

Multipath Routing for Video Delivery Over Bandwidth-Limited Networks

Jiancong Chen, *Student Member, IEEE*, S.-H. Gary Chan, *Senior Member, IEEE*, and Victor O. K. Li, *Fellow, IEEE*

Abstract—The delivery of quality video service often requires high bandwidth with low delay or cost in network transmission. Current routing protocols such as those used in the Internet are mainly based on the single-path approach (e.g., the shortest-path routing). This approach cannot meet the end-to-end bandwidth requirement when the video is streamed over bandwidth-limited networks. In order to overcome this limitation, we propose multipath routing, where the video takes multiple paths to reach its destination(s), thereby increasing the aggregate throughput. We consider both unicast (point-to-point) and multicast scenarios. For unicast, we present an efficient multipath heuristic (of complexity $O(|V|^3)$), which achieves high bandwidth with low delay. Given a set of path lengths, we then present and prove a simple data scheduling algorithm as implemented at the server, which achieves the theoretical minimum end-to-end delay. For a network with unit-capacity links, the algorithm, when combined with disjoint-path routing, offers an exact and efficient solution to meet a bandwidth requirement with minimum delay. For multicast, we study the construction of multiple trees for layered video to satisfy the user bandwidth requirements. We propose two efficient heuristics on how such trees can be constructed so as to minimize the cost of their aggregation subject to a delay constraint.

Index Terms—Bandwidth-delay constraints, multicast routing, multipath routing, quality-of-service (QoS) routing, video scheduling.

I. INTRODUCTION

THE Internet in recent years has migrated from the best-effort service model to an integrated service network for voice, data, and video applications. As high-bandwidth applications such as video conferencing and movie delivery become increasingly popular, supporting their quality-of-service (QoS) requirements become increasingly important. Several recent advances in QoS architectures such as integrated services (Intserv), differentiated services (Diffserv), and multiprotocol label switching (MPLS) can potentially transform the Internet into an infrastructure supporting QoS guarantees in terms of, for example, bandwidth, delay, or cost. Given a certain set of end-to-end QoS requirements, routing protocols have to be

devised to support it. This is the so-called routing with QoS constraints, or simply QoS routing.

In traditional data networks, routing primarily focuses on connectivity. The network is usually characterized by a single cost metric such as hop-count or delay, and routing tables are built with the objective of minimizing the cost of each path. For some multimedia applications such as video applications, a certain minimum end-to-end bandwidth with bounded start-up delay have to be guaranteed in order to offer high user satisfaction. For example, in an MPEG-I video-on-demand system, a streaming bandwidth of about 1.5 Mb/s has to be guaranteed [1]. In a bandwidth-limited network such as the Internet today, such a high end-to-end bandwidth usually cannot be offered when the traditional single-path approach is used, leading to a degradation in video quality. In order to meet the QoS requirements, a multipath approach can be adopted, where the source (i.e., the server) delivers the data to its destinations via multiple paths, thereby supporting an aggregated transfer rate higher than what is possible with any one path [2], [3]. In this paper, we investigate such routing protocols and their related scheduling issues for video applications, in both unicast and multicast environments.

For unicast, we seek a routing algorithm, which finds a multipath set P with the following properties:

- 1) a certain minimum end-to-end aggregate bandwidth has to be guaranteed;
- 2) the startup delay is minimized over all the feasible paths satisfying 1.

Since single-path routing is a special case of multipath routing, our proposed multipath routing algorithms reduce to shortest-path routing when single-path bandwidth is sufficient.

As an example, we show in Fig. 1 a network with six routers labeled from v_0 to v_6 , interconnected by a number of links. Each link has its link state indicated by the tuple (w, d) , where w is the link bandwidth (e.g., in unit of 100 kb/s) and d is the propagation delay (e.g., in unit of 10 ms). (Note that we do not assume that the network metrics have to be stable during the entire video delivery session. If the metrics change, the algorithm can be rerun to meet the bandwidth requirement.) Suppose that a video stream with a bandwidth requirement of 1.5 Mb/s is to be transmitted from node v_0 to node v_6 . Note that for single-path routing, the maximum bandwidth that can be provided is only 10 units [i.e., 1 Mb/s, along the path $(v_0v_1v_4v_5v_6)$], less than the bandwidth required. On the other hand, when multipath routing is used [indicated by paths $(v_0v_1v_4v_5v_6)$ and $(v_0v_2v_3v_5v_6)$], a high end-to-end bandwidth of 15 units (or 1.5 Mb/s) can be achieved.

Manuscript received September 1, 2003; revised April 15, 2004. This work was supported in part by the Areas of Excellence (AoE) Scheme of the University Grants Council under Grant AoE/E-01/99, and in part by the Competitive Earmarked Research Grant funded by the Research Grants Council under Grant HKUST6156/03E and Grant HKUST6199/02E) in Hong Kong. This paper was presented at Globecom 2001, San Antonio, TX, November 2001.

J. Chen and S.-H. G. Chan are with the Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China (e-mail: kentchen@cs.ust.hk; gchan@cs.ust.hk).

V. O. K. Li is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: vli@eee.hku.hk).

Digital Object Identifier 10.1109/JSAC.2004.836000

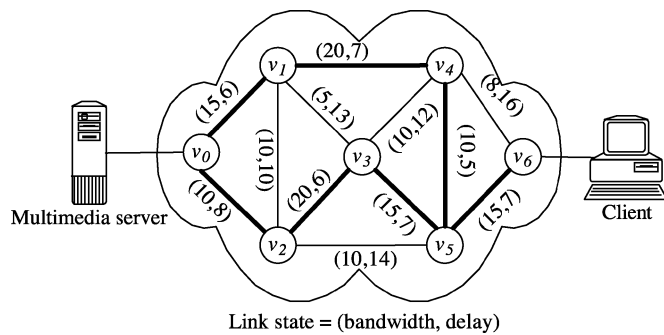


Fig. 1. Multipath routing provides high bandwidth.

Note that since the lengths of the paths are different in multipath routing, the end host needs to do reassembly, hence increasing the end-to-end delay (for the same reason, the buffer requirement at the end-host is also increased). In order to reduce the delay (and, hence, the buffer requirement), we propose a scheduling algorithm for stored video at the source based on the knowledge of the path lengths. In this algorithm, video data are segmented and multiplexed in a specific manner over different paths so that the end-host can assemble the data and play back the video at the earliest time. We prove that our scheduling algorithm achieves the theoretically minimum delay possible, i.e., no other scheduling algorithm can achieve lower delay than ours given the set of path lengths. We show that the delay is very close to the shortest-path algorithm. With the above example, the start-up delay using our video scheduling scheme can be reduced from 280 to 260 ms (the shortest-path delay is 250 ms), and user buffer is reduced from 30 to 10 kbits.

It is difficult to derive a general and exact solution for the multipath problem, therefore, we present in this paper an efficient heuristic, which can be easily applied to today's Internet. The heuristic is shown to achieve high end-to-end bandwidth with a delay close to the shortest path. Under the special case of a network with unit bandwidth in each link, multipath routing reduces to edge-disjoint path routing, where flows do not share links (but can share nodes) [4]. In such a case, with our video scheduling algorithm at the source, an exact and efficient solution (of complexity $O(|V|^2)$, where $|V|$ is the number of routers in the network) can be obtained. Refer to Fig. 1 but with unit link bandwidth, two shortest-disjoint paths $(v_0v_1v_4v_6)$ and $(v_0v_2v_3v_5v_6)$ with delay 260 and 280 ms can be found. By applying the video scheduling scheme discussed in Section III, the user delay can be further reduced from 280 to 270 ms.

When delivering video to a group of users, multicasting is generally employed to achieve bandwidth efficiency. Obviously, the traditional single-tree approach which offers a rate equal to the minimum bandwidth of the links in the tree is no longer sufficient to offer quality video. Therefore, we study the aggregation of multiple trees to achieve it. Different optimization goals can be used in multicast to define what constitutes a good multicast tree. One such goal is to guarantee the minimum delay from the source to the destinations in the tree, which is important for delay-sensitive multimedia applications, such as real-time teleconferencing. Dijkstra's shortest-path algorithm can be used to compute such a shortest-path tree in polynomial time. Another

optimization goal is to minimize the cost of the multicast tree, which is important in managing network resources efficiently. The computation of such minimum cost tree (known as minimum Steiner tree) is NP-hard.

A more interesting direction is to minimize both delay and cost, assuming that delay and cost functions are identical. Schemes have been proposed to balance a minimum spanning tree and a shortest-path tree [5], [6]. Recently, there have been increased interests in good algorithms for computing low-cost delay-constrained multicast trees. Parsa *et al.* proposed an algorithm which refines a shortest-path tree to a low-cost delay-constrained multicast tree by iteratively replacing the most expensive super-edge by a less expensive path, without violating the delay constraints [7], [8].

Furthermore, the users in the multicast environment may even have heterogeneous bandwidth requirements depending on their classes. A natural way to address this problem is to use layered encoding. In such a system, the video is encoded into a base layer and a number of enhancement layers, which are all transmitted along the multicast trees. The users in the network join the base layer and as many enhancement layers as their end-to-end bandwidth require [9]–[12]. In such an environment, the cost of the multicast trees taken becomes an important issue. We investigate, given a network with its link bandwidth, delay and cost, the construction of multicast trees so as to meet the user bandwidth requirements with minimum cost and a delay constraint of the *aggregation*. We formulate the problem and, as the problem is NP-hard, propose and compare two heuristics for tree construction, one based on multiple trees, while the other one based on point-to-point multipath algorithm.

Note that for point-to-point video applications, we have considered bandwidth and delay as our performance parameters as they are in general regarded as important to offer quality service. For multicast applications, we have introduced the cost metric, i.e., we have addressed the problem of finding trees with bandwidth and delay constraints as well as cost optimization. Although our simulation uses an Internet topology, our multipath routing can be applied as well to other networks such as corporate networks, virtual private networks, mobile ad hoc networks and multihop wireless networks, etc., [13]. For example, multipath routing protocols have been proposed in the ad hoc network seeking to support image and video applications [14], [15].

The routing algorithms proposed in this paper have the following properties.

- 1) Efficient—Our algorithms have similar complexity as the shortest-path routing algorithm and exhibit fast route convergence.
- 2) Achieving high bandwidth—The algorithms achieve high end-to-end bandwidth, up to the max-flow of the network.
- 3) Amendable/Implementable—The current Internet routing algorithms can be easily extended to implement our algorithms.

There are, hence, three contributions of this paper.

- 1) We propose an efficient heuristic for point-to-point video transmission to meet user bandwidth requirements by means of multipath routing.

- 2) We present a scheduling algorithm for a stored video sequence, which achieves the theoretical minimum delay given a multipath set. The scheduling algorithm, when combined with disjoint-path routing, provides an exact solution of multipath routing with a network with unit link bandwidth.
- 3) We present multicast heuristics based on multiple trees to meet the user bandwidth requirements with minimum cost and a delay constraint.

Multiple path transport schemes have been proposed in the past for increased connection capacity, as well as for reliability. Much of the previous work on multipath routing focuses on data replication, addressing its reliability issue or path delay issue [16]–[19]. Our approach differs from theirs in considering data *aggregation* (instead of replication) to meet user bandwidth requirements with low user delay. While some other work has considered joint constraints on bandwidth and delay [20]–[23], it focuses on single path instead of multiple paths. The issue of resource allocation in multipath routing has been discussed in [24] and references therein. All these studies consider a point-to-point system (i.e., with a single source and a single destination) without considering meeting the bandwidth requirements of the users, and, hence, cannot be directly extended to the multicast case we consider here. Multipath traffic engineering schemes are also proposed for MPLS networks in which delay is not considered [25]. Packet scheduling is proposed to minimize the re-ordering overhead at the end-host in [26]. Our work differs from [26] in that we focus on the scheduling of video to minimize the start-up delay.

The design of multicast algorithms has been studied extensively. Traditionally, the focus was on how to minimize the cost of the tree [27], [28]. These algorithms are mostly heuristic in nature as the problem is NP-hard. Recently, QoS multicast has been gaining attention, focusing on routing to meet a certain QoS requirement (in terms of, for example, the delay or bandwidth) with minimum cost [7], [8], [23], [29]–[31]. Most of this previous work has focused on building a single multicast tree, which clearly limits the probability of meeting the QoS requirements of all the users. Our work overcomes this limitation by considering multiple multicast trees with minimum aggregated cost.

A moderate amount of recent work has been dedicated to use content distribution networks (CDNs) and overlay networks to support value-added network service [32]–[34]. CDNs are a mechanism to deliver content to end users on behalf of origin servers. Content distribution offloads work from origin servers by serving some or all of the contents [35]. An overlay network is formed by a subset of underlying physical nodes. As the overlay applications are usually built at the application layer, it can effectively use the Internet as a lower level infrastructure to provide higher level services to end users [34]. Our routing scheme can be extended to CDNs and overlay networks if the network topology and delay is known, and node-to-node bandwidth is reserved by the service providers.

Network state information in fact can be measured and exchanged by, e.g., link-state routing protocol. Our algorithms are based on the existence of such information. How to precisely

measure and exchange such information has been extensively discussed in the literature (e.g., see [36]) and is beyond the scope of this paper.

The rest of our paper is organized as follows. We first formulate the multipath routing problem for unicast (point-to-point) transmission and propose a heuristic based on max-flow and shortest-path algorithms in Section II. In Section III, we discuss the theoretical minimum delay achievable given a multipath set and propose a video scheduling algorithm to achieve it. Using the scheduling algorithm, we show in Section III-C an exact and efficient solution of the multipath routing problem with unit link bandwidth. The issue of multicast networks is dealt with in Section IV, and heuristics are proposed to minimize the aggregate tree cost. We conclude in Section V.

II. MULTIPATH HEURISTIC FOR POINT-TO-POINT TRANSMISSION

In this section, we formulate the multipath routing problem for point-to-point (unicast) video transmission and present an efficient heuristic to solve it.

A. Problem Formulation

We model the network as a graph $G = (V, E)$, where V is the set of nodes in the network and E is the set of links or edges. Each link $e = (v_i, v_j) \in E$ has two associated positive metrics.

- The available bandwidth $w(e) \in \mathbb{Z}^+$ in some appropriate unit determined by the granularity of traffic splitting (say, in 100 kb/s).
- The link delay $d(e) \in \mathbb{R}^+$ also in some appropriate unit (say, in 10 ms). In general, the link delay has two components, the queuing delay and propagation delay. Our algorithm is general enough that it is independent of the relationship between link bandwidth and link delay. Although we use delay as our optimization metric in this paper, such metric can also be another additive metric such as loss or jitter.

A (simple and loop-free) path p in G is defined as a list of nodes (v_1, \dots, v_n) such that $\forall i, 1 \leq i < n, (v_i, v_{i+1}) \in E$, and no node appears more than once. Denote by $w(p)$ and $d(p)$ the bandwidth and delay of path p , respectively. Define the bandwidth of a path as the minimum of the residual bandwidth of all links on the path (the so-called “bottleneck bandwidth”). We clearly have

$$w(p) = \min_{i \in \{1, \dots, n-1\}} \{w(v_i, v_{i+1})\} \quad (1)$$

and

$$d(p) = \sum_{i=1}^{n-1} d(v_i, v_{i+1}). \quad (2)$$

Further define the set of multiple paths (termed “multipath set”) $P = \{p_1, p_2, \dots, p_K\}$, where K is the number of paths taken. The aggregate bandwidth W of the K paths is, therefore

$$W(P) = \sum_{i=1}^K w(p_i). \quad (3)$$

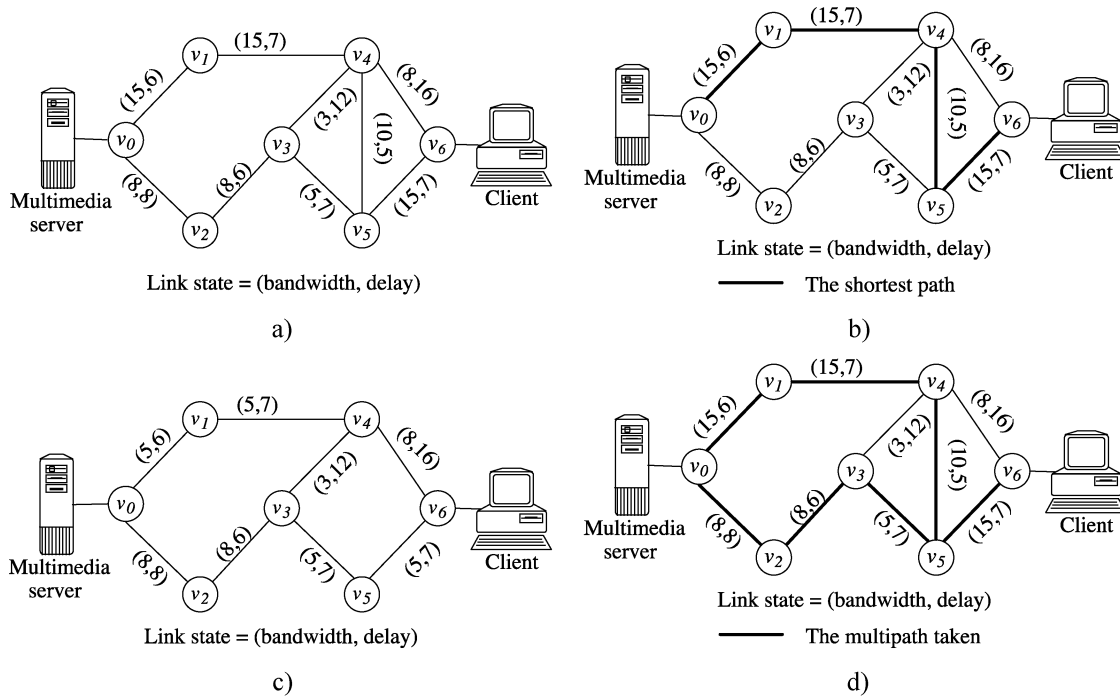


Fig. 2. Example on multipath computation. a) Max-flow graph G' . b) Shortest path p_1 in G' . c) Relabeled network G' . d) Resultant multipath.

The video is streamed via the K paths. If we consider that the end-host plays back the video after the data from the longest path arrive, the start-up delay is then equal to $D(P)$, where

$$D(P) = \max_{p_i \in P} \{d(p_i)\}. \quad (4)$$

Generally, those data received before playback have to be buffered and the corresponding buffer requirement R to buffer the difference in path-lengths is obviously

$$R = \sum_{p_i \in P} w(p_i) (D - d(p_i)). \quad (5)$$

The multipath routing problem can now be presented as follows.

Definition 1: Bandwidth-constrained delay-optimized multipath (BDM) problem: Consider a network represented by a graph $G = (V, E)$ and a bandwidth constraint B , find a multipath set P from the source node s to the destination node t such that:

- 1) $W(P) \geq B$;
- 2) $D(P)$ is minimized over all feasible paths satisfying 1.

In the following, we first present a heuristic to solve the BDM problem, followed by some numerical illustrative results of its performance.

B. Multipath Heuristic

In this section, we present a multipath heuristic given arbitrary bandwidth and delay of each link (the general case). The heuristic is based on max-flow and shortest-path algorithms. First of all, we find, via the preflow-push algorithm, the maximum aggregate bandwidth from the source s to the destination t . If such bandwidth is less than the video bandwidth required, then no route is available to meet the end-to-end

bandwidth requirement. Otherwise, we can use the following multipath heuristic to find the path set P , which meets the bandwidth requirement with low delay.

Algorithm 1 A Multipath Heuristic

- Step 1) Let the multipath set P be empty initially, i.e., $P = \emptyset$.
- Step 2) Run the preflow-push algorithm on network G to compute the max-flow M . We assume in the following that $M \geq B$. Add up all the augmenting paths to form the max-flow graph $G' = (V', E')$.
- Step 3) Take the shortest path p in G' and add p to P

$$P \leftarrow P + \{p\}. \quad (6)$$

- Step 4) Subtract the bandwidth of the shortest path p from the available bandwidth of each link along p in G' , i.e.,

$$w(e') \leftarrow w(e') - w(p), \quad \forall e' \in p. \quad (7)$$

- Step 5) Repeat Steps 3) and 4) until the total bandwidth offered by P is sufficient for the video application, i.e., $\sum_{p_i \in P} w(p_i) \geq B$. \otimes

Note that since each iteration increases the end-to-end bandwidth, the algorithm converges in at most B steps, i.e., $|P| \leq B$.

We analyze the complexity of the heuristic as follows. First of all, observe that the preflow-push algorithm in Step 2) is of complexity $O(|V|^3)$ [37]. Given G' , the shortest-path algorithm in Step 3) is of complexity $O(|V| \log |V| + |E|) \leq O(|V|^2)$. Since it iterates $|P|$ times, where $|P|$ is the number of paths, the complexity of Step 5) is given by $O(|V|^2 \times |P|)$. As a result, the complexity of our heuristic is given by $O(|V|^3 + |V|^2 \times |P|) = O(|V|^3 + |V|^2 \times B) = O(|V|^3)$.

As a simple example, consider the network in Fig. 1 again with a bandwidth requirement B of 15 units (i.e., $B = 1.5$ Mb/s). We obtain the following.

- Step 1) The multipath set P is empty.
- Step 2) The max-flow graph G' of the network in Fig. 1 is shown in Fig. 2(a), where the max-flow $M = 23$ units.
- Step 3) We show the shortest path $p_1 (v_0 v_1 v_4 v_5 v_6)$ obtained from G' in Fig. 2(b).
- Step 4) Subtract the bandwidth of the shortest path from the available bandwidth of each link along path p_1 in G' . The residual graph is then shown in Fig. 2(c).
- Step 5) From Fig. 2(b), the bandwidth offered by p_1 is 10 units, which does not meet the bandwidth requirement. Hence, we repeat Steps 3) and 4) to yield the shortest path $(v_0 v_2 v_3 v_5 v_6)$ in the relabeled G' , which is shown in Fig. 2(c). Since the total bandwidth is equal to B , the algorithm terminates with the resultant multipath shown in Fig. 2(d). The total delay in this case is the longest path in the set P , i.e., 28 units (or 280 ms).

C. Illustrative Simulation Results

To evaluate the performance of the multipath routing over the Internet, we generate the Internet topology by GT-ITM, which models the topological structure of internetworks [38]. There are two hierarchies in the network: AS level and intra-AS topologies. The network generator places a certain number of nodes on a $10\,000 \times 10\,000$ plane. The propagation delay of a link between two connected nodes is defined as the distance (in miles) over light speed. The residual bandwidth of each link is uniformly distributed between 1 and 10 units. The source and destination are randomly chosen and the bandwidth requirements of the users are 5 units.

Using the network model described above, we generate a number of networks with 4000–12 000 routers and compare the performance of the multipath routing with single-path routing in terms of the bandwidth achieved, delay, and the probability of finding sufficient bandwidth for the application (i.e., the success rate). The single-path routing algorithms considered are the shortest-path algorithm and the shortest-feasible path algorithm [23]. For the shortest-feasible path, we first filter out the links with capacity less than the bandwidth requirement, and then run shortest-path algorithm on the residual network.

We compare in Fig. 3, the success rate of the algorithms versus network size (number of routers in the plane). Clearly, the success rate of multipath and shortest-feasible path routing increases with the network size, due to the higher chance of

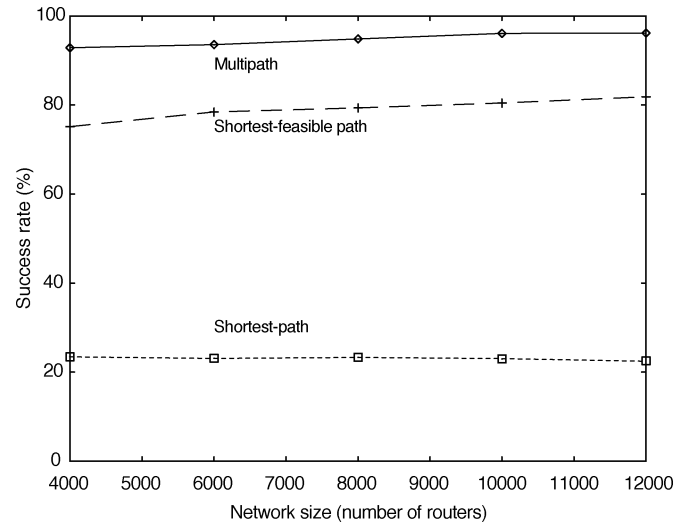


Fig. 3. Success rate versus network size for the multipath and single-path algorithms given N .

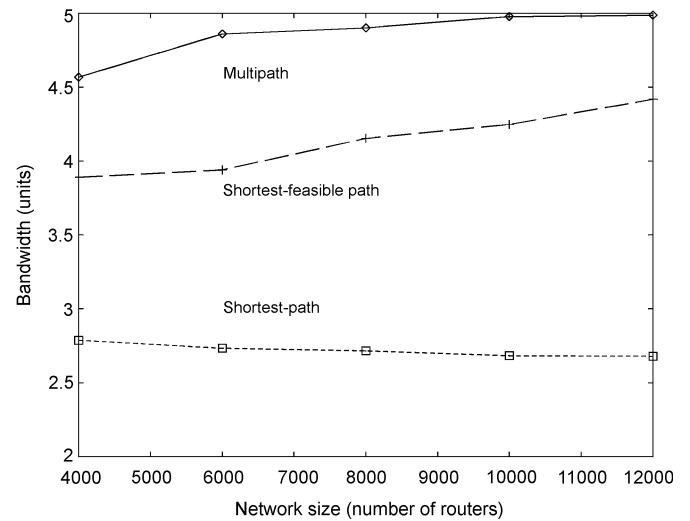


Fig. 4. Comparison of end-to-end bandwidth versus network size between the multipath heuristic, constrained single path, and shortest-path.

finding paths with high bandwidth. On the other hand, the success rate of the shortest path remains more or less the same, because the routing mechanism does not consider the bandwidth issue. It is obvious that shortest-path routing is not appropriate to meet the bandwidth requirement, and the multipath heuristic achieves much higher success than the shortest-feasible path routing. The overall success rate of our heuristic is quite high (more than 90%), showing that it is much more able to find paths than the single-path approach. Actually, given a network and a source-destination pair (s, d) , we can prove that 1) the shortest path has a bandwidth of at least B implies that the shortest-feasible path exists and 2) the existence of a shortest-feasible path implies the success of the heuristic algorithm.

Our multipath heuristic in general achieves much larger bandwidth than the single-path routing algorithms. In Fig. 4, we plot the average bandwidth achieved by the three routing algorithms versus network size. Clearly, the bandwidth obtained by the multipath heuristic is very close to the bandwidth requirement

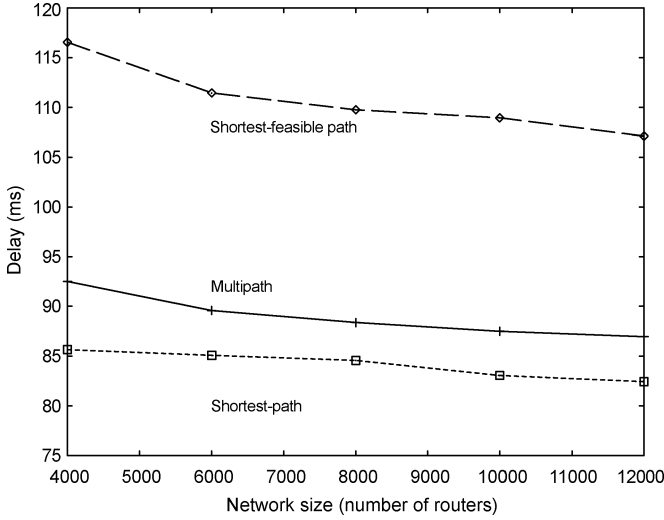


Fig. 5. Comparison of end-to-end delay versus network size for multipath, constrained single path, and the shortest path.

and increases when the network size increases, while that of the single-path approaches remain low. We can prove that the bandwidth of the heuristic algorithm is not less than the bandwidth of the shortest-feasible path, and the bandwidth of the shortest path is less than the bandwidth of the shortest-feasible path.

We show in Fig. 5 the corresponding average end-to-end delay of the algorithms versus network size. Since the plane size remains the same (the Euclidean distance between two nodes is constant), the delay of the shortest-path is more or less constant. On the other hand, the delay of the multipath and shortest-feasible path decreases with network size, mainly because alternate paths with a higher bandwidth are more likely to be found. The shortest-feasible path sacrifices much delay to achieve higher bandwidth, while our multipath heuristic achieves a delay comparable to the shortest path, showing that it does not compromise the high bandwidth attained with delay. For the end-to-end delay, we can easily prove that 1) the shortest path has the minimum delay and 2) when the shortest-feasible path exists, the heuristic algorithm will find the shortest-feasible path, therefore, the delays are the same; when the shortest-feasible path does not exist, the heuristic algorithm may find a set of paths but the delay of the longest path is longer than the delay of the shortest path.

III. VIDEO SCHEDULING ALGORITHM TO ACHIEVE THE MINIMUM DELAY

In multipath routing, the start-up delay of the user can be as long as the longest path. Furthermore, since the end-host has to buffer the data due to the path difference, the buffer requirement is correspondingly increased. We propose a video scheduling algorithm for stored video as implemented at the source given the knowledge of the path-lengths to reduce the start-up delay (and, hence, the buffer requirement). Although the buffer requirement is calculated at the server, such information can be sent to the client along with the data requested so that the receiver knows about it. With our scheduling algorithm, the end-to-end delay is reduced to the theoretical minimum achievable, i.e., no other

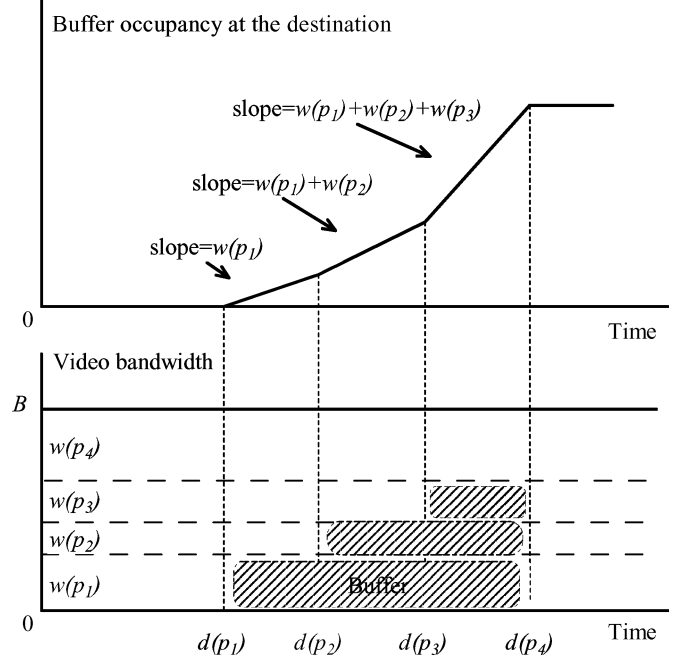


Fig. 6. Buffer occupancy and bandwidth profile versus time at an end-host for multipath routing (the simplest case).

algorithm can achieve a lower delay than it. With this algorithm, we show that an exact and efficient solution for the BDMP problem can be obtained when the link bandwidth is of unit capacity. Before explaining the algorithm, we first discuss the theoretical minimum achievable delay given the set of path lengths.

A. Theoretical Minimum Achievable Delay

We discuss here the theoretical minimum achievable delay given a set of path lengths. In the simplest “naive” case, given K paths $P = \{p_1, p_2, \dots, p_K\}$, where $d(p_1) \leq d(p_2) \leq \dots \leq d(p_K)$, with aggregate bandwidth $B = \sum_{i=1}^K w(p_i)$, the video can be divided into K substreams with bandwidths $w(p_1), w(p_2), \dots, w(p_K)$ and transmitted along the paths p_1, p_2, \dots, p_K , respectively. The end-host does the reassembly and plays back the video. We plot in Fig. 6 the buffer occupancy at the destination versus time, and the received bandwidth versus time for a video stream starting at the source at time 0 ($K = 4$). After a time $d(p_1)$, the first substream arrives with a bandwidth of $w(p_1)$ [and, hence, the buffer increases with slope $w(p_1)$], followed by the second substream with a path length of $d(p_2)$ and bandwidth $w(p_2)$, and so on. The end-host buffers all the data from the paths with shorter delay and, hence, its buffer requirement increases until the last substream arrives (at time $d(p_K)$), when the video is played back. The start-up delay is clearly equal to the delay of the longest path in the multipath set P . It is not difficult to see that the buffer requirement at the end-host is equal to the shaded area, which is given by

$$\begin{aligned} \hat{R} &= \sum_{i=1}^{K-1} w(p_i) (d(p_K) - d(p_i)) \\ &= B \times d(p_K) - \sum_{i=1}^K w(p_i) d(p_i). \end{aligned} \quad (8)$$

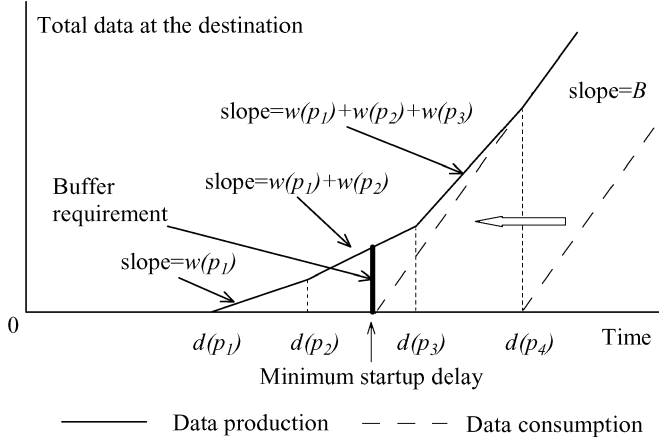


Fig. 7. Data production and consumption versus time at the end-host.

To achieve the theoretical minimum delay, one important observation to make is that, if we can guarantee playback continuity, the start-up delay in fact can be reduced by a time \hat{R}/B . Therefore, the theoretical minimum start-up delay \hat{D} is given by

$$\begin{aligned}\hat{D} &= d(p_K) - \frac{\hat{R}}{B} \\ &= \frac{1}{B} \sum_{i=1}^K w(p_i) d(p_i).\end{aligned}\quad (9)$$

\hat{D} is, hence, the minimum start-up delay achievable without buffer starvation. We illustrate this case in Fig. 7 for $K = 4$, where we show the data production curve (solid line) and consumption curves (dashed lines with slope B) versus time at the end-host (the video is started at time 0). The simplest case corresponds to playback point at time $d(p_4)$, where the buffer requirement stands at its maximum at \hat{R} . When the start-up delay is reduced, the consumption curve moves to the left until it touches the data production curve (as shown by the arrow). Further reduction is not possible without affecting playback continuity. Clearly, such minimum start-up delay is given by the intersection of the consumption curve with the time axis, as shown in (9). Obviously, the buffer requirement at \hat{D} is the bold vertical line, which is the lowest achievable buffer requirement, and the buffer occupancy, given by the difference of the production and consumption curves, is zero at time $d(p_4)$. The buffer requirement at \hat{D} can be obtained by

$$\sum_{i=1}^{K-1} w(p_i) \left(\hat{D} - d(p_i) \right) I \left(\hat{D} - d(p_i) \right) \quad (10)$$

where $I(x)$ is an indicator function where

$$I(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

B. Video Scheduling Algorithm to Achieve the Minimum Delay

In order to achieve \hat{D} , the server has to transmit the video in a specific manner so that the video can be played back at the

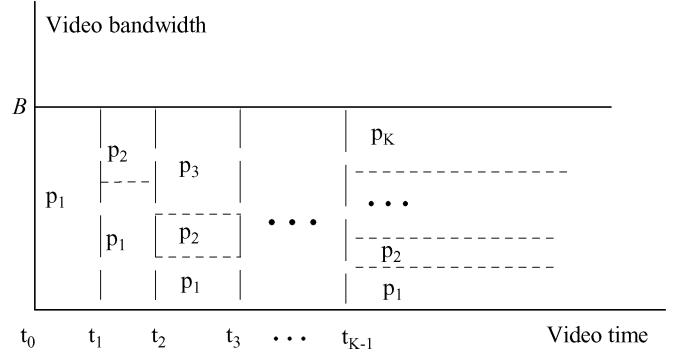


Fig. 8. Video transmission schedule at the source given K .

time \hat{D} , while guaranteeing continuity. The way to do that is to divide the video file into segments and transmit these segments along the paths in such a way so that the end-host does not need to wait for all the substreams to arrive before playback. This is illustrated in Fig. 8, where we show a video sequence being divided into segments in time t_1, t_2, \dots, t_K , where K is the number of paths with $d(p_1) \leq d(p_2) \leq \dots \leq d(p_K)$. Each of the segments is transmitted along the paths as indicated in a multiplexed manner, and data are transmitted in different paths in parallel (i.e., segments of p_1 are transmitted at the same time with the segments of p_2 , even though the segments of p_2 correspond to video sequence t_1 s later). Clearly, this transmission “pushes” the beginning segments of the video into the shorter paths, so that the video can be played back earlier.

To guarantee continuity, t_i 's ($0 \leq i \leq K - 1$) have to be chosen appropriately according to

$$t_i = \frac{\sum_{l=1}^i w(p_l) (d(p_{l+1}) - d(p_l))}{B} \quad (12)$$

for $i = 1, 2, \dots, K - 1$, and $t_0 = 0$.

The scheduling algorithm can then be stated as follows.

Algorithm 2 A Video Scheduling Algorithm

- Step 1) Segment the video from the beginning according to (12);
- Step 2) For $i = 1, 2, \dots, K - 1$, multiplex video segment (t_{i-1}, t_i) into i paths given by p_1, p_2, \dots, p_i according to the bandwidth ratio $w(p_1) : w(p_2) : \dots : w(p_i)$, i.e., segment (t_0, t_1) is transmitted along path p_1 , and segment (t_1, t_2) is transmitted along paths p_1 and p_2 in the ratio of $w(p_1) : w(p_2)$, and so on. The segment after t_{K-1} is multiplexed along the K paths with bandwidth ratio $w(p_1) : w(p_2) : \dots : w(p_K)$. \otimes

In the following, we prove that using our scheduling algorithm, data of the same segment transmitted along different paths arrives at the end-host at the same time, so that the video can be played back with continuity. Consider the segment

(t_{i-1}, t_i) , where $i = 1, 2, \dots, K - 1$. Since it is transmitted along i paths, the transmission time τ_i for it is given by

$$\begin{aligned} \tau_i &= \frac{(t_i - t_{i-1}) \times B}{\sum_{l=1}^i w(p_l)} \\ &= d(p_i) - d(p_{i-1}). \end{aligned} \quad (13)$$

The portion of the segment transmitted along path p_m ($1 \leq m \leq i$) arrives at the end-host at time (taking into account all the previous data transmitted along the path)

$$\begin{aligned} \sum_{l=m}^{i-1} \tau_l + d(p_m) &= (d(p_i) - d(p_m)) + d(p_m) \\ &= d(p_i). \end{aligned} \quad (14)$$

Therefore, the arrival time for the video segment (t_{i-1}, t_i) is $d(p_i)$. Since the end-host gets the prefix of the video earlier, it does not need to wait till $d(p_K)$ but can start the playback at \hat{D} with continuity.

C. Exact Solution for Networks With Unit Link Capacity

We show in this section that, using the scheduling algorithm and under the case that each link in the network is of unit capacity, an exact and efficient solution for the BDM problem can be obtained. Clearly, if the bandwidth requirement is k units, k paths is needed to deliver the video. Since the links are of unit bandwidth, the paths found are disjoint. Therefore, the BDM problem is reduced to finding a set of edge-disjoint paths with minimum start-up delay. We show that, with our video scheduling algorithm, the k -shortest-disjoint paths (k -SDP) is the optimal solution for the BDM problem (k -shortest-disjoint paths are k disjoint paths with the shortest total path length). The algorithm of finding the k -shortest-disjoint paths is well known. Interested readers may refer to [19] for more detail.

Theorem 1: The k -shortest-disjoint paths with video scheduling minimize the start-up delay.

Proof: Consider a network G and k -shortest-disjoint paths $P = \{p_1, p_2, \dots, p_k\}$. For any other k -disjoint paths $P' = \{p'_1, p'_2, \dots, p'_k\}$ in G , we have $\sum_{i=1}^k d(p_i) \leq \sum_{i=1}^k d(p'_i)$. The start-up delay of the k -shortest-disjoint paths with video scheduling is $\sum_{i=1}^k d(p_i)/k$ [see (9)]. Therefore, we have

$$\frac{\sum_{i=1}^k d(p_i)}{k} \leq \frac{\sum_{i=1}^k d(p'_i)}{k} \leq D(P') \quad (15)$$

i.e., the start-up delay of the k -shortest-disjoint paths with video scheduling is never larger than that of the other path sets with or without video scheduling. This completes the proof.

Using the aforementioned network simulation model with the same parameters (except for unit-capacity links), we compare the delay of disjoint path routing with shortest-path routing. We have used the bandwidth requirements of 2–4 units, i.e., 2–4 disjoint paths are needed, respectively. We compare in Fig. 9, the end-to-end delay versus network size of the SDP without scheduling (and, hence, the delay of the longest path), SDP with video scheduling, and the shortest-path algorithm. Clearly, the delay

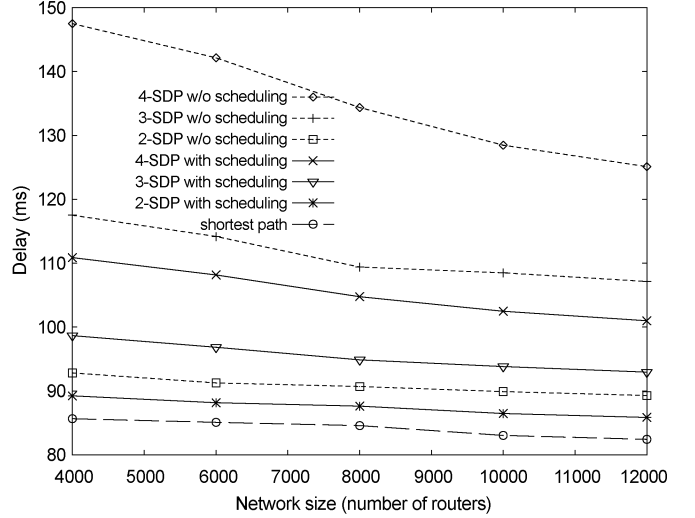


Fig. 9. Comparison of end-to-end delay versus network size between SDP and shortest-path routing.

of the shortest-path algorithm does not change much with respect to the network size. When the number of paths required increases, the delay of SDP increases accordingly since the additional path is always longer than the previous selected path. However, the scheduling algorithm reduces the start-up delay of SDP to be comparable to that of the shortest-path. The SDP algorithm with video scheduling, therefore, provides a substantially larger bandwidth at the cost of slightly higher delay. The trend of the success rate of SDP is very similar to that of multipath (Fig. 3) and, hence, would not be reproduced here. One point to note, though, is that given the same number of paths, the success rate of SDP is slightly lower because of its stronger “disjoint” constraints.

IV. MULTICAST EXTENSION OF MULTIPATH ROUTING

In this section, we consider the use of multiple trees to meet user bandwidth requirements. In multicast, the management or monetary cost of the trees is usually an important concern, therefore, we seek to minimize the cost of the tree-aggregate (which is essentially an irregular “mesh”) subject to a delay constraint. We propose two heuristics to address the problem, one of which is based on the point-to-point multipath heuristic. Both heuristics are shown to achieve much better QoS with slightly higher cost than the single-tree approach.

A. Problem Formulation

Our video multicast problem can be formulated as follows. Given a network $G = (V, E)$, with each link characterized by its bandwidth $w(e) \in Z^+$, delay $d(e) \in \mathbb{R}^+$, and a (positive) cost $c(e) \in \mathbb{R}^+$, where $e \in E$. A source s in the network transmits video to m users given by the set $Y = \{y_1, y_2, \dots, y_m\}$ with corresponding bandwidth requirements $B = \{b_1, b_2, \dots, b_m\}$ units, where $b_i \in Z^+$. Clearly, each b_i is upper bounded by the total video bandwidth $B_0 \in Z^+$.

Let L be the number of trees constructed to meet the bandwidth requirements of the users, and $T = \{t_1, t_2, \dots, t_L\}$ be the multiple multicast trees constructed. Further denote $M =$

$t_1 \cup t_2 \cup \dots \cup t_L$ as the union of the trees, or simply the “tree-aggregate.” Then, the cost of T is given by

$$c(T) = \sum_{e \in M} c(e). \quad (16)$$

The bandwidth of a tree t_l is the bandwidth of the bottleneck link, i.e.,

$$w(t_l) = \min_{e \in t_l} \{w(e)\}. \quad (17)$$

The aggregated bandwidth provided to a user y_i is the sum of the bandwidth of all paths from s to it, i.e.,

$$w(T : s \rightarrow y_i) = \sum_{l=1}^L w(t_l : s \rightarrow y_i) \quad (18)$$

$$= \sum_{l=1}^L w(p_{\in t_l} : s \rightarrow y_i). \quad (19)$$

The delay of a tree-aggregate T is the longest delay from the source s to the users in T , i.e.,

$$d(T) = \max_{i=1 \dots m} d(p_{\in T} : s \rightarrow y_i). \quad (20)$$

The multipath routing problem for video multicast is, therefore, posed as follows.

Definition 3: Video multipath multicast (VMM) problem:

Find: $T = \{t_1, t_2, \dots, t_L\}$;

To minimize: $c(T)$;

Subject to: $w(T : s \rightarrow y_i) \geq b_i$ and a delay bound $d(T) \leq D$.

Note that since the VMM problem is an extension of the delay-constrained minimum Steiner tree problem, it is NP-hard. We, hence, propose the following two efficient heuristics (of complexity $O(m|V|^3)$ and $O(mB_0|V|^2)$, respectively).

B. Multicast Heuristics

The first heuristic is based on the point-to-point multipath algorithm. It starts with constructing individual multipaths from the source to each destination in the multicast group to meet the bandwidth requirement subject to the delay constraint. Based on the sets of multipaths obtained, the algorithm then constructs multiple multicast trees for the video.

Algorithm 3 Multipath Extension

- Step 1) For $i = 1$ to m , find the shortest (in terms of cost) multipath set P_i with bandwidth requirement b_i for each destination $y_i \in Y$ by the multipath heuristic;
- Step 2) Check the delay of each path p in path set P_i , $i = 1 \dots m$. If $d(p)$ violates the delay constraint D , replace it by the shortest path in terms of delay with bandwidth no less than $w(p)$;
- Step 3) Initially, let $T = P_1$. For $i = 2$ to m , add multipath set P_i to T as follows:
 - i) Find a path $p_{ij} \in P_i$ such that

in T , it intersects a tree not covering y_i with the most links. Call this tree \hat{t} . If all the trees in T cover y_i , then add the rest of the paths in P_i to T as new trees and end;

- ii) If $w(p_{ij}) \geq w(\hat{t})$, then add p_{ij} to \hat{t} to form a new tree and subtract $w(\hat{t})$ from the bandwidth of the path p_{ij} , i.e., $w(p_{ij}) \leftarrow w(p_{ij}) - w(\hat{t})$;
- iii) Otherwise, break \hat{t} into two trees with bandwidth $w(p_{ij})$ and $w(\hat{t}) - w(p_{ij})$. Add p_{ij} to the new tree with bandwidth $w(p_{ij})$. Remove p_{ij} from P_i ;
- iv) Repeat i) to iii) until $P_i = \emptyset$.

In Algorithm 3, Steps 1) and 2) correspond to the low cost delay-constrained multipath construction. We show in Fig. 10(a) an example to demonstrate the tree construction procedures of Step 3). There are two path sets P_1 and P_2 from the source s to the destinations y_1 and y_2 , where $P_1 = \{p_{11}, p_{12}\} = \{s \rightarrow y_1, s \rightarrow v \rightarrow y_1\}$ with bandwidth 3 and 2 units as labeled, and $P_2 = \{p_{21}, p_{22}\} = \{s \rightarrow v \rightarrow y_2, s \rightarrow y_2\}$ with bandwidth 3 and 2 units, respectively. In Fig. 10(b), we show the tree construction procedure by the resultant T and P_2 after each repetition in Step 2).

- 1) Initially, T is P_1 and P_2 is not changed.
- 2) Path p_{21} intersects t_2 with the most links and its bandwidth is larger than that of the tree. According to Step 2ii), add p_{21} to t_2 and subtract the bandwidth of t_2 from p_{21} .
- 3) There is only t_1 which does not cover y_2 . The two paths in P_2 do not intersect it. Therefore, add any one of them to t_1 . Here p_{21} is added to t_1 . Its bandwidth is smaller than that of t_1 . According to Step 2iii), break t_1 into two new trees with bandwidths 1 and 2 units and add p_{12} to the new tree with a bandwidth of 1 unit and remove it from P_2 .
- 4) Add path p_{22} to t_1 and remove it. Then, P_2 is empty. This ends the algorithm.

Note that since the multipath algorithm is of complexity $O(|V|^3)$, the multipath construction cost for m users is then $O(m|V|^3)$. Clearly, the cost of multicast tree construction is $O(mB_0^3)$ because the number of paths and trees are both bounded by B_0 . Therefore, the complexity of the heuristic is $O(m|V|^3 + mB_0^3)$. However, B_0 is usually a very small number such as 5 such that $|V|$ is much larger than B_0 . Therefore, the complexity of the heuristic is $O(m|V|^3)$.

Clearly, the heuristic tries to meet the bandwidth requirements of the clients first before minimizing the cost. Therefore, it achieves good success rate at the expense of some increase in cost. We propose in the following yet another heuristic based on the idea of constructing multiple low-cost delay-constrained trees successively in order to meet the bandwidth requirements:

