Dynamic Multi-Path Communication for Video Traffic *

Hao-hua Chu, Klara Nahrstedt
Department of Computer Science
University of Illinois
h-chu3@cs.uiuc.edu, klara@cs.uiuc.edu

Abstract

Video-on-Demand applications emerge on campus-wide networks. Compressed video streams pass switches which do not have any bandwidth reservation mechanism, nor any other mechanism against congestion. The switches, shared among video servers, WWW servers and clients, get easily congested in critical situations, hence the switches drop packets and the quality of end-to-end delivery service degrades. To provide soft bandwidth (QoS) guarantees for video, we designed and tested through simulation a **dynamic end-to-end multi-path protocol** *for video traffic.*

We compared our dynamic multi-path protocol performance with a dynamic single-path protocol and the experiments show significant advantages of the dynamic multi-path communication with respect to provision of soft bandwidth guarantees for video traffic. A disadvantage of our approach is the reordering overhead. However, the use of the higher level frame information helps to simplify reordering and the overhead is acceptable.

1 Introduction

Almost all VOD applications require certain level of network Quality of Service (QoS). Given that the network resources are shared, congestion in the network will occur and they can cause unacceptable performance degradation for the VOD applications due to *unavailability of bandwidth.* The current solutions use either a connection flow control and feedback mechanism (e.g., [4]), or a reservation mechanism for a connection (e.g., RSVP, Tenet RCAP, ST-II [8].

The former approach uses a flow control mechanism at the source to regulate sending video data over the network via a single path, and if congestion still occurs in the network, feedback is sent to decrease the sending rate at the source. This approach provides *best effort service* when considering the QoS parameter - *bandwidth availability.* Although, widely used, this is not an optimal solution for continuous video traffic. Having such an underlying network service means that applications must be adaptive and must recover from large changes in bandwidth availability. This also means more complexity for the applications.

The latter approach admits and reserves/allocates network resources for video traffic along the single path if all switches on this path have enough resources. This approach provides *deterministic guaranteed services* with hard bandwidth guarantees. This is the best solution for continuous media; however, deterministic guarantees come at a certain cost because:

- All switches must support the same distributed reservation protocol.
- The reservation protocols consider the worst-case scenario even though it may rarely occur. This generally results in an underutilization of the resources [4].
- Most of reservation protocols require fixed-routing (e.g., ST-II). The route is decided at its reservation phase and it cannot be changed. However, the network traffic is dynamic and it may change to a state where this fixed path can cause resource fragmentation and poor network load balance. This is called the *network availability* problem described in [6]. An example in Figure 1 illustrates this problem. The available bandwidth is fragmented into 50 Mbps on the upper path and 25 Mbps on the lower path. The network has a combined 75 Mbps of available bandwidth but it cannot accommodate a new 60 Mbps connection without moving connection A to the upper path.
- The VOD application must submit its flow specification and its desired QoS during the reservation phase and it must behave according to its flow specification during the transmission phase. This is the *application flexibility* problem described in

*This work is supported by **National Science Foundation** Career Grant, under contract: NSFCCR96-23867

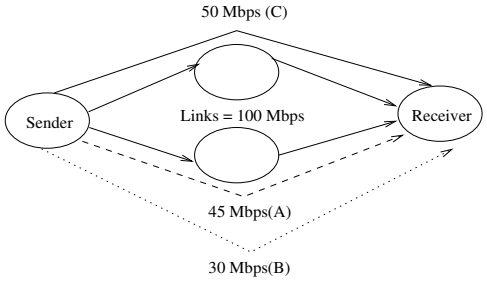


Figure 1: Example of Network Availability Problem. The total bandwidth capacity of the upper path is 50 Mbps, the lower path has the total bandwidth of 100 Mbps. Already two connections A, requiring 45 Mbps, and B, requiring 30 Mbps, are reserved on the lower path. Therefore, the remaining available bandwidth is 50 Mbps on the upper path and 25 Mbps on the lower path.

[6]. The VOD application may want to provide support for the users to adjust their desirable QoS (e.g. frame rate and resolution) interactively or according to the current network traffic. The same example in Figure 1 can illustrate this problem. Suppose that connection B wants to increase its QoS to 60 Mbps. Again the network cannot accommodate this change without moving connection A to the upper path.

In this paper, we describe a *dynamic multi-path approach*, which is an alternative approach in-between the best effort and the deterministic guaranteed services for VOD applications. Our approach addresses the availability and flexibility problems described above, and since our approach is not reservation-based, it offers only *soft bandwidth (QoS) guarantees*[1] instead of *hard guarantees*.

The multi-path approach exploits the ability of a connection to utilize bandwidth of potentially multiple unique paths from the sender to the receiver for concurrent data transmission. The protocol groups fragmented bandwidth on the multiple paths into one logical connection for the VOD applications. There is no reservation during the connection establishment phase. Hence, transmitted data may use a path where congestion occurs. The congestion (degradation of bandwidth availability) is detected by the receiver which sends a feedback information back to the sender with specific information about the congested path. Based on the feedback information, the sender provides load balancing to re-distribute the load from the congested path to other free paths. During the re-balancing process of available bandwidth, there may exist a time interval during which the required bandwidth availability may be briefly violated; therefore our protocol provides soft guarantees only. However, we consider an application class that can tolerate soft guarantees.

The outline of the paper is as follows: Section 2 discusses related work with detailed presentation of the tradeoffs between different approaches to solve the congestion problem and violations of bandwidth availability for video traffic. Section 3 presents the design of our protocol architecture and services. Section 4 describes the implementation of our simulator and tested protocol. Section 5 presents results and evaluation of our simulation. Summary in Section 6 concludes our paper.

[1]Soft bandwidth guarantee means that there exists a time interval when guarantees are violated, however violation is bounded by a specified value.

2 Related Work

The network availability and application flexibility problems due to network congestion and dynamic requirements are addressed in several protocols as follows:

- *Single-Path Protocol with Network Feedback*

  In the adaptive real-time video transport protocol [4] (no reservation exists), the congestion is resolved by sending feedback information from the congested switch to the source and the source reacts to the feedback by decreasing its sending rate. This protocol is an example that in the absence of resource reservation, this type of feedback control scheme can provide a graceful degradation in video quality (packet loss rate) during periods of congestion.

- *Single-Path Protocol with Dynamic Multi-Path Routing*

  Another possibility to work around the congestion problems (no reservation mechanism exists), and provide the network bandwidth availability in a protocol is to apply a dynamic multi-path routing algorithm during the connection setup phase when choosing the proper path [1]. Routing decisions can significantly affect network congestion and load balance. The dynamic multi-path algorithm considers several potential paths for the selection and classifies each of the candidates according to their utilization or load. The path selection is based on a combined heuristics of the shortest path and path utilization. The simulation results in [1] show that the protocol with the dynamic multi-path routing algorithm outperforms other

routing algorithms, resulting in a lower number of dropped packets and a more balanced network.

- *Single-Path Protocol with Dynamic Re-routing*

  A third possibility to solve the network bandwidth availability problem and the application flexibility problem is to apply a reservation mechanism during the connection setup and when changes on bandwidth availability occur during the transmission phase, dynamic re-routing is applied. The Tenet RCAP protocol carries this idea on their real time channels [6] with the Dynamic Connection Management (DCM). The DCM decides if a real-time channel should be rerouted to an alternative route, called the *shadow channel*. Recall the example in Figure 1. When a new 60 Mbps connection request arrives, the DCM will reroute connection A from the lower path to the upper path so that the lower path will have enough bandwidth to support the new connection. The same action is taken when connection B wants to change its QoS to 60Mbps. A channel administration algorithm is responsible for the establishment and the reservation of the alternative route which has the identical QoS requirements as the primary route. A transition algorithm replaces the primary channel with the alternative channel. The DCM rerouting is done transparently to the VOD applications and without service disruption and QoS violations. In addressing the network load balance (or the availability) problem, the Tenet protocol allows dynamic rerouting of its channels during its transmission phase, whereas [1] multi-path routing algorithm operates only during the connection setup phase.

Our *multi-path protocol with receiver feedback* takes the solution for network bandwidth availability and flexibility one more step further than the previous protocols as described in the Section 3.

3 Protocol Architecture Design

The multi-path protocol with receiver feedback is an adaptive connection-oriented end-to-end protocol designed to support video traffic on top of packet(cell)-switched high-speed networks. The basic principle of our approach is based the multi-path connection (see Figure 2). The *multi-path connection* contains multiple paths from a sender to a receiver. Each path is likely to be unique, meaning that it goes through different switches and links in the network. One important distinction between our multi-path approach and the others is that a packet is not duplicated among the paths. One packet is transmitted only through one chosen path and multiple paths work simultaneously to transmit packets from the sender to the receiver.

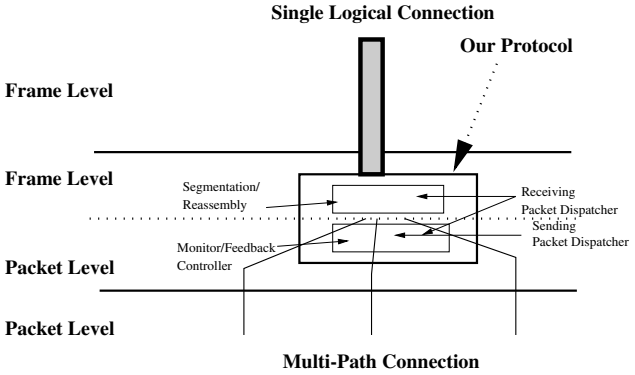


Figure 2: Multiplexing/De-multiplexing of Multiple Paths onto One Logical Connection

By the combination of the fragmented bandwidth with multiple paths for one logical connection, the fragmentation loss in the availability problem is completely overcome. Consider again the example in Figure 1. The new 60 Mbps connection can be accommodated with a split of 40 Mbps on the upper path and 20 Mbps on the lower path. Our protocol can accommodate a new 75 Mbps connection with a 50-25 Mbps split, whereas the Tenet protocol can not. The flexibility problem can be solved in a similar fashion. If connection B wants to increase its QoS to 60 Mbps, it can open a new upper path with 30 Mbps.

