Two source-controlled mechanisms for reducing traffic congestion

Chyan Yang
National Chiao Tung University
Institute of Information Management
MB307, 1001 Ta Hsueh Rd.
Hsin Chu 300 Taiwan R.O.C
cyang@cc.nctu.edu.tw

Chen-Hua Fu
National Chiao Tung University
Institute of Information Management
MB317A, 1001 Ta Hsueh Rd.
Hsin Chu 300 Taiwan R.O.C
u8634804@cc.nctu.edu.tw

Abstract

The characteristics of TCP and UDP lead to different network transmission behaviors. TCP is responsive to network congestion whereas UDP is irresponsible. This paper proposes two mechanisms that operate at the source node to regulate TCP and UDP flows and provide a differential service for them. One is the congestion control mechanism, which uses congestion signal detected by TCP flows to regulate the flows at the source node. Another is the time slot mechanism, which assigns the different number of time slots to flows to control their flow transmission. Based on the priority of each flow, different bandwidth proportions are allocated for each flow and differential services are provided. We use the ns-2 network simulator in which the different scenarios are set up to simulate the transmission operations of these two mechanisms. Simulation results show some insights of these two mechanisms. Moreover, we summarize the factors that may impact the performance of these two mechanisms.

Keyword: *differential service, network congestion, congestion control mechanism, time slot mechanism, transmission performance*

I. Introduction

The TCP and UDP are the two major protocols over the Internet. These two protocols have different traffic transmission operations. TCP is connection orientated whereas UDP is connectionless. TCP uses "slow start" mechanism [1] as the end-to-end congestion control mechanisms to prevent traffic congestion. UDP simply uses "store and forward" mechanism to transfer its data. If there is no proper control mechanism to handle UDP's transmissions, UDP traffic will share most of bandwidth over the Internet. These characteristics of TCP and UDP lead to different network transmission behaviors. The drop-and-run UDP is unfavorable to self-controlled TCP when the proportion of UDP is relatively higher than that of TCP.

Since most of Internet applications are based on TCP. The performance of TCP will impact the Internet efficiency. How to improve the TCP transmission performance and restrict too much bandwidth shared by UDP is the focus of this study. In this study, a differential service mechanism will be proposed to handle the transmission of TCP and UDP traffic. Priority setting for each TCP or UDP traffic ensures each traffic flow gets the different bandwidth share. The higher priority is, the better transmission performance it gets.

Two flow control mechanisms are proposed to control the TCP and UDP traffic transmissions in this study. One is congestion control mechanism. It uses the TCP traffic congestion signal and the priority of each traffic flow to control the transmissions. The other is time slot control mechanism. According to the priority of each traffic flow, the different amount of time slots will be assigned. Network simulator is used to simulate the traffic transmission operations of these two mechanisms and the transmission performance of them is collected. With an analysis of simulation results, we find that both of the two control mechanisms can provide a differential service for the larger transmission data, such as 1M bytes, and the congestion control mechanism has a better transmission performance than the time slot mechanism. Moreover, some parameter settings, may impact the transmission operations and performance of the proposed control mechanisms.

II. Analysis of TCP/UDP traffic transmission

The transmission protocols of TCP/UDP are quite different. The difference between TCP and UDP calls for different transmission behaviors and therefore they have different transmission performance over the Internet.

1. TCP/UDP protocol operation

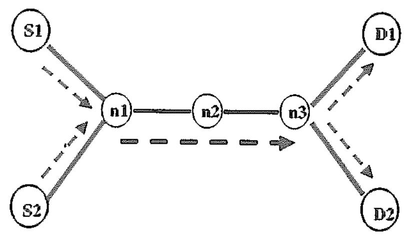
TCP is a connection-orientated transmission protocol. Before traffic can start, TCP needs to set up a connection between the source node and the destination node. The established connection guarantees the network resource is available for TCP to transmit packets. Moreover, TCP keeps a hand shaking mechanism to maintain the correctness and proper sequence of traffic transmission during protocol operation. Having completed the data transmission, TCP disconnects the connection for the session. TCP also uses the "slow start" algorithm to handle the operation of traffic transmission and prevent the traffic congestion. So TCP traffic is a "responsible flow" over the Internet [2].

UDP is a connectionless transmission protocol. It is unnecessary for UPD to establish a connection from source node to destination node before start a data transmission. UDP uses a "store-and-forward" mechanism to transmit data. When each node receives the packets from the previous hop node, it stores the data at local memory or disk first. Then based on the routing path information, the node forwards the stored data to the next hop or node. For UDP, there is no end-to-end control mechanism to guarantee the correctness of traffic transmission. That is, UDP traffic is a kind of best-effort traffic and an "unresponsive flow" over the Internet [2].

