World Multiconference on Systemics, Cybernetics and Informatics

July 22-25, 2001
Orlando, Florida, USA

PROCEEDINGS

Volume XIX

Cybernetics and Informatics: Concepts and Applications (Part III)

Organized by IIIS
International
Institute of
Informatics
and Systemics

Member of the International
Federation of Systems Research
IFSR

Co-organized by IEEE Computer Society
(Chapter: Venezuela)

EDITOR
Nagib Callaos
Maria Sanchez

CONTENTS

VOLUME XIX

CYBERNETICS AND INFORMATICS: CONCEPTS and APPLICATIONS (PART III)

Modelling and Description Techniques for Systems Management – Invited Session
Organizer: Manuel Rodríguez (Spain)

Synchronization in the Brain as a Neural Basis for Synthetic Intelligence – Invited Session
Organizer: Yoko Yamaguchi (Japan)

Computer Applied in Engineering – Invited Session
Organizer: Qingying Zhang (China)

Information Systems Development, Methods and Methodologies

Mobile / Wireless Computing

Communication Systems

Control Systems

Communication Systems

Bandwidth And Service Quality Management In the Internet – Invited Session
Organizer: Junaid Zubairi (USA)

Robotics, Sensory Systems for Robotics and for Human-Machine Interface – Invited Session
Organizer: Milan Kvasnica (Czech-Republic)

Collective Algorithms and New Heuristic for Problem Solving

SCI in Biology and Medicine

Control Systems

Modeling, Graphs, Algorithms and Programming Techniques

Control Systems

Virtual Engineering

Enhancing Quality of Service in TCP/IP Using Active Networks*

M.A Badhusha, M.I. Buhari, S.B. Junaidu, M.H. Saleem
{akbar,mibuhari,sahalu,msaleem}@ccse.kfupm.edu.sa
Information and Computer Science Department
King Fahd University of Petroleum and Minerals, Saudi Arabia.
Dhahran 31261, Saudi Arabia.
Phone: 966 3 860 4374
Fax: 966 3 860 2174

Abstract:

Good quality of service (QoS) in packet-switching networks is becoming a high priority in network design and research due to ever-growing network bandwidth and intensive network applications. Failing to address QoS issues may lead to problems like long delay of message delivery, waste of system resources, and possible network collapse. In this paper we propose a method of enhancing QoS in computer networks by making use of the traditional source quench algorithm of congestion control in an active networks setting.

Closed-loop congestion control algorithms (like the source quench approach) generate feedback information in response to the traffic conditions in the network, such as when the queue length in a switch rises beyond certain limit. The feedback information is sent between the destination and the source. Although these schemes can explicitly distribute indications of resource utilization and traffic conditions throughout the network, they act slowly specifically in the case of high channel propagation delay found in ATM networks.

We use active networks to enhance the responsiveness of these schemes by applying the source quench algorithm at the router levels inside our network model. OPNET-based simulation of our active network model exhibits about 30% performance enhancement over the same network model with passive routers.

1. Introduction

The emergence of Active Network technology has attracted many academics and researchers to become involved in the development of this technology. Active network has potentially much to offer the Information Technology world. First, exploiting this technology can reduce the time required for the standardization process of new network services. Second, active network shifts the conventional network paradigm: from a passive node that only transfers bits to a more general processing engine like an end station which supports customized computation on user's data. Furthermore, active networks can also be used for enabling on-the-fly modification of network functionality, for example to adapt to changes in link conditions.

The use of active networks potentially introduces a high amount of computations taking place inside network routers. Parallel processing can be used to speedup those potential computations. We envisage that parallel processing will be a powerful tool when used in conjunction with active networks especially in a LAN environment: when the load distribution on the nodes in a LAN is skewed with some machines heavily loaded and others lightly loaded these two technologies can be used to good effect taking advance of low-communication latency of a typical LAN.

In this article, we set out to use a synergy of these two techniques in a LAN environment to investigate the prospects of adding user-customized computations inside networks. Another goal of this paper to use parallel processing to improve the performance and resource utilization of the underlying active networks system in a LAN setting.

