國 立 清 華 大 學

碩 士 論 文

題目：以太網路上 TCP 速率控制方法之研究

A Novel Rate Control Scheme for TCP over Ethernet

所別：資訊工程研究所

學號姓名： 884347 黃 力 行

指導教授： 黃 能 富 教授

中華民國九十年六月

摘要

本論文主要在提出以太網路 TCP/IP 頻寬管理系統核心之設計方法。在端點對傳的 TCP/IP 資料連線中，此系統設計定位為穿透式網路裝置並提供主動式的網路控制架構。所有穿過裝置的封包會暫存在系統佇列中，然後依照管理者所制訂頻寬管理之政策決定釋放的時間點，達到控制頻寬的目的。

目前常見的頻寬管理方法，如延遲回應（delayed-ACKs）和變更 TCP 滑動視窗（Sliding window），均需與 TCP 擁擠管理演算法緊密結合。使用本設計不需要改變 TCP/IP 協定，就可以精確地控制以太網路上 TCP/IP 連線速率，使其遵循網路頻寬政策或服務品質保證（Quality of Service, QoS）並且能和其他擁擠管理協定完全相容（如 Congestion Management Protocol 和 ECN）。

ABSTRACT

The thesis proposes a kernel design of bandwidth control system for TCP/IP over Ethernet. This system acts as a transparent device in the end-to-end uni-cast flow path and provides active network control architecture. The received packets will be queued inside first and then according to the bandwidth management policy, made by the MIS manager, these packets will be forwarded in a suitable time to achieve the allocated bandwidth.

The proposed scheme precisely controls the rate of TCP/IP flows over Ethernet without modifying the TCP/IP protocol. The famous rate control schemes, such as delayed-ACKs scheme or changing window-size scheme, are tightly bounded with the ways of TCP congestion avoidance. Nevertheless, the proposed scheme is independent with TCP congestion managements. This novel feature makes the proposed scheme be completely compatible with any congestion avoidance protocols, such as CM or ECN.

CONTENTS

FIGURES

1. INTRODUCTION

With the booming use of Internet, the requirement of QoS on TCP/IP traffics is more and more important. The general TCP/IP has its flow control scheme, but it can't be controlled in centralized fashion and it is also too simple to various hot applications in Internet, especially real-time applications. In LAN environment, local machines will compete with each other for the bandwidth of network. In this kind of network environment, all applications have the same priority in ideal, and can't ensure their quality. This problem is more and more important today. In recent years, network services in LAN are increasing so quickly and many enterprises use Ethernet/LAN to communicate between the internal departments. Many traditional paper works already became electronic signals in network. This trend makes the traffic of LAN so busy and large. For commercial reasons, industry must make sure the quality of important services.

Today, people cannot control TCP/IP traffic very well as they want over Ethernet. The behavior of TCP/IP in LAN is mainly decided by probability of the congestion happened. So users cannot make sure how many bandwidths allocated to them. It is a terrible thing for network services, especially when some services need a stable bandwidth environment.

Therefore, we need to classify those applications, and hope the important services have higher priority and ensure their quality. This means that the services in future LAN, which should have priority in allocating bandwidth and take care those traffic flows in network. In order to add the QoS on TCP/IP and improve the utilization of bandwidth, we design a new system to act as a transparent device in traditional TCP traffic path. Using our algorithm to control packet rate, we can make sure they will fit the QoS class. Our goal is to control TCP/IP traffic over Ethernet and

design a policy-based environment.

1.1 Ethernet Background

Ethernet was originally a shared-medium broadcast bus technique using CSMA/CD (Carrier Sense Multiple Access with Collision Detection). First standardized version used a bit rate of 10Mb/s, and now the speed is already up to 100Mb/s (Fast Ethernet), 1000 Mb/s (Giga-bit Ethernet). As originally designed, all stations were attached to the broadcast bus, which formed the shared "Ether" used for communication.

All stations transmitted on, and listened to, the same bus. This type of medium is now known as Unshielded Twisted Pair (UTP, 10BaseT) Ethernet (10BaseT is like “star” structure but it is topologically the same as shared coax bus, 10Base 5). The attachment unit is the network interface card, and led to the development with a centrally located hub that propagated the transmit signals of each station to the receive inputs of all the others. Each station has a transmit connection and a receive connection. Collisions are detected by having each station listen for its own signal to return from the central hub. [9]

Having too many collisions is a big problem in hub structure. If one station has collisions, the signal will broadcast to all stations that connected to this shared bus.

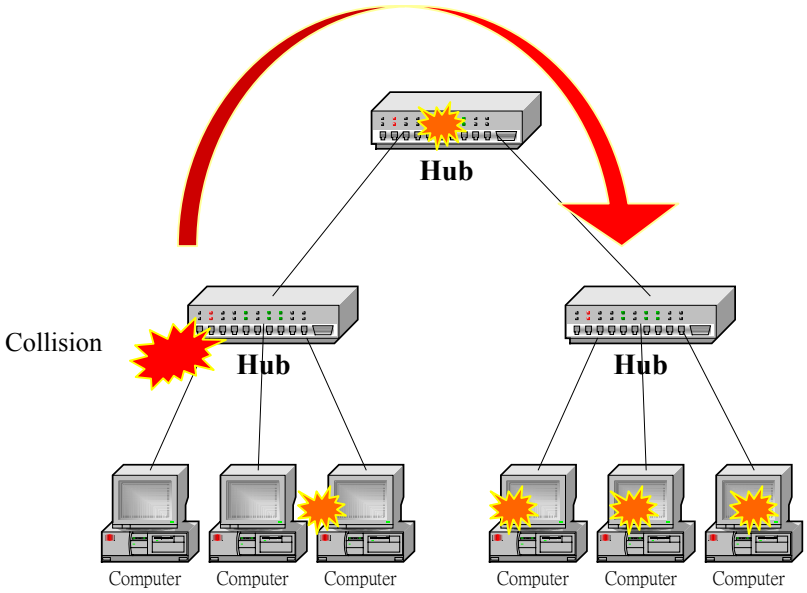

Figure 1.1 Collision in hub infrastructure

In recent years, the layer-2 LAN switching is more and more popular. Using a switched structure rather than shared connection to the hub, each station has the same possibility of sending and receiving 10Mb/s, for an aggregate of 20Mb/s total. This actually eliminates the "collision detection" part of the Ethernet protocol. LAN switching reduces the effect of collision and provides a more efficiency utilization of bandwidth.

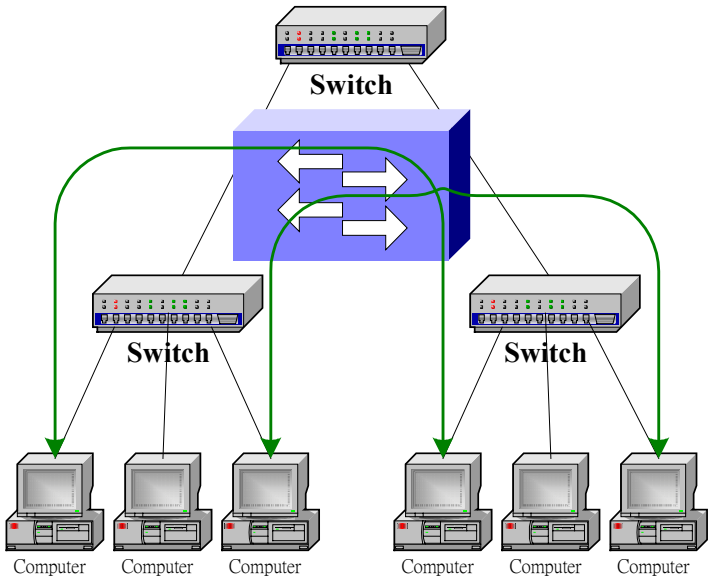

Figure 1.2 Traffic flow of switching infrastructure

Although LAN switching is better then traditional hub infrastructure, it still

cannot control the bandwidth assignment. LAN switching can provide a more smooth traffic path, but most LAN environments only have one outbound port, all network traffic that connects to outside still need to share the outgoing channel. So, LAN user cannot control the bandwidth or request the committed rate. The bandwidth distribution maybe fair but cannot be dynamic allocated for users.

1.2 Active TCP Control

Transmission Control Protocol (TCP) is an end-end protocol; it provides a connection-oriented and reliable service. TCP assumes that the network is no reliance, so it uses a windows-based flow control mechanism and indirectly detect network status by setting time-out or checking the duplicate ACKs. If the network status is not good enough, the TCP will limit the number of packets sent to the network.

There are many ways that can actively control TCP connections. We group those methods to two parts, one is “Protocol-based Solution” and the other is “Network-based Solution”.

- Protocol-based Solution

Here we define “Protocol-based Solution” is to modify TCP/IP protocol on sender and receiver. The spirit of “Protocol-based Solution” is to avoid the cost of dropping packets. In current TCP/IP networks, an IP router when congested simply drops packets. But the TCP sender can’t detect how serious congestion happened until packet loss is inferred by the receipt of 3 duplicate ACKs or detected by the timeout. It would be a vicious circle in congested network and reduce the utilization of bandwidth. There are many ways to change the original TCP flow control scheme, and detect the network status directly. Like sending ICMP (Internet Control Message Protocol) or ECN (Explicit Congestion Notification) packets, which uses the TOS

field of the TCP header, explicitly reports the network status to the TCP source.

ECN (Explicit Congestion Notification) is an end-to-end congestion avoidance mechanism. When routers detect congestion before the queue overflows, routers are no longer limited to packet dropping as an indication of congestion. It could instead set a Congestion Experienced (CE) bit in the packet header of packets from ECN-capable transport protocols.[6]

When the TCP source got the exact information of network, it can regulate the rate of sending data and achieve the rate control. Because TCP is weakness on detecting network congestion, the goal of protocol–based solutions is to provide a strong congestion-detecting interface, like CM (Congestion Manager). CM is to modify TCP/IP protocol stack and insert CM Protocol between TCP and IP. It can improve the congestion handling ability of TCP and carefully increases the traffic rates by feedback mechanism. So applications can obtain an unprecedented degree of control over what they can do in response to different network conditions.[4] It also can be integrated with ECN.

This kind of rate control method is good for enhancing the utilization of network bandwidth. But they are not designed for exactly controlling the rate of per flow. In other word, they cannot provide the QoS (Quality of Service) for applications. Because “Protocol-based solution” has a major disadvantage, it cannot change other machine’s behaviors. TCP sender has to compete bandwidth with those hosts who are sharing the backbone, it can slow down its speed and avoid congestion, but if it wants to allocate more bandwidth; it is very difficult to snatch other hosts’ bandwidth in busy LAN environment. In order to guarantee the bandwidth for TCP applications, we need new network-based solution.

- Network-based solution

We define “Network-based Solution” is that it doesn’t need to modify TCP protocol and do active TCP control from network devices, like IP router or switch. The control scheme of “Network-based solution” is built-in at router or transparent network device. As the RED (Random Early Detection) scheme, it intentionally discards packets in a probabilistic manner when the number of stored packets in the buffer exceeds a certain threshold, that indicating buffer congestion. When packets are dropped, it actives the traditional TCP flow control and forces the TCP flow to slow down its transmission rate.

RED (Random Early Detection) router is different with traditional routers. The congestion of network is monitored by the average queue size. The probabilistic manner of dropping packet can avoid the burst drops and drawbacks. In fact, the extension of RED can integrate with ECN by marking the IP header instead of dropping packets. Although the general design of RED doesn’t keep per-flow state, it has the ability to monitor per-flow state. Because routers can catch all information of connections passed through, it also can make different probabilistic manners for different traffic flows and control the rates of traffic.[14]

There are already many commercial products for active TCP control today. In general, those products were designed as a transparent device and act as Layer-4 switches. The control scheme of those products is directly and explicitly controlled the TCP flows. Like delaying ACKs or modifying window size of packets, those methods need to catch packets passed through and count the delay time to slow down the ACK stream of TCP flow or change the window size to speed up/down the rate of traffic.

ACK delaying approach needs to modify the contents of the ACK packet. It

records per-flow status, which include the rate of ACKs and the number of ACKs. The rate of ACKs is relative to TCP traffic rate. Decreasing the rate of ACKs can reduce the TCP rate. Changing TCP window-size is also relative to TCP traffic rate, because with TCP's sliding-window protocol the receiver does not have to acknowledge every received segment. The number of segments, which is sent by TCP sender, is according to the window-size in packet. The advantage of delaying ACKs and controlling window-size is that the network device doesn't need to queue packets. It only needs to modify the ACKs and window-size. In general, this kind of rate control system can exactly control per-flow rate.

Our rate control scheme also belongs to "Network-based Solution". We implement it on a Layer-4 switch developed by our lab. And we design this scheme on network-based; this is because it can handle all TCP flows over the LAN in this structure. It also has more freedom on rate controlling. Because it can handle all TCP flows, the bandwidth distribution could change according to people's policy and we could speed up/down any TCP flow as we wish. Even if the LAN is so congested, we still can allocate the bandwidth we need and redistribute bandwidth to TCP flows.

2. SYSTEM MODEL

Suppose the environment is a simple LAN. All connections have the same opportunity in competing bandwidth with each other. In this environment, the bandwidth distribution is uncontrolled. When a TCP connection loses some packets, this connection will follow the slow-start mode and suffer the unfair chance to share the bandwidth.

If we focus on whole process of a connection's life, the variety of traffic speed is huge. In other words, the connection wouldn't have a smooth process on the Ethernet. If we can create a transparent device, which could catch all traffic passed through; this device can regulate those flows and provide quality of service (QoS) for each connection.

In fact, there are many proposed methods to date in terms of network control. Those methods can be classified into two main classes. One is explicit TCP network control and the other is implicit TCP network control, which already has been discussed in previous chapter.

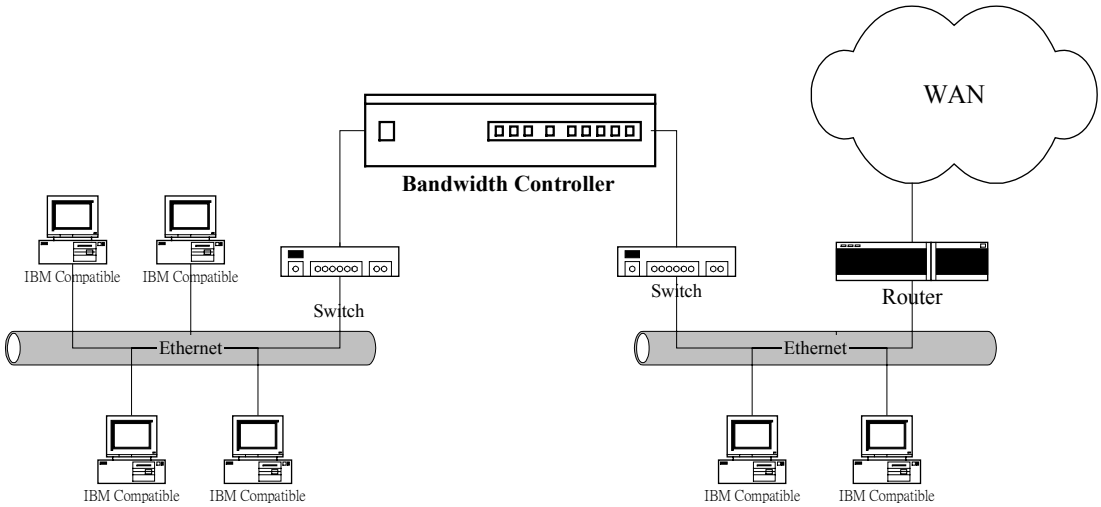

Figure 2.1 System model

Protocol-based methods need to modify TCP protocol, but this is a very difficult mission to replace TCP software on so many machines, especially TCP is

already so popular today. We propose an implicit method, which does not need to change the original TCP devices and software. It should be a better way.

Using transparent device to control network traffic has some advantages:

- End machines don't need to change. What we need is a wire-speed rate control device and it is totally unnecessary to modify the original setting of original network environment.
- Efficiency and cost down.
- Centralized control. We can import policy-based manager system and it is easy in maintenance and configuration for manager.

But it also has some disadvantages:

- Must be high-end devices
- Packets may be dropped when connection over the rate policy.

2.1 Local Area Network technology

The structure of Ethernet in LAN has two types: one is traditional share bus (half-duplex mode) and the other is switch-based (full-duplex mode). The difference is in switch function, which actually eliminates the "collision detection" part of the Ethernet protocol. In a switched environment, each station has the possibility of sending and receiving in full rate (10/100 Mb/s), for an aggregate of double rate (20/200 Mb/s) total. But using hub, the congestion would be a big problem and the TCP traffic would run in unstable environment. It is very easy to be interfered by other traffic on the shared bus.

In general, using switches in a LAN can avoid collisions and get better efficiency of network. But if there is still only one outgoing port, like most business or campus, which connected to Internet, congestion is always a big problem. The bottleneck is just moved to the path between outbound router and first stage switch.

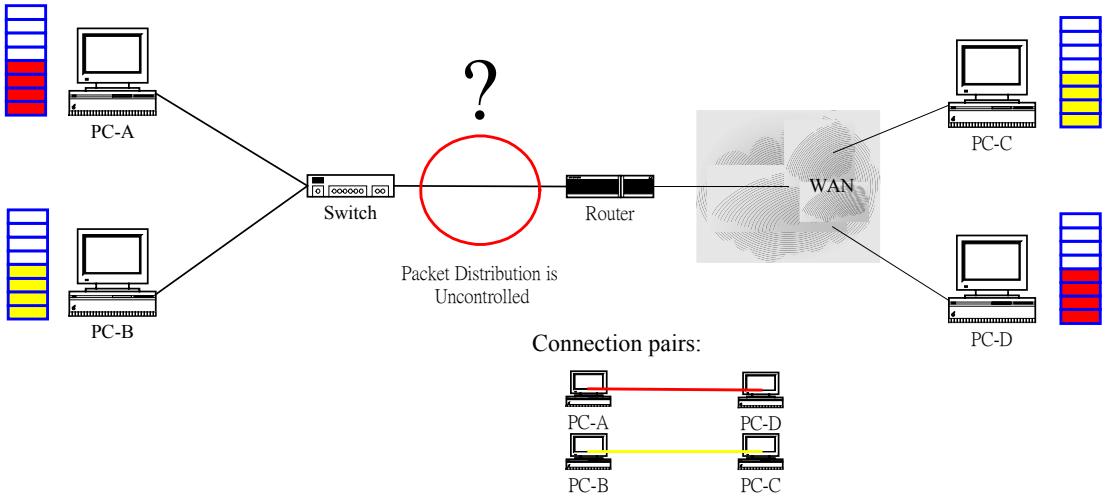

Figure 2.2 The general LAN environment

The way of control rate is to place a transparent device between switch and router. In this critical position, the device needs to handle all traffic, which pass through it. This device can redistribute the bandwidth for each TCP connections by using the rate control scheme proposed in this thesis.

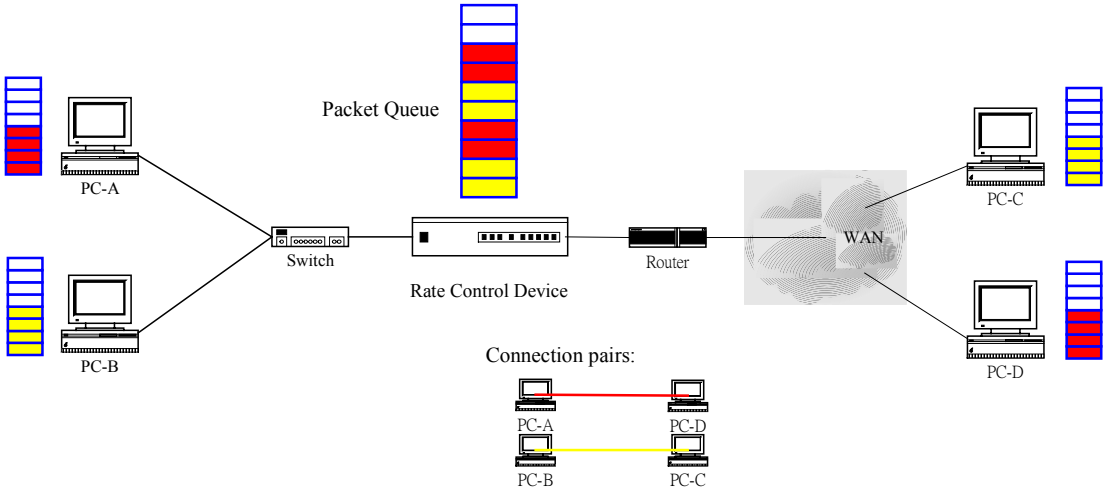

Figure 2.3 The position of rate control device

The key point is how to reach wire speed when the traffic load is heavy. For getting better performance, much complex algorithm should not be employed and the number of system clocks to handle each packet should also be reduced.

3. RATE CONTROL SCHEME

RCS (Rate control scheme) is a network-based solution, which is designed to maintain all connections passing through the device. The main concept is to divide the network bandwidth into time slots. Using those slots, many virtual channels are created for different flows. For each flow, if the allocated number of time slots is 20% out of the available, it just like running on a virtual channel with 20% bandwidth allocated.

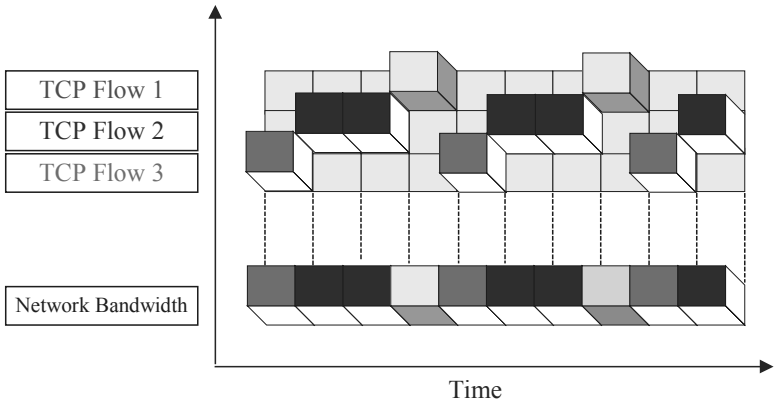

Figure 3.1 Time divisions slots of RCS

In other words, RCS introduces the time division concept. RCS divides the network bandwidth into many short time slots. If every slot is a very short time, we can rebuild the network traffic exactly.

There are two main parts of this rate control system:

- Determine the current rate of a flow.
- Decide which time slot this packet belongs to.

If each TCP flow's rate is controllable, this system can provide at least three QoS types:

1. Committed Rate Control

   It can guarantee the rate of a flow.

2. Maximum Rate Control

It defines the maximum rate of a flow.

3. Minimum Rate Control

It defines the minimum rate of a flow. The actual rate allocated for a flow can over this minimum value, as the free bandwidth is available.

When the traffic flows come in, RCS (Rate Control Scheme) system will determine the rate for them by flow classification module and the arranged policies. RCS will dispatch packets of this flow according the allocated rate.

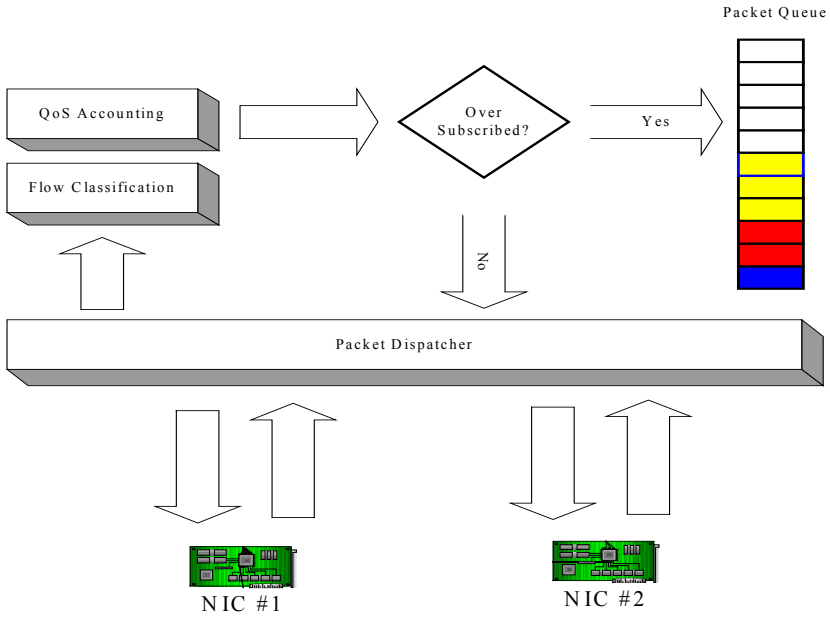

Figure 3.2 Structure of rate control scheme

Rate Control Scheme works as a scheduler in the kernel of a device. After looking up the flow tables, the packet will be passed to RCS with its flow information. RCS will transfer this packet according to its legal transmission rate described in flow information.

If flows are already over the limited rates, the followed packets of these flows will be assigned to future time slots. Actually, packets are temporary stored in the packet queue of RCS and wait until more bandwidth of this flow is available. Because the RCS delays the packet in its packet queue, the whole TCP flows control will slow

down to fit the network status that was created. Actually, the goal of RCS is to create different network environment for each TCP flow. If we define the rate of TCP flow is 200Kbps, the RCS will make a 200Kbps channel for this flow exactly. The expected and assigned bandwidth will activate TCP flow control automatically. It can limit the number of packets allowed for the senders/receivers of TCP flows to enter the network, and, the goal of smoothly controlling TCP flows rate is achieved.

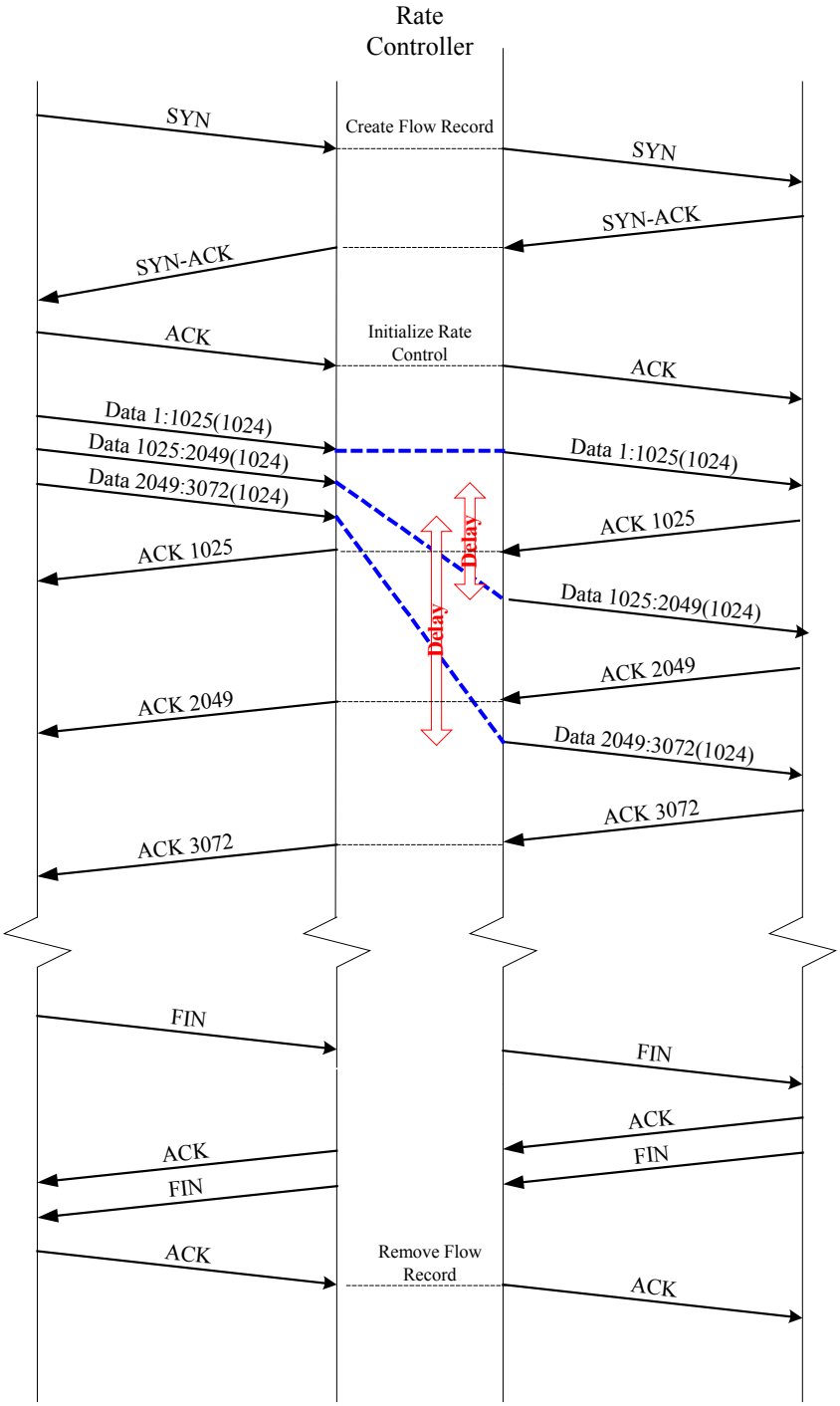

Figure 3.3 Time line of connection in delay data packets

This scheme needs a large packet queue to buffer the flow packets, especially when a burst of packets happens. The formula is shown below.

Queue Size = Network Bandwidth * Buffering Time

Ex.

Under an 100Mbps full-duplex Fast Ethernet environment, the network bandwidth is 200M bit/sec. If we define the Buffering Time is 1 sec, the queue size is

200M/8 = 25MBytes

The Buffering Time cannot be set too small. Otherwise, the packet-dropping rate will increase. This is because in this case, it is easy for the incoming traffic to overflow the packet queue. But the Buffering Time also cannot be set too long for the memory cost issue.

If the queuing time between any two consecutive packets of a TCP flow is longer than 1.5 sec, then the TCP retransmission may be happened. For example, if the time between two consecutive packets is too long, it may occur duplicate ACKs, one is in our RCS' packet queue and the other is just sent due to the TCP retransmission. This will reduce network performance. So RCS will check the time and drop packets instead of queuing them after certain period.

3.2 Time Division Rate Controller

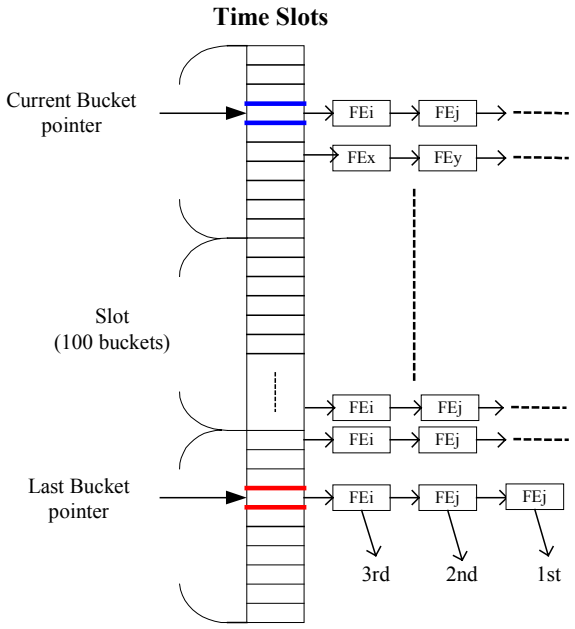

Figure 3.4 Structure of time division queue

After the NIC driver announces packets arriving, the dynamic rule table will be checked to see if any information about this flow is already existed. The system will fetch the corresponding information if any record is found or, do “Flow Classification” and construct a new one.

Rate Controller will check whether this flow over-subscribes the bandwidth by record the octets of flow in connection information table. If it is not over-subscribed, Rate Controller forwards this packet directly and logs the octets of this flow. But if it is over-subscribed, Rate Controller would store this packet in a proper bucket of the time division queue by calculating the position of the queue according to the fraction of packet size and reserved bandwidth of this TCP flow.

Every certain time (10 ms), a system timer is triggered to transmit packets in the corresponding bucket and clear all the entries that were sent. The “Current bucket pointer” will be increased to next bucket and waits for the next sending time. If there is no packet comes in, a permanent task will wake up and start to clear the flow

entries in the last non-empty bucket in Timer Ring, and then transmit the corresponding packets. We call this “Bandwidth Borrowing” which is useful to improve the network utilization.

3.3 MSS Control

Sometimes it is necessary to limit the packet size for the convenience to manage the network bandwidth. Here is one tough condition usually happens: if the packet size is relatively larger; the committed rate is relatively lower and the sender might transmit a packet with huge size. The packet will be queued for a long time and it may result in TCP retransmission. For example, a 1500bytes in a 1kbps flow might be queued for 12 seconds and results into a retransmission. The first retransmission time of popular TCP/IP protocol stacks is usually 6 seconds.

In order to solve this kind of problem, huge size packets need to be chop into smaller pieces. This job can be achieved by modifying the TCP option header – MSS. MSS just appears with SYN, which can be modified during connection setup. After modifying the MSS value, the sender payload size will be bounded by this value. And since there is a congestion window mechanism in TCP protocol, the problem of burst number of packets will be avoided.

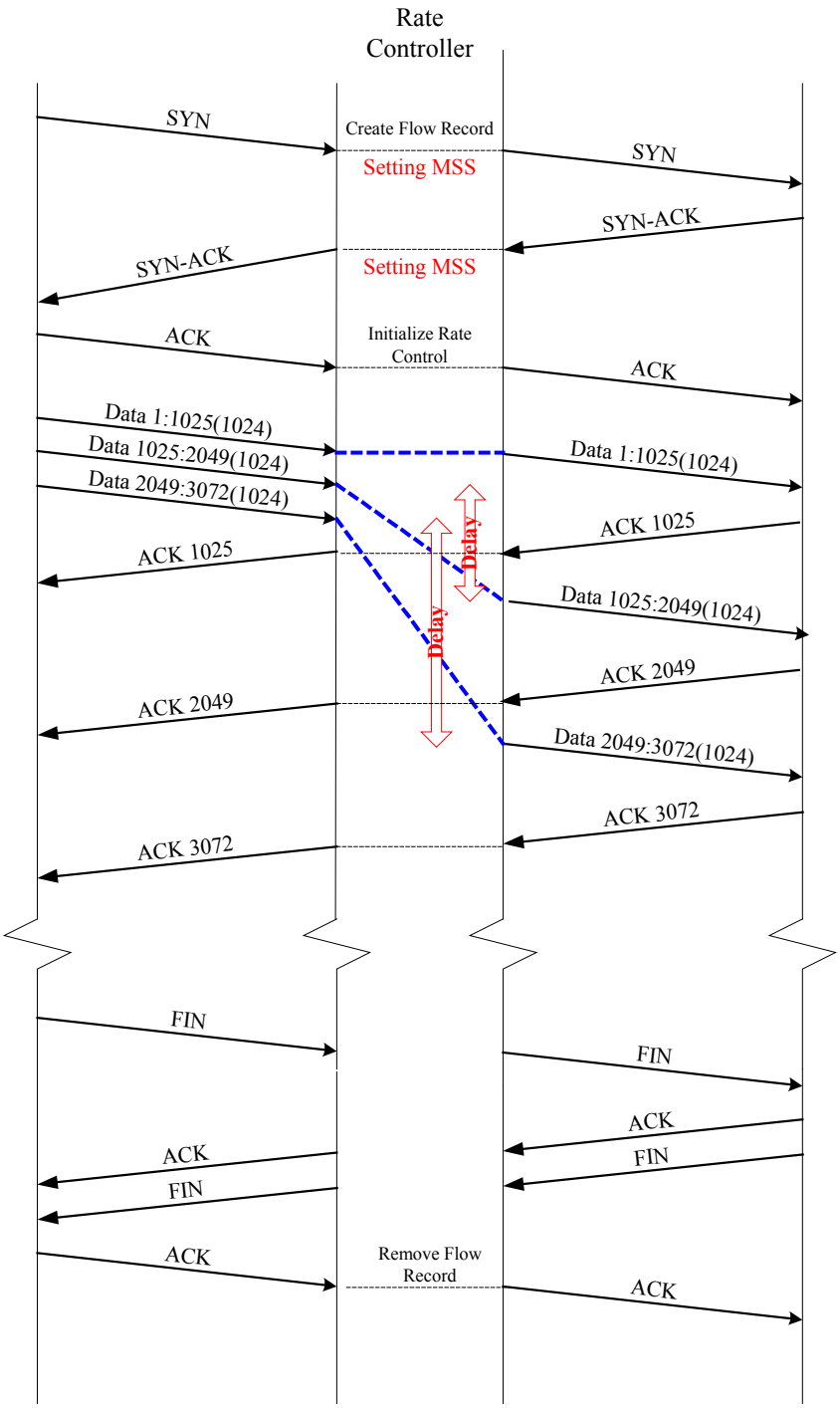

Figure 3.5 Time line of setting MSS

If the payload size is set to a small value, say 10 bytes. There will be a problem that the header size (at least 54 bytes) is always much greater than the payload size, and this means low utilization. There is a disadvantage of setting MSS for active network control. It sacrifices the flexibility of the bandwidth control, because once the MSS is determined during the connection setup phase, it is not allowed to be changed

anymore. Thus, it can only be assigned at connection startup. If a small MSS is determined for a connection at beginning, we will lose the chance to speed up its rate relatively. Because the packet size is small, the utilization of per packet cannot be increased.

4.IMPLEMENTATION

The implementation of the RCS kernel is constructed on Microsoft Embedded NT system with 2 Intel I82559 NICs (Network Interface Card). There are two NICs on the device; one stands for an inbound port and the other stands for an outbound port. The Embedded NT is employed in our development platform due to quickly programming and building the RCS system is possible. This OS platform can fully support all X86-structured hardware and network card and avoid the time waste in hardware compatibility problems.

The spirit of whole system is the timer of kernel. Timers control all the function's flow and depend on those timers to trigger system working. The duration of a timer is tight with performance. A smaller duration will have a more precisely rate control. But, if the duration is too small, the system load will be increased. In our implementation system, 10ms is used as the duration. The network bandwidth is divided into many small time slots (the length is 10ms) and RCS will assign the transmission order of incoming packets according to the rate of each TCP flow.

4.1 Memory Resource Requirement

The resource requirement in memory has two parts; one is in system packet queue and the other is in Time Division Queue.

- Memory usage of system packet queue

  In 100Mbps full-duplex Fast Ethernet, the network bandwidth is 200M bit/sec. If we define the Controlling Time be 1 sec, the queue size is 200M/8 = 25MBytes

- Memory usage of Time Division Queue

  A Time Division Queue has 100 slots and one slot has 100 buckets. If the

Time range of one bucket is 10 ms and double link list entry size is 8 bytes:

In 100 Mbps environment:

- Packet length: 64 bytes; There are 2048 entries per 10 ms

  2048 * 8 * 100 * 100 = 156.25 Mbytes

- Packet length: 1514 bytes; There are 87 entries per 10 ms

  87 * 8 * 100 * 100 = 6.6 Mbytes

It has been experienced in MS Windows that the packet payload size can be set from 10 to 1460. In the worst case, we need about 180MB for RCS. The total system (include keeping OS works) needs at least 212MB RAM. In our implementation, 256MB RAM is used.

4.2 Packet Driven Event

When device receives packets, the "Packet Driven Event" will be invoked. The incoming packets will be processed according to layer-2 information first. System will handle the ARP, ICMP or broadcast packets and forward them. Then the TCP packets will enter "Flow Classification Module". On next step, system will find the packets' limit rate from rule table and count its current rate. After checking the bandwidth, if packets are over-subscribed, the pointers of the packets will be added to Time Division Queue. If not, the packet will be directly sent forward.

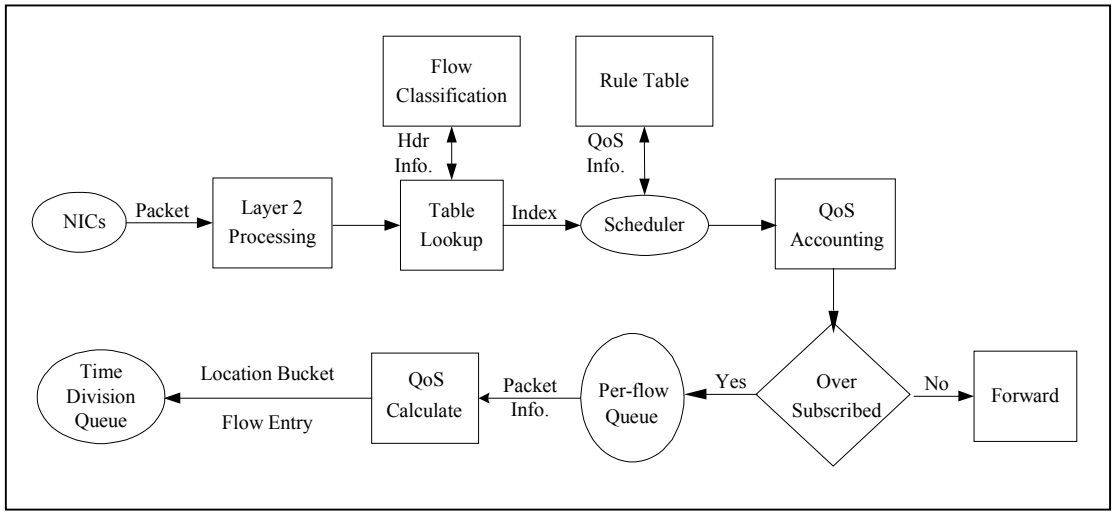

Figure 4.1 Packet driven event flow

4.2 Timer Trigger Event

System timer is set to 10ms. Every time it wakes up, system will check which bucket shall be sent out at this time. All packets link to this selected bucket will be sent out from the system packet queue.

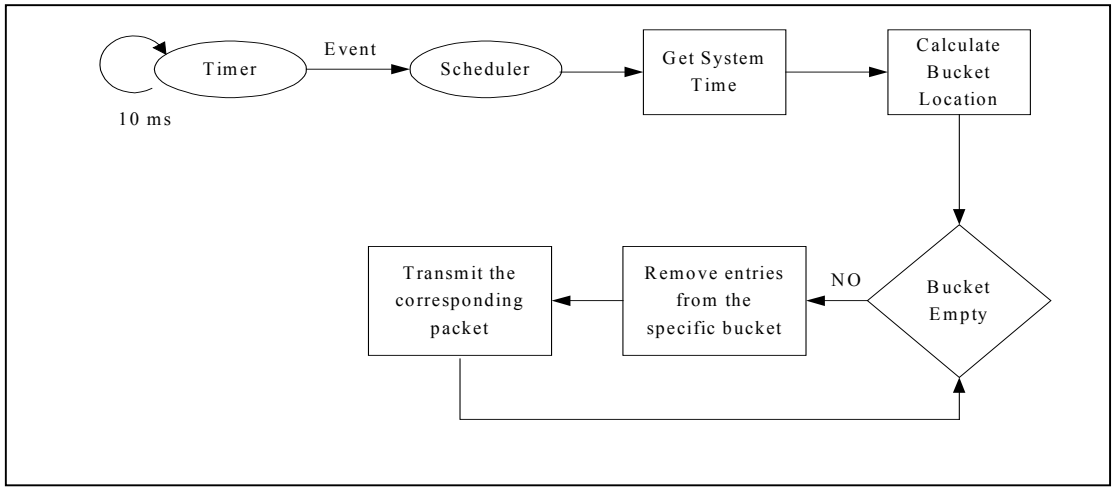

Figure 4.2 Timer trigger event flow

4.4 Bandwidth Reassignment

Bandwidth reassignment is to increase the bandwidth utilization. For example, if we define 400kbps for a TCP flow, but it only needs 200Kbps or its traffic is suspend at some time, RCS can redistribute the remain bandwidth of this flow to other TCP flows.

The difficulty to provide bandwidth reassignment is: how can we predict the rate of this flow at next second? If we have to do many works for predicting the traffic rate, it means that we cannot count how many bandwidth should be reassigned to other TCP flows as soon as possible.

For solving the problem, we define and provide the service of minimal bandwidth guarantee. It is different with committed bandwidth guarantee. This class, minimum rate guarantee, is allowed to transmit its packets when there is more

available bandwidth in the network. Because our goal is to level up the utilization of network bandwidth, system detects the number of remaining packet-time in one time slot instead of counting the remaining bandwidth of some TCP flows at real-time. In this way, we do not need to guess the bandwidth of some TCP flows at next second (Although according the traffic logs of TCP flows, a good predictive value can be provided, the system loading may be heavy). System will check the outgoing packet queue every 10ms or longer. When the number of remaining packet-time in one time slot is not zero, system will start to send packets of “minimum rate guarantee” class as possible, and the packets of “minimum rate guarantee” class will have chance to over the rate limit.

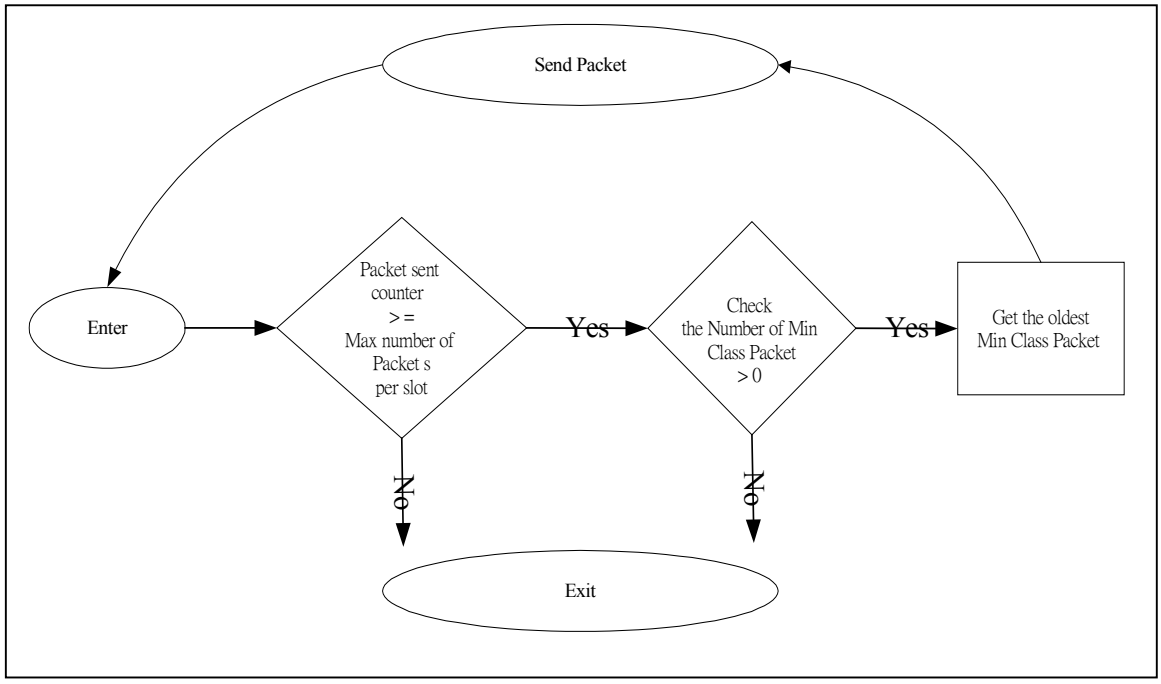

Figure 4.3 Bandwidth reassignment flow

5.SYSTEM BEHAVIOR

In order to evaluate the performance of the proposed Rate Control System, a testing tool, called “Catapult” is developed. Catapult is a Server/Client-based standard network BSD/Socket program. We can run Catapult on one computer as a client, and also start another Catapult on other computer as a server. A client program requests service and send data and a server program accepts connection request from clients and monitor the status of rate changing.

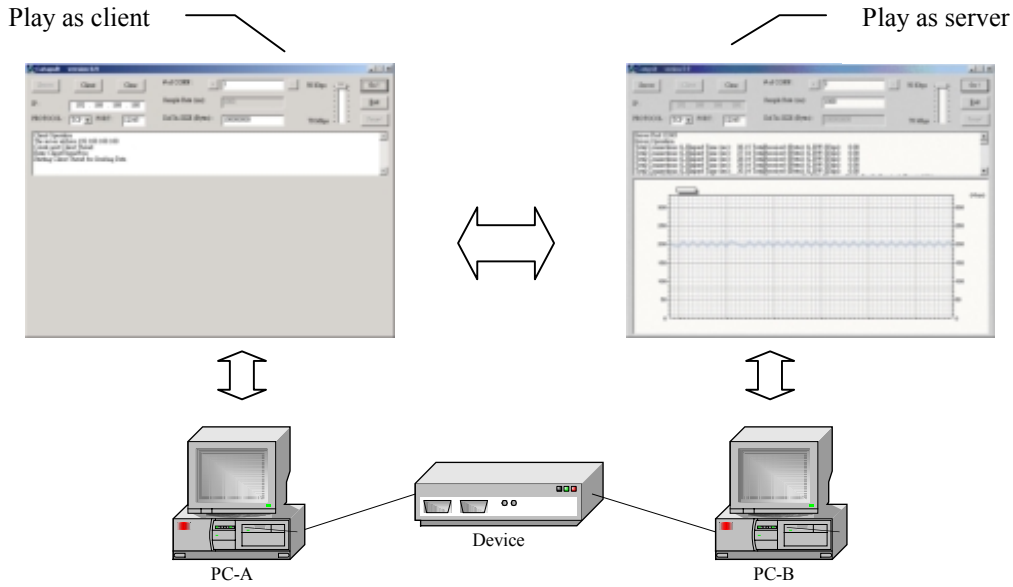

Figure 5.1 Run-time environment of Catapult

Catapult has three parts. The program option lists in top frame, the middle frame is console, and the last part is chart of real-time bandwidth rate. A server Catapult shows the number of established connections and the rate chart as well. Using Catapult, the traffic status can be monitored clearly and help us to understand whether the system behavior is correct as we expected or not.

Catapult has the following characteristics:

- User can define connection’s IP/Port.
- User can define the Total Octets allowed to transmit for one connection.

- Establish multiple connections simultaneously.
- Provide easy configuration and graphical reporting.

5.1 Example of Standard Rate Control in One Connection

We use an example to show the standard rate control. For simplicity, only one connection is shown here, Client->Server. Background traffic (FTP) is generated and the desired rate is defined. The rate is controlled according to the source IP address of the connection. This example shows the result after rate control working and the detailed information about this example is listed below (Figure 5.2).

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	225 Kbps

(Sampling rate: 1000 ms)

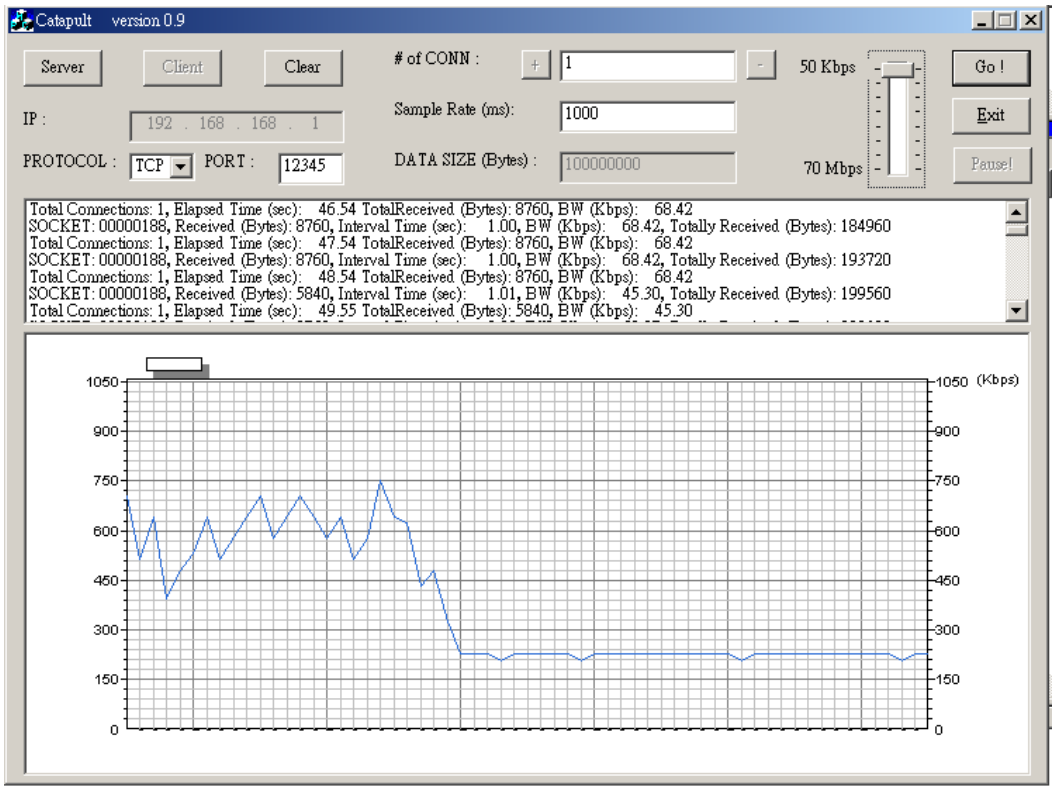

Figure 5.2 Standard rate control of one connection

5.2 Example of Changing Rate Control in One Connection

We use an example to show the changing rate control. Again, for simplicity,

only one connection is shown here, Client->Server. Background traffic (FTP) is generated and the desired rate is defined. The rate is controlled (changed) according to the source IP address of the connection. This example shows the result after rate control working and the detailed information about this example is listed below (Figure 5.3).

Original QoS of one connection:

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	200 Kbps

(Sampling rate: 1000 ms)

New QoS of one connection:

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	400 Kbps

(Sampling rate: 1000 ms)

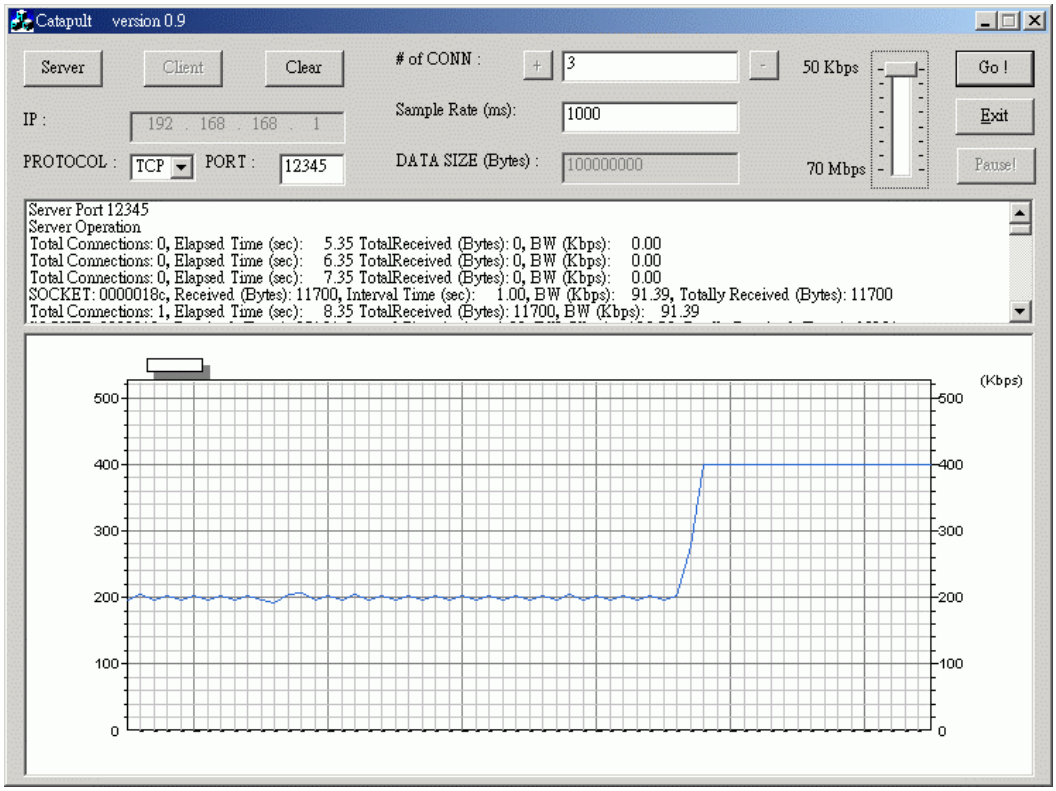

Figure 5.3 Changing rate control on one connection

5.3 Example of Standard Rate Control on Multiple Connections

We also use an example to show the standard rate control with multiple connections. For simplicity, three connections, Client1->Server, Client2->Server and Client3->Server are established. According to the source IP address of each connection, we define the desired rate and control the rate as well. The detailed information about this example is listed below (Figure 5.4).

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	300 Kbps
192.168.168.2	192.168.168.160	TCP	12345	200 Kbps
192.168.168.3	192.168.168.160	TCP	12345	100 Kbps

(Sampling rate: 1000 ms)

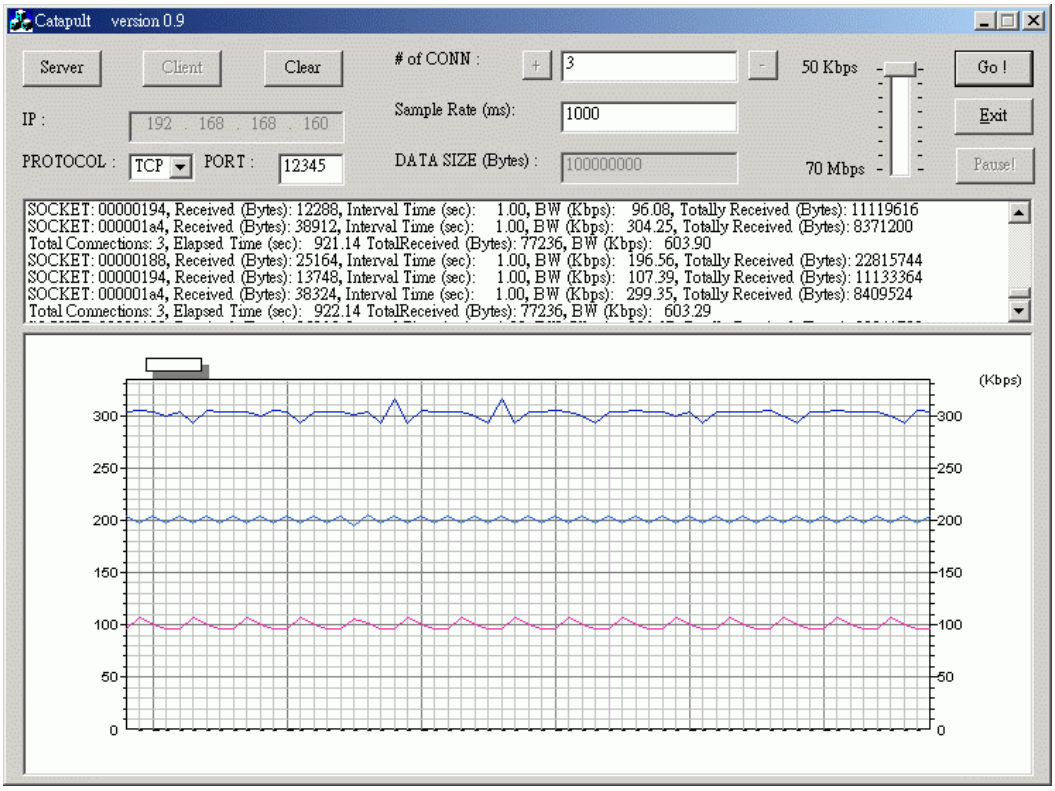

Figure 5.4 Standard rate controls of multiple connections

5.4 Example of Changing Rate Control on Multiple Connections

This is an active network control example. There are three connections, Client1->Server, Client2->Server and Client3->Server, running in the testing

environment. We define and control their rates according to source IP address of the connection. This example shows the active control ability of RCS. The original QoS for these three connections are 300Kbps, 200Kbps, and 100Kbps, respectively. These QoS are updated to 200kbps on the fly. Then these QoS are restored to the original QoS again. The result after rate control working and the detailed information about this example is listed below (Figure 5.5 and Figure 5.6).

Original QoS of connections:

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	300 Kbps
192.168.168.2	192.168.168.160	TCP	12345	200 Kbps
192.168.168.3	192.168.168.160	TCP	12345	100 Kbps

New QoS:

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	200 Kbps
192.168.168.2	192.168.168.160	TCP	12345	200 Kbps
192.168.168.3	192.168.168.160	TCP	12345	200 Kbps

(Sampling rate: 1000 ms)

Figure 5.5 Changing rate control on multiple connections

Restored QoS:

Source IP	Destination IP	Protocol	Port	QoS
192.168.168.1	192.168.168.160	TCP	12345	300 Kbps
192.168.168.2	192.168.168.160	TCP	12345	200 Kbps
192.168.168.3	192.168.168.160	TCP	12345	100 Kbps

(Sampling rate: 1000 ms)

Figure 5.6 Restoring rate control on multiple connections

5.5 Example of Virtual Channel Control on Multiple Connections

To show the ability of controlling the virtual channels on multiple connections, two virtual channels, Client1->Server, and Client2->Server, with channel rates 20Mbps and 10Mbps, respectively, are created. For each virtual channel, 25 connections are established. The rates of these connections are controlled by the RCS according to the source IP address. The result is shown in Figure 5.7, where we can see that allocated bandwidth for each virtual channel is shared fairly among those 25

connections included. For 20Mbps (10Mbps) virtual channel, the shared bandwidth for each involved connection is 800Kbps (400Kbps).

QoS of virtual channels:

Source IP	Destination IP	Protocol	Port	Connections.	QoS
192.168.168.1	192.168.168.160	TCP	12345	25	20Mbps
192.168.168.2	192.168.168.160	TCP	12345	25	10Mbps

(Sampling rate: 3000 ms)

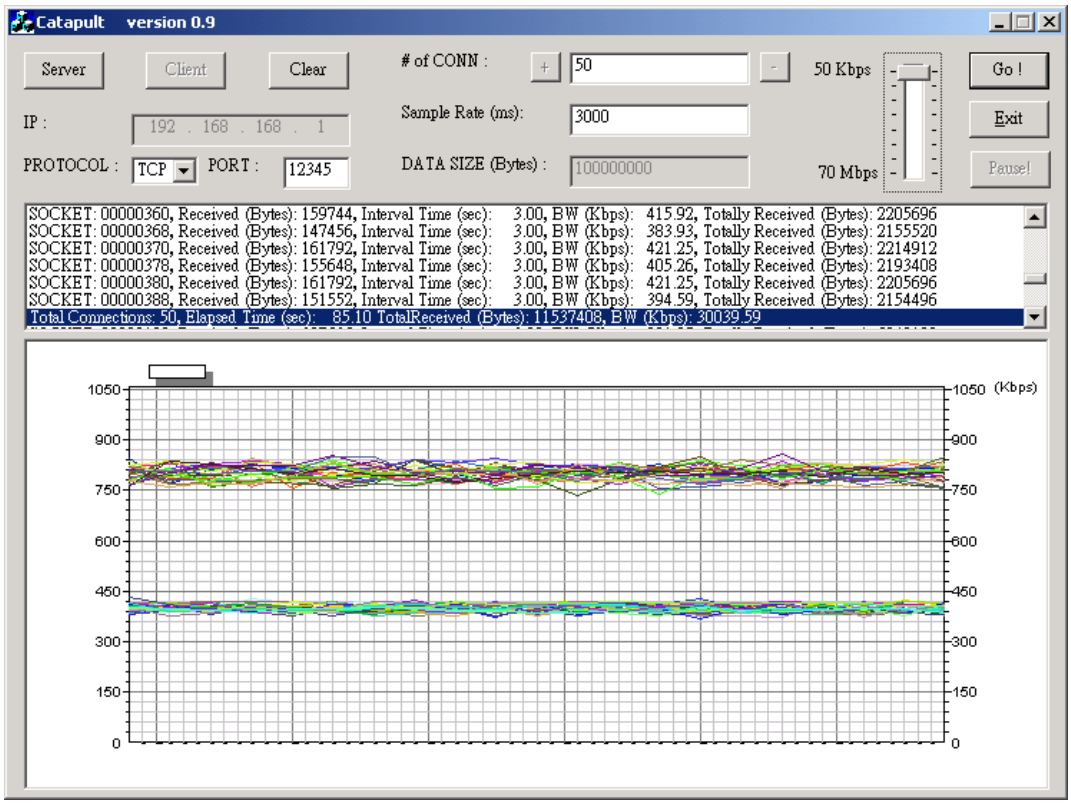

Figure 5.7. Virtual Channel Control on Multiple Connections

6. PERFORMANCE EVALUATION

For performance evaluation, the NetPerf is used as the testing tool. Netperf is a benchmark that can be used to measure various aspects of networking performance. Its primary focus is on bulk data transfer and request/response performance using either TCP or UDP and the Berkeley Sockets interface [5].

6.1 Throughput Performance

NetPerf provides two types of TCP benchmarks, TCP stream performance and TCP request/response performance. We use TCP stream performance function to test the device throughput performance.

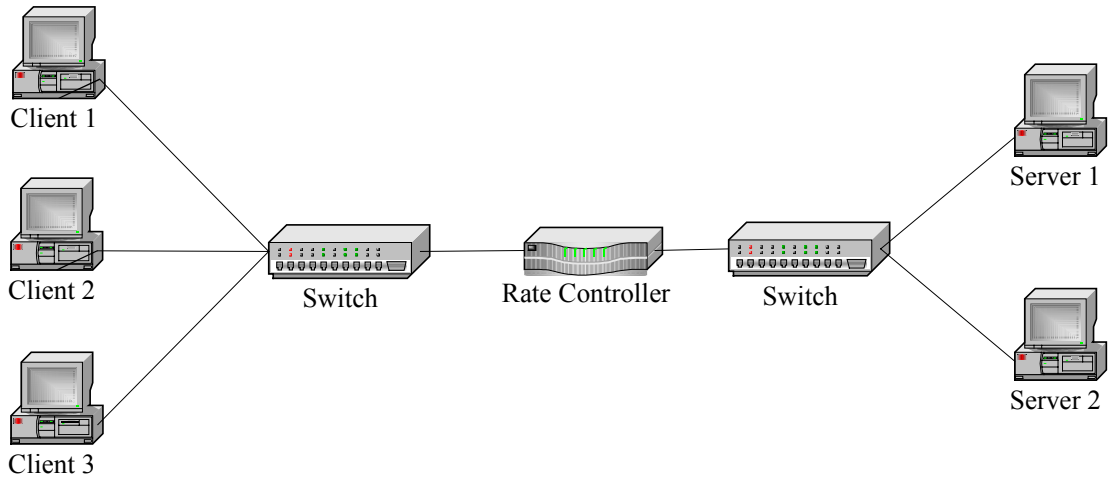

Figure 6.1 The environment of "throughput performance" testing

Bypass mode (no rate controller): (MB)

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	AVG.
S1-C1	17.62	17.64	19.29	17.38	15.87	17.56
S1-C2	19.27	18.03	19.08	17.66	18.40	18.488
S1-C3	27.94	17.35	21.17	22.56	21.19	22.042
S2-C1	14.06	17.89	18.17	14.97	11	15.218
S2-C2	11.55	10.25	13.21	16.73	15.77	13.502
S2-C3	18.45	19.99	13.27	14.23	19.98	17.184

Total	108.89	101.15	104.19	103.53	102.21	103.994

No limit mode (Any to Any): (MB)

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	AVG.
S1-C1	13.25	22.15	16.6	19.23	13.59	16.964
S1-C2	16.29	19.7	16.52	22.81	21.97	19.468
S1-C3	18.66	12.15	19.95	16.21	15.49	16.492
S2-C1	17.29	15.55	12.13	11.87	21.84	15.736
S2-C2	14.18	15.45	18.78	16.83	13.55	15.758
S2-C3	19.61	15.8	16.52	13.44	13.07	15.688
Total	99.28	100.8	100.5	100.39	99.51	100.096

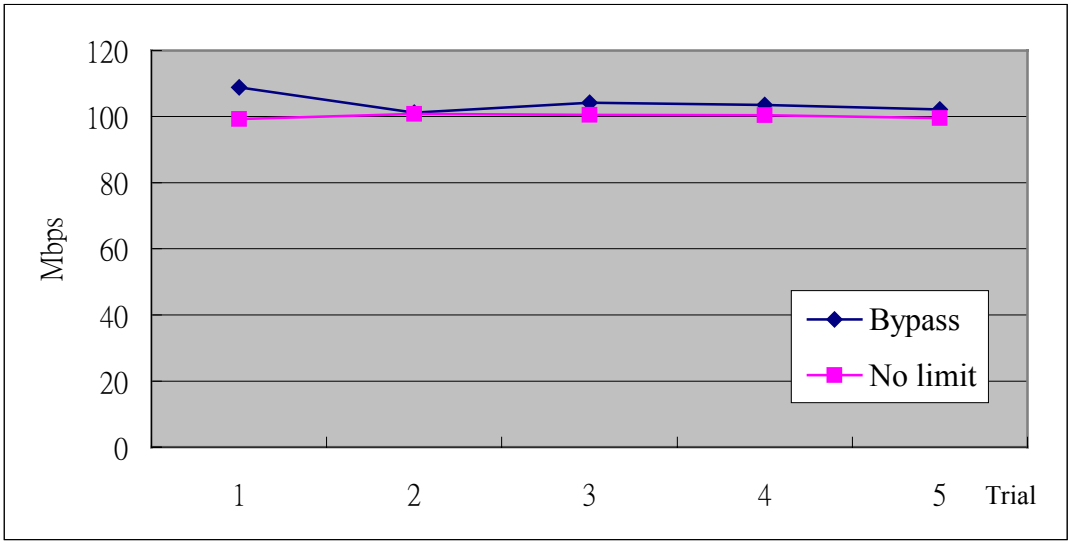

Figure 6.2 The result of “Throughput Performance” testing

7. CONCLUSION

This thesis proposed an efficient rate control system (RCS) over the Ethernet. The RCS is a simple bandwidth control scheme that provides a total solution for "Active TCP Control". This system furnishes the ability of partitioning the network bandwidth and creating virtual channels for TCP flows. Each TCP flow will naturally follow to send the data according to the allocated bandwidth.

Thus, the rate (bandwidth) of TCP flows can be allocated and controlled in a very efficient and accurate way. The software specification of the RCS system is as follows:

1. Manage and control network resources efficiently.
2. Provide guarantee bandwidth for TCP flows.
3. Provide minimal bandwidth guarantee, maximal bandwidth guarantee and committed bandwidth guarantee.
4. Provide bandwidth reassignment function to improve network utilization.

The famous rate control schemes, such as delayed-ACKs scheme or changing window-size scheme, are tightly bounded with the ways of TCP congestion avoidance. Nevertheless, the proposed RCS is independent with TCP congestion managements. This novel feature makes the RCS to be completely compatible with any congestion avoidance protocols, such as CM or ECN. Although the proposed RCS may need a larger memory, but since the memory cost is dropping dramatically, it is a good choice for active TCP control.

8. REFERENCES

[1] Arata Koike, "Active TCP Control by A Network, "*In Proceedings of IEEE GLOBECOM'99*, pp. 1921-1925, Rio de Janeiro, Brazil, December 1999.

[2] David Andersen, Deepak Bansal, Dorothy Curtis, Srinivasan Seshan, and Hari Balakrishnan, "System support for bandwidth management and content adaptation in Internet applications," In *Proceedings of 4th Symposium on Operating Systems Design and Implementation*, pp. 213-226, San Diego, CA, October 2000. USENIX Association.

[3] D. Bansal and H. Balakrishnan, "TCP-friendly Congestion Control for Real-time Streaming Applications," *Technical Report, MIT-LCS-TR-806*, May 2000.

[4] Hari Balakrishnan, Hariharan S. Rahul and Srinivasan Seahan, "An Integrated Congestion Management Architecture for Internet Hosts," *In Proceedings of ACM SIGCOMM*, Cambridge, MA, September 1999.

[5] Information Network Division Hewlett-Packard Company, "NetPerf: A Network Performance Benchmark," http://www.netperf.org/netperf/training/Netperf.html, February 1995.

[6] K.K.Ramakrishnan, and S. Floyd, "A Proposal to add Explicit Congestion Notification (ECN) to IP," *IETF Internet Experimental RFC 2481*, January 1999.

[7] Lampros Kalampoukas, Anujan Varma, and K. K. Ramakrishnan, "Explicit Window Adaptation: A method to enhance TCP performance," *In Proceedings of IEEE INFOCOM '98*, San Francisco, CA, April 1998.

[8] Jamal Hadi Salim and Uvaiz Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks," *IETF Internet Experimental RFC 2884*, July 2000.

[9] Jeffrey D. Carter, "Full Duplex Switched Ethernet (FDX 10BaseT) with Linux,"

http://www2.shore.net/~jeffc/fdse/, September 1999.

[10] Jin Tang, Giacomo Morabito, Ian F. Akyildiz and Marjory Johnson, "RCS: A Rate Control Scheme for Real-Time Traffic in Networks with High Bandwidth-Delay Products and High Bit Error Rates," *In Proceedings of IEEE INFOCOM '2001*, Anchorage, Alaska, April 2001.

[11] Jonas Andren, Magnus Hilding, Darryl Veitch, "Understanding End-to-End Internet Traffic Dynamics," *In Proceedings of IEEE GLOBECOM'98*, Sydney, Australia, November 1998.

[12] Mohit Aron, Peter Druschel, "Soft timers: efficient microsecond software timer support for network processing," *In Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP'99)*, pp. 232-246, Kiawah Island Resort, SC, Dec 1999.

[13] Packeteer Inc., "White papers on TCP rate control," *TCP/IP Bandwidth Management Series*, *http://www.packeteer.com/*

[14] Prasad Bagal, Shivkumar Kalyanaraman, Bob Packer, "Comparative study of RED, ECN and TCP Rate Control," *Technical Report, Dept of ECSE*, RPI, March 1999.

[15] Rich Seifert, "Issues in LAN Switching and Migration from a Shared LAN Environment," *Technical Report, Networks and Communications Consulting*, November 1995.

[16] W. Richard Stevens, TCP/IP Illustrated: The Protocols, Volume 1, Addison Wesley, 1994.

[17] Zhiruo Cao and Ellen W. Zegura, "Utility Max-Min: An Application-Oriented Bandwidth Allocation Scheme," *In Proceedings of IEEE INFOCOM'99*, New York, March 1999.