2. A TCP/UDP traffic transmission simulation

TCP uses the "slow start" mechanism [1] to control the traffic transmission: it increases the packets' transmission size when there is no congestion in the network, and it reduces the transmission load to response the network congestion. UDP traffic transmission does not response network congestion and have no end-to-end congestion control mechanism. It does not reduce its transmission load during the network congestion occurs. It exists an extreme unfairness situation among TCP and UDP traffic flows when the network is congested: the bandwidth should be shared by the TCP traffic flows are occupied by the UDP traffic flows. This unfairness results from responsive and irresponsible flows completing for bandwidth. The UDP flow effectively "shuts out" the responsive TCP traffic [2].

In this study, two simple simulation scenarios are established with ns-2 network simulator [4]. These two simulations can be used to observe the transmission situation of TCP and UDP traffic. The simulation network refers to the simulation network simulated by Sally Floyd and Kevin Fall in [2], and it can help us examine the simulation results. The network topology is shown as Figure 1. The S1, S2, D1 and D2 are the source and destination nodes. There are four TCP flows and three UDP flows from S1 to D1, and three TCP flows and UDP flows from S2 to D2. Each TCP flow uses FTP type traffic as its traffic source, and the traffic source of each UDP flow is the constant bit rate traffic.



Bandwidth : S1-n1 10MBps S2-n1 10MBps n1-n2 -n3 1MBps
n3-D1 10MBps n3-D2 10MBps

Figure 1. A topology of simulation scenario

The first simulation scenario is that each TCP/UDP flow starts at the same time to transmit the assigned data size. One can then observe the transmission performance of each TCP and UDP flow. The length of assigned data size is from 10KBytes to 100MBytes. The simulation result of the transmission time of each TCP and UDP flow is listed in Table 1. Table 1 shows an important observations: all transmission time of the UDP flows are shorter than the TCP flows and the transmission performance differences between TCP and UDP flows increase as the transmission size increases. This phenomena show that the UDP flows always have larger shares of bandwidth than that of TCP flows and therefore receive a better transmission performance. This is especially true when the transmission size is large because UDP does not response to the network congestion. On the other hand, TCP flows regulate their transmission by the "slow start" mechanism when the network is congested, they do not transmit packets until UDP flows finish their transmission. This is a reason why the TCP flows have the longer transmission time than the UDP flows.

Table 1. A transmission performance of TCP/UDP flows in a simulation scenario

Flows \ Size		10KB	100KB	1MB	10MB	100MB
S1 I D1	TCP1	2.6	13.5	122.9	1080.1	10462.1
	TCP2	2.7	14.7	121.7	1066.7	10475.8
	TCP3	2.7	16.4	207.5	1089.9	7488.9
	TCP4	2.7	14.8	121.8	1073.1	10436.5
	UDP1	0.8	5.0	48.5	474.7	4769.6
	UDP2	1.4	6.0	48.0	480.8	4794.8
	UDP3	0.7	5.3	48.1	481.0	4802.3
S2 I D2	TCP5	2.6	14.5	122.8	1079.1	10482.2
	TCP6	2.7	14.6	124.8	1080.7	9851.1
	TCP7	2.8	15.5	129.9	710.3	7544.9
	UDP4	0.8	4.2	46.5	479.2	4797.0
	UDP5	0.6	3.2	47.8	480.5	4782.4
	UDP6	0.9	5.1	49.2	471.6	4788.6
						Unit : second

The second simulation scenario is that the TCP and UDP flows have a same period to transmit traffic and record the traffic sizes they transmitted. This simulation tries to record the transmission performance of the TCP and UDP flows in a same period. This would be more helpful for us to understand how the bandwidth is shared by TCP and UDP flows during a same period. The range of a simulation transmission period is from 1 second to 10000 seconds and nine check points are selected to collect the transmission size of the TCP flows and UDP flows. Figure 2 shops the results of this simulation, the ratio variability of bandwidth shared by TCP flows and UDP flows.

