Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1995

Effective Strategies for Multi-Media Message Transmission: UDP Cooling and TCP

Xiangning Sean Liu

Lebin Cheng

Bharat Bhargava
Purdue University, bb@cs.purdue.edu

Report Number:
95-067

Liu, Xiangning Sean; Cheng, Lebin; and Bhargava, Bharat, "Effective Strategies for Multi-Media Message Transmission: UDP Cooling and TCP" (1995). *Department of Computer Science Technical Reports.* Paper 1240.
https://docs.lib.purdue.edu/cstech/1240

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

EFFECTIVE STRATEGIES FOR MULTIMEDIA MESSAGE TRANSMISSION: UDP COOLING AND TCP

Xiangning Sean Liu
Lebin Cheng
Bharat Bhargava

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR-95-067
October 1995

Effective Strategies for Multi-Media Message Transmission: UDP Cooling and TCP*

Xiangning Sean Liu, *Lebin Cheng*, and *Bharat Bhargava*

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907
E-mail: `{xl,lcheng,bb}@cs.purdue.edu`

Abstract

Multi-media messages are typically very large and vary from tens to thousands of kilobytes in size. This paper investigates the transmission of large multi-media messages on an Ethernet LAN, Internet WAN, and an ATM LAN. We proposed a UDP cooling method by introducing a short delay between each datagram for large message transmissions. Experiments show that the round-trip UDP datagram loss rate can be reduced from 85% to 8% with a slight performance downgrade for messages larger than 200 Kbytes on the Ethernet LAN. We also conducted experiments by applying UDP cooling method to WAN and an ATM LAN. Experiments with regular UDP and TCP transmission were conducted for comparison of performance and reliability. On the basis of these experiment results, we provided guidelines about using TCP, UDP, and UDP cooling for multimedia data transmission.

1 Introduction

Current applications, such as digital libraries, electronic market, digital world, and video conference, involve large volume of multi-media data transmitted in large size messages to multiple remote sites [Ste95]. For example, NASA images are accessed by many scientists remotely through Internet. Image files of 1 Mbytes are very common. Video conferencing requires a large amount of data being transmitted in real time. One frame typically may involve 250 Kbytes of data for displaying. There are two major protocols that are used at the transport layer in the current Internet, TCP and UDP [Com95]. TCP (Transmission Control Protocol) is a connection-oriented, flow-controlled, end-to-end

---

*This research is partly supported by a grant from NSF under NCR-9405931

transport protocol that provides reliable and ordered stream delivery of data [Pos81]. UDP (User Datagram Protocol) is a simple connectionless datagram transport protocol that provides peer-to-peer addressing and fast but unreliable, unordered delivery of data [Pos80]. The two protocols are suited to different purposes and applications.

The development of new applications has brought with it increasing demands for shorter delays, lower datagram loss rates, and higher throughput. Recent technologies such as Asynchronous Transfer Mode(ATM) [Vet95], have made high performance hardware available. Experiments show that the capacities of the physical network may not be completely exploited by high-level software if the applications do not choose appropriate and properly-configured protocols [CL94, LCBZ95].

We have conducted experiments that investigate the transmission of small and large message transmission on both an Ethernet LAN (Local Area Network) and a WAN (Wide Area Network) with TCP and UDP. The relative merits of each protocol are assessed. We conducted similar experiments on a high speed ATM LAN, which gave us a rather different assessment.

For large message transmission on LANs with relatively low bandwidth, such as a traditional Ethernet LAN, dividing the messages to 8 Kbyte UDP datagrams and transmitting one datagram immediately followed by another will result in high datagram loss rate. Introducing short delay between transmissions of the UDP datagrams can reduce the datagram loss rate significantly. For messages over 200 Kbytes, experiments show that the round-trip UDP datagram loss rate can be reduced from 85% to 8% by this UDP cooling method. Experiments are repeated with the same set of test data using TCP for comparison. The results show that the cooled UDP transmission still has overwhelming efficiency over TCP in this situation.

On WAN, the advantage of the cooled UDP is not so obvious. In the WAN environment, introducing small cooling interval is not enough to reduce datagram loss rate significantly. Introducing large cooling interval, however, may downgrade the speed of transmission to the same speed as or sometimes even slower than that of TCP. Therefore, TCP is recommended for large message transmission on WAN in most cases.

Research on an ATM LAN shows that UDP datagram loss rate is high because the machine on the receiver side can not keep up with the high speed of incoming messages. In this situation, synchronization of the sender and receiver becomes crucial for the performance of the system. Experiments show that neither UDP nor cooled UDP is viable in this case. Instead, TCP proves to be the protocol of choice in this high speed, high throughput LAN environment for large messages.

The experiments of this paper are organized as follows. In Section 2, UDP cooling and TCP on an Ethernet LAN are presented. Section 3 discusses UDP cooling and TCP on WAN. In Section 4, UDP cooling is again compared with TCP in a high speed ATM LAN environment. Section 5 surveys some related research. Finally, in Section 6 we draw conclusions and provide directions for future research.

2 UDP Cooling vs. TCP on an Ethernet LAN

The research was pursued by experiments on data communication performance of TCP and UDP first on an Ethernet LAN for 1) short messages, mostly found in traditional applications and 2) large messages in applications such as multimedia systems. Measurements of message round-trip time and message loss rate in the course of these investigations permit an objective assessment of the tradeoff between the performance and reliability between UDP and TCP.

2.1 Experimental Setup and Input Data

All experiments in the local area network were conducted between two Sun Sparc workstations `raid9.cs.purdue.edu` and `raid11.cs.purdue.edu` connected by 10 Mbps Ethernet. TCP and UDP echo utilities of extended `ping` programs originally authored by Muuss [Ste90] were employed for the experiments. In this set-up, a sender process running on one machine sends messages to the echo receiver located on another machine. The receiver does not send the message back to the sender until it receives the entire message. Round-trip time is then calculated by the sender using the time interval between the delivery of each message and the arrival of the corresponding echo message. Each experimental trial consisted of 50 message echoes, and the same experiment was repeated 20 times. The experiments were conducted using two TCP modes and two UDP modes.

- **Individual-connection TCP**: A connection is established and closed for each message transmission. Some applications can send several messages through one TCP connection, while in other instances the number of messages that pass through a given connection is unpredictable. A system can maintain only a limited number of simultaneous connections. Closing a connection may increase the system overhead of each message sent but allows other applications to use the limited connection resources.

- **TCP** or single-connection TCP: Regular TCP messages, by default, are all sent within a single connection.
- **Cooled UDP**: Large messages are divided into 8 Kbyte datagrams which are sent at short intervals using UDP. A busy wait was used to ratchet up an integer for certain times, generating the interval between datagrams.
- **UDP** or uncooled UDP: For large messages, each 8 Kbyte datagram is sent immediately following the previous datagram.

