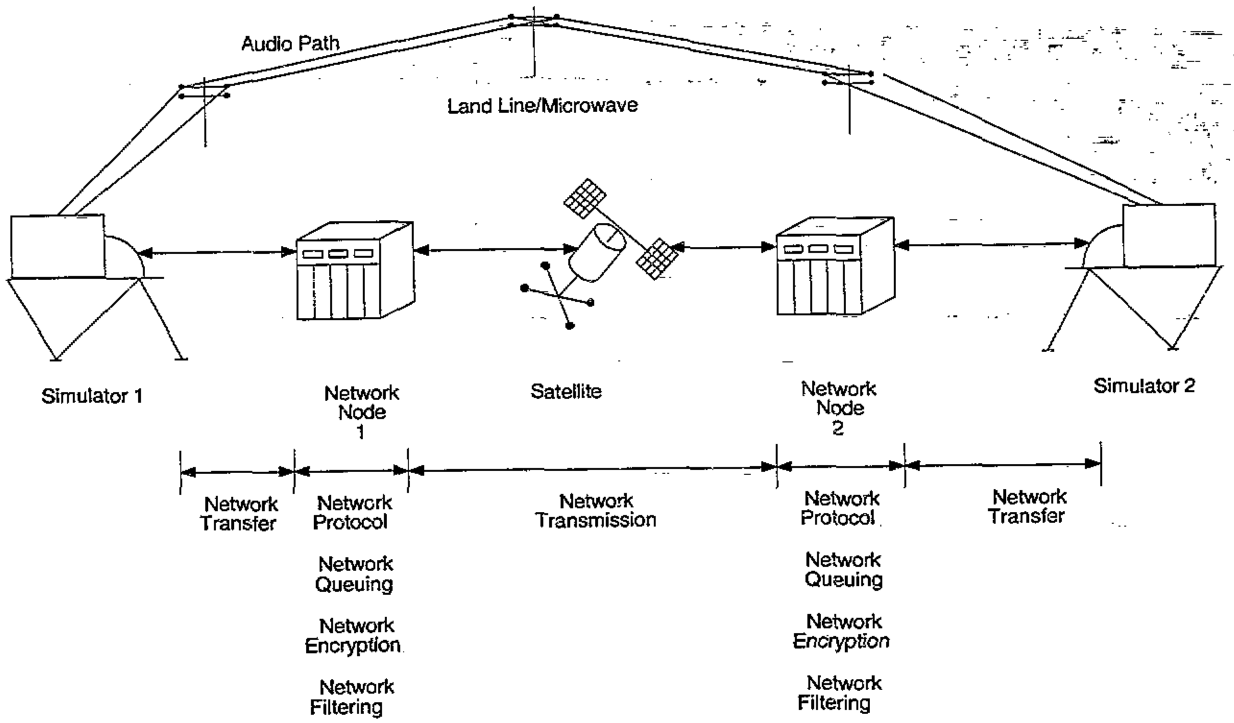


Figure 5
Typical Time Delays for Long-Distance Networked Systems

In order to determine the maximum number of entities which a network can support, we must first determine the worst case rate at which PDUs will be issued. This is done on an entity by entity basis In the networks that we studied, six PDU's (entity state, detonation, emission, transmitter, signal, and laser) make up the bulk of the message traffic. Based on an analysis of the data we studied, we found that this traffic accounted for an average of 98.3 per cent of the network traffic. On the average, PDU size was 1384 bits. Using the issue rates obtained from our sample exercises, we determined that a worst case average of 200 bits per second is required for all other PDUs. We can therefore aggregate all of these other PDUs into one representative PDU with an issue rate of 200 bits/second. We intend to adjust this aggregate representation as we analyze more data. Using the formulae published in the proposed IEEE Standard Draft version 2.0.3,[15] we can get a rough idea of the predicted load. These formulae are similar to those derived by Doris and Loper[16] in 1993, but include DIS 2.0.3 Draft Protocols. Our intent was to refine these equations to also take into account the worst case rate of issue, R, of the PDUs.

R is determined by looking at the threshold values set for the dead reckoning algorithm, an entity's capabilities and the entity's dimensions. PDUs are issued whenever the difference between an entity's dead reckoned position (P_{dr}) and its actual position (P_{act}) exceeds the positional tolerance (T_p). The tolerance can be represented in terms of the entity's velocity by substituting velocity-time products in place of *instantaneous positions:*

$$T_p = |P_{dr} - P_{act}| = \Delta t(V_0 - V_{n-1})/\text{frame rate}$$

where V_0 is the velocity at the time that the last entity state PDU was updated for a given entity, V_{n-1} is the velocity calculated by the last pass of the real-time simulation of the entity, and the frame rate is the iteration rate of the simulation. The worst case occurs at maximum entity velocity (V_{max}). Solving for Δt and substituting V_{max} for V_0 yields:

$$\Delta t = 1/V_{max} (T_p + V_{n-1}/\text{frame})$$

In the worst case, tolerance is simultaneously broken in both position and orientation ($T\omega$). In this case, the maximum value of V_{n-1} is the projection of the dead reckoned velocity onto the actual velocity vector, where ω is the angle between the two vectors. Substituting the projected V_{max} for V_{n-1} we get:

$$\Delta t = T_p /V_{max} + \sin (T\omega)/\text{frame} \text{ and } R = 1/\Delta t$$

Our table of formulae, then, is shown in Figure 6:

PDU	FORMULA for size estimate	where:
Entity State	R(1152 + 128A)	A = # of articulated parts
Detonation	800 + 128H	H = # of articulated parts hit
Emission	R(192 + E(160+ 8(416+64T)))	E = # of emitters
Transmitter	R(768 + S_mM)	B = # of beams per emitter
Signal	256 + L	T = # of targets in beam
Laser	576	M= # of Modulation parameters
Other PDUs	200	S_m = Size of modulation pattern m
		L = Length of data stream

Figure 6 PDU Sizing Formulae

We next sum all PDU issue rates (in bits/second) over all of our entities in order to determine the loading of the network:

$$\text{Virtual Load} = \sum_{\text{entities}} \text{PDU bits/second}$$

Finally, we determine the maximum number of "average" entities which can be supported by this network.

$$\text{Max Entities} = BW_{eff} /\text{Virtual Load}$$

This number can be used for planning purposes. It represents an average worst case for the network, given the physical constraints of the network and the exercise goals. The network designer can now assess alternatives and their impact on the network's physical and virtual design. For example, the designer may choose to improve the accuracy of geo-positioning., but does so at the expense of increased latency.

