Service-Specific ABR Routing Algorithms

Youngsu Chae Ellen W. Zegura
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{yschae, ewz}@cc.gatech.edu

Abstract

A wide variety of applications, including traditional data transfer and adaptive real-time multimedia, can potentially use the ATM available bit rate service. While these applications all share the ability to adapt their rate, they otherwise may have very different characteristics. The goal of this paper is to examine routing algorithms in the context of two different classes of best-effort applications: data-driven and real-time. We propose three new routing algorithms and compare them to algorithms from the literature using global and per-connection metrics. We demonstrate that the performance of a given routing algorithm is quite sensitive to the class of application. For real-time connections, routing algorithms that prefer shorter routes show better results. For data-driven applications, routing algorithms that try to balance allocated bandwidth and length of routes perform better. We further show that our new subgraphing algorithm consistently performs best for real-time applications on nearly all metrics.

1. Introduction

The popularity of the Internet, and the best-effort bandwidth-sharing service provided by TCP over IP, has led to the development of a wide variety of applications that are potentially well served by the ATM available bit rate (ABR) service [1]. These applications include traditional data transfer applications, such as FTP [11] and HTTP [3], along with newer multimedia applications such as nv [4] and vic [10]. This collection of applications share the characteristic that they can adjust their output bit rate based on available bandwidth information.

The current TCP/IP best-effort service treats all connections similarly. However, the growing diversity of best-effort applications, and the opportunity provided by connection-oriented ABR service, raise the issue of different mechanisms — and specifically, different *routing algorithms* — for different types of best-effort applications. The goal of this paper is to examine routing algorithms in the context of two different classes of best-effort applications: data-driven and real-time.

A somewhat similar study was undertaken by Ma et al. [9]. The paper by Ma et al. is significant in that it identified the importance of ABR routing and considered the use of different routing algorithms for applications in two classes: high-bandwidth applications (characterized by a large amount of data and little sensitivity to delay) and low latency applications (characterized by a smaller amount of data and sensitivity to delay).

We expand upon the work of Ma et al. in three dimensions. First, we consider a refinement of the low latency class to model the important class of real-time adaptive applications. We term this class "real-time" and the high-bandwidth class "data-driven". Second, we consider a wider range of performance metrics to include global metrics and route lengths, and we apply these metrics to both classes of traffic. Third, we explicitly note a tradeoff between the per-connection average throughput and the variance across connections. We also propose several new algorithms and compare them to those from Ma.

Data-driven applications such as FTP have a fixed amount of data to transmit and can typically make use of as much bandwidth as is available (up to the peak supported by the sending end system or access link). Data-driven applications are also typically tolerant of long network delays. Real-time applications, on the other hand, may be sensitive to network delay, since long delays can increase the possibility of delay jitter and decrease the interactivity that some real-time applications require. Further, real-time applications often have an upper bound on peak bandwidth imposed by the application itself, rather than end system limitations. Finally, the duration of real-time applications is not typically affected by the available bandwidth [1].

[1] Clearly the duration of the application will sometimes be affected by the available bandwidth, particularly when that bandwidth causes very good or very poor quality of the application.

In this paper, we investigate how routing algorithms affect the performance of data-driven and real-time applications that use an ABR service with max-min fair-share bandwidth allocation. In addition to routing algorithms from the literature, we propose several new algorithms. These new algorithms make use of a simple abstraction of the max-min rate information to compute link costs and achieve load balancing. The performance of the routing algorithms is evaluated based on network-wide and per-connection metrics. In addition to standard metrics (e.g., average connection rate), we include the *variance* in rates across different connections. This measure captures a form of inter-connection fairness that can be affected by routing algorithms.

The remainder of this paper is organized as follows. In Section 2, we describe main ideas of the paper, including the performance considerations for data-driven and real-time applications, the set of routing algorithms, and the basic simulation setup. Sections 3 and 4 contain simulation results for real-time and data-driven applications respectively. In Section 5 we discuss related work. Conclusions are provided in Section 6.

2. ABR Routing

2.1. Performance Considerations

For ABR services, the available bandwidth is dynamically shared among the connections. One of the important issues is how to fairly allocate available bandwidth among connections sharing the same link. A number of different definitions of fairness are possible. In this paper, we assume the commonly used criterion of max-min fairness [6].