The remainder of this paper is organized as follows. We outline the related work in Section 2. Our work is described in Section 3. Our network model and processing part of the host and the hub are then presented in Section 4. The Experimental analysis and results are discussed in Section 5. Section 6 concludes.

2. Related work

Studies on congestion control, which outline measures for controlling network traffics in order to prevent, avoid, or recover from network congestion, have long been considered significant for the future development of network communications. A large number of various congestion control schemes have been proposed, and a few mechanisms have been implemented in real networks, such as the control methods in IBM's System Networking architecture (SNA) [1], Digital's Networking Architecture (DNA) [2], and the Internet [3, 4]. However, despite years of research efforts, the problem of network congestion control remains a critical issue and a high priority, especially given the prospective of the continually growing speed and size of future networks.

The existing approaches for network congestion control cover a broad range of techniques, including window (buffer) flow control [5], source quench [6], slow start [3], schedule-based control [7], binary feedback [8], rate-based control [9], etc[10].

It is often difficult to characterize and compare various features among different congestion control schemes. Current literature in the field classifies most congestion control approaches into two categories: approaches for congestion avoidance, and approaches for congestion recovery. Such a simple classification only provides a very general picture of common properties between separating groups of approaches. A detailed taxonomy is required in order to help researchers and engineers understand the similarities and differences among

* This work was supported by King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia.

various schemes, and to decide which techniques are best suited for particular designs.

In order to take any action against the problem of congestion and take necessary actions, we need to use the amount time of time taken for the data to be transferred from the source to the destination and so on. But, this makes the whole job difficult due to the following reasons.

3. Our Work

To develop the IP/SQ algorithm, we have used the OPNET simulation tool. We have modeled and simulated two scenarios where the network processing is done at the router level and where the network processing is done at the host level.

4. Our Network Model

The packet format in our model is made up of a source address, a destination address and the type of packet being transmitted. Type 0 is the normal packet that travels from one node to another. In case of heavy traffic or when the packet arrival rate is more than the capability of the receiver, the receiver sends a source quench (SQ) packet that is of type 1. Upon receiving the SQ packet, the source reduces the pace at which it sends packet.

The basic model of the network is made of 4 nodes named from 0 to 3, which are connected to each other through a hub [as shown in Figure 1]. The host computers handle the packet that carries destination address of the packet. The packet generation is handled by the hosts, which posses the ideal packet generators [src in Figure 2] in them. The generated packets are then processed and transmitted through the transmitter. The hub is made up of a processing node connected to many transmitters and receivers, depending on the number of hosts that ought to be connected to it. The node part of the host and the hub are shown in Figure 2.

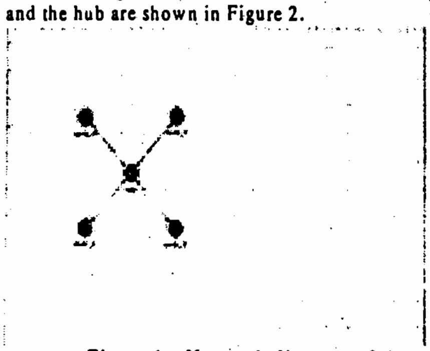


Figure 1. Network diagram of the system

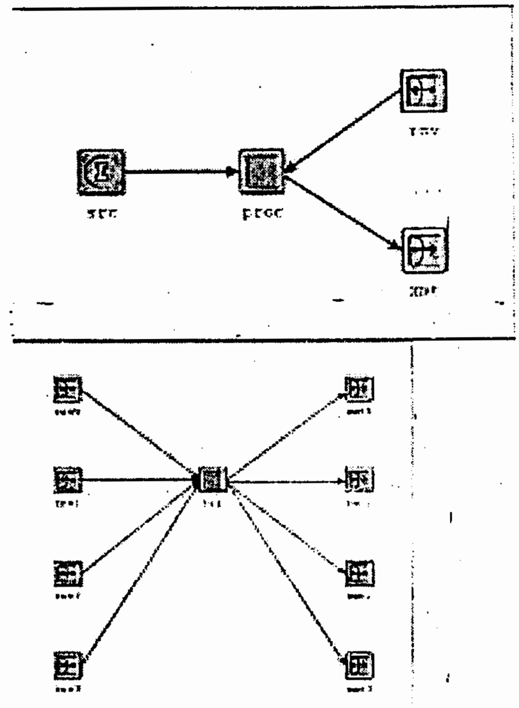


Figure 2. Node diagram of the host and the hub

- **Processing part of host and hub**

The processing portion of the host and the hub are shown in Figure 3. The host's "init" part is used to make the packet destination address generation to be between 0 and 3 because the network has 4 nodes only. The generated packets are then forcibly sent to the idle part. The transmitter then transmits the packet along any one stream depending upon the destination address specified.

At the hub, the checking for the destination address is done and packet is forwarded in the respective transmitter stream. Once the packet reaches the destination, the packet is now processed to identify the time it has taken to reach the destination from the source. Depending upon the time duration taken the SQ message is generated to the source.

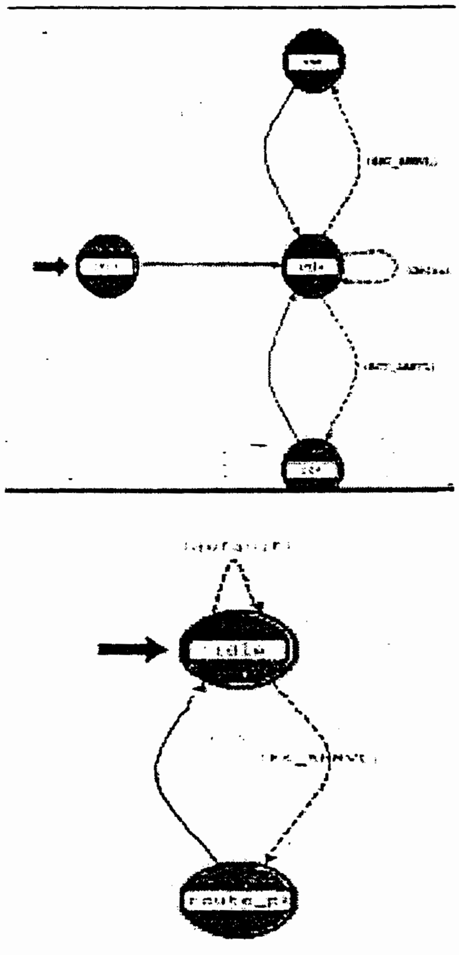


Figure 3. Process module of the host and the hub

To identify the state of congestion, we calculate the time taken for the packet to move from the source to the destination. If it takes long time, that means that the network is congested and there is more likelihood of collision and loss of packets. To handle this situation, we need to check for the time that the packet takes to transfer from source to the destination. Based on the identification of the timing, the source needs to be informed if the destination identifies that the network is congested. After receiving the packet at the destination, the first task done is the calculation of the time taken for the packet to reach the destination from the source. The source time is present in the packet header. This time is subtracted from the current time at arrival to obtain the time taken. The source quench message is generated if the packets are generated at a faster rate by the source. In this case, if the packets are received faster than the receiver can cope with, the receiver will send a SQ message to the source and so the source has to reduce the speed of transmitting packets.

5. Experimental Analysis and Results

Initial analysis was done by changing the time at which SQ message has to be generated along with varying the bits per second. Later, the processing was done at two levels and compared in terms of the recorded performances. In order to see that the situation of applying the active networks concepts of processing the data on the fly helps the IP/SQ algorithm, we have done the processing at the host level and also at the hub level. The processing time was analyzed.