Since our protocol does not assume that the underlying network provides reservation during connection setup, network congestion may occur and the network bandwidth availability changes. We handle the congestion with receiver feedback to adapt the sending rate on the congested path(s). The difference to single-path protocol with network feedback is that the adaptation (decrease in sending rate) on the congested path is compromised by adaptation (increase in sending rate) on other paths. Hence, to the higher protocol layer our protocol maintains its overall requested bandwidth (QoS) requirement. Since our approach assumes that the underlying packet level supports the establishment of unique paths when requested, two joint establishment requests will provide two paths (e.g., virtual circuits) with minimal resource overlapping on their routes.

3.1 Protocol Overview

The protocol architecture consists of several major components as shown in Figure 3. At the source, the *packet dispatcher* sends and controls the concurrent transmission of packets through its multiple paths using the load distribution information of multiple paths. The *QoS monitor* receives the feedback information from the destination which carries the congestion level of every path of a connection. The QoS monitor then uses the feedbacks to adjust the load distribution information and the degree of the multiple paths employing load balancing algorithm and routing algorithm to find a set of multiple paths with as few overlaps as possible.

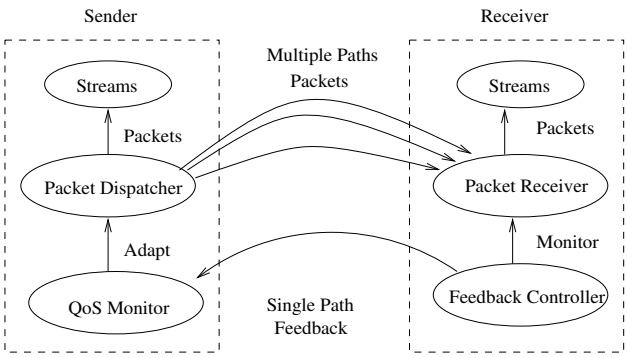


Figure 3: Multi-Path Protocol with Receiver Feedback. Transmission.

At the destination, the *packet receiver* receives the the concurrent transmission of packets on multiple paths. Since multiple paths may have varying end-to-end delay, the receiving dispatcher needs to *reorder* the packets in their sending sequence. The arrival of packets is monitored with respect to the loss of packet by the *feedback controller*. When a specified value of lost packets is detected over certain period of time[2] (loss rate QoS required value), the feedback controller signals the source with feedback information about the loss rate. The QoS monitor receives this information and reacts adequately.

Our protocol uses two adaptive mechanisms to achieve the soft QoS guarantees. The first one is the *load distribution* among multiple paths to avoid network congestion. When congestion occurs on one or more paths, our protocol will shift the load from the congested paths to the other paths. It will result in a better network load balance, therefore higher *network availability*. The second one is the dynamic *adjustment of the degree (or number) of multiple paths* to suit the application's changing bandwidth usage. Higher/lower degree of multiple paths means higher/lower possible available bandwidth. Because the VOD application can change their QoS parameters and flow specification, the *application flexibility* is increased.

In the next subsection, we describe in detail major components of our protocol connection/path management (see Figure 3).

3.2 Connection Setup and Routing

In a packet-switched VC (virtual circuit) network, the multiple paths are established per connection during its setup phase. The connection request (type 1) is sent to a *centralized routing server* that computes, online, the multiple routes based on a simple *dynamic state-dependent multi-path routing* algorithm. The routing server also accepts a (type 2) routing request from an existing connection that is in need of additional bandwidth and path.

Upon receiving the routing request, the routing server will probe each switch for the instantaneous available bandwidth on its incoming and outgoing links. We assume that the switches can monitor the traffic loads on its links. For each link, the switch computes the $available\_bandwidth = link\_capacity - link\_load$. Then a weighted directed graph is constructed; the weight on the edge corresponds to the available bandwidth of the link. We have designed a modified bottleneck Dijkstra algorithm and apply it to the weighted graph. There are 2 modifications: (1) the weight of the path is defined as the minimal weight of its edges, or the available bandwidth of the bottleneck link, (2) instead of finding the shortest path, we find the path with the maximum weight, or the highest available bandwidth.

The goal of our routing algorithm is to find a set of paths with the minimal number of shared links. Large number of shared links in the multi-path connection makes it more vulnerable to congestion when congestion hits the shared links. Shared links also lessen the ability of the load distribution mechanism to balance the traffic among the physical links in the network. For example, a multi-path connection with all paths going through identical links is equivalent to a single-path

[2] We are not interested in using receiver feedback to resolve short term network congestion. As pointed out in [4] the receiver feedback is not useful for short term congestion. By the time a receiver feedback arrives, the congestion might have been resolved, i.e., the feedback information at that point is obsolete. However, for network congestion of longer duration (longer than the round trip delay between source and destination) and of less frequent occurrence (mostly once a second or even larger time intervals), the receiver feedback is a valid mechanism to control bandwidth availability for video traffic. This is currently done at the application level with MPEG-compressed video over VDP/UDP/IP [2] or over RTP/TCP/IP. In our protocol we provide this feedback mechanism at a much lower level of the protocol stack.

connection, and the load distribution mechanism cannot shift the load outside the congested links. Note that our simple algorithm is not the focus of this paper. Its design may not be optimal or competitive in many aspects [7].