When trying to maximize bandwidth, it seems natural to select a path with the highest max-min fair rate. For a max-min fair share network, however, there is no guarantee that the highest rate will be sustained for the entire lifetime of the connection. Since the max-min fair rate of a connection is the minimum of the fair rates available on each link along the route, the chance that the max-min fair rate will go down increases with the length of the route. Moreover, longer routes consume more network resources, which might reduce the max-min fair rate of future connections. Thus, shorter routes are preferred over longer routes.

Since ABR connections share the available bandwidth on links, the actual *distribution* of rates is also important. In general, a small variation in rates is preferable over a large variation. A max-min fair share mechanism has only indirect control over the distribution of rates since it only enforces max-min fairness on the fixed route configurations of connections. Thus, routing algorithms that "balance" the load to achieve lower variance in rates are preferred over algorithms that cause high variance.

The service requirements for the data- and real-time ABR connections are quite different from each other. The data-driven ABR connections utilize as much bandwidth as is available; the connection lifetime depends upon the allocated bandwidth. The data-driven services also can typically tolerate longer network delays. For the data-driven services, therefore, providing a larger bandwidth is one of the major objectives of routing algorithms. The real-time services, however, get somewhat less benefit from a larger bandwidth allocation as compared to the data-driven services. In our model, the connection lifetime is an application specific characteristic, which is not dependent upon the allocated bandwidth. Furthermore, real-time applications such as nv and vic require a maximum delay to maintain interactivity of the communications. Thus, limiting the length of the routes for the applications is desirable. The different service requirements for the data- and real-time connections represent different characteristics of the routing algorithms used for each type of connections.

2.2. Routing Algorithms

In this paper, we propose a new routing algorithm based on a subgraphing technique. We also propose two new algorithms that use polynomial and exponential functions of the number of connections per link as link-cost metrics. These algorithms have some ability to achieve load-balancing. Specifically, the three new algorithms are :

- Subgraphing : Assign large link cost to all the links that have more connections than a specific threshold and unit cost to the remaining links. Select a shortest path in the graph. The threshold used in our algorithm is $mean_{con} + \alpha * std_{con}$, where $mean_{con}$ is the average number of connections per link and std_{con} is the standard deviation of the number of connections per link. The value of α used in our simulation study is 3.0.

- Polynomial(n) : Make a graph using the subgraphing method, then select a path with the minimum cost

  $$cost(P, n) = \sum_{i=1}^{k} C_i^n$$

  where $C_1, \ldots, C_k$ are the number of connections on links on the path P with k hops. The value of the exponent n used in our simulation study is 1.0.

- Exponential(n) : Select a path with the minimum cost

  $$cost(P, n) = \sum_{i=1}^{k} n^{C_i}$$

where $C_1, \ldots, C_k$ are the number of connections on links on the path P with k hops. The value of the base n used in our simulation study is 1.1.

We compare our new algorithms to the following routing algorithms.

- Shortest-widest : Select a path with the maximum approximate max-min rate [9]. If there are multiple such paths, select one with the minimum hop count.

- Widest-shortest : Select a path with the minimum hop count. If there are multiple such paths, select one with the maximum approximate max-min rate.

- Shortest-dist(P,n) [9] : Select a path with the shortest distance

$$dist(P, n) = \sum_{i=1}^{k} \frac{1}{r_i^n}$$

where $r_1, \ldots, r_k$ are the approximate max-min rates of links on the path P with k hops. The value of the exponent n used in our simulation study is 1.0. In Ma et al. [9], each r_i is further approximated by a fixed number of discrete rate intervals. In our evaluation, we use an exact value of r_i instead of their discrete-approximated value.

2.3. Simulation Details

We have designed and implemented a call-level event-driven simulator. Parameters for the simulator include a network topology, connection arrival process, connection lifetime distribution for real-time applications, and distribution of the amount of data traffic for data-driven connections. Three different topologies, G1, G2 and G3, are used for our simulation study (Figure 1). The topologies G1 and G2 are from Ma et al.[9]. Each node in the topologies represents a multiplexing/demultiplexing point for a set of end systems attached to it. The link capacities are all 155 Mbps.

