Virtual Path Management in ATM Networks

Ying-Dar Lin†, Wen-Jong Su†, Chi-Chun Lo*
†Department of Computer and Information Science
*Institute of Informatiion Management
National Chiao Tung University
Hsinchu, Taiwan
ydlin@cis.nctu.edu.tw
cclo@cc.nctu.edu.tw

ABSTRACT: In ATM networks, VPCs (Virtual Path Connections) could be setup permanently or semi-permanently by network administrators to form a virtual private network or temporarily by applications to group several VCCs (Virtual Channel Connections) of users. Virtual path management is of great importance in maintaining ATM network service quality. In this paper, we propose our heuristics for two problems of virtual path management: (1)Virtual Path Bandwidth Resizing Algorithm, which conducts utilization-driven virtual path bandwidth adjustment, and (2)Virtual Path Rerouting Algorithm, which is based on a dynamic prioritized route search with path overwriting feature. We develop simulation programs to examine the performance and effectiveness of the proposed heuristics.

Keywords: ATM virtual path, virtual path management, bandwidth resizing, path rerouting, path restoration

1. Introduction

The VP (Virtual Path) concept has been standardized in CCITT [1] for ATM networks. A VPC is broadly defined as a labeled unidirectional path between virtual path terminators. A VPC can be established permanently, semi-permanently or dynamically based on different purposes, namely, [2-4] User-user application (VPCs extend between a pair of UNIs), User-network application (VPCs extend between a UNI and a network node), and network-network application (VPCs extend between network nodes). Advantages over traffic control and resource management by the use of VPs include: (1)Simplification of CAC (Call Admission Control), (2)Separation of logical transport network from the physical transmission network, (3)Direct multiplexing of VPs to physical transmission links, and (4)Statistical multiplexing between VCs of a VP.

VP management functions could be implemented in a NMS (Network Management System) for network administrators to (1)monitor the performance of VPCs, (2)adjust the bandwidth of VPCs to efficiently use the bandwidth and (3)reroute the affected VPCs as soon as possible to maintain network reliability in case any failure occurs. There has been some researches on VP management or related topics including: (1)VP layout design[5], (2)VCC routing algorithm in VPC[6,7], (3)VP bandwidth resizing algorithm[4,8], (4)VP rerouting algorithm[9-13], and (5)VP performance and fault management with OAM (Operation, Administration,and Maintenance) mechanism [14]. With VP layout design, permanent or semi-permanent VPCs could be established to accommodate estimated traffic flow. With VCC routing algorithm, VCC could be routed to use network resource efficiently during VCC setup. NMS could use OAM mechanism to monitor the performance of VPCs or to detect failures by OAM mechanism. NMS could also adjust the bandwidth of VPCs by VP bandwidth resizing algorithm, and reroute affected VPCs in any failure by VP rerouting algorithm.

We propose two algorithms in VP management : (1)VP bandwidth resizing algorithm which conducts utilization-driven VP bandwidth adjustment, and (2)VP rerouting algorithm which is based on a dynamic prioritized route search with path overwriting feature.

Section 2 describes the proposed VP management algorithms. Section 3 presents the simulation results of our algorithms, and section 4 concludes this paper.

2. Proposed VP Management

2.1 Proposed VP Bandwidth Resizing Algorithm

The scheme we propose is a distributed bandwidth resizing mechanism, other than a centralized scheme proposed by Hisaya Hadama [8], because a distributed scheme is usually more reliable than a centralized scheme when network failures occur. The purpose of bandwidth resizing is to balance the load of each VPC and make better use of trunk capacity under certain quality of service (i.e. call blocking probability), by dynamic adjustment of allocated bandwidth for each virtual path. The management architecture consists of ***a bandwidth manager*** and ***agents*** for virtual paths(See Figure 1). The dynamic bandwidth adjustment are determined by the

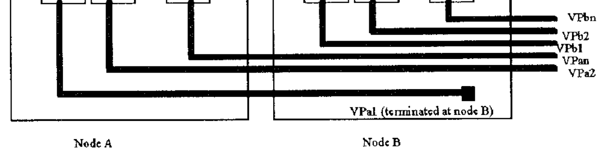

Figure1. Architecture of Proposed Distributed Bandwidth Management

0-7803-3250-4/96$5.00©1996 IEEE

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on July 03,2023 at 06:57:02 UTC from IEEE Xplore. Restrictions apply.

messages exchanged between bandwidth managers or between bandwidth manager and agents. Each node could be (1)source node of a VPC, (2)destination node of a VPC, or (3)intermediate node of a VPC.