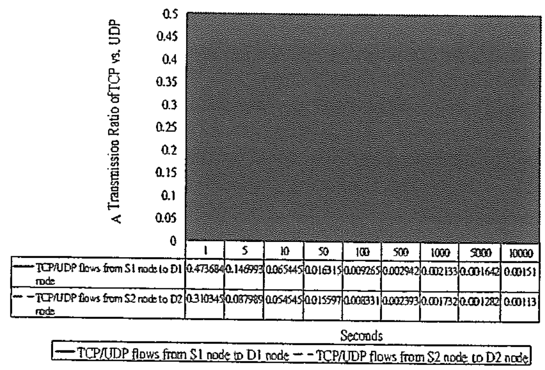


Figure 2. A Variability of TCP / UDP Transmission Ratio

From the curves shown in Figure 2, the ratio of bandwidth shared by TCP flows and UDP flow is inversely with the transmission periods. The curves also show the bandwidth share of TCP flows drop dramatically from the first second to the 50^{th} second, and the bandwidth share of TCP flows is less than 1 % after the 100th second. These results exactly show why the UDP flows have the better performance than the TCP flows in the first simulation scenario, because most of bandwidth is shared by the UDP flows during the transmission period.

The results from the second simulation demonstrate the TCP flows have higher bandwidth share at the beginning of transmission period because the network is not so congested. When the UDP flows continue sending their packets on the network and gradually the network congestion becomes more congested. The TCP flows continue to response the network congestion by reducing their transmission rates, and their bandwidth shares continue to decrease. If the UDP flows keep their traffic transmission, UDP traffic will receive more bandwidth share. This in turn will cause TCP flows to continue slowing down TCP transmissions and the bandwidth share of TCP is going lower than that of UDP. This would be a serious problem for TCP protocol since it supports connection-oriented applications. Almost all real-time applications are connection-oriented and TCP based.

The above two simulation results show that an unfairness situation exists between the transmissions of the TCP flow and the UDP flow: most of the bandwidth is shared by UDP flows, when two protocols are competing for the limited bandwidth. This is mainly due to that the UDP protocol adopts "store and forward" mechanism to transmit packets and has no provision to response the network congestion. UDP sends data packets there is available bandwidth on the network link. TCP protocol, however, uses "slow start" algorithm [1] as its end-to-end congestion control mechanism. TCP regulates its traffic transmission when it detects there is a congestion on the network. By reducing the packet transmission rate, the TCP flows effectively release bandwidth share to UDP traffic. If there are sufficient UDP traffic, bandwidth released by TCP does not reduce network congestion. It only prevents the congestion from worsening. So it is obvious that a proper control mechanism is needed to regulate the transmission of UDP flow and prevent the bandwidth share overused by the UDP flows. This is why the two source-based traffic flow control mechanisms are studied in this study.

III. Source-based traffic flow control mechanisms

Congestion of the Internet becomes more serious as more network applications emerged over the Internet. The Internet bandwidth is limited. If there is no proper mechanism to control the congestion and a congestion collapse may happen over the Internet. To satisfy flows' transmission requirements with limited bandwidth, only higher priority applications get more bandwidth share and have a better transmission performance that can satisfy most users' requirements. It is a reason why several differential service mechanisms are proposed [3, 4].

Differential service mechanisms provide different levels of services to different users over the Internet. Generally speaking, traffic management or bandwidth control mechanism that treats different users or traffic flows differently - ranging from simple Weighted Fair Queueing to RSVP and per-session traffic scheduling [4]. Most of proposed differential service mechanisms involve the capability of the gateways on a routing path, These mechanisms need gateway devices to support its operations.

In this study, two source-based traffic flow control mechanisms are proposed: one is congestion control mechanism and the other is time slot mechanism. These two mechanisms operate at the source node to regulate TCP and UDP traffic flows. They allocate different bandwidth proportions for different traffic flows according to their priorities. Priority overwrites the types of protocols. That is, the priority 1 of UDP traffic takes higher preference to the priority 2 of TCP traffic. For the same type of protocol, priority determines the preference. The higher priority traffic flows could get the more bandwidth shares and better transmission performance. With these two control mechanisms, the transmissions of TCP and UDP flows can be regulated and the differential services can be provided at the source node to enhance the transmission performance of higher priority traffic flows.

1. Congestion control mechanism

The congestion control mechanism is a source-based traffic flow control mechanism. It operates at the source node, when it detects the network congestion, it reacts with a slower transmission in hoping not to worse the network congestion. The idea of this control mechanism comes from the characteristics of TCP and UDP flows and the simulation results from [2]. The characteristics of TCP and UDP protocols could lead to UDP flows get most of bandwidth shares and only a little bandwidth is available for TCP flows. This result is shown in the section II simulations. If the congestion signal from TCP flows can be used as a congestion indicator for the source node, this could help the source node control the TCP and UDP traffic transmissions. When the transmission path is congested, the source node can stop the transmissions of lower priority flows and let higher priority flows keep their transmissions. With regulated transmission, higher priority flows can have the better transmission performance.