The nature of traffic injected into the network is controlled by the connection arrival process, connection lifetime distribution, distribution of quantity of data for connections, and the connection destination distribution. In our simulation study, the connection arrival process follows a Poisson distribution. The connection lifetime and the quantity of data for a connection follow an exponential distribution. A uniform distribution is used for connection destination distribution. The network load is controlled by the mean connection arrival rate. We use different measures of the network load for real-time and data-driven connections. For real-time connections, the network load is measured by the average number of current connections per node, which is the average connection arrival rate multiplied by the average

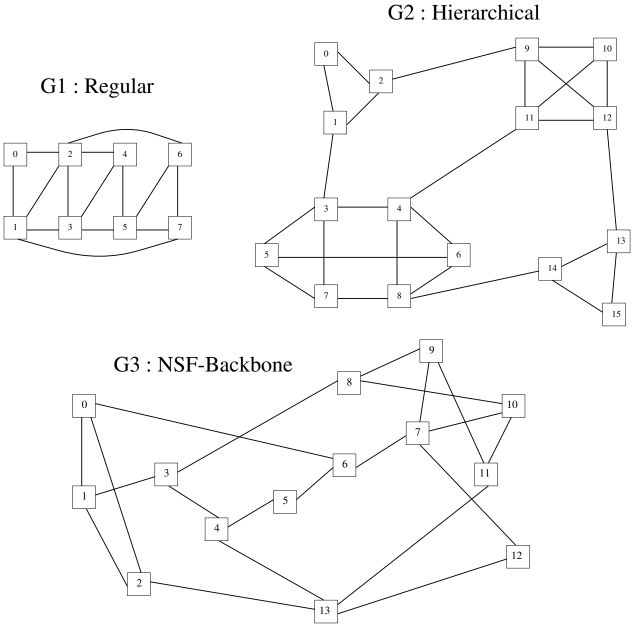


Figure 1. Topologies used by simulator

connection lifetime. For data-driven connections, the network load is measured by the average data injection rate per node, which is the average connection arrival rate multiplied by the average amount of data per connection.

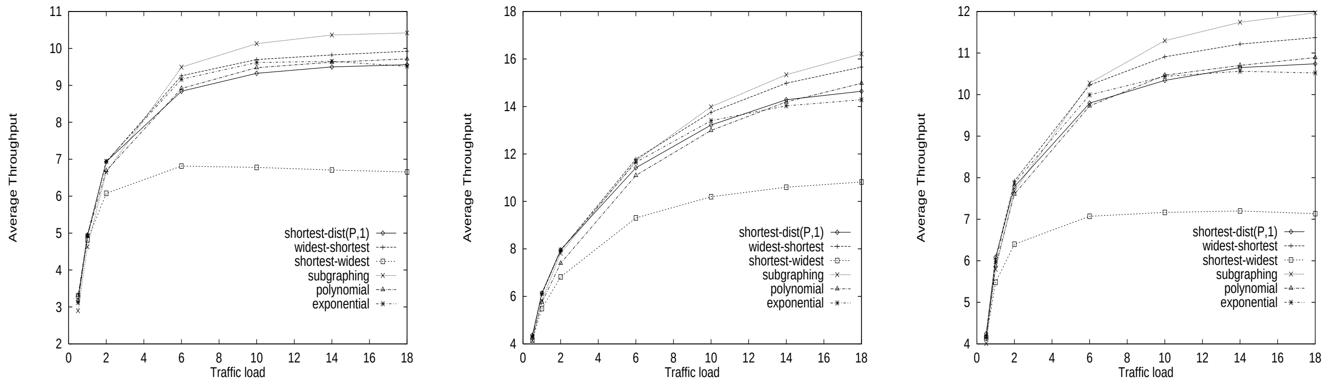
The main performance measures reported by the simulator are the average network throughput, the average per-connection throughput, the standard deviation of the connections' throughput, and the average length of the routes for the connections. The standard deviation of the average bandwidth partially reflects the fairness of the bandwidth distribution among connections. The average length of routes is also provided so that we can determine how the different routing algorithms operate upon different topologies and different traffic loads, resulting in different length routes.

3. Real-time ABR Service

We have compared the performance of the six routing algorithms with different topologies and traffic load distributions. In this section, we report on performance for real-time applications; data-driven applications are covered in the next section.

3.1. Network Throughput

One of the main performance measures is the overall network throughput. The network throughput achieved for the real-time connections for all three topologies is shown



(a) G1 : Regular topology (b) G2 : Hierarchical topology (c) G3 : NSF-backbone

Figure 2. Average network throughput of real-time connections

in Figure 2. The traffic load is the product of the average connection arrival rate and the average connection lifetime.

We first focus on the topology G1. For all routing algorithms, the network throughput initially increases linearly as the network load increases. After a certain threshold value, however, the network starts to be saturated. In a very lightly loaded network, all the routing algorithms show similar results for network throughput, however, the shortest-widest method performs slightly better than the other methods. The shortest-widest method tries to select the widest route, which may utilize more links, but results in better load-balancing.

While load-balancing is good for very lightly loaded networks, it shortly becomes a burden as the traffic load increases. This is strongly related to the length of routes selected by the routing algorithms and the structure of the network. In networks that provide a large pool of alternative paths, a routing algorithm that does load-balancing can get benefits in terms of more bandwidth allocation by bypassing congested links. The benefits, however, lead to longer routes, which means more consumption of network resources. While the larger network resource consumption may not be a problem in lightly loaded networks, it is a problem as the traffic load becomes high and network resources become more scarce. The exponential algorithm shows a slightly earlier saturation point than the others, and starts to degrade after a certain threshold value of traffic load.

The subgraphing algorithm performs best over almost the full range of traffic load. The limited load-balancing results in less consumption of network resources while bypassing congested regions by excluding over-congested links.

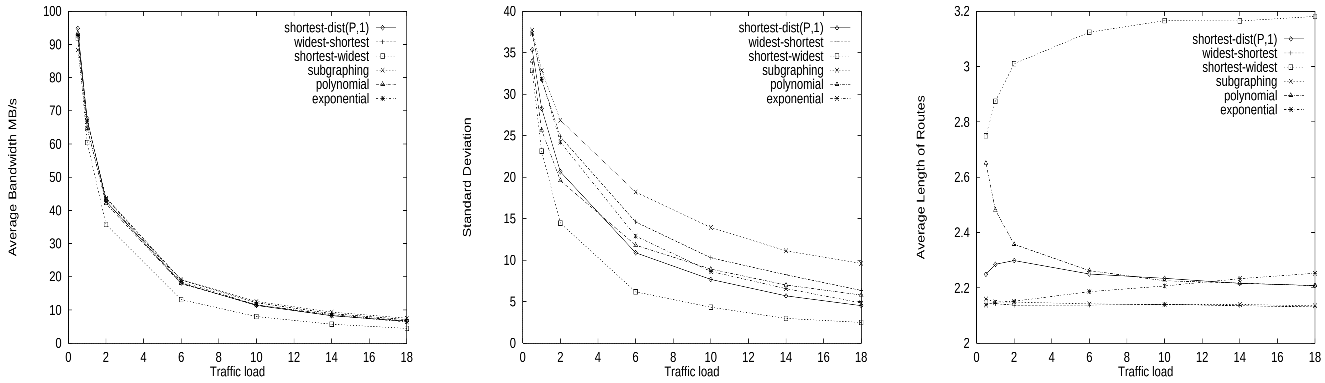
Topology G2 has a hierarchical structure, thus the links between sub-domains are bottlenecks. The hierarchy also limits the number of the alternative routes for a connection, reducing the opportunity for load-balancing. Figure 2(b) shows that even in a lightly loaded network, the shortest-widest algorithm performs poorer than the others. Topology G2 achieves higher average network throughput than the other topologies and saturates at a higher traffic load. This is because the topology G2 has more nodes and links than the other topologies, giving more traffic carrying capability. As in topology G1, the subgraphing algorithm outperforms the others over a wide range of traffic load.

Although topology G3 has more nodes and links than G1, the average network throughput of G3 is similar to that of G1. This is because G3 is less regular and has a lower degree of connectivity than G1. The shortest-widest and the exponential algorithms also show similar phenomena as in topology G1.

In summary, the subgraphing algorithm consistently performs well across the different topologies and the wide range of traffic load. The shortest-widest algorithm tends to perform best under light loads, but worst under moderate to heavy loads.