Therefore, to produce a source quench message, the time difference between the source and destination of packets was setup at different numbers of seconds for different data rate systems. The implication of the data from these tables is that when we issue the Source Quench when the time taken for the data to reach the destination is low, then there is less likelihood of collision. It's a tradeoff between placing the IP/SQ time and the total time. Minimizing the IP/SQ time will safeguard the system from congestion but at the same time affects the efficient usage of the network. On the other hand, fixing the IP/SQ timer at a high value makes the network prone to congestion while resulting in high network utilization. The use of source quench in an appropriate way [in accordance with the network capability] reduces the probability of collisions and congestion.

The system was tested with varying interarrival time [the time gap between the transmission of two consecutive packets] between packets. There was an increase in the End-to-End (ETE) delay when packets where transmitted with short interarrival time. This caused an increase in the network utilization when we transmit packets faster, as shown in Figure 4.

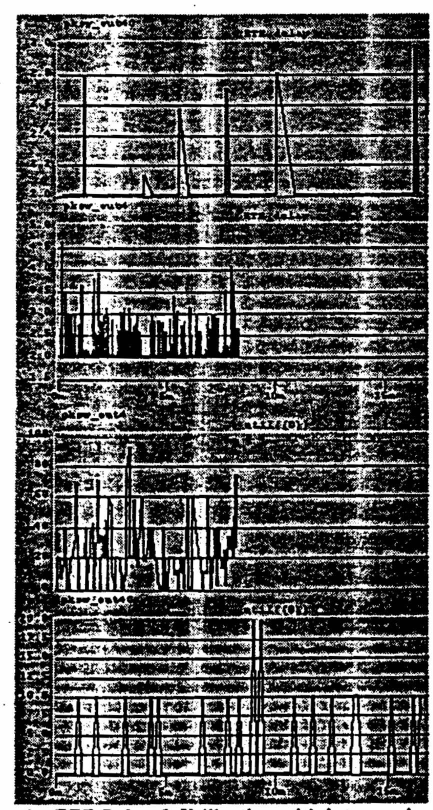


Figure 4. ETE Delay & Utilization with inter-arrival time as 4 and 40 sec.

The ETE delay graph is drawn with simulation time as X-axis and End-to-End time as Y-axis. In the case of utilization, the Y-axis indicates the percentage usage of the link is mentioned and the X-axis indicates the simulation time. The above graph helps us to infer that if the packets are sent with high inter-arrival, there is no more congestion and the network utilization is high. The process of identification of the situation of congestion in the network was done at the destination end as well as at the hub and compared. The impact of the network congestion found at the end hosts has an equivalent impact on the hub too as shown in Figure 5. This makes it clear that we can make an action at the hub on the congestion control that will reduce the impact of congestion throughout the network. The analysis was done when packets where generated at various inter-packet arrival time differences. As the packets are being forwarded faster when the inter-arrival rate is less, the whole processing completes earlier with more density within a short time period.

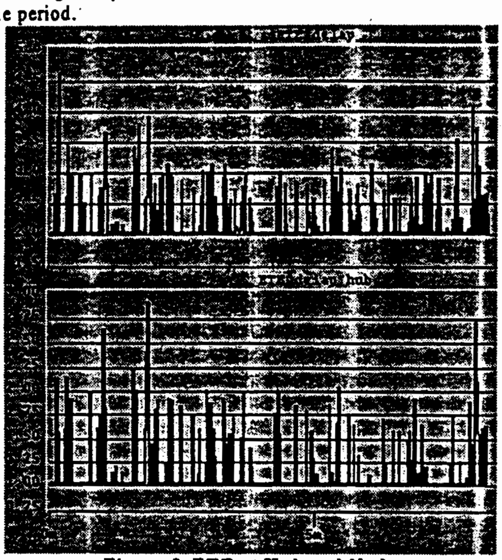
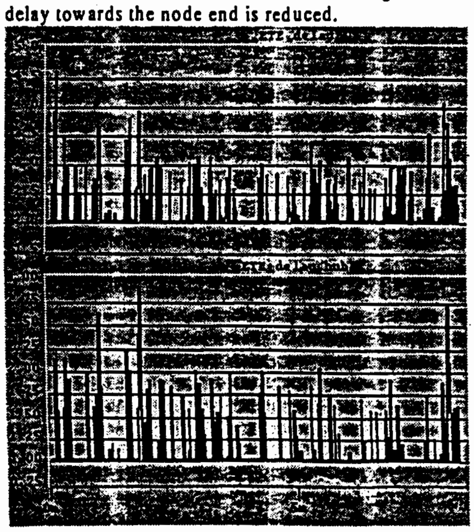