Depending on a transmission requirement of each flow and its importance, network administrators assign a transmission priority to TCP/UDP flow at a source node. A critical flow can get a higher transmission priority and a fair flow may get a lower transmission priority. The congestion control mechanism collects flows' priority information. TCP and UDP flows can concurrently transmit their packets. This mechanism will routinely check the congestion signal issued by TCP flows to detect congestion on the transmission path. If the network is congested, it stops the transmissions of lower priority flows to release some bandwidth share for higher priority flows to get a better transmission performance. Otherwise, if there is no congestion, perhaps more bandwidth available on the network, it starts the transmission of higher priority flows to enhance the bandwidth utilization. To prevent the transmission starvation of lower priority flows, this mechanism uses a priority aging method to upgrade the priority of lower priority flows. After a transmission period elapsing, lower priority flows can have the higher priority and allocate more bandwidth share to transmit packets. Figure 3 is the diagram of congestion control mechanism.

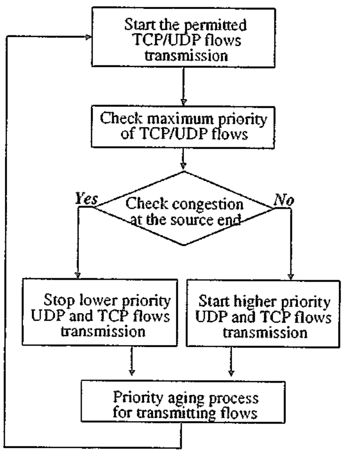


Figure 3. A Diagram of Congestion Control Mechanism.

2. Time slot mechanism

The time slot mechanism is an application of time sharing concept. The bandwidth is divided into many transmission units. Each transmission unit is a time slot. In each time slot, the source node only allows one flow to transmit its packets and this flow can use all available bandwidth as much as it can. All other flows must yield the right of way during the time slot. How many time slots can a flow get? It depends on the priority, assigned by network administrators at a source node, of a flow. The higher priority a flow has, the more time slots it gets. A higher priority flow has a longer period to transmit its packets and get a better transmission performance. With the time slot mechanism controlling, the transmission behaviors of TCP and UDP flows will be regulated. UDP flows can no longer occupy the bandwidth share irresponsibly. Moreover, the transmission performance of each flow can be ensured with its priority.

The time slot mechanism can adopt the first-comes-first-serves principle to append a flow to a round robin scheduling queue and transmit its packets by turns. When a flow take turns at transmitting its traffic, the time slot mechanism assigns a number of time slots to the flow according to its priority. Then a transmission token is assigned to the flow to start its transmission and all available bandwidth is given to this flow in the assigned period. A round robin scheduler used by the time slot mechanism to arrange each flow's transmission. With time slot mechanism, although each flow can get an assigned period to send its packets, but a transmission starvation situation may happen to the lowest priority flow when higher priority flows continue arriving. A priority aging method is also incorporated to upgrade the long waited priority flow. To provide possibility of upgrading lower priority flow will prevent the starvation. Figure 4 is the time slot mechanism.

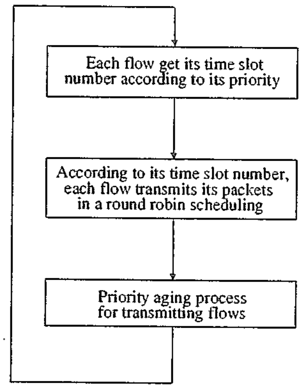


Figure 4. A Diagram of Time Slot Mechanism

IV. A simulation of congestion control and time slot mechanisms

Several simulation scenarios are simulated to illustrate the operations of these two mechanisms. With the simulation results, one can obtain some transmission performance about these two mechanisms. Factors that may affect the algorithms we also investigated.

The Network Simulator - ns (version 2 beta release 5) [5] is used as the simulation tool. It is implemented on a PC based environment with an Intel pentium-III-500 CPU, 256MBytes RAM and a linux operating system. For the required simulation function, an internal variable binding in the ns TCP agent module is made by us to get the congestion action of TCP flows. With this variable binding [6], the detection of a congestion signal from TCP flows can be supported in Otcl and simplifies the coding of a simulation scenario in the ns environment. The topology of simulation is same as the simulation in the section 2 (see Figure 1). A number of TCP/UDP traffic flows are simulated to transmit packets from the S1 and S2 source nodes, all traffic flows have the same routing path and share the same bandwidth from N1 node to N4 node, then reach the D1 andD2 destination nodes.