3.2. Average Throughput

Figure 3 (a) shows the average per-connection throughput for real-time connections for topology G3. The trends are similar to the average network throughput. All routing algorithms give fairly similar performance, although the subgraphing and widest-shortest algorithms achieve slightly higher average throughput, and the shortest-widest algorithm achieves slightly lower average throughput. The difference across the algorithms is less than for the average network throughput. The average network throughput of the subgraphing algorithm is somewhat better than for the other algorithms (Figure 2 (c)), even though it is only slightly better in average per-connection throughput. This is because at high traffic load there are more in-transit connections in the network. Summing of the small average per-connection throughput over a large number of connections gives the larger difference for network throughput. This can be easily verified by noting that the difference of network throughput becomes large as the traffic load becomes high. The results



(a) Average per-connection throughput (b) Standard deviation of bandwidth (c) Average length of routes

Figure 3. G3 NSF-backbone : Performance for real-time connections

for the other topologies are all similar to that for topology G3.

The subgraphing algorithm shows the best performance in terms of the average per-connection throughput over the different topologies and traffic loads. The shortest-widest algorithm shows the worst performance.

3.3. Variance in Bandwidth

We now turn to a measure that reveals greater differences between algorithms and illustrates a tradeoff between connection performance and inter-connection fairness.

Figure 3 (b) shows the standard deviation of the bandwidth allocation for real-time connections for topology G3. As the traffic load becomes high, all links tend to be congested, providing a small range of bandwidth allocations and a decrease in standard deviation. The shortest-widest algorithm consistently shows the best standard deviation results across the different topologies and network loads because of its active load-balancing property. The shortest-dist(P,1) algorithm also shows a good result in the standard deviation metric. The algorithms with very limited load-balancing, such as the subgraphing and the widest-shortest algorithms, show a higher value of standard deviation of bandwidth allocation. The results for the other topologies are all similar to that for topology G3.

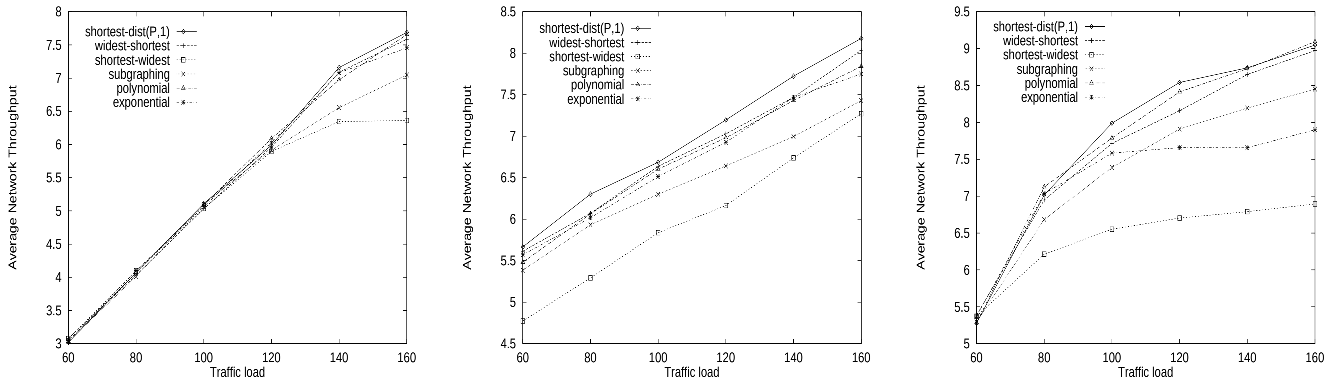
An ideal routing algorithm for ABR connections will simultaneously achieve good network-wide and per-connection throughput and low variance of bandwidth allocation. The standard deviation results, however, show a tradeoff between throughput and variance. Routing algorithms that achieve good throughput results, such as the subgraphing and the widest-shortest algorithms, tend to give higher standard deviation results. The shortest-widest routing algorithm is at the other end of the spectrum. It gives the smallest variance and the smallest average network throughput and average per-connection throughput. The load balancing achieved by a routing algorithm largely determines the position of the routing algorithm on the performance spectrum. Algorithms that achieve a good load balancing tend to have small standard deviation. Algorithms with a little load balancing tend to have high average throughput.

3.4. Route Length

The last metric we consider is the length of the routes. As mentioned earlier, shorter routes may be prefered for real-time applications with jitter and/or interactivity constraints.

Figure 3 (c) shows the average length of routes selected by different routing algorithms for real-time connections for topology G3. The figure shows that route length depends heavily on the particular routing algorithm. The widest-shortest and the subgraphing algorithms avoid longer routes, providing the shortest length routes among all the routing algorithms. Furthermore, these algorithms provide almost a constant length of routes over wide range of network load for all topologies. These characteristics are especially good for delay-sensitive real-time applications since they provide less chance of delay jitter and smaller end-to-end delay. Although the two algorithms achieve similar length routes, the subgraphing algorithm provides better performance over different topologies and traffic loads. Therefore, it has been shown to be very effective for real-time applications.

The shortest-widest algorithm shows the longest average route length. Furthermore, it selects longer routes as traffic load becomes high. The exponential algorithm also selects longer routes as traffic load becomes high, though the increase in length is not significant. The polynomial algorithm prefers longer routes in a lightly loaded network. As the network load becomes high, however, it starts to utilize shorter routes. In lightly loaded network, the shortest-dist(P,1) algorithm initially selects longer routes as the network load increases. As the load exceeds a certain threshold, however, it starts to select shorter routes. The results for the other



(a) G1 : Regular topology (b) G2 : Hierarchical topology (c) G3 : NSF-backbone

Figure 4. Average network throughput of data-driven connections

topologies are all similar to that for topology G3.

In conclusion, an active load-balancing property is not suitable for real-time applications that require short routes. Algorithms that prefer shorter routes at the expense of some load balancing provide better performance results for this type of applications.

4. Data-driven ABR Service

The performance of a given routing algorithm for data-driven applications may be quite different from that for real-time applications. For data-driven applications, the lifetime of a connection is dependent upon both the amount of data to be transfered and the allocated bandwidth to the connection. Therefore, the duration of connection cannot be determined at the arrival instance of the connection and dynamically changes as the network status changes. Further, connections may exist for a very long period if they are consistently allocated low bandwidth. It should be noted that the traffic load for data-driven connections is expressed as the rate of traffic injected into the network at each node.

4.1. Network Throughput

Figure 4 shows the average network throughput for the three different topologies. The figure shows very different performance results compared to the results for real-time connections.

For topology G1, the shortest-dist(P,1) algorithm performs slightly better than the other algorithms. The shortest-widest algorithm shows the same early saturation phenomenon as for real-time connections and performs worst. The subgraphing algorithm, which performs best for real-time connections, does not perform as well in this case. The polynomial algorithm, a counterpart of the shortest-dist(P,1), shows very similar performance to that of shortest-dist(P,1).

The average network throughput for this traffic is largely dependent on the structure of underlying topology. The figure shows quite different behavior for the three topologies. It should be also noted that while topology G2 has provided the highest network throughput at the same traffic load for real-time connections, topology G3 shows the best performance results for data-driven connections. The performance drop for topology G2 comes from its hierarchical structure; the links between sub-domains act as bottlenecks and the connections using these links may have very long lifetimes. These long-lived connections may affect the network throughput adversely. For topology G3, the exponential algorithm shows a saturation feature that does not appear as dramatically in the other topologies. The shortest-dist(P,1) algorithm also performs well for this topology.

For data-driven connections, the shortest-dist(P,1) algorithm consistently performs well over different topologies and different range of traffic load. It also has been proven to be very effective for carrying high-bandwidth applications in max-min fair share networks [9]. The subgraphing method, which performs best for real-time connections, shows relatively poor performance results for data-driven connections. The average per-connection throughput for data-driven applications shows similar results to the average network throughput.

4.2. Variance in Bandwidth

Figure 5 shows the standard deviation results for data-driven ABR connections in topology G3. As in the real-time connections, the standard deviation decreases as the traffic load becomes high. The shortest-widest routing algorithm shows the best results as in the real-time case. The subgraphing algorithm also gives the worst results for medium and high traffic loads.

