Virtual Path Routing for ATM Networks

Yiu Kwok Tham and Åke Arvidsson
Department of Telecommunications and Mathematics
University of Karlskrona/Ronneby
S-37179 Karlskrona
Sweden

Abstract

Virtual paths facilitate the rapid movement of end-to-end traffic streams in an ATM network by keeping processing at intermediate nodes en route to the minimum. There may exist, however, some virtual paths in an ATM network with low volumes of traffic on them. Balancing between efficient utilization of transmission resources en route and keeping intermediate switching to the minimum, lightly loaded virtual paths are decomposed into at most two logical hops, which require only one intermediate switching for an end-to-end traffic stream. The decomposition procedure and data structure for efficient implementation are described. For a twenty-node network with between three and four hundred virtual paths, experimental results show that the average number of lightly loaded virtual paths that cannot be decomposed by our procedure is about 6.5. Work in progress and future work lie in simulating network performance and investigating improved network dimensioning techniques for direct and two-hop routes in ATM networks.

1 Introduction

Virtual path is an important concept in ATM networks. It recognizes the distinct identity of a traffic stream between an origin node and a destination node. In the case where the end-to-end traffic matrix is symmetric, one may simply consider the amount of traffic between a pair of end nodes; we assume in this work that the end-to-end traffic matrix is symmetric.

Furthermore a virtual path specifies the route (a sequence of transmission links) to be traversed by a traffic stream between a pair of end nodes. This allows for rapid movement of traffic (ATM cells) with minimal processing at intermediate nodes (which is related to switching costs at each node).

Because of the number of routes that traffic between a pair of end nodes may traverse and the number of all possible pairs of end nodes with specified traffic requirements, this may give rise to an inordinately large number of virtual paths. This may cause problems in two areas of ATM network management. One problem area is in the management of virtual path identifiers. The maximum number of virtual paths is limited by the field length allowed for virtual path identifiers. The other problem area is in bandwidth management or efficient utilization of the transmission capacity on a transmission link. This may give rise to inefficiency particularly when transmission capacities are reserved for virtual paths with small volumes of traffic to satisfy given levels of quality of service.

In this work we assume that the amounts of traffic between all pairs of nodes are given (or previously measured) in the form of a (symmetric) end-to-end traffic matrix and routing computations have been performed. This means that we are given a potentially large number of virtual paths with transmission capacities allocated (in the form of number of transmission circuits). We will decompose lightly loaded virtual paths into components of virtual paths of shorter length, which in turn can be used to carry other traffic that passes through a common component virtual path (or virtual path link). In this way a virtual path will have shorter length and carry a higher volume of traffic.
2 Motivation

In network planning, design and management, there often exist competing goals and objectives, for which network planners have to strike a balance and arrive at a happy mean.

In virtual path routing, ideally one would like to provide end-to-end routes for a traffic stream between a pair of end nodes without any intermediate switching as switching incurs processing time and requires processing logic that translates into additional equipment costs. The switching dimension of the problem is further aggravated in high-speed networking, as is typical for ATM network deployment.

A virtual path may simply identify a route for a traffic stream between a pair of end nodes or additionally it may have transmission capacity committed along its route for its associated service class to satisfy a given level of quality of service. In this work we have only one class of service (which one may think of as voice calls and this provides further justification of our assumption of symmetric end-to-end traffic matrices) and transmission circuits are reserved along the route of a virtual path. Our treatment of a single service class may also be viewed as simplification of network management by segregating service classes and dedicating sufficient transmission resources to each service class.

The idea of separate management may be pushed further by treating each virtual path and its committed transmission circuits as a distinct inviolable entity and not allowing any sharing of transmission circuits between virtual paths. Network management and management of individual virtual paths are indeed simplified, and this is justified if there are high volumes of traffic on all virtual paths as might happen during busy hours of a business day or at peak calling seasons like Christmas and one is only concerned with getting as much traffic through the network as possible with minimal switching and processing. Under normal operating conditions, however, there bound to exist some virtual paths with low volumes of traffic. Since we already assume segregation of service classes with no sharing of resources among service classes, it behoves us to manage the resources of a service class as tightly and efficiently as possible.

Resources on transmission links are best utilized when there is maximum sharing among all virtual paths traversing common transmission links. Sharing should only be allowed, however, to the extent that it is necessary as it complicates traffic management since congestion on a traffic stream may adversely affect the quality of service of other traffic streams traversing common transmission links under a complete sharing regime. So only virtual paths with low volumes of traffic are targeted for sharing of transmission capacity along common transmission links.

If we now confer a distinct identity on a sequence of common transmission links, which may be called a virtual-path link, the routing of a virtual path (with low volume of traffic) may be decomposed into a sequence of virtual-path links. This would require intermediate switching. While transmission link-by-link switching may yield maximum benefits of sharing of transmission circuits, the amount of switching involved may be deemed excessive. So we allow a virtual path to be decomposed into at most two virtual-path links (or not at all if there is sufficient traffic on the virtual path). That is, the end nodes of a virtual path are at most two logical hops away.