Figure 5. ETE at Hub and Node

After identifying that the presence of some actions at the hub makes an impact on the network, we added the code to send the SQ message from the hub to source in case there occurs a very low ETE delay. Because of the presence of SQ action at the hub itself, we can see from figure 6 that the ETE delay towards the node end is reduced.



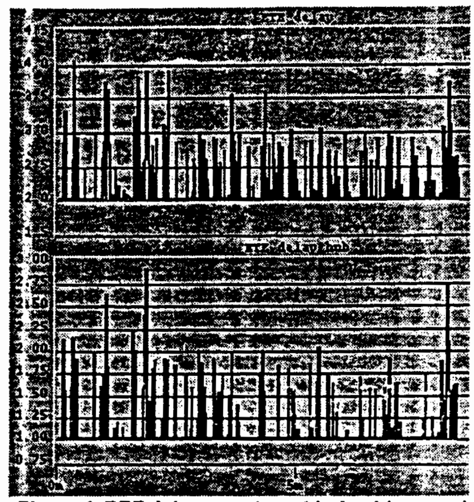


Figure 6. ETE delay at node and hub without and with SQ at hub

A highly congested network may end up in collision and loss of packets. To avoid this situation, we handled the congestion scenario at the router level itself. Therefore, it is clear that the performance of the system improves with the inclusion of Source Quench at the router level.

6. Conclusion

The more a network provides better Quality of Service, the greater its efficiency would be. Quality of Service enhancement of Transmission Control Protocol (TCP) is used to analyze and improve network performance. The usage of Source Quench to control the speed at which the source sends data to the receiver was implemented. The retransmission time for the network was set by first analyzing the time taken by a packet to reach the destination from the source, along with the processing time. The total network is now mostly prevented from congestion by using a Quench message. This is because the possibility of congestions is identified earlier at the routers, and the necessary action is taken.

7. References

[1] F. George and G. Young, "SNA Flow Control: Architecture and Implementation," IBM System Journal, vol. 21, no. 2, 1982, pp. 179-210.

[2] Digital Equipment Corporation, DECnet Digital Network Architecture (phase IV) General Description, Order AA-N149A-TC, Digital Equipment Corporation, 1982.

[3] V. Jacobson, "Congestion Avoidance and Control," Proc. of SIGCOMM '88, 1988.

[4] J. B. Postel, C. A. Sunshine, and D. Cohen, "The ARPA Internet Protocol," Computer Networks, vol. 5, no. 4, 1981, pp. 171-261.

[5] S. Lam and M. Reiser, "Congestion Control of Store-and-Forward Network by Input Buffer Limits Analysis," IEEE Trans. on Commun., COM-27, no. 1, Jan. 1979.

[6] W. Prue. and J. Postel, "Something a Host Could Do with Source Quench," RFC-1016, Network Working Group, July 1987.

[7] U. Mukherji, "A Schedule-Based Approach for Flow-Control in Data Communication Networks," Ph.d. Thesis, Massachusetts Institute of Technology, Feb. 1986.

[8] K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," ACM Trans. on Computer Systems, vol. 8, no. 2, May 1990, pp. 158-181.

[9] D. Comer and R. Yavatkar, "A Rate-Based Congestion Avoidance and Control Scheme for Packet Switched Networks," Proc. of 10th ICDCS, IEEE, 1990. pp. 390-397.

[10] L. Zhang, "Congestion Control in Packet-Switched Computer Networks," Proc. of 2nd Int'l Conf. on Computers and Applications, 1987, pp. 273-280.