The data-driven connections, however, give a somewhat different performance spectrum as compared to the real-time

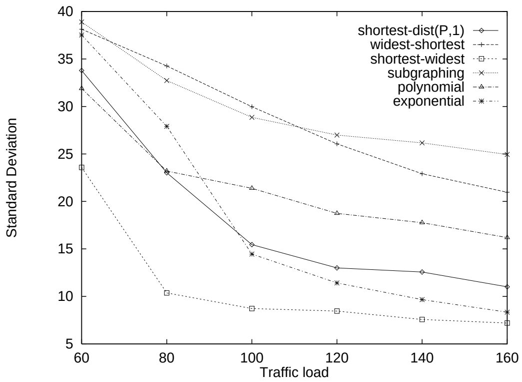


Figure 5. G3: Standard deviation of bandwidth for data-driven connections

connections. Although the subgraphing algorithm shows the highest standard deviation, its average throughput is not the best. The shortest-dist(P,1) gives good performance both in average throughput and in standard deviation. The shortest-widest algorithm gives the best standard deviation result and gives the worst average throughput. The performance results shows that little or excessive load balancing is not good for data-driven connections. The shortest-dist(P,1) algorithm that balances length and bandwidth of routes performs best both in average throughput and in standard deviation. Thus, the performance tradeoff of average throughput with variance is not as clear for data-driven traffic.

4.3. Route Length

Figure 6 shows the average length of routes selected by the different routing algorithms. For topology G1, the shortest-widest algorithm shows a similar result as in real-time connections, utilizing more links as traffic load becomes high and selecting longest routes. The exponential algorithm utilizes more links than in the real-time connections and selects shorter routes as traffic load becomes high. But, as traffic load goes beyond a certain threshold value, it starts to select longer routes. The widest-shortest and the subgraphing algorithms show almost the same shape, utilizing the fewest links and providing almost constant length of routes over the different traffic loads. For the data-driven connections, however, the widest-shortest algorithm uses the network resources more efficiently than the subgraphing algorithm, providing better network throughput while using the same amount of resources. This shows that a routing algorithm can provide different efficiency depending on the type of connections.

For topology G2, the polynomial algorithm utilizes fewer links than for real-time connections. The shortest-widest algorithm selects the longest routes also in this topology. For G2, however, it slightly decreases as traffic load increases unlike for G1. For data-driven connections, the exponential algorithm utilizes more links than for real-time connections under all three topologies. For topology G3, the exponential algorithm shows a sharp increase beyond the traffic load of 80Mbps. The increase is reflected in its saturated network performance in that region (Figure 4(c)).

Routing algorithms show different route selection properties in terms of average length of routes for different types of connections. In addition, although the widest-shortest and the subgraphing algorithms consume the same amount of network resources for both real-time and data-driven connections, they provide very different performance results for each type of connections. For data-driven connections, the widest-shortest algorithm utilizes network resources more efficiently, while the subgraphing algorithm performs better for real-time connections.

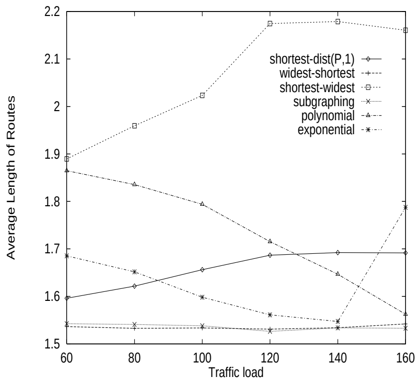
5. Related Work

Most of work on the ABR service class has been devoted to congestion control algorithms [2, 7, 8], as opposed to routing. In this section, we review work on routing algorithms both in max-min fair share networks and in related guaranteed service networks.

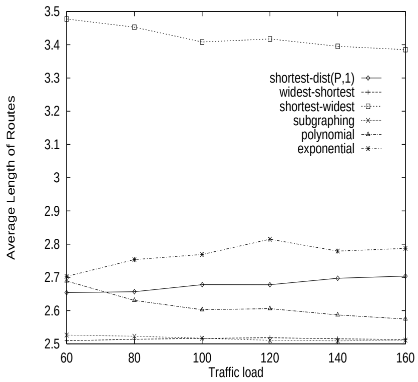
Ma et al. [9] classified best-effort services into three different classes — low-latency traffic, continuous-rate traffic and high-bandwidth traffic — and studied how to use routing to improve the performance of high-bandwidth applications in max-min fair share networks. For high-bandwidth traffic, they proposed the dist(P,n) function as a link-cost metric for shortest-path routing. Their experiments show that shortest-dist(P,1) performs best for high-bandwidth applications over a wide range of situations. Their results focus on performance for high-bandwidth applications; they do not indicate how network throughput is distributed to the other applications and how the network performs overall. In addition to single path routing, they also provide a prioritized multi-path routing algorithm in which low priority paths share the bandwidth left unused by higher priority paths.