As mentioned above, the routing server accepts 2 types of routing request. The type 1 request contains the sender/receiver pairs, its bandwidth QoS (*req_bandwidth*), and an initial degree of multi-path (*init_deg*). Our algorithm finds a minimum of *init_deg* number of paths for this new connection. The weight of each path, *weight*(*path*) is also computed. If the total available bandwidth from the *init_deg* number of paths does not satisfy the *req_bandwidth*, it will obtain more paths. To minimize shared links among the multiple paths of the same connection, our algorithm uses a penalty scheme. The *weight*(*link*) on a new path is reduce by the amount of the *weight*(*path*). This penalty scheme ensures that successive paths will avoid going through the shared links unless the shared links have very high available bandwidth. Figure 4 shows an example of our algorithm with a type 1 request and *init_deg* = 3. The type 2 request contains the sender/receiver pairs and its additional bandwidth QoS. This algorithm is applied in a similar fashion as for a type 1 request. Similar algorithms are described in [6].

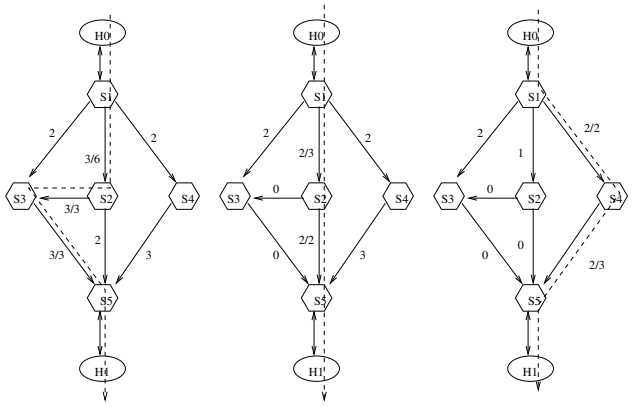


Figure 4: This is a sample network that uses the multi-path routing algorithm to find unique paths. The dotted lines are the paths computes. The value on the links indicate their available bandwidth.

The computed paths and their available bandwidth are returned to the requesting connection for initialization or update of its path table. The path table is described in the next section.

3.3 Packet Dispatcher and QoS Monitor

The *packet dispatcher* resides between the frame layer and the packet layer. At the sender side, after frames are segmented into network packets, the packets are input into the packet dispatcher. For every packet that comes through, the dispatcher assigns a suitable path before sending it into the network. The suitable paths are stored in a path table, and each path is assigned a weight value. The weight value is normalized to a number between 0 and 1 and it indicates the probability of packets going through this path. The sum of all path weights is 1. The weights are adjusted by the QoS monitor.

The feedback information, coming from the receiver, is processed by the *QoS Monitor* at the sender side. In a typical flow control scheme, the feedbacks are used to adapt the sending rate on the source. If the protocol indicates a large number of dropped packets, the sender decreases its sending rate; and hence decreases the total bandwidth availability. If the feedback indicates no dropped packets, the sender increases its sending rate. In our approach, the feedback information is not used to decrease the total bandwidth availability. Instead QoS monitor uses the feedback information as input to the two adaptive mechanisms, (1) load distribution and (2) adjustment of the degree of multi-path.

3.3.1 Load distribution

The QoS monitor balances the load among the multiple paths so that a minimum number of packets are lost. It is stated, without proof, that this minimum occurs when the lost percentage on each path is equal. Load distribution service is performed by adjusting the path weights. The path weight is the probability of packets being sent through this path. By decreasing the weight of the congested path and increasing the weight of the free path, the dispatcher does load balancing by sending fewer packets into the congested path and more packets into the free path. The goal of the load distribution is to avoid network congestion and to adapt to the bandwidth availability changes measured through loss rate performance metric. This service is done transparently to the VOD applications.

A *weight adjustment scheme* to reach this minimum state is briefly discussed: first, an average loss rate percentage is computed. This is our specified acceptable QoS (bandwidth availability) value. Second, for all paths with a loss rate percentage above the acceptable (e.g., average) value, the path weights are decremented by a small constant amount $0.01 \sim 0.001$. The decreased weight value is then distributed evenly to paths with loss rate percentage below the specified QoS

value (e.g., average). The small constant ensures that the adjustment is a gradual process.

An example in Figure 5 illustrates the weight adjustment scheme. Let us assume that the logical connection's bandwidth request is 50 Mbps. It is distributed evenly among two paths, the upper and lower path, hence the *weight vector* is (upper_path=0.5, lower_path=0.5). Now a congestion occurs on the upper path, and the upper link bandwidth decreases to only 20 Mbps. This causes dropped packets on the upper path. As a result of the feedback and load balancing, the weight on the upper path will decrease until weight vector becomes (0.4, 0.6) or the load splits to (20, 30) Mbps. The system is again in a stable state.