Figure 5 shows the basic end-to-end transmission procedure in each simulation scenario. The simulator starts to transmit TCP and UDP packets from the source nodes to the destination nodes at the beginning of the simulation and maintains an individual time scheduler. After a period transmission, the simulator checks the transmission size of each TCP and UDP flow to see whether it finishes its transmission job or not. If the transmitted data size is larger than the specified transmission size, it means the TCP/UDP flow finishes its transmission job and the transmission time will be recorded. Otherwise, it continues its transmission job. Once a TCP or UDP flow finishes its transmission job, the simulator checks whether all flows that have finished their transmission job or not. If they do, then the simulation ends. All simulation runs are based on this same procedure. In each simulation scenario, two group traffic flows are based on this same procedure with the same control mechanism or the best-effort traffic to simulate traffic flows' transmission concurrently. One group is from the S1 node to the D1 node and the other is from the S2 node to the D2 node,

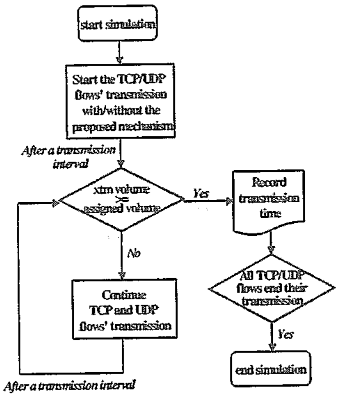


Figure 5. A flow chart of simulation scenario

Most of Internet applications are composed of TCP and UDP applications. The ratio of Internet TCP/UDP traffic flow is the basic ratio for our scenario to simulate the transmission of Internet flows. From the MCI/NSF's very-high-performance Backbone Network Service (vBNS) project [7], we find the ratio of TCP and UDP traffic flows: if there are 100 traffic flows over Internet, then approximated 90 traffic flows are TCP flows and approximated 10 traffic flows are UDP flows. Base on this, with 100 traffic flows the TCP may vary from 81 to 99 whereas UDP varies from 19 to 1 during simulation. With different TCP/UDP traffic flow combinations, the transmission behaviors and performance of the proposed mechanisms can be analyzed further. The transmission size is another factor that may impacts the transmission behavior of traffic flows. We use a 10 Kbytes file as the smaller traffic flow source and a 1 Mbytes file as the larger traffic flow source on the network, and these two transmission lengths are used as the traffic sources of TCP/UDP flows in the simulation scenarios. For the differential service simulation, the four different priorities, from the 1st highest priority to the 4th highest priority, are assigned to the TCP flows and two different priorities, the 3rd highest priority and the 4th highest priority, also are assigned to the UDP flows. These scenarios can help us realize the transmission characteristics and differential service performance of the two proposed mechanisms.

When there is contention for network bandwidth, a queueing discipline is used to schedule network resources. With different queueing discipline over the network, network applications may have different transmission behaviors and performance. Queueing disciplines are also important since they may impact the transmission performance of the two proposed control mechanisms. In our simulations, four different queueing disciplines: FCFS (First Come First Serve), SFQ (Stochastic Fair Queue), RED (Random Early Detection) and DRR (Deficit Round Robin) are implemented to schedule network applications' transmissions.

Many network applications use different control mechanisms to transmit their packets on the Internet. The Internet is no more a simple network environment. The Internet's operations are complicated. It is impossible only one transmission control mechanism operates on the network. Usually, several mechanisms co-work with others on the Internet. The congestion control mechanism or time slot mechanism will coexist with other transmission control mechanisms or with the best-effort traffic, without any control mechanism. Co-working with a different control mechanism, the proposed mechanisms may have a different transmission performance and behaviors. To simulate different combinations, there are two groups of end-to-end traffic transmissions with the individual flow control mechanism and time scheduler in all simulation scenarios. The five group transmission environments are used in our simulation scenarios.

V. Results and analysis

Parameters used for the simulation include transmission size, queueing disciplines, transmission performance, TCP/UDP ratios, environment setting and parameter settings. Several simulation results are illustrated in the following paragraphs.

Two different sizes of data transmission, 10 Kbytes and 1 Mbytes, are tested to measure the size sensitivity of congestion control algorithms. The simulation results show that the congestion control mechanism provides a significant differential service among the TCP/UDP flows at the transmission size of 1 Mbytes. Most of the 1 Mbytes TCP/UDP flows receive different transmission performance based on their transmission priorities. In both pure and mixed congestion control mechanism environment, the priority of the data dictates the transmission performance. For 10 Kbytes TCP/UDP flows, the congestion control mechanism also provides a differential service. But, there are cases that the transmission performance is inconsistent with their transmission priorities. Some traffic flows without higher priority showed better transmission performance. In other words, the congestion control mechanism exhibit differential service when the transmission size is large.