Details are described as follows: *The agent* is responsible for call admission control and monitor the bandwidth of its corresponding VPC. If it is necessary to resize the bandwidth of a VPC, its corresponding agent sends a request to the local bandwidth manager. *A bandwidth manager* is built in each ATM node. The bandwidth manager creates an agent for each VPC originating from the node. It waits for any bandwidth adjustment request from both upstream or downstream links. When a bandwidth adjustment request for downstream links of a VPC is received by the bandwidth manager, we have the following two cases: *(1)*If the node is not the destination node of the VPC and the residual capacity in downstream link is enough, the bandwidth manager will reserve requested bandwidth and propagate the request downstream; otherwise, if the residual capacity is not enough, it will respond upstream with negative indication to reject the request. *(2)*If the node is the destination of the VPC, the bandwidth manager will respond upstream with positive indication to accept the request. Both positive and negative response messages will be propagated to the source node, and previously reserved bandwidth will be allocated for the VPC if the response message is positive; otherwise, previously reserved bandwidth will be released.

Unlike the algorithm, which adjusts the bandwidth of a VP when the number of VCC reaches a certain number [4], we adjust the bandwidth according to the degree of necessity of the VP adjustment in our proposed heuristics. The two key points to adjust the bandwidth in our algorithm are : *(1)*it might be more urgent to increase bandwidth than to decrease bandwidth, *(2)*it might need to increase more bandwidth when bandwidth is not enough but it doesn't need to decrease a lot of bandwidth when allocated bandwidth is more than needed. In our proposed heuristics, the time to launch a bandwidth adjustment is according to the usage of the allocated bandwidth; and the bandwidth change step to increase(S_{inc})is larger than the step to decrease (S_{dec}). Three parameters are used to evaluate the performance of the bandwidth management algorithm: (1)Bandwidth Utilization(BU), (2)Call Blocking Rate(CBR), and (3)Normalized Processing Load(NPL) which is defined as the ratio of the frequency of bandwidth adjustment request and the frequency of VCC setup.

Our bandwidth adjustment rules are described below:

1. ***If any call blocks***, the agent sends a bandwidth adjustment request to its bandwidth manager to increase the allocated bandwidth .
2. ***High Utilization***: Under the condition that no call blocks, if the bandwidth utilization at some time instant gets close to or greater than the pre-determined upper-threshold (BU_u) of bandwidth utilization, for T_u time units, the agent will send a request to its bandwidth manager to increase the allocated bandwidth.
3. ***Low Utilization***: Under the condition that no call blocks, if bandwidth utilization remains close to or lower than the pre-determined lower-threshold (BU_l) for T_l time units, the agent will send a request to its bandwidth manager to decrease the allocated bandwidth.
4. ***Medium Utilization***: Under the condition that no call blocks, if bandwidth utilization remains between BU_u and BU_l for T_m time units , and the residual bandwidth is less than a certain value, the agent will send a request to its bandwidth manager to increase the allocated bandwidth.

2.2 Proposed VP Rerouting Algorithm

Conventional restoration can be decomposed into three phases: route searching, response, and confirmation. The basic mechanism is based on Sender-Chooser concept[10]. When a failure occurs, one side of path terminators becomes a Sender, which broadcasts restoration messages toward other side of path terminators, i.e. the Chooser. The Chooser then selects a path by the information contained in the received restoration messages from Sender. To reduce the scale of the broadcasting area, Hiroyuki Fujii et al.[13] proposed a two-Sender-two-Chooser Double-Search algorithm to restore a bidirectional VP failure. We adopt this concept of Double-/Search as a basic scheme of our restoration algorithm. The heuristic algorithm is described as follows:

(1)*. Failure Detection Phase* (See Figure 2): The detection phase is based on OAM mechanism. Intermediate node of a VPC which detects a failure will generate VP-AIS(Alarm Indication Signal) to alert the downstream nodes that a failure has been detected upstream. The node terminates the failed VPC will generate VP-FERF(Far-end Received Failure) to alert upstream nodes that a failure has been detected downstream. The status of VPC endpoints which received either VP-AIS or VP-FERF then enters the next phase.

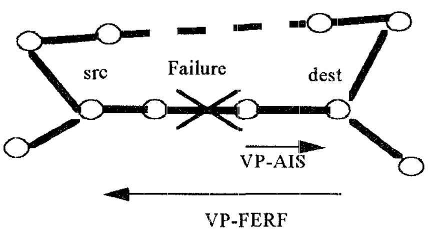

Figure 2. Failure Detection Phase

(2). Prioritized Double-Search Phase with Path Overwriting(See Figure 3): To start *double search* for a restoration path, both endpoints of affected VPCs will reserve the capacity of the failed VP on their neighboring links and broadcast the restoration messages to their neighboring nodes. In each restoration message, it contains a *priority number* indicating the preference degree of the link. A restoration message also contains the node ID (identifier) of its upstream node.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on July 03,2023 at 06:57:02 UTC from IEEE Xplore. Restrictions apply.

In this phase, each node which receives any restoration message from its adjacent nodes records the information contained in the *first* restoration message for any VPC and then forward the restoration message to other adjacent nodes. But if a node receives other new restoration messages for the same VPC later, it will not broadcast those new messages. But the node will check the priority number contained in the new messages. If the priority of new message is higher than the previous one, the node will *overwrite* the path information kept for previously received message. Thus, a better upstream node is recorded. As mentioned, this new message will not be re-broadcasted from the node again.

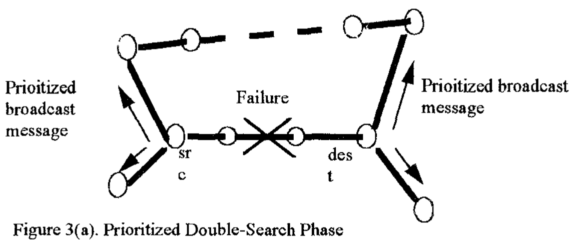

Figure 3(a). Prioritized Double-Search Phase

Priority x > y > z
Node A will be the next upstream node of Node D

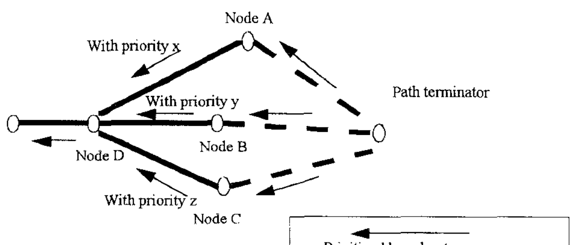

Figure 3(b) ***Path Overwriting***

(3). Response Phase(See Figure 4): Any node that receives the route searching messages from both end-points of a failed VPC proves that there exists restoration routes for the VPC. The node combines the rerouting information contained in the broadcasting messages from end-points of the failed VPC and then sends out a response message back to the source node of the failed VPC. Each node which receives a response message will forward the response message to its upstream node until the source node of the failed VPC receives the response message.

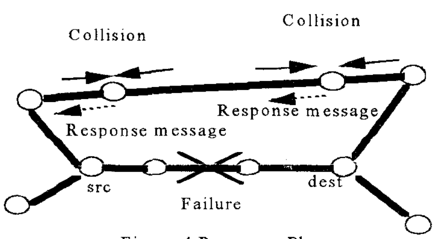

Figure 4.Response Phase

(4). Confirmation Phase(See Figure 5): When the source node of the failed VPC receives the first response message, it then sends confirmation message along the new route contained in the response message to update routing table of each intermediate node. Regarding the other alternate routes, the node also sends out cancellation messages to deallocate other unused but reserved bandwidth.

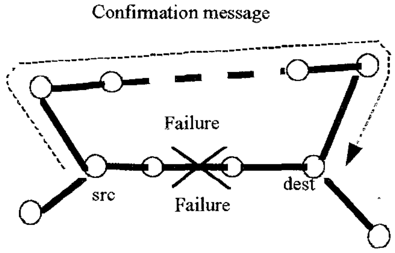

Figure 5. Confirmation Phase

3. Simulation studies

3.1 Bandwidth Resizing Algorithm

A single VPC is considered in our simulation. VCCs comes according to a Poisson distribution and the holding time is exponentially distributed. The required bandwidth of VCCs has a normal distribution. The maximum bandwidth that a VP could allocate is 50 units, initial allocated bandwidth is 20 units. $T_u = 5$ (time unit), $T_m = 10$ (time unit), $T_l = 20$ (time unit), S_{inc} = twice the Mean bandwidth of incoming VCCs, and S_{dec} = mean bandwidth of VCCs.

We first observe the results of different pairs of (BU_u , BU_l) while service rate and arrival rate are equal to one unit. We examine the performance of our algorithm in terms of call blocking rate, normalized processing load and the utilization of allocated bandwidth.

	CBR	NPL	Utilization
(0.7, 0.2)	0	0.005982	0.370041
(0.7, 0.3)	0	0.004985	0.530569
(0.7, 0.4)	0	0.008973	0.485572
(0.7, 0.5)	0.00099	0.018962	0.481927
(0.7, 0.6)	0.00099	0.021956	0.499571
(0.6, 0.3)	0	0.01695	0.34426
(0.8, 0.3)	0	0.003988	0.512216
(0.9, 0.3)	0	0.003988	0.512216
(0.5, 0.3)	0	0.003988	0.422928
(0.4, 0.3)	0	0.001944	0.395553
(0.8, 0.4)	0.00099	0.008982	0.499337
(0.9, 0.4)	0.00099	0.011976	0.439887
(0.6, 0.4)	0	0.015952	0.451707
(0.5, 0.4)	0	0.007976	0.423864

Table 1. Evaluation Parameters of (BU_u , BU_l)

From Table 1, we choose (BU_u , BU_l) = (0.7,0.3) as the base parameters in the following analysis because they produce a lower call blocking rate, a lower normalized processing load and a higher utilization in Table 1.

Figure 6, 7 and 8 show the average call blocking rate, average normalized processing load, and average utilization of a virtual path for 2000 arrivals and departures of VCCs at different mean bandwidth. Different curves with various *(Service Rate, Arrival Rate)* are plotted to reflect the conditions when a virtual path is lightly loaded, medium loaded and highly loaded, respec-

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on July 03,2023 at 06:57:02 UTC from IEEE Xplore. Restrictions apply.

tively.

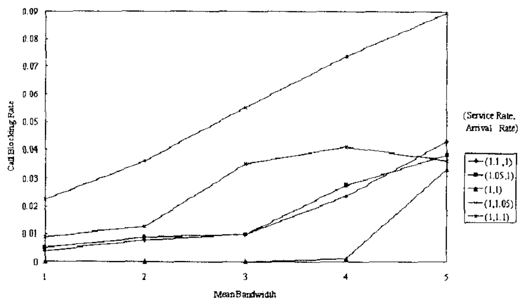

Figure 6. Call Blocking Rate v.s. VCC Mean Bandwidth

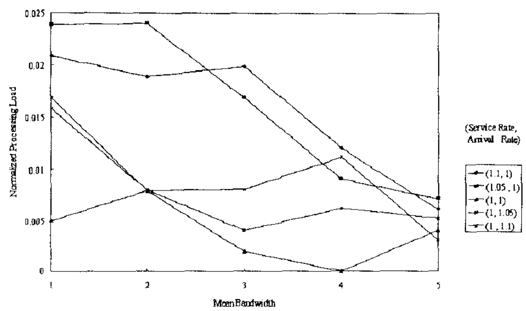

Figure 7. NPL v.s. VCC Mean Bandwidth

In Figure 6, our algorithm performs well in call blocking rate when the call arrival rate equals service rate and mean bandwidth of VCCs is small compared to the initial allocated bandwidth of the VPC. We note that when (1)the mean bandwidth of incoming calls becomes larger *or* (2)the arrival rate is larger than the service rate, the call blocking rate will increase. Because in these two situations, it is easy for the VPC to resize to its maximum allocated bandwidth. If the allocated bandwidth has been resized to its maximum value, average call blocking rate will increase because incoming VCCs might be kept blocked due to the residual bandwidth is less than the required bandwidth.

In Figure 7, the normalized processing load of our algorithm is low on the average. From this figure, we could see that when the arrival rate is greater than the service rate, the normalized processing load is less than that when arrival rate is smaller than its service rate. The reason is that when the arrival rate is greater than the service rate or the mean bandwidth of incoming VCCs is larger, more calls will be blocked after the VP bandwidth has been resized to its maximum value. Although our bandwidth management algorithm performs a bandwidth adjustment every time when any call blocks, few bandwidth adjustment will be performed after the allocated bandwidth of a virtual path has been resized to its maximum bandwidth. Hence, the normalized processing load becomes lower on the average. However, when the service rate is equal to the arrival rate, few bandwidth adjustment will be performed after the usage of bandwidth stays in a stable state. We could see in Figure 7 that when the service rate is eaual to the arrival rate, the proposed bandwidth algorithm has the lowest normalized processing load on the average. In Figure 8, we show that the proposed algorithm controls most of the average bandwidth utilization in the range (BU_u , BU_l)=(0.7,0.3). We also observe that the average utilization is higher when the arrival rate is larger than the service rate. Again, this is because that the maximum bandwidth of the VPC is reached, thus more bandwidth is used.

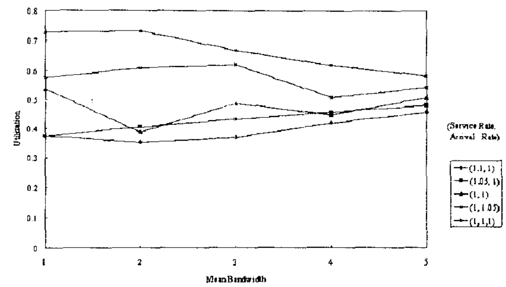

Figure 8. Utilization v.s. VCC Mean Bandwidth

To summarize, the proposed bandwidth adjustment algorithm does control the average utilization to run between the upper-threshold and the lower-threshold and the normalized processing load and average call blocking rate are low if the arrival rate is not greater than the service rate or the mean bandwidth of incoming calls is not large.

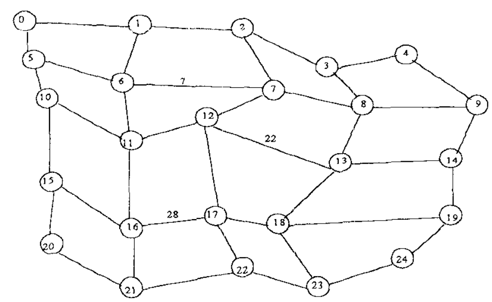

Figure 9. Network Model 1

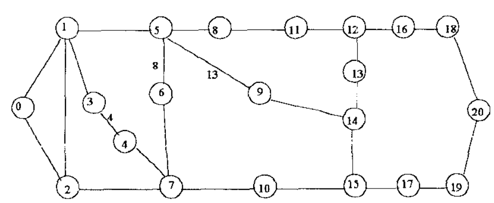

Figure 10. Network Model 2

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on July 03,2023 at 06:57:02 UTC from IEEE Xplore. Restrictions apply.

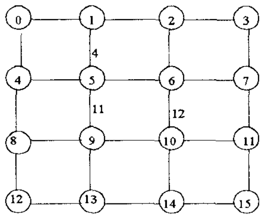

Figure 11. Network Model 3

3.2 Rerouting Algorithm

The simulation is based on the three network models shown in Figure 9, 10 and 11. Again, bandwidth of VPCs is determined according to a normal distribution. The processing time is assumed to be 5 ms for each node. We list the restoration results in network model 1, 2 and 3 in Table 5, 6 and 7.

Failed Link ID	Restoration Time	Restoration Ratio
28	65 ms	100%
7	85 ms	100%
22	65 ms	100%

Table 2. Restoration Results in Network Model 1

Failed Link ID	Restoration Time	Restoration Ratio
8	75 ms	99%
13	85 ms	98%
4	65 ms	100%

Table 3. Restoration Results in Network Model 2

Failed Link ID	Restoration Time	Restoration Ratio
4	65 ms	100%
12	70 ms	100%
11	45 ms	100%

Table 4. Restoration Results in Network Model 3

In each simulation, there exists 3000 virtual path connections, and each failed link contains about 80 virtual paths. The restoration is fast and most of the failed virtual paths are restored in simulations. Note that in network model 2, failed link 8 and 13 are not fully restored because these two links contains more virtual path connections and the alternative paths are not as many as those in network model 1 and 3.

4. Conclusions and Future Works

In this paper, we proposed VP bandwidth resizing algorithm, and rerouting algorithm. The way that the proposed VP bandwidth management algorithm adjusts VP bandwidth is based on the utilization of the bandwidth and the different degrees of urgency between increase and decrease of the bandwidth. From the simulations, our algorithm performs best when (1)the arrival rate of the incoming connections approaches service rate of virtual path, *and* (2)the required bandwidth of the incoming VCC is not large compared to the allocated bandwidth. Our rerouting algorithm is proved, in simulations, to be capable of a fast and near complete restoration of VPCs with failed links.

Although our VP management algorithms perform effectively in most situations, some studies are still needed to compare our resizing and rerouting algorithms with other schemes. We also intend to implement some of the management functions on a campus ATM testbed to evaluate their feasibility and effectiveness.

References

[1] CCITT Recommendation I.113, Vocabulary of terms for broadband aspects of ISDN, 1990.

[2] Ken-Ichi Sato, et al., "Broad-Band ATM Network Architecture Based on VirtualPaths", IEEE Transactions on Communications, Vol. 38, No.8, pp.1212-1222, August 1990.

[3] The ATM Forum, ATM User-Network Interface Specification, Version 3.1, The ATM Forum, 1995.

[4] Satoru Ohta, et al., "Dynamic Bandwidth Control of the Virtual Path in an Asynchronous Transfer Mode Network", IEEE Transactions on Communications, Vol. 40, No.7, pp.1239-1247, July 1990.

[5] Imrich Chlamtac, et al., "How to Establish and Utilize Virtual Paths in ATM Networks", Proceedings of the IEEE ICC'93.

[6] Gerald R. Ash, "Design and Control of Networks with Dynamic Nonhierarchical Routing", IEEE Communication Magazine, pp.34-40,October 1990.

[7] Yu Watanabe, et al., "Dynamic Routing Schemes for International Networks", IEEE Communication Magazine, pp.70-75, Oct. 1994.

[8] Hisaya Hadama, et al.,"Direct Virtual Path Configuration in Large-Scale ATM Networks", Proceedings IEEE INFOCOM'94, pp.201-207, Vol. 1, 1994.

[9] Ryutaro Kawamura, et al., "Self-Healing ATM Networks Based on Virtual Path Concept, IEEE Journal on Selected Areas in Communications, Vol.12, No.1, pp.120-127, January 1994.

[10]W.D.Grover, "The SELFHEALING™ Network", Proc. GLOBECOM'87, pp.1090-1095, Nov. 1987.

[11]Hiroaki Komine, et al., "A Distributed Restoration Algorithm for Multiple Link and Node Failures of Transport Networks", Proceedings of ICC, 1990.

[12]Noriaki Yoshikai, et. al., "Control Protocol and Its Performance Analysis for Distributed ATM Virtual Path Self-Healing Network", IEEE Journal on Selected Areas in Communications,Vol.12, No.6, pp.1020-1029, August 1994.

[13]Hirojuki Fujii, et al., "Restoration Message Transfer Mechanism and Restoration Characteristics of Double-Search Self-Healing ATM Network", IEEE Journal on Selected Areas in Communications,Vol.12, No.1, pp.149-157, January 1994.

[14]Breuer, "ATM-Layer OAM: Principles and Open Issues", IEEE Communications Magazine, pp.75-78, Sept. 1991.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on July 03,2023 at 06:57:02 UTC from IEEE Xplore. Restrictions apply.