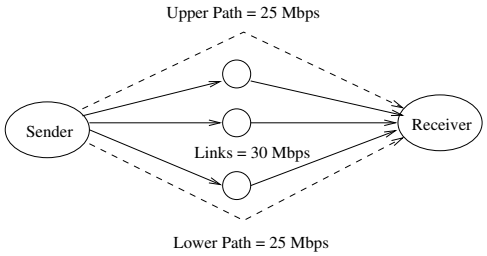


Figure 5: Example for the Weight Adjustment Scheme

A more interesting result is that the load distribution done by each multi-path connection on its multiple paths can produce better load balancing for the entire network. Consider the example in Figure 6. There are 3 connections, each connection has assigned two paths. Let us assume that a congestion occurs on the switch $S2$. The load distribution mechanism avoids the congested switch as follows:

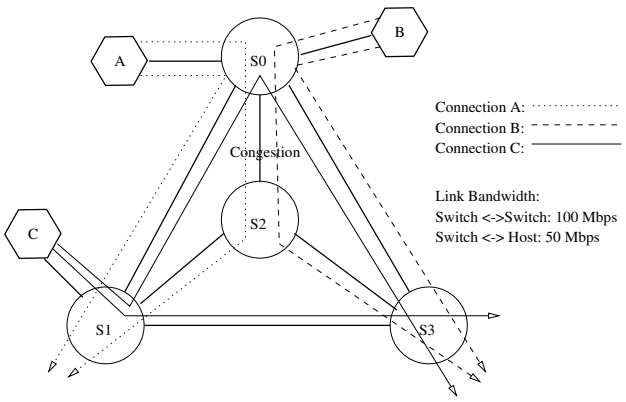


Figure 6: Inter-domain Load Balancing

- Connection A is affected, and it detects congestion on its right path $S0 \rightarrow S2 \rightarrow S1$. Connection A responds by shifting part of its load from its right path to its left path $S0 \rightarrow S1$. If the available bandwidth on its left path cannot accommodate the shifted load, the left path will become congested[3].

- Even after resolving bandwidth availability problem for A, the congested switch $S2$ may affect connection B. It detects congestion on its left path $S0 \rightarrow S2 \rightarrow S3$, which shares the link $S0 \rightarrow S2$ with the right path of connection A. In response, connection B will shift its load from its left path to its right path $S0 \rightarrow S3$.

- Connection C may be still affected. It will detect congestion on its upper path $S1 \rightarrow S2 \rightarrow S3$, which shares the link $S2 \rightarrow S3$ with the left path of connection B. In response, connection C will shift its load from its upper path to its lower path $S1 \rightarrow S3$.

In the multi-path protocol, we can think of the network resources broken down into many small domains of the multiple paths. Each connection is the master in its domain of multiple paths, i.e., each connection is responsible for the load balance in its own domain. This is called *intra-domain load balance*. Due to the uniqueness of each path in the routing algorithm as discussed in 3.2, domains are likely to share links and paths with other domains. When a domain overlaps with another domain, the effect of load balance is propagated across domains. This is called *inter-domain load balance*. With a lot of such inter-domain load balance relations, intra-domain load balance will propagate its effects throughout the entire network.

As shown in the previous example, at first, the intra-domain load balance has an effect on the domain of connection A, link $S0 \rightarrow S1$, and $S0 \rightarrow S2 \rightarrow S1$. Then the inter-domain load balance relations between domain $[A,B]$ and between domain $[B,C]$ propagate the load balancing throughout the entire network.

The higher the number of inter-domain relations, the faster a intra-domain load balance propagates its effect throughout the network. A higher number of paths per connection results in higher number of inter-domain relations. In our simulation results shown later, a high number of inter-domain relations is a double edged sword. It causes very rapid propagation in the network load balancing. But at the same time, the network may become unstable for a certain period of time (wave behavior) until the load is balanced.

[3] Degradation of bandwidth availability occurs, which means that the multi-path approach case falls back into the single path approach and the only solution to resolve congestion is to decrease the sending rate. This also shows that the single path protocol is a special case of our protocol.

3.3.2 Adjustment of the degree of multi-path

In some situations, all the existing paths for a connection can be congested. The QoS monitor finds that the overall multi-path performance does not meet the QoS requirements and load distribution does not improve QoS. In this situation, the connection can increase the number of paths in the hope that it will be able to find a new congestion-free path. Consider the example in Figure 5. Let us assume that congestion occurs on both the upper and lower paths, and bandwidth decreases to only 20 Mbps. Packets are lost on both paths. The QoS monitor will open the additional new path (in the middle).

The reverse situation can occur as well. If no paths are experiencing congestion, the connection can close a few paths. Closing paths will free the network resources, as the closed channels can be used by other connections. Fewer paths also mean less complexity and table-keeping.

The appropriate number of paths per connection is an adaptive process depending on the congestion level. The opening and closing of paths does not interfere with the continuity of data transmitted from the sender to the receiver and the probing for new paths can happen concurrently with the data transmission on the current paths.

In some unfortunate situations, the whole network is congested and every possible path is congested. In this case, our protocol falls back to the single-path protocol policy, which means decreasing the sending rate.

3.4 Packet Receiver and Feedback Controller

In a typical end-to-end feedback protocol, the receiver sends feedback information through a feedback channel to the sender to reflect the congestion level on the path. The decision when to send feedback is commonly based on the number of dropped packets over a fixed period of time. Our feedback model is similar. The only exception is that the packet receiver gets packets from multiple paths in our protocol, and the *feedback controller* needs to detect losses to provide feedback for each path.

Two mechanisms are needed: (1) loss detection, and (2) reordering.

3.4.1 Reordering and Loss Detection

Due to the different end-to-end delay in the multiple paths, packets sent in order by the dispatcher on the sender can arrive out of order at the receiver side. As a result, the receiver needs a large buffer to store the incoming packets and then it sorts the packets in the buffer according to their sending order before reassembling them into a frame. We call it the *re-ordering problem*. The receiver may need a large buffer if the end-to-end delays among the multiple paths cannot be bounded to a small number. This would also lower the throughput which is not feasible for high-speed networks.