The time slot mechanism has different simulation results from that of the congestion control mechanism. The differential service transmission behaviors only occur in the 1 Mbytes TCP/UDP flows. When the transmission size is of 10 Kbytes, the transmission performance of each TCP/UDP flow behaves as a first-come-first-serve transmission. The first TCP/UDP flow starts its data transmission at the beginning of transmission operation.

When the first TCP/UDP flow ends its transmission, then the second TCP/UDP flow starts its data transmission. Such a transmission operation does not end until all TCP/UDP finish their data transmissions sequentially. This transmission behavior of the time slot mechanism causes that the first transmitted TCP/UDP flow gets the best transmission performance and the last transmitted TCP/UDP flow gets the worst transmission performance. The first-come-first-serve transmission behaviors occur in both of a pure and mixed time slot mechanism environments. This is why the drawings of TCP/UDP flows' transmission performance shown in Figure 7 are straight. There is a particular situation in the pure time slot mechanism environment, the TCP/UDP flows start to transmit at the same time slot, no matter the TCP/UDP flow ratio is high or low, and they get an almost same transmission performance. And many drawings of TCP/UDP flows transmission performance are straight and overlapped in a pure time slot mechanism environment.

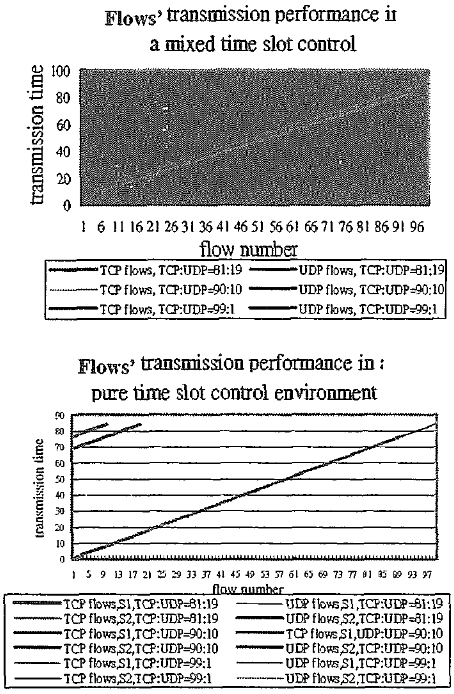


Figure 6. A performance of time slot mechanism in 10 Kbytes file size

From the information shown in Figure 7, it shows the time slot mechanism can not work properly to provide a differential service for TCP/UDP flows when the transmission size is relatively small to each assigned transmission time. Comparing with the congestion control mechanism, the time slot mechanism is more sensitive to a transmission size. This also reveals that these two proposed mechanisms have better differential service behaviors for flows when transmission size are large.

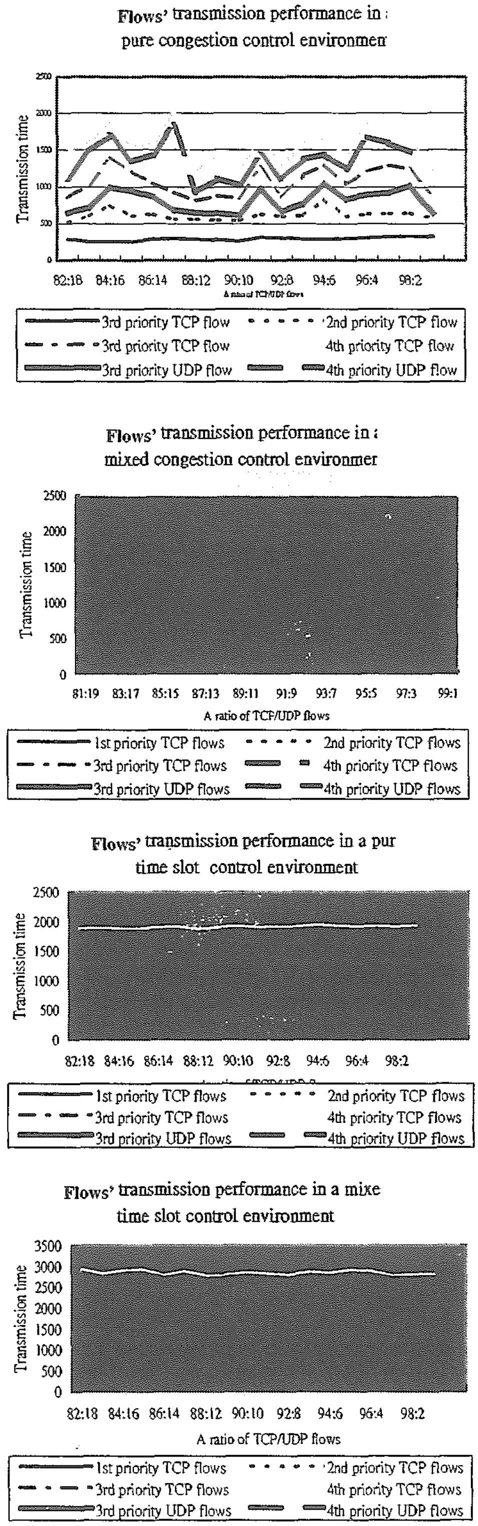


Figure 7. Performance of congestion control mechanism and time slot mechanism in the pure and mixed transmission environment