The formula for max entities is reciprocal and can be used to derive the required bandwidth given a desired number of entities with known capabilities:

$$BW_{eff} = \text{Max Entities} * \text{Virtual Load}$$

CONCLUSIONS

We have developed a method for predicting the maximum number of entities that can play in a network exercise given the constraints of the physical network, the exercise objectives, the characteristics of the virtual network, and latency requirements of the exercise. The observations and derivations which we have made in this paper are based solely upon five experimental exercises conducted over the last several years and published research.

A great deal of research remains to be conducted in this area. Our data set was limited and the data was not collected with the expressed purpose of developing prediction methodologies for network performance. Future experimentation concerning the effects of bandwidth, latency, delay dispersion, data synchronization, interrelationship between PDUs and implications of bandwidth reduction methods (such as dead reckoning) remain to be conducted. Specific future areas of research concerning network performance include:

- Protocol Delay
- Queuing Delay
- Encryption Delay
- Transmission Delay
- Non-simulation Network Traffic
- Interrelationship between PDU types
- Impact of varying latency
- Analysis of operational systems (as opposed to experimental or demonstration)
- Comparison between live, virtual, and constructive simulations.
- Impact of network performance on fidelity
- Impact of performance on transfer of training

As a further step in this study, we intend to apply our methodology toward prediction of upcoming DIS exercises (including the 1994 I/ITSEC demonstration) and fine tune this algorithm as needed.

Prediction of network performance will provide only part of the information that the network designer must know prior to building a simulation network. Even the best prediction algorithm will not allow the designer to develop successful network exercises unless it is accompanied by knowledge of how the network implementation affects transfer to the real-world. When equipped within this knowledge, simulation networking can reach its true potential.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the invaluable assistance of Dr. Sandra Cheung from the University of Central Florida Institute for Simulation and Training and Dr. Joshua Seeger from BBN Systems and Technologies. The authors also acknowledge the efforts of STRICOM in supporting DIS research initiatives.

REFERENCES

[1]IEEE Standard 1278-1993, IEEE Standard for Information Technology -- Protocols for Distributed Interactive Simulation Applications Entity Information and Interaction, IEEE Computer Society, May, 1993.

[2]Hays, R. T., "Fidelity: A Concept Paper", Army Technical Institute Technical Report 490, Alexandria, VA, November, 1990.

[3]Dees, J.W. and Cornett, T.R., "Simulator Networking in Helicopter Air-to-Air Combat Training", AIAA Flight Simulation Technologies Conference, Boston, MA, September, 1989.

[4]Monette, R., Knight, S.K., and Goodwin, P., "Sustainment Training for Army Aviation Readiness through Simulation", Interservice/Industry Training Systems Conference, Orlando, FL, December, 1991.

[5] Loper, M., Goldiez, B., and Smith, S., "The 1992 I/ITSEC Distributed Interactive Simulation Interoperability Demonstration", Interservice/Industry Training Systems and Education Conference, Orlando, FL, December, 1993.

[6]Seeger, J., "Networked Oriented Scalability", Proceedings of the 10th Workshop for the Interoperability of Defense Simulations, Orlando, FL, March, 1994.

[7]Cheung, S.E., "I/ITSEC 1993 DIS Interoperability Demonstration Traffic Analysis", Proceedings of the 10th Workshop for the interoperability of Defense Simulations, Orlando, FL, March, 1994.

[8]Sowler,R.J., and Matusof, R., "Issues Concerning Cue Correlation and Synchronization of Networked Simulators", AIAA Flight Simulation Technologies Conference, New Orleans, LA, August, 1991.

[9]Panzitta, M.J., and Moore, R.G., "Visual System Interoperability Between CCTT and SIMNET", IMAGE VII, Tucson, AZ, June 12-17, 1994.

[10]Economy, R., Ferfuson, R., and Pollak, E., "Geographic Position Accuracy in a Distributed Simulation Environment", IMAGE VII, Tucson, AZ, June 12-17, 1994.

[11]Ibid.

[12]Miller, D., Pope, A., Waters, R., "Long-Haul Networking of Simulators", 10th Interservice/Industry Training Systems Conference, Orlando, FL, December, 1988.

[13]Doris, K. and Loper, M., "DIS Network Traffic Analysis Estimation Techniques", Interservice/Industry Training System and Education Conference, Orlando, FL, December, 1993.

[14]Golovko, N.I., "Message Delay Time at Network Node for the Case of Incoming Traffic of Variable Intensity", Avtomatika i Vychislitel'naya Tekhnika, Volume 23, Number 2, 1989.

[15]Proposed IEEE Standard Draft, IEEE Standard for Information Technology -- Protocols for Distributed Interactive Simulation Applications Version 2.0, Third Draft, IST CR-93-15, Institute for Simulation and Training, May, 1993.

[16]Doris, K. and Loper, M., "DIS Network Traffic Analysis Estimation Techniques", Interservice/Industry Training System and Education Conference, Orlando, FL, December, 1993.