Our solution is to dispatch packets within the same frame on the same path, so they will arrive in their sending order as in the case of a single-path. The reordering problem is reduced to re-ordering the frames. Buffering is still needed to sort the frames, however this is done at the frame level where we assume that content (semantic) information can be used to speed up the reordering of frames. This solution implies that at the packet level the implementation becomes relatively easy and inexpensive. The packet loss detection can also be done during the reassembling of packets into a frame.

4 Implementation of the Multi-Path Protocol

We implement the multi-path protocol within our simple network simulator. The simulator is modeled according to an ATM network as shown in Figure 7.

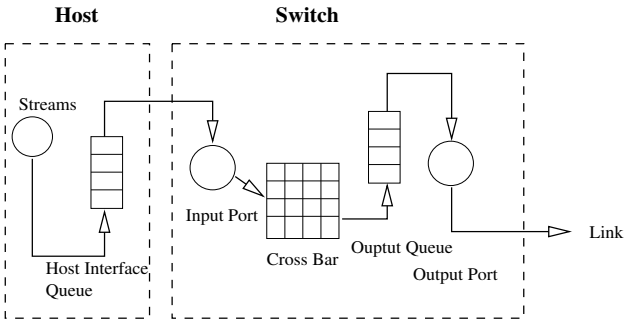


Figure 7: Network Simulation Model

The network consists of switches, hosts, and links. One host can connect to one switch only. Every host contains a host interface queue. Packets (Cells) are queued in the host interface queue before sending them to the network. All connections are unidirectional. A switch can connect to multiple hosts or switches. The switch is based on the output-queuing model. A cell enters a switch through an input port. All input ports are connected to a switching fabric, e.g. a cross bar, where cells are switched to the output queues. Each output link has its corresponding queue where cells are queued

up before transmitted over the link. Every queue has a finite size of 1000 cells. When the queue reaches its maximum length, cells are dropped based on their priority[4]. The feedback cells have higher priority than data cells. Hence, data cells are dropped first before any feedback cells can be dropped.

To simplify our simulation, we do not consider contention in the cross bar, only on the output queues, which translates into a change of bandwidth availability.

The simulator reads a user input network configuration file to initialize the simulation environment. The user specifies the following information in the file: (1) hosts, (2) switches, (3) interconnecting links with parameters for latency, bandwidth, and the maximum queue size, and (4) streams. The streams can be CBR, VBR, defined with burst parameters, or a name of a video trace file.

In the current version, the number of paths per connection is compiled statically inside the code. The simulator runs on a SUN sparc 10 machine under Solaris. It was written in C++ language.

5 Experiments, Results and Evaluation

We evaluate the performance of our protocol by running and comparing the single-path protocol and the multi-path protocol in our simulation. The video traffic that is fed into this simulation consists of motion JPEG streams with characteristics: 640x480 pixels and 30 frames per second. The compressed video stream bandwidth is approximately 4 Mbps.

5.1 Experiment Description

Our experiments test two networks topologies. The *first network*, shown in Figure 8, consists of fours switches configured into a ring. The switch-switch links run at 6 Mbps (available bandwidth), and the switch-host links at 12 Mbps (available bandwidth). The video traffic flows between hosts $H0$ and $H1$ attached to the switches $S0$ and $S3$. Using this topology we experiment with two video streams. Both streams flow from $H0$ to $H1$.

The *second network*, shown in Figure 9, consists of four switches configured into a full-connected graph. In this architecture, we consider three hosts $H0, H1, H2$ each connected to a switch $S0, S1, S3$. The host-switch links run at 12 Mbps (available bandwidth), the outer switch-switch links at 3 Mbps, and the inner switch-switch links at 4 Mbps (available bandwidth). The second network topology is tested with 6 video streams concurrently. Every host has two streams that go to the other hosts.

[4] Our switch includes *simple priority mechanism* which differs between cells carrying data and cells carrying control information such as feedback.

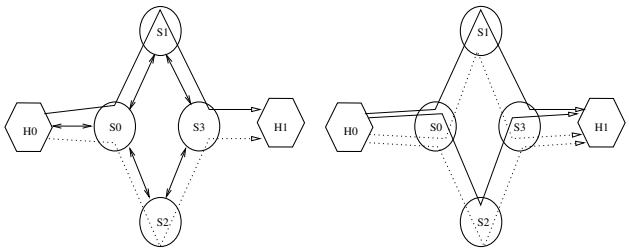
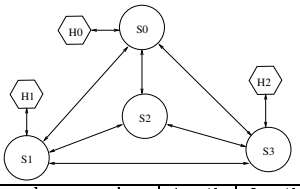


Figure 8: Ring Network Topology. The left figure shows traffic flow (two streams) using a single-path protocol. The right figure shows traffic flow (two streams) using the two-path protocol. The solid lines represent the single/multiple paths used by the first stream. The dotted lines represent the single/multiple paths used by the second stream.



stream	sender → receiver	1-path	2-path	
1	H0 → H1	0341	0341	03641
2	H0 → H2	0352	0352	03652
3	H1 → H0	1430	1430	14630
4	H1 → H2	1452	1452	14652
5	H2 → H1	2530	2530	25630
6	H2 → H3	2541	2541	25641

Figure 9: Fully-connected Network Topology (top). Path Allocation in our Fully-connected Network Topology (bottom). The path is specified: (Source, Switch,..,Switch, Destination). For example, for stream 1 the 1-path route is 0341 which means H0-S3-S4-H1.