The following extensions were made to the **ping** programs for the TCP and UDP echo utilities:

- Provision of options to support separate connection experiments (for individual-connection TCP) and no-delay transmission of small messages.
- Support of large messages of up to 4 megabytes. This is an extension from the 8 Kbyte maximum of the original programs.
- Measurement of the precise round-trip time. The echo receiver awaits arrival of the entire message before returning it to the sender. The original **ping** program starts to echo back when a part of the message has arrived.
- Automatic adjustment of the time interval between the individual messages sent out by the sender. If the interval (1 second by default) is too short and the messages are large, the system may break down if a new message is sent before the previous messages return.
- Avoidance of intermediate output. Under UDP, large messages are divided into small datagrams. If the sender must output the round-trip time after the arrival of each datagram, the accumulated round-trip time measured for a large message will be artificially increased by the lag of the print statements which involves slow device I/O.
- Addition of an option to adjust the inter-datagram interval for cooled UDP data transmission.

In addition to the messages of various size generated by the TCP and UDP echo utilities, a set of NASA image files are selected as benchmarks. These files has various sizes and multimedia contents which occur typically in the retrieval of NASA information. The size and descriptions of the NASA image files are listed in Table 1.

Table 1: The image-message benchmark (large NASA image files)

Size (bytes)	File Content
6988	**earth-round.gif:** Sharp contours, green on blue globe. Res 187x158
7708	**earth1.gif:** Very sharp contours, green on blue globe. Res: 160x160
17027	**gal_line.gif:** Red on black, a whole line of only dots. Res: 450x450
29668	**gal_green.gif:** Green on green, lots of dots, striation of colors. Res: 384x330
35543	**comet.gif:** White eye, blue tail, tail fades into background. Res: 512x480
60379	**mars.gif:** Huge circle of light brown shades. Res: 340x340
74058	**surface.gif:** Sloping surface of white and blue, sharp contours, shades. Res: 550x450
80385	**jupiter.gif:** Huge circle of red and yellow shades, yellow text on black. Res: 710x765
97835	**gal_blue.gif:** Blue on blue, some dots. Res: 607x373
104365	**hubble.costar.gif:** Shades of concentric red, orange, yellow colors; shading, text. Res: 566x384
114323	**earth_detail.gif:** Blurred contours; text; pink color; black bg.. Res: 1152x864
135701	**eclipse2.gif:** A huge number of red shades. Res: 784x630
153634	**4gal_red.gif:** Bright red, orange; black and white dots; shading. Res: 441x400
175405	**sf.gif:** Sharp boundary contours, blue, white and red colors. Res: 500x500
205747	**ast_spray.gif:** Black background, lots of small particles. Res: 701x659
236199	**mitwave1.gif:** Orange and white shades, delicate, multicolored ridges. Res: 1024x1024
279786	**earth_highres.gif:** Blurred contours; text; blue color; black bg.; Res: 1152x864
406851	**text+image.gif:** Text, many dots, subtle shading. Res: 936x867
486430	**eclipse1.gif:** A huge number of orange and yellow shades. Res: 1280x1024

2.2 Measured Data and Results

Experiment results are presented as follows:

1. Figures 1.a and 1.b show the round-trip times of messages with size under 512 bytes and 2 Kbytes, respectively. No UDP datagram loss was observed in this case.

2. Figure 2 provides results for larger messages of between 4 and 80 Kbytes in size. Performance of the individual-connection TCP is almost identical to that of the regular TCP, which implies that the overhead of connection re-establishment becomes negligible in this case. UDP outperforms TCP and the gap increases with the message size. UDP datagram loss rate remains low (less than 4.5%).

3. Figure 3 shows the results obtained by sending very large messages of size between 50 and 1000 Kbytes. Datagram loss rate of regular UDP becomes unacceptably high (increases to 90% rapidly). Cooled UDP is shown effectively reducing the datagram loss rate to less than 8% while keeping the round-trip time almost as short as that of regular UDP.

4. Figure 4 shows the enlarged UDP portion of Figure 3.a. Unlike the regular UDP, round-trip times of cooled UDP grow more steadly with the message size. Though slightly slower than the regular UDP, cooled UDP is more predictable than regular UDP.

5. Figure 5 provides the results of sending large multimedia NASA files. The results are very similar to those obtained for very large messages in Figure 3.

2.3 Discussion

For small messages, the experiments show that UDP is the most efficient mechanism, with almost no observed datagram loss. Due to the low datagram loss rate, TCP needs to undertake fewer retransmissions and its performance is comparable to that of UDP. Connection establishment has been noted to add significant overhead when sending small messages. Therefore, re-establishment of connections should be avoided when sending small messages.

While UDP is much faster than other mechanisms for messages of larger size, its reliability, however, gradually declines when message size increases. Ethernet employs the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) mechanism to avoid collisions of transmitted signals. When messages become larger and more IP datagrams must then be generated for each message,

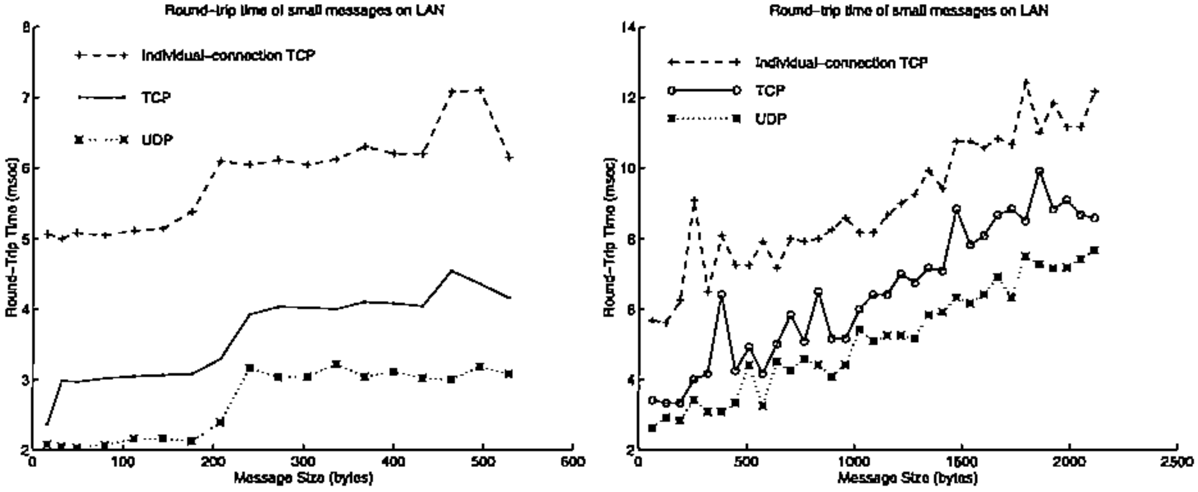

Figure 1: Round-trip time of small messages of size: (a) smaller than 512 bytes; (b) smaller than 2 Kbytes on a LAN