Wang and Crowcroft [12] studied routing algorithms for guaranteed QoS services. The shortest-widest algorithm was proposed for a guaranteed bandwidth service. The first "widest" phase of the algorithm searches the widest (maximum) bottleneck bandwidth route, and if there are multiple such routes, the one having minimum delay is chosen at the second "shortest" phase. However, they do not provide any performance results on the shortest-widest routing algorithm.

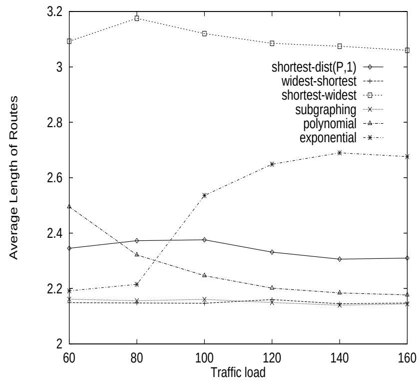
Gawlick et al. [5] studied routing algorithms for permanent virtual circuits on reservation-based networks and used ideas from competitive analysis and combinatorial optimization. An exponential routing algorithm was proposed,



(a) G1 : Regular topology



(b) G2 : Hierarchical topology



(c) G3 : NSF-backbone

Figure 6. Average length of routes selected by different routing algorithms

where the cost of a link is exponential in the bandwidth currently routed on a link. The use of exponential function is based on research that shows that a cost metric based on an exponential function leads to routing algorithms that have an optimal competitive ratio. The exponential routing algorithm was shown to perform best in terms of routed bandwidth over a network. In this paper, we also proposed a similar exponential routing algorithm for ABR service, where the max-min congestion information is used as an exponent instead of the routed bandwidth.

6. Conclusion and Future Work

In this paper, we have evaluated the impact of routing algorithms on the network performance for different types of ABR applications: real-time applications and data-driven applications. For the real-time applications, routing algorithms which prefer shorter routes show better performance results. For data-driven applications, routing algorithms that try to balance allocated bandwidth and length of routes perform better. We have shown that the subgraphing algorithm performs best for real-time applications. It provides the best average network throughput and average per-connection throughput. It also provides the shortest length of routes. The only drawback is a large standard deviation of per-connection rates. For data-driven applications, the shortest-dist(P,1) performs best. The subgraphing routing algorithm, however, does not perform well for the data-driven applications.

Our work can proceed in several directions. There is a need to study the performance with both real-time and data-driven connections. Thus far, we have considered each class separately. We also assumed that routing information is instantly available for all nodes. We can study how delays in routing information forwarding affect the network performance in different routing algorithms.

References

[1] *ATM Forum Traffic Management Specification 4.0.* ATM Forum, 1996.

[2] F. Bonomi and K. W. Fendick. The rate-based flow control frameworks for the available bit rate ATM service. *IEEE Network*, pages 25–39, Mar. 1995.

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext transfer protocol - HTTP/1.1. *RFC2068*, Jan. 1997.

[4] R. Fredrick. *Network Video*[1] *(nv)*. Xerox Palo Alto Research Center.

[5] R. Gawlick, C. Kalmaneck, and K. G. Ramakrishnan. On-line routing for permanent virtual circuits. *Proc. of IEEE INFOCOM'95*, pages 278–288, 1995.

[6] J. M. Jaffe. Bottleneck flow control. *IEEE Trans. Commun.*, 29(7):954–962, July 1981.

[7] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. An efficient rate allocation algorithm for ATM networks providing max-min fairness. *Proc. of 6th IFIP International Conf. on High Performance Networking, HPN '95*, Sept. 1995.

[8] C. Lefelhocz, B. Lyles, S. Shenker, and L. Zhang. Congestion control for best-effort service: Why we need a new paradigm. *IEEE Network*, pages 10–19, Jan. 1996.

[9] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. *ACM SIGCOMM'96*, 1996.

[10] S. McCanne and M. Vetterli. vic[2] : A flexible framework for packet video. *Proc. of ACM Multimedia'95*, Nov. 1995.

[11] J. Postel and J. Reynolds. File transfer protocol (FTP). *RFC959*, Oct. 1985.

[12] Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE JSAC*, 14(7):1228–1234, July 1996.

[1] ftp://ftp.parc.xerox.com/net-research
[2] ftp://ftp.ee.lbl.gov/conferencing/vic