The overview of the path distribution for the 6 streams in our second network for 1-path, 2-path and 3-path approaches is shown in Figure 9.

5.2 Performance Metric

To see how well our protocol reacts to congestion (the availability problem), the simulation generates a congestion by randomly picking a switch-switch link in the network and decreasing its capacity. The amount decreased is significant enough so that both intra-domain and inter-domain load balancing are required. For the ring network, the capacity of the congested switch-

switch link is reduced by half to 6 Mbps[5]. For the fully-connected network, the capacity of the congested link is reduced by 1 Mbps for the switch-switch links. A congestion is generated at every 15 seconds interval.

To test how our protocol reacts to changes in application QoS (the flexibility problem) in addition to congestion[6], the simulation randomly picks a stream and increases its QoS from 30 frames/sec (4 Mbps) to 40 frames/sec (5.3 Mbps). The simulation models both a congested link and a QoS-changing stream.

The metric information which is gathered to evaluate the protocol during the simulation is the *packet loss number*. In our simulation, there is a statistic server which collects and outputs statistical metric information into a profile periodically.

5.3 Results and Evaluation

5.3.1 Ring Network Topology

The graphs in Figure 10 show the results from the simulation run for the ring network with congested links. The y axis presents the number of dropped cells (packets), the x axis shows the simulation time. The graphs in Figure 11 show the results with change in application QoS in addition to congestion.

For the single-path protocol, there are large numbers of dropped cells due to congestion and the loss of cells persists. For the two-path protocol, we see a jump in the number of dropped cells at every 15 seconds when congestion hits. The load is gradually redistributed to avoid the congested links. This process takes about 2-3 seconds, the number of dropped cells reaches zero and the network is in stable state until the next congestion is generated. The same behavior is valid for the QoS change graph (see Figure 11). Hence, the two-path protocol outperforms the single-path protocol significantly.

5.3.2 Full-connected Network Topology

The graphs in Figure 12 show the results from the simulation run for the second sample network. The y axis presents the number of dropped cells, the x axis shows the simulation time.

The results for the single-path and the two-path runs in the second sample network are similar to the results in the first sample network. The two-path outperforms the single-path as shown in Figure 12.

[5]The availability change could be caused by adding another compressed video stream of capacity 3 Mbps.

[6]This test is only done on the ring network. The availability of the capacity is reduced by 4 Mbps, instead of 6 Mbps.

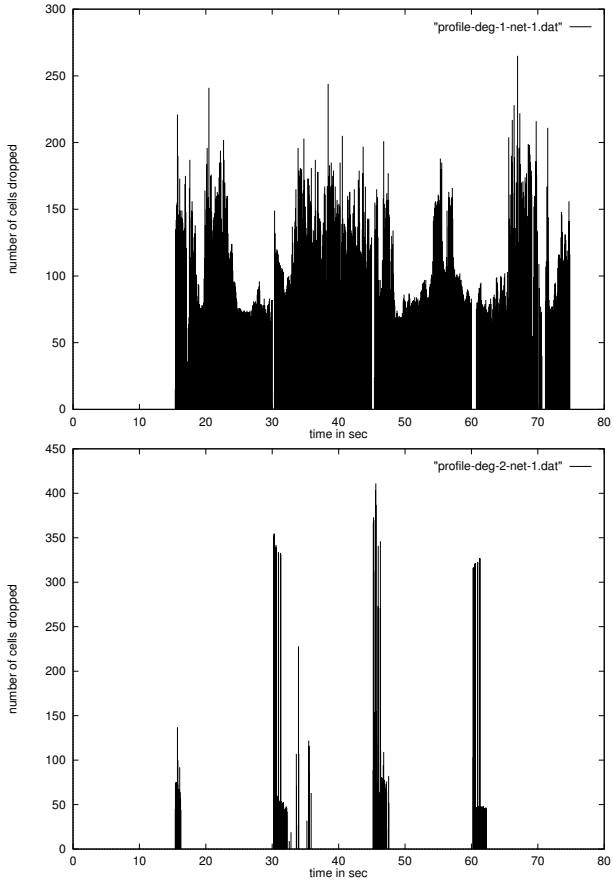


Figure 10: Lost Cells during congestion in the Ring Network Topology. The top graph shows the number of dropped cells for the single-path run, and the bottom one shows the two-path run.

6 Conclusion

A connection can better utilize the network resources by using multiple unique paths from the sender to the receiver because by having each connection doing load distribution among its multiple paths, the network can achieve a better load balance. To balance the load properly, we utilized a routing algorithm to find unique paths for a given sender and receiver and when congestion occurred, we invoked a feedback congestion control algorithm for the multi-path communication. None of the algorithms by itself are optimal and sufficient. However, in an integrated form they can provide a feasible service for load balancing. We demonstrated this behavior with our simulation. Although our simulation scenarios are simple, the preliminary results are promising for compressed video traffic (VBR) to achieve soft bandwidth guarantees.

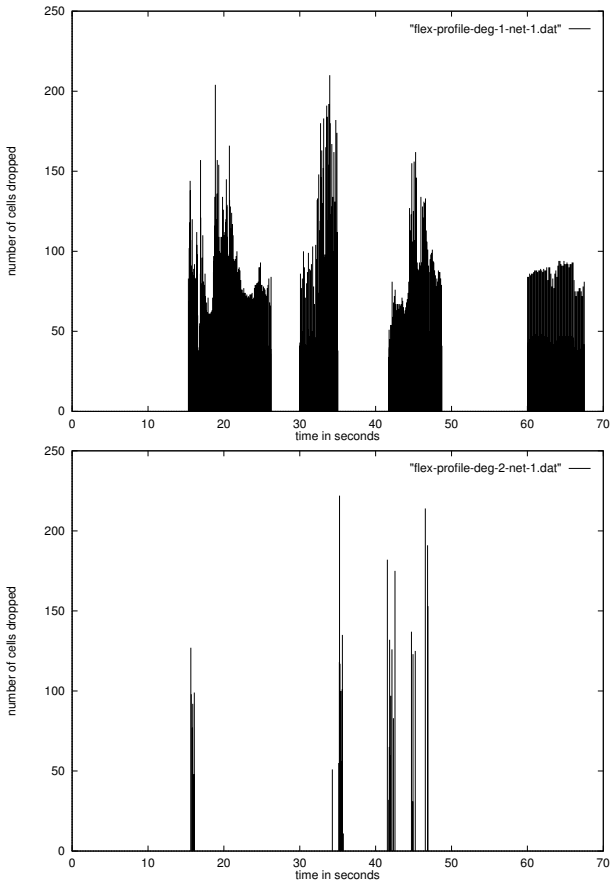


Figure 11: Lost Cells during QoS change and congestion in the Ring Network Topology. The top graph shows the number of dropped cells for the single-path run, and the bottom one shows the two-path run.

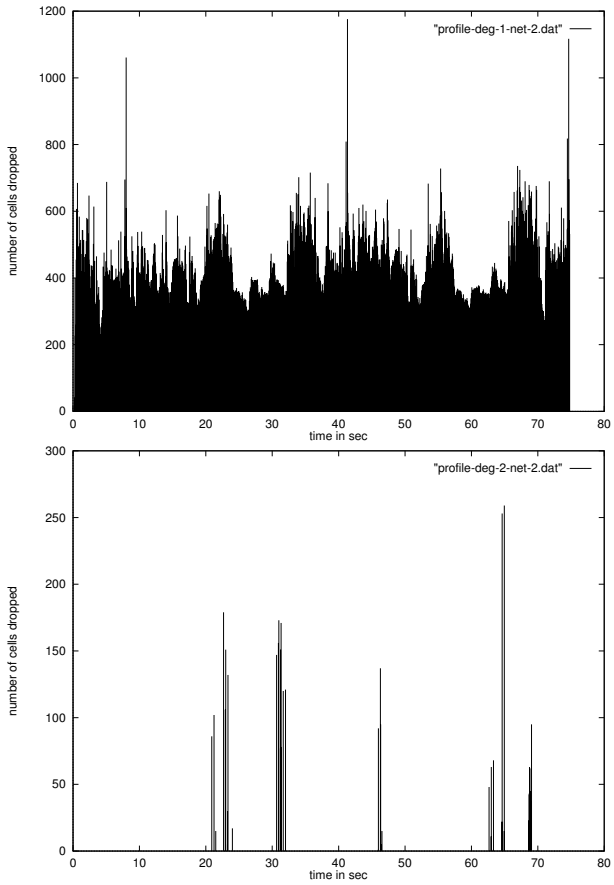


Figure 12: Lost Cells during congestion in the full-connected Network Topology. The top graph shows the number of dropped cells for the single-path run. The bottom one shows the two-path run.

References

[1] S. Bahk, and M. E. Zarki. "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks", *Proceedings of the ACM SIGCOMM*, October 1993.

[2] Z. Chen and S-M. Tan and R.H. Campbell and Y. Li "Real Time Video and Audio in the World Wide Web", WWW 95, (1995).

[3] T. Faber, and L.H. Landweber. "Dynamic Time Windows: Packet Admission Control with Feedback", *Proceedings of the ACM SIGCOMM*, August 1992.

[4] H. Kanakia, P. P. Misra, and A. Reibman. "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport", *Proceedings of the ACM SIGCOMM*, October 1993.

[5] H.T. Kung, T. Blackwell, and A. Chapman. "Credit-Based Flow Control for ATM Networks: Credit Update Protocol, Adaptive Credit Allocation, and Statistical Multiplexing", *Proceedings of the ACM SIGCOMM*, October 1994.

[6] C. Parris and D. Ferrari. "A Dynamic Connection Management Scheme for Guaranteed Performance Services in Packet-Switching Integrated Services Networks", *Technical Report TR-93-005*, International Computer Science Institute, Berkeley, CA, January 1993.

[7] S. Plotkin. "Competitive Routing of Virtual Circuits in ATM networks", *invited paper to IEEE J. Selected Areas in Communications*, 1995.

[8] C. Topolocic, "Experimental Internet Stream Protocol Version 2 (ST-II)", Internet Network Working Group, RFC 1190, October 1990