To sum up, a virtual path is decomposed only if the amount of traffic on it (or the number of allocated circuits) is less than a threshold value — we consider a threshold value of thirty circuits. To minimize additional switching and processing time, a virtual path is decomposed into at most two virtual-path links.

Next we look at the decomposition procedure, which only applies to virtual paths with less than the threshold value of traffic.

3 Decomposition Procedure

Given the virtual paths for an end-to-end traffic matrix, we can immediately separate virtual paths with traffic greater than or equal to the threshold value from those with less than the threshold value of traffic, and concentrate on decomposing virtual paths with low volumes of traffic.
The routing for virtual paths and the circuit allocation may be the outcome of routing and dimensioning calculations for a given end-to-end traffic matrix. This may be performed at network planning and design stage when traffic matrices have been obtained by traffic measurement and forecasting techniques and virtual paths and circuit allocation are to be configured. Different virtual path configurations may be put into effect for various time periods of a working day and even for different times of a year so as to be able to best meet daily and seasonal variations in user traffic demands. Some degree of central planning and administration have been implicitly assumed, though this is not entirely necessary.

The network may be managed on a faster time scale with greater intelligence and processing logic placed on key nodes of the network. Under distributed and dynamic network management regimes, traffic flowing through various parts of the network may be periodically measured, virtual paths may be dynamically assigned and circuit allocation may be continually updated. That is, the routing for virtual paths and the circuit allocation may gradually evolve with time. But, from time to time, there may exist virtual paths with low volumes of traffic that may benefit from re-organization and decomposition into virtual-path links.

3.1 Terminology

For the ease of reference, we call virtual paths with traffic volume greater than or equal to a given threshold value primary virtual paths (or primary virtual-path routes) while those with traffic volume less than the given threshold value thin virtual paths, which are objects for decomposition. A primary virtual-path route may, on the other hand, carry transit traffic in addition to its own end-to-end traffic, for example, as a result of the decomposition of a thin virtual path into virtual-path components. When a virtual-path component has no end-to-end traffic of its own, it is called a virtual-path link. Furthermore a thin virtual path may be able to carry transit traffic from longer thin virtual paths such that the total of its own end-to-end traffic and the transit traffic that it carries reaches or exceeds the given threshold value. In this case the thin virtual path becomes a primary virtual-path route and its own end-to-end traffic benefits from the aggregation of transit traffic from other thin virtual paths in that its own end-to-end traffic goes straight through without incurring any intermediate switching. It stands further to benefit from better statistical multiplexing gain due to the higher volume of traffic that it carries, or from better level of quality of service if, for simplicity of operation, the total number of allocated circuits is not correspondingly reduced (for example, when aggregate traffic characteristics are not well understood or extensive re-computation is required, and network operators choose to err on the conservative side).

From the foregoing discussion, it should be clear that a primary virtual-path route or a virtual-path link would have aggregate traffic (or aggregate number of circuits) greater than or equal to the threshold value. When we employ the number of circuits instead of the volume of traffic, this may be due to post-processing after first-cut routing and dimensioning have been performed. Post-processing may be required, for example, when there are constraints on the minimum module size for interfacing periphery equipment while initial network routing and dimensioning can be more efficiently computed by not considering these constraints.

3.2 Traffic Aggregation in Recursive Decomposition

The decomposition procedure for a thin virtual path is a traffic aggregation procedure such that the aggregate traffic on the component virtual-path links or virtual-path routes reaches or exceeds the threshold value. That is, a virtual path is decomposed into two logical hops only if the aggregate traffic on each hop is not less than the threshold value. Each logical hop in turn would become a leg of a two-hop route for other thin virtual paths or the logical hop itself is a primary virtual-path route.

The decomposition of a thin virtual path is a recursive procedure. By itself, a candidate logical hop would not have sufficient traffic from a thin virtual path under decomposition. In order for a candidate logical hop to become a primary virtual-path route or a virtual-path link,
it must be able to accumulate traffic from other virtual paths such that the aggregate traffic reaches or exceeds the threshold value. For a thin virtual path to be successfully decomposed, a pair of complementary candidate logical hops are considered at the same time: a candidate logical hop together with its complementary logical hop would form a two-hop end-to-end route for the thin virtual path under decomposition, subject to satisfaction of traffic threshold values.

This implies that in the process of accumulating traffic for a candidate logical hop of a thin virtual path under decomposition, traffic from other thin virtual paths can be collected only if the contributing thin virtual path can be successfully decomposed into the same logical hop and its own complementary logical hop. Therefore the process of accumulating traffic to reach the threshold value propagates to the complementary logical hop of a potential contributing thin virtual path.

The propagation terminates when a resulting candidate logical hop is an end-to-end route for either a primary virtual path or a thin virtual path, and thus there is no further complementary logical hop to consider and ensure that the traffic threshold value is satisfied.

Starting from one thin virtual path and a complementary pair of logical hops, a number of thin virtual paths have to be successively considered that share overlapping candidate logical hops. Only when all the affected thin virtual paths and candidate logical hops satisfy the traffic threshold value, can the starting thin virtual path be successfully decomposed into the candidate pair of complementary logical hops.

For a thin virtual path consisting of $h$ transmission links, there are $h - 1$ possible pairs of complementary logical hops. The whole virtual path itself may be considered as a candidate logical hop for other longer thin virtual paths. Let the thin virtual path be $a - b - \ldots - d$ (where $a, b, \ldots, d$ are node names) and a longer thin virtual path be $a - b - \ldots - d - \ldots - f$. Then $a - b - \ldots - d$ is said to be a prefix of $a - b - \ldots - d - \ldots - f$. The notion of suffix can be similarly defined. Since we are dealing with bidirectional virtual paths with the same bandwidth in both directions, it is not necessary to make the distinction between prefix and suffix. It is convenient, however, to refer to a complementary pair of candidate logical hops as a prefix and its matching suffix.

### 3.3 Prefix Processing

Treating a virtual path as a string of node names, decomposing a thin virtual path is equivalent to finding a pair of prefix and matching suffix satisfying the traffic threshold requirement. To ensure that a candidate logical hop satisfies the traffic threshold requirement, one successively examines virtual paths with the common prefix, which is the candidate logical hop, and attempts to add up traffic on these virtual paths so that the accumulated traffic satisfies the traffic threshold requirement.

To facilitate searching for strings with a common prefix, the virtual paths may be lexicographically sorted by their string of node names, and may be organized into data structures suitable for rapid searching, for example, binary trees or other search tree structures so that locating the starting point of virtual paths with a certain prefix takes $O(\log n)$ time for a tree of size $n$.

The size of the search tree may be further reduced to speed up the look-up operations, which are core to the recursive decomposition of thin virtual paths, by partitioning the set of strings of node names for virtual paths into smaller sets. For instance, separate search trees may be formed for strings of node names with the same end node. From a given end node, the search trees may be further partitioned, for example, into sets of strings of node names with a common second-node name. This is like organizing virtual paths from an end node according to their outgoing transmission link from the given end node, which is a natural thing to do in a distributed implementation for dynamic network traffic management. As a starting point, we consider centralized implementation for this work.
3.4 Longest Path First

An advantage of centralized implementation is that one has better control over the order of processing of the thin virtual paths. A good heuristic is to decompose the thin virtual paths in descending order of the path length (that is, the number of transmission links in a virtual path) because in general it is harder to decompose a long virtual path into two logical hops. So these difficult cases should be dealt with first when more alternatives are available by way of recursive decomposition of other thin virtual paths. In addition to processing the thin virtual paths in descending path length, for the same path length we process the thin virtual paths in ascending order of path length because in general it is harder to accumulate traffic to satisfy the traffic threshold requirement for a virtual path with a smaller volume of traffic.

Finally it may be added that we always treat a virtual path consisting of only one transmission link as a primary virtual path even if the traffic on it may be less than the traffic threshold requirement. In a way, these unit-path-length virtual paths may be viewed as feeder links into the network core where the volume of carried traffic and allocated transmission capacity are high.

4 Test Results

We have tested our decomposition procedure on 64 network configurations where the routing and circuit allocations have been computed for the virtual paths in the network. Each of the networks consists of twenty nodes. Initial routing calculations yield between 300 and 400 virtual paths. We use a threshold value of 30 circuits. With this threshold value, the numbers of thin virtual paths lie in the range of 100 to 150. After applying the decomposition procedure, the numbers of undecomposed thin virtual paths lie in the range of 0 to 16; the average number is about 6.5. The numbers of new primary virtual paths formed lie in the range of 10 to 31; the average number of new primary virtual paths formed is about 19. This means that for these originally thin virtual paths, no intermediate switching of their end-to-end traffic is required because of success in traffic aggregation from longer thin virtual paths that use these virtual paths as one of their two logical hops. The numbers of new virtual-path links formed lie in the range of 0 to 9; the average number is about 3 (recall that virtual-path links carry only transit traffic).

For two test cases, we have inverted the order of processing thin virtual paths, that is, we process the thin virtual paths in ascending order of path length. In one test case, the number of undecomposed thin virtual paths has gone up from 6 to 10. In another case, the number of undecomposed thin virtual paths has gone up from 7 to 12.

One could extrapolate the worst-case performance of a distributed implementation over an ATM network where there is little control over the order of processing thin virtual paths.

5 Work in Progress

Based on the decomposition results, work is in progress to simulate network performance and will be reported in the extended version of this paper.