Table2 is the summary of average transmission performance of traffic flows from the S1 source node to the D1 destination node. These flows are controlled by either the congestion control mechanism or time slot mechanism. The transmission size of each flow is 1 Mbytes.

Table 2 S1 Traffic flows' average transmission performance

Flows' priority / Transmission Environment	1st priority TCP flows	2nd priority TCP flows	3rd priority TCP flows	4th priority TCP flows	3rd priority UDP flows	4th priority UDP flows
1	290.4	612.0	1084.6	1581.1	802.7	1390.2
2	450.4	1114.7	1447.6	1910.9	1533.6	1719.1
3	582.2	1232.3	1575.5	1937.2	1402.8	1771.3
4	1214.9	1812.0	2266.4	2839.1	1741.3	2058.6
5	1004.3	1031.0	1001.1	1028.3	145.6	151.8

Legend:

Transmission environment:
1 : A pure congestion control environment
2 : A pure time slot environment
3 : A mixed congestion control environment
4 : A pure time slot environment
5 : A best-effort traffic environment

Unit: Seconds

Table 2 shows that the transmission performance of congestion control mechanism in each simulation scenario is better than the time slot mechanism. The time slot mechanism receives a poorer transmission performance, it may result from two possible factors. One possible factor is the traffic flow can't fully utilize all the available bandwidth since a required burst bandwidth of a flow is less than the offered available bandwidth in a pure time slot controlled transmission environment,. The other is most of bandwidth in a mixed time slot controlled transmission environment is shared by best-effort traffic, only a little bandwidth available for the time slot mechanism to regulate flows' transmission. In additions, the congestion control mechanism always keeps higher priority traffic flows to share the bandwidth as possible as they can. This may be another reason why the congestion control mechanism gets a better transmission performance than the time slot mechanism.

Comparing the transmission performance of the third and fourth priority TCP flows with the transmission performance of third and fourth priority UDP flows in Table 2, we can find, with the same priority, the transmission performance of UDP flows is better than the TCP flows. This result coincides with the transmission characteristics of TCP and UDP flows. It also gives us a hint: To assign the lower priority to the UDP flows if the transmission performance requirement of them is not so critical, otherwise they will share extra bandwidth that should be shared by higher priority TCP flows and slow down higher TCP flows' transmission performance.

The transmission performance of these two control mechanisms in the best-effort traffic environments is interesting. From the 3rd row, 4th row and 5th row in Table 4, one can find the congestion control mechanism has a better transmission performance than the time slot mechanism when they operate with best-effort traffic flows. But comparing the TCP flows' transmission performance in the mixed congestion control environment with the best-effort traffic environment, only the highest priority TCP flows with the congestion control mechanism is better than the best-effort traffic flows, a guaranteed service is provided. The other priority TCP flows do not outperform the best-effort traffic flows. In other words, there is no need to have too many priority levels in a differential service mechanism. Two levels are enough. High priority flow will receive a guaranteed service and a better performance than others. On the other hand, the low priority or the best-effort traffic flows will compete bandwidth and there is no guaranteed transmission.

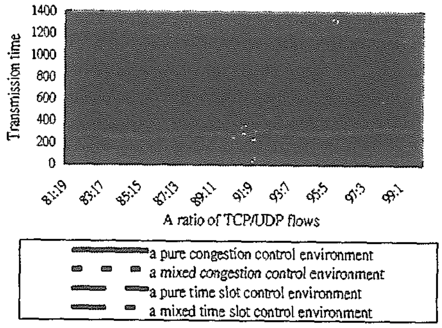


Figure 8. A relationship between the transmission performance of flows and a ratio of TCP/UDP flows

Various ratios of TCP/UDP data flows are simulated. The TCP/UDP ratios do not show significant effects on the traffic performance. Figure 8 uses the first priority TCP flows in the four kind of transmission environments as an example to demonstrate the variations of transmission performance of flows as the TCP flow number increases and the UDP number decreases. From the transmission performance curve fluctuations, it is no obvious evidence show there is a relationship between a ratio of TCP/UDP flows and their transmission performance.