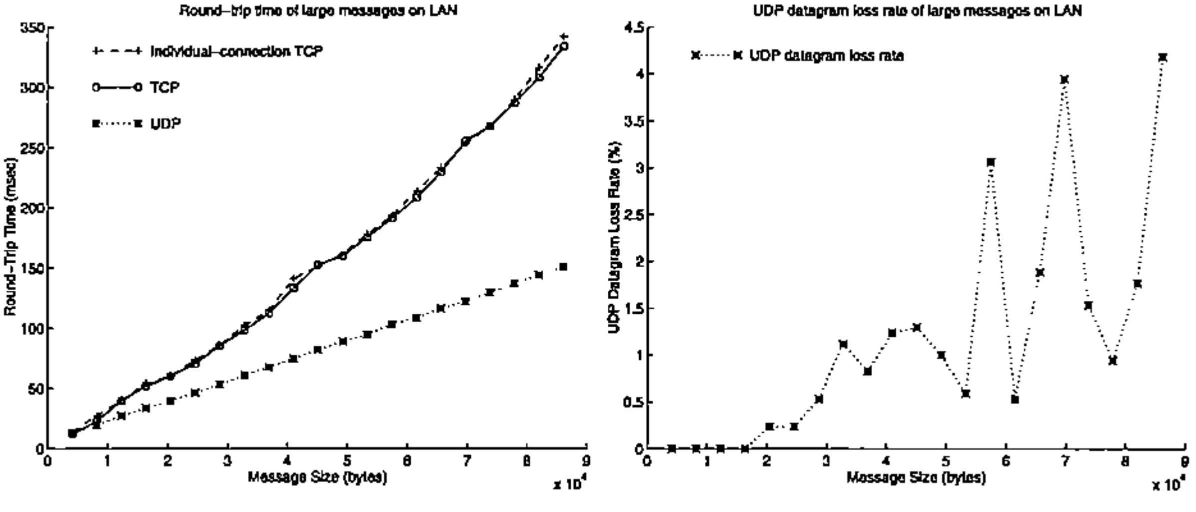

Figure 2: (a) Round-trip time; (b) UDP datagram loss rate of large messages on LAN

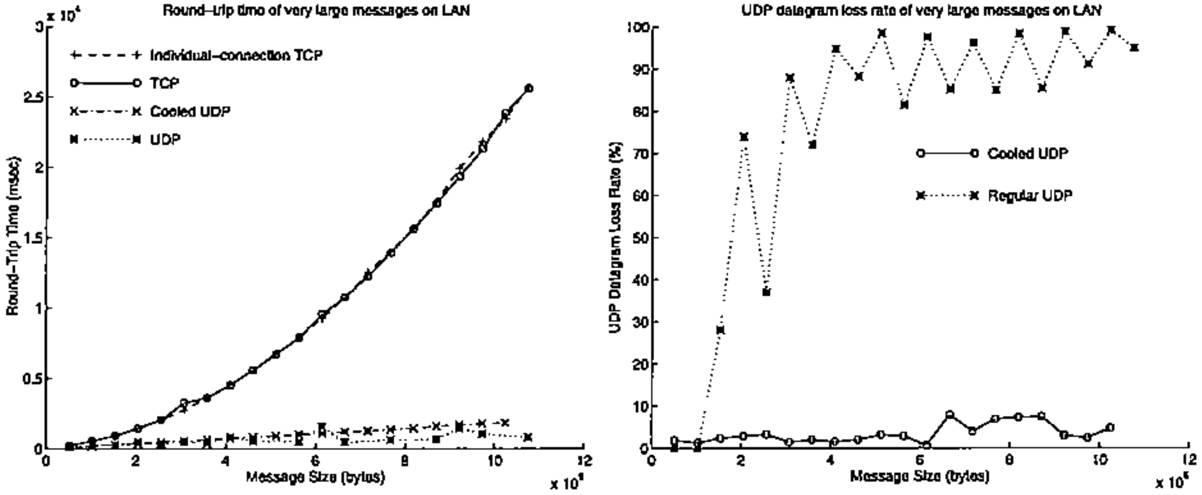

Figure 3: (a) Round-trip time; (b) UDP datagram loss rate of very large messages on LAN

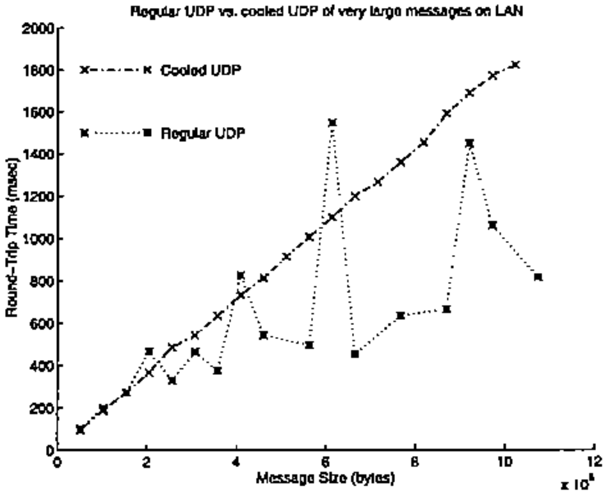

Figure 4: UDP round-trip time of very large messages on LAN

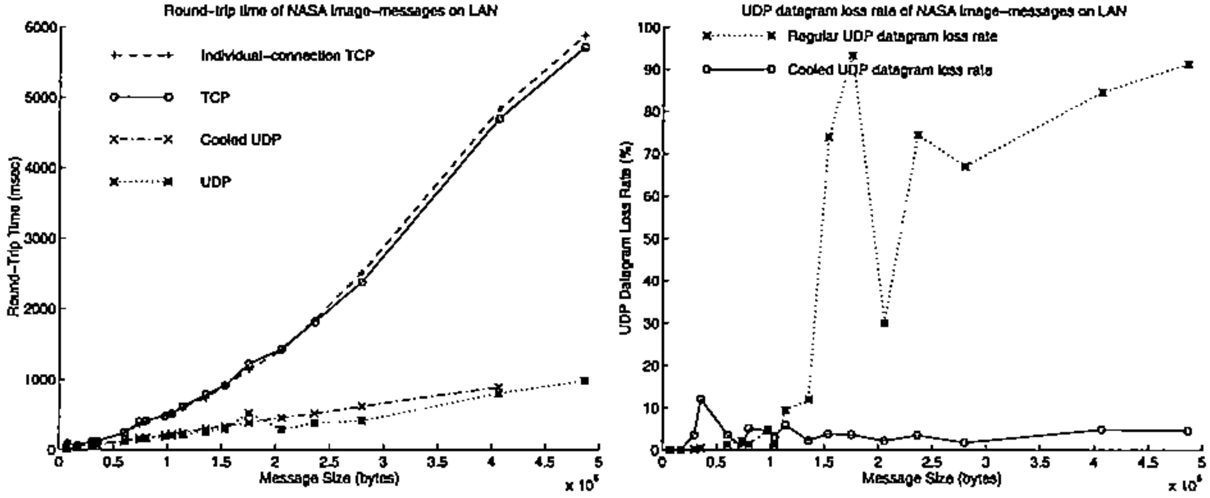

Figure 5: (a) Round-trip time; (b) UDP datagram loss rate of NASA image files on LAN

the probabilities of collisions and out-of-order deliveries become significantly higher. Under these circumstances, TCP exhibits significant performance downgrade because it involves many slow retransmissions. The extent of the out-of-order data and data loss increases even more significantly when message size exceeds 200 Kbytes. Unlike the small message case, overhead caused by connection re-establishment becomes negligible in this situation.

For very large messages over 200 Kbytes, cooled UDP transmission can reduce the round-trip datagram loss from 85% to less than 8% while causing little increase in speed-related overhead. In our experiments, busy waiting was employed to generate the fixed interval between each UDP datagram. Experiments show that long intervals may increase transmission delays with little further improvement in reliability. Intervals that are too short, on the other hand, may not be good enough to bring significant improvement. It is worth mentioning that the optimal length of intervals is system-dependent.

When sending time-critical messages of large size in an Ethernet LAN environment such as for multimedia applications in video conferencing, large number of UDP datagrams must be transmitted across the underlying network in a very short period of time. Given the computing power of today's Sun Sparc workstation, a 10Mbps Ethernet LAN can easily be saturated if the sender continuously transmits packets into the network at full speed. As a result, datagram loss rate becomes unacceptably high. Even though replacing UDP with TCP can solve the reliability problem, it is not desirable because TCP, in this situation, becomes very inefficient in speed. TCP relies on the acknowledgment(ACK) message to control data flow. The congestion avoidance algorithm of TCP employs the exponential back-off scheme to slow down packet transmission rapidly whenever a message is not acknowledged. Consequently, the TCP protocol experiences significant performance downgrade when very large messages are transmitted. Cooled UDP proves to be a viable solution in this environment because the sender is tuned by the user to send packets at optimal pace. Essentially, data flow is preset "manually" to avoid congestion by cooled UDP.

3 UDP Cooling vs. TCP on a WAN

The TCP and UDP echo utility programs described in Section 2 were again employed for experiments in this section. An echo server was installed on a Sun Sparc workstation at Stanford University. Unlike the common ICMP echo server on port 7, this server has the capacity to handle large messages involved in our experiments. The same input data for the LAN experiments are used here.

The experiment results for messages of different sizes are as follows:

1. Figure 6 shows round-trip time for messages under 1 Kbytes. The UDP datagram loss rate is low (less than 7%).
2. Figure 7 provides results obtained by sending large messages of size between 4 and 80 Kbytes. Even though cooled UDP can reduce datagram loss rate, its performance is downgraded to the same level as TCP.
3. Figure 8 shows the results of sending very large messages between 50 and 1000 Kbytes. Due to the unacceptably high UDP datagram loss rate, we did not gather sufficient samples to measure UDP round-trip times in this case.
4. Figure 9 provides results obtained by sending the same large NASA multimedia data files discussed in Section 2.

In a WAN environment, unlike the Ethernet LAN, datagram loss is caused by the intermediate networks instead of just one saturated local network. Datagram loss rate can be affected by various factors such as buffer overflow, malfunctioning routers, outrage of intermediate network, etc. Tuning the pace of the sender will not significantly affect the overall quality of the transmission. Therefore,

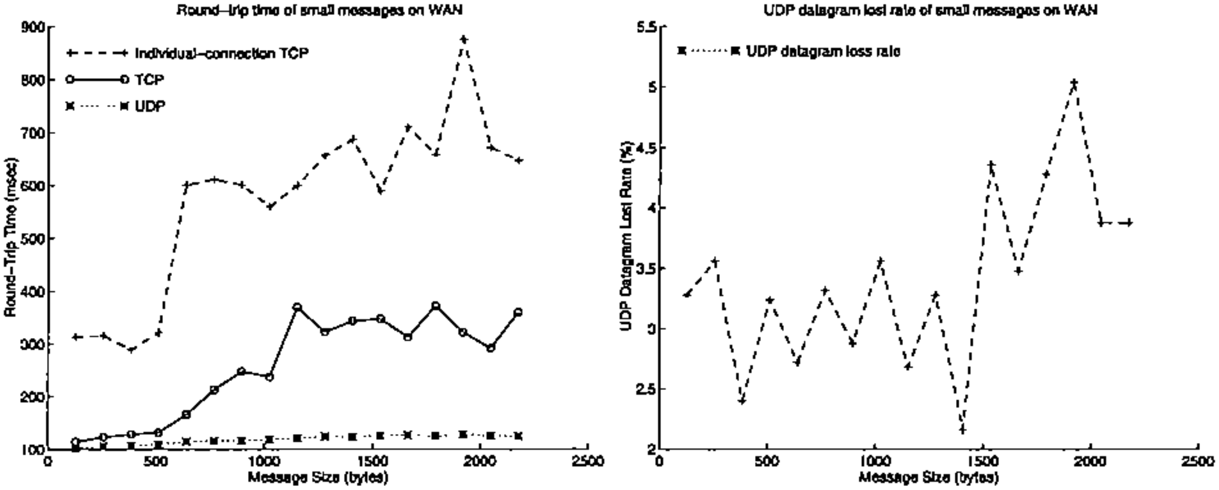

Figure 6: (a) Round-trip time; (b) UDP datagram loss rate of small messages on WAN

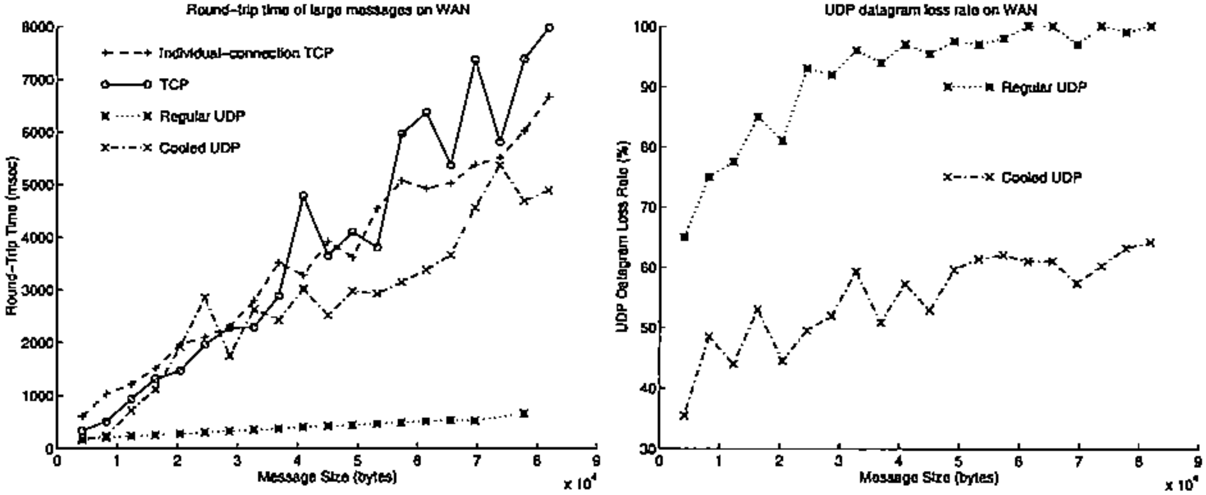

Figure 7: (a) Round-trip time; (b) UDP datagram loss rate of large messages on WAN

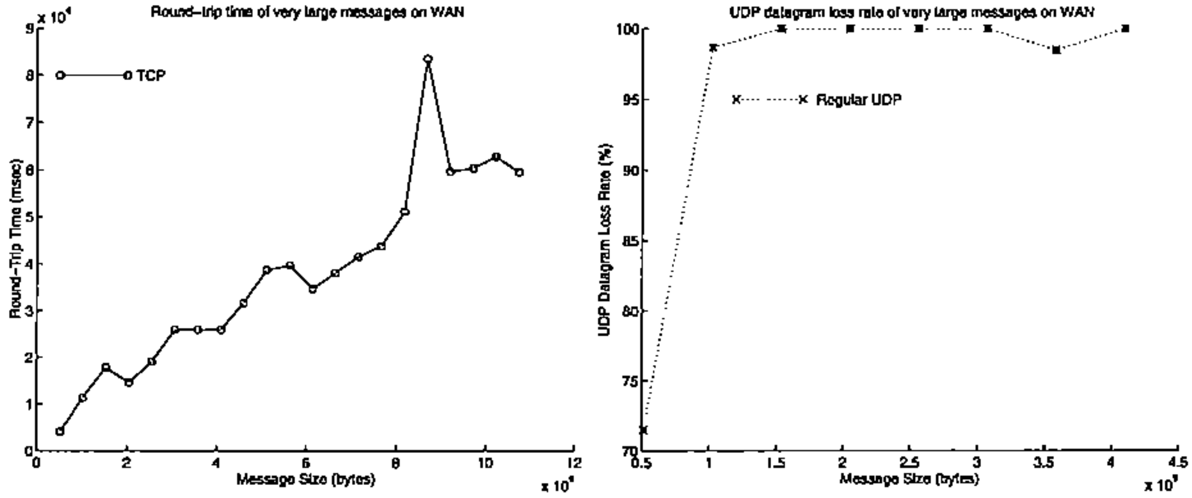

Figure 8: (a) Round-trip time; (b) UDP datagram lost rate of very large messages on WAN

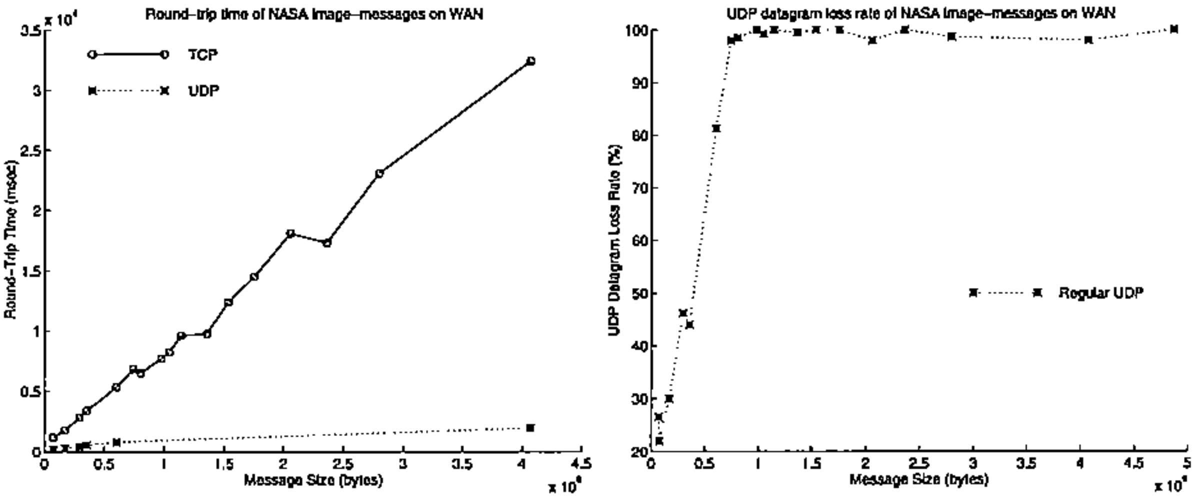

Figure 9: (a) Round-trip time; (b) UDP datagram loss rate of NASA image files on WAN

it is recommended to use TCP in general for its guaranteed reliability. However, for applications that can tolerate high data loss, UDP can be used to improve the speed of transmission, especially when the message size is relatively small (less than 8 Kbytes).

4 UDP Cooling vs. TCP on an ATM LAN

ATM (Asynchronous Transfer Mode) is a family of protocols supporting both the circuit and packet-switching services [Vet95]. ATM cells, or fixed-length packets, form the basic data units. An ATM cell, as defined by ITU (formerly CCITT) recommendation I.361, contains 48 octet of data and 5 octet of control information. This fixed-length cell and the early binding of routing information during the connection setup make the ATM suitable for high-speed data communication. Its bandwidth reservation and graceful multiplexing also render it suitable for multi-media traffic. Four major benefits of the ATM discussed in [KW95] are scalability, statistical multiplexing, traffic integration, and network simplicity. Because of its high bandwidth, low delay and high flexibility, ATM network has been described as the network of choice for data communication in the future.

We conducted experiments similar to those described in the previous sections to investigate the applicability of the UDP cooling method to an ATM LAN.

4.1 Experimental Setup and Input Data

Our experiments involved two Sun Microsystems IPX workstation, called `isabella` and `michelangelo`, running Solaris 2.3. As shown in Figure 10, each host connects to a SynOptics LattisCell 10114 ATM switch via OC3 155.52 Mb/s multi-mode fiber cables, which supplement the conventional 10 Mb/s Ethernet. Both hosts use S/ATM 4615 ATM adapter cards of InterPhase Corp., the driver of which supports ATM Adaptation Layer 5 (AAL5) [GL92]. The SynOptics ATM switch comes with a dedicated processor and special-purpose software for user configurations. In one experiment, we used the same TCP and UDP echo utilities as that of the previous experiments to measure the performance of both cooled UDP and TCP. Note that the cooled UDP is tuned by the interval with 100 iterations of integer increases in these experiments. In another experiment, we used a program called `ttcp` originally written by Slattery [CL94], which simply transmits large number of packets from one host to another through the ATM switch, to measure throughput and UDP datagram loss rate of the network. In an attempt to find the optimal busy waiting interval, we recorded the UDP throughput and datagram loss rate tuned with various numbers of iterations.

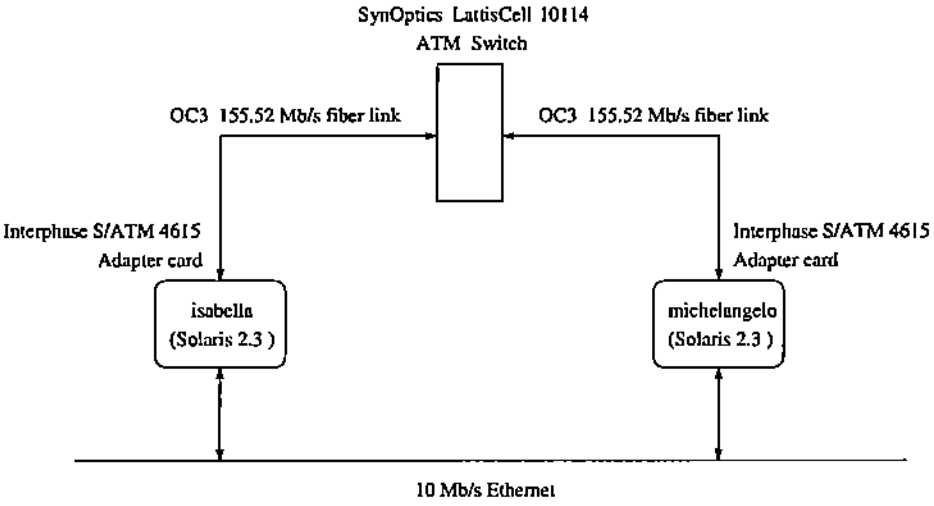

Figure 10: Experimental environment setup of an ATM LAN

4.2 Measured Data and Results

Results of experiments using the TCP and UDP echo utilities are presented as follows:

1. Figure 11 shows the round-trip time measured by sending small messages. Connection time is a significant overhead.
2. Figure 12 presents the results of sending large messages. Cooled UDP, regular UDP, and TCP without connection overhead show almost identical performance. Datagram loss rates of both cooled UDP and regular UDP increase rapidly with the size of the message.
3. Figure 13 shows the results of sending very large messages. Even though both cooled UDP and UDP result in short round trip times, the datagram loss rates of both UDP mechanisms are higher than 50%.
4. Figure 14 shows results of sending NASA image files. Datagram loss rates of both UDP mechanisms are approaching 90% for messages of size larger than 300 Kbytes.
5. Results of experiments using the `ttcp` program are presented in Figure 15. The throughput of data transmission using TCP is also present in the figure for comparison. Although longer cooling intervals can reduce the datagram loss rate, the measured throughput of UDP will be downgraded to the same level as TCP when the datagram loss rate is reduced to an acceptable level (less than 10%).

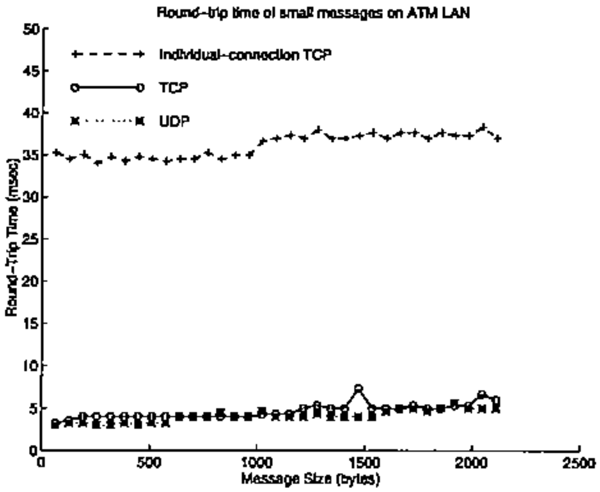

Figure 11: (a) Round-trip time of small messages on ATM LAN

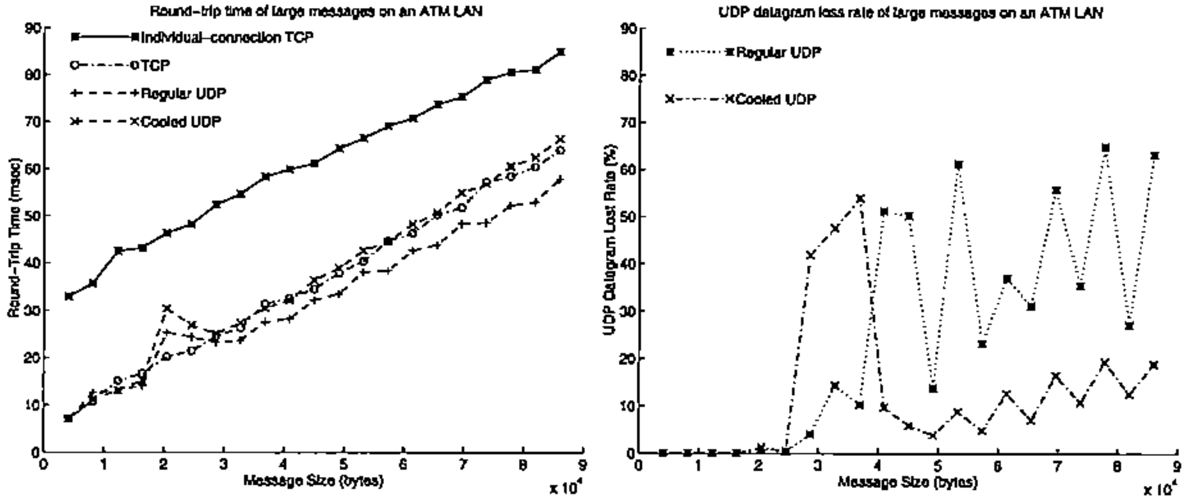

Figure 12: (a) Round-trip time; (b) UDP datagram loss rate of large messages on ATM LAN

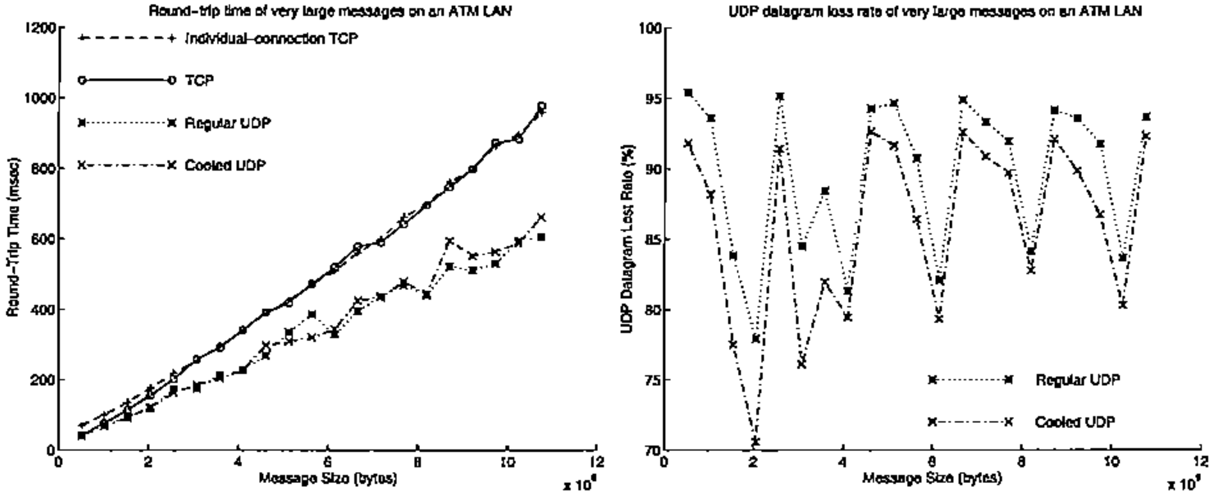

Figure 13: (a) Round-trip time: (b) UDP datagram lost rate of very large messages on ATM LAN

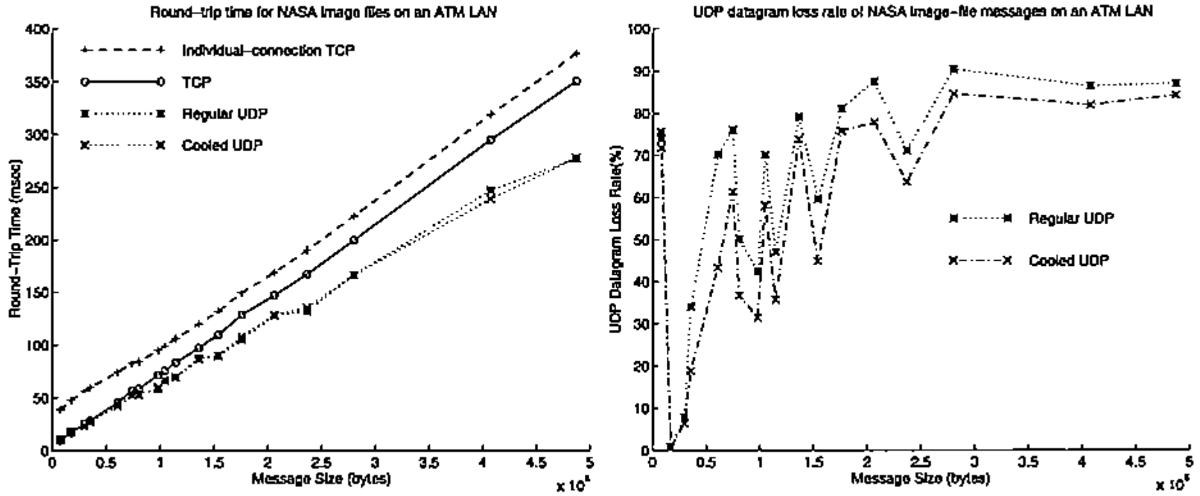

Figure 14: (a) Round-trip time; (b) UDP datagram loss rate of NASA image files on ATM LAN

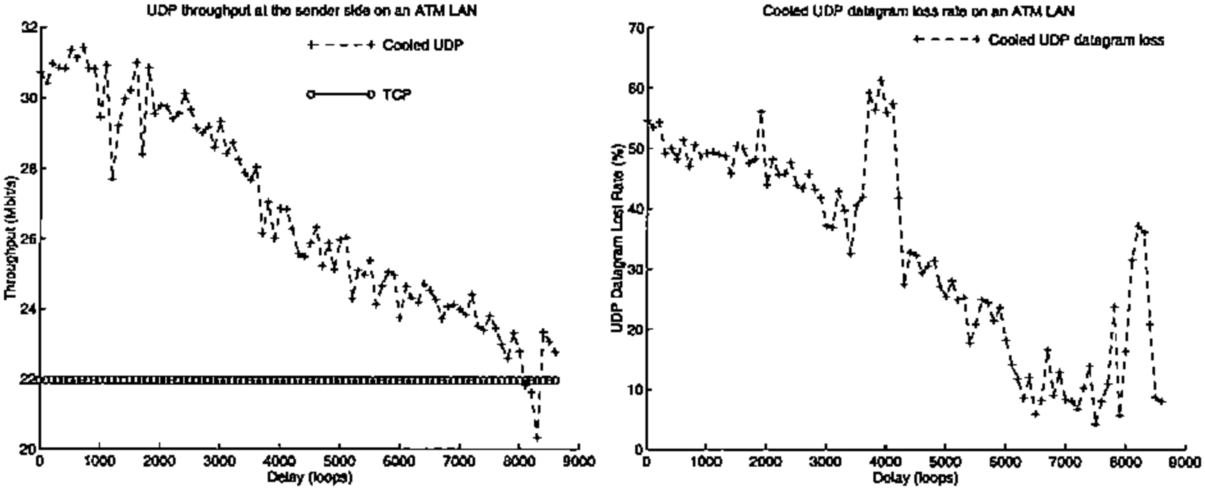

Figure 15: (a) UDP and TCP throughput; (b) UDP datagram loss rate on ATM LAN

4.3 Discussion

In all experiments, a special purpose program was employed to monitor the ATM switch and to report ATM cell loss. No cell loss is detected in any of the experiments. It is concluded that datagram loss is due to the end hosts, not the ATM network. Experiments show that, unlike the low speed LAN such as the 10 Mbps Ethernet LAN, an ATM LAN operates at a much higher speed and the host machine, instead of the network, becomes the bottle-neck. The dedicated link on an 155Mb/s ATM network provides enormous data transmission capacity, capable of sending all packets generated from the sender machine to the receiver instantly. In this situation, if the sender machine generates packets faster than the receiver machine can handle, the receiver will eventually drop packets due to the buffer overflow. Therefore, slowing down on the sender side can reduce packet loss.

Although the experiment results show that UDP cooling scheme can reduce datagram loss, the throughput of the data transmission, however, is downgraded to overlap the throughput of data transmission using TCP. Consequently, the cooled UDP ceases to be a viable solution to the datagram loss problem because it significantly downgrades the performance of UDP. TCP proves to be the protocol of choice in this high bandwidth LAN environment.

In the ATM LAN environment, both acknowledgment message of TCP and data packets are sent with little delay. Transmission of packet is very reliable on the otherwise idle ATM network. TCP, therefore, can rely on the ACK message to effectively control data flow to achieve optimal performance. In essence, the effective flow control of TCP eliminates the need for "manual" adjustment by programmers.

5 Related Research

Partridge and Pink discussed optimizations at system level to provide fast UDP with 30% cost reduction [Pat94]. Papadopoulos and Parulkar presented performance study on SUNOS IPC and TCP/IP implementation in [PP93]. These research activities focus on lower system level improvement. Improvement of UDP at both lower system level and higher application level for transaction processing, which incurs small messages, has been studied by Bhargava, Mafla, and Zhang [BZM91, MB91] and 70% performance increase was obtained by the combined optimizations. ATM for multimedia data communication is studied by Iwata et al. in [IMI+95]. Communication issues for digital library has been discussed in a recent paper [BA95].

6 Conclusions and Guidelines

Our research shows that cooled UDP is suitable for time-critical applications involving large amount of data transmission such as in video conferencing applications, especially in a low speed LAN environment where delay or loss of TCP datagram makes TCP data transmission very inefficient. In general, TCP is suitable for applications that require high reliability such as file transfer and database transaction processing, especially on WAN. In a high speed and high capacity LAN, such as an ATM LAN, TCP has been found to be both very reliable and not so inefficient compared with UDP.

On Ethernet LAN, while UDP exhibits almost no datagram loss, the round-trip time of TCP differs from UDP by less than 30% for small messages (less then 16 Kbytes). For messages between 16 Kbytes and 100 Kbytes, UDP loss rate is still less than 5%, while the performance of TCP is comparable with UDP. In these two cases, applications can select UDP or TCP according to specific performance and reliability requirements. By introducing short interval in cooled UDP data transmission, experiments show that for messages over 200 Kbytes, round-trip UDP datagram loss rate can be reduced from 85% to 8% with small performance cost. Appropriate slow down on the sender side reduces the probability of collisions and effectively prevent the sender from saturating the Ethernet LAN. Thus, when transmitting messages of large size, cooled UDP outperform both the regular UDP by low data loss and TCP by short delay.

In a WAN environment, UDP data loss is not very big when messages are less than 8 Kbytes. When message sizes are larger than 8 Kbytes, UDP loss rate is high and no obvious improvement was observed by using the cooled UDP. When transmitting datagram on WAN, intermediate networks are the determining factors that contribute to the datagram loss. Therefore, tuning merely the sender have little effect on the overall performance.

Though cooled UDP can reduce datagram loss rate in a high capacity ATM LAN, TCP proves to achieve acceptable throughput while at the same time guarantee reliability for large messages. The key factor that affects the performance of TCP in this situation is the instant ACK made possible by the high speed ATM network, with which TCP can effectively adjust the data transmission speed on the sender side to control data flow. UDP has high datagram loss when message size is larger than 16 Kbytes since receiver can not keep up with the fast incoming data.

It is advisable for application developers and system administrators to choose the most suitable protocol for data transmission based on the characteristics of the underlying physical network and user requirement. Usually, one should not expect a system to perform well when the system is operating near its full capacity. Care must be taken by the application developers to avoid saturating the underlying network. UDP Cooling has been shown outperform other protocols in an Ethernet LAN environment because it effectively prevent the applications from saturating the underlying Ethernet. The recommended strategies of data transmission in different environments are summarized in Table 2.

Table 2: Recommended Strategies

Network type	Small message (<8 Kbytes)	Large message (8 Kbytes – 100 Kbytes)	Very large message (>100 Kbytes)
Ethernet LAN	UDP/TCP	UDP/TCP	Cooled UDP
WAN	UDP/TCP	TCP	TCP
ATM LAN	UDP/TCP	TCP	TCP

Based on the previous discussion, we suggest experiments of the following ideas:

- Though we have shown that TCP works well in a high speed ATM LAN environment with no congestion, we can not predict the performance of TCP in the situation when the ATM LAN is saturated. Due to the limitation of equipments, we were not able to assess the relative merit of TCP and UDP in a saturated ATM LAN.
- We obtained the appropriate interval or UDP cooling by manually conducting the experiments and tuning. A self-adaptive UDP, which may choose the optimal parameter dynamically based on the behavior of the underlying physical network, is desirable for a more robust system. Future research should address this issue.

Acknowledgment

The authors would like to thank Professor Douglas E. Comer for providing support for our experiments. Professor Abdelsalam (Sumi) Helal provided substantial encouragement and suggestions for this research. Melliyal Annamalai provided NASA image files for our experiments. Yue Zhuge at Stanford University helped us with the WAN experiments.

References

[BA95] B. Bhargava and M. Annamalai. Communication Costs in Digital Library Databases. In *Lecture Notes in Computer Science Series (LNCS) 978, Database and Expert Systems Applications (DEXA '95)*, pages 1–13. Springer-Verlag, September 1995.

[BZM91] B. Bhargava, Y. Zhang, and E. Mafla. Evolution of Communication System for Distributed Transaction Processing in Raid. *USENIX Journal Computing Systems*, 4(3):277–313, Summer 1991.

[CL94] D. E. Comer and J. C. H. Lin. TCP Buffering and Performance over an ATM Network. *Journal of Internetworking: Research and Experience*, 10(4):70–80, October 1994.

[Com95] D. E. Comer. *Internetworking with TCP/IP Vol I: Principles, Protocols, and Architecture*, volume I. Prentice Hall, Inc, Englewood Cliffs, New Jersey, third edition, 1995.

[GL92] D. Greene and B. Lyles. Reliability of Adaptation Layers. In *Proceedings of IFIP 6.1/6.4 Workshop (Protocols for High-Speed Networks, III)*, 1992.

[IMI+95] A. Iwata, N. Mori, C. Ikeda, H. Suzuki, and M. Ott. ATM Connection and Traffic Management Schemes for Multimedia Internetworking. *Communications of the ACM*, 38(2):72–89, February 1995.

[KW95] B. G. Kim and P. Wang. ATM Network: Goals and Challenges. *Communications of the ACM*, 38(2):39–44, February 1995.

[LCBZ95] X. Liu, L. Cheng, B. Bhargava, and Z. Zhao. Experimental Study of Data Communication for Scalability in Distributed Databases. Technical Report CSD-TR-95-046, Department of Computer Sciences, Purdue University, July 1995.

[MB91] L. E. Mafla and B. Bhargava. Communication Facilities for Distributed Transaction Processing Systems. *IEEE Computer*, pages 61–66, August 1991.

[Pat94] C. Patridge. *Gigabit Networking.* Addison-Wesley, Reading, Massachussetts, 1994.

[Pos80] J. B. Postel. User Datagram Protocol. *Request for Comments*, (RFC-768), August 1980.

[Pos81] J. B. Postel. Transmission Control Protocol. *Request for Comments*, (RFC-793), September 1981.

[PP93] C. Papadopoulos and G. M. Parulkar. Experimental Evaluation of SUNOS IPC and TCP/IP Protocol Implementation. *IEEE/ACM Transactions on Networking*, 1(2):199–216, April 1993.

[Ste90] W. R. Stevens. *Unix Network Programming.* Prentice-Hall, Inc., 1990.

[Ste95] M. J. Stefik. *Internet Dreams – Archetypes, Myths and Metaphors for Inventing Our Information Networks.* 1995. (to appear).

[Vet95] R. J. Vetter. ATM Concepts, Architectures, and Protocols. *Communications of the ACM*, 38(2):30–38, February 1995.