Proper parameter settings would let the control mechanisms have better control operations to satisfy transmission requirements. With numerous simulation settings, one can find that the priority aging time is important. The priority aging time will impact the transmission behaviors of two proposed mechanisms. A short priority aging time allows the lower priority flows to be upgraded sooner than otherwise. In that case, soon all the flows become the highest priority. This traffic in turns degenerate into a best-effort traffic. The differential service is no more to be supported by these two mechanisms. A priority aging time, however, may cause a flow with lowest priority to starve because other higher priority flows may keep coming and jump ahead of the queue.

For different priority flows, we also find it is important for the time slot mechanism to have a proper transmission period as a length of time slot unit. A proper time slot will allow the time slot mechanism to provide a better transmission operation for TCP and UDP flows. If the transmission time is too long, the first-come-first-serves transmission operation will happen just like the case specified in Figure7. If the transmission time is shorter than the round trip time, TCP flows can not get their ACKs from the destination node, the retransmission of TCP flows will happen repeatedly and their transmission performance will be poor. Additionally, if differences among time slot length of different priority flows are too large, it may cause that lower priority flows starve. But, if they are to small, a differential service behavior is not obvious. For the time slot mechanism, a proper setting of time slot length is important.

VI Conclusion

With different transmission mechanisms, such as TCP and UDP, traffic flows show different transmission behaviors and their performance. TCP flows are responsive to the network congestion whereas UDP flows are irresponsible. UDP flows often deteriorate the network congestion, sometimes even cause a congestion collapse. The uncontrollable of a UDP flow is a major problem over the Internet.

The congestion control mechanism and time slot mechanism are the two different source-based flow control mechanisms proposed in this paper. These two mechanisms are applied at the source node to regulate the transmissions of TCP and UDP flows. With these two mechanisms, UDP flows are no more irresponsible to the network congestion and they are a little bit more controllable.

The source-based control mechanisms regulate the TCP and UDP flows at the source node. They are compatible with the current transmission operation environment over the Internet. No additional device, protocol, or control mechanism is needed to implement these two mechanisms. The only operation cost of these two mechanisms is the execution time at the source node.

Simulation results show that: 1. The transmissions of TCP and UDP flows can be controlled at the source node and one can provide the differential service for each TCP and UDP flow. 2. The congestion control mechanism has a better transmission performance than the time slot mechanism. 3. The ratio of TCP/UDP flows doesn't significantly impact the transmission performance of these two mechanisms. 4. It is unnecessary to have too many priority levels for the differential service. Two levels are enough. 5. It is helpful to set the lower priority to the UDP flows. 6. A proper operation setting of these mechanisms could improve their performance.

Acknowledgement

This research is sponsored in part by NSC 89-2416-H-009-011

Reference

[11] Andrew S. Tanenbaum, "Computer Networks" third edition, Prentice-Hall Inc., 1996

[13] David D. Clark and Wenjia Fang, "Explicit Allocation of Best Effort Packet Delivery Service", ACM Transactions on Networking, Vol. 6, No. 4, 362-373, August 1998

[12] Douglas E. Comer and David L. Stevens, "Internetworking with TCP/IP Volume II : Design, Implementation, and Internals" second edition, Prentice-Hall Inc., 1994

[8] K.Thompson, G. Miller, and R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", IEEE Network, 11(6):10-23, Nov.1997

[5] Kevin Fall and Kannan Varadhan, "ns Notes and Documentation", May 11,1999

[4] S. McCanne and Sally Floyd., "ns-Network Simulator", URL http://www-mash.cs.berkeley.edu/ns/

[2] Sally Floyd and Kevin Fall, "Promoting the Use of End-to-End Congestion Control in the Internet", To appear in IEEE/ACM Transactions on Networking, May 3, 1999

[7] Sally Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, 24(5):10-23, Oct. 1994

[9] Sally Floyd and Van Jacobson, "Random Early Detection Gateway for Congestion Avoidance", IEEE/ACM Transactions on Networking, 1(4):397:413, Aug. 1993 URL http://www-nrg.ee.lbl.gov/nrg-papers.html

[10] Sally Floyd and Kevin Fall, "Router Mechanisms to Support End-to-End Congestion Control", Network Research Group Lawrence Berkeley National Laboratory, Feb. 15, 1997

[1] Van Jacobson and Michael J. Karels, "Congestion Avoidance and Control", Proceeding of ACM SIGCOMM, 1988,

[3] URL http://diffserv.lcs.mit.edu/

[6] URL http://www.vbns.net/stats/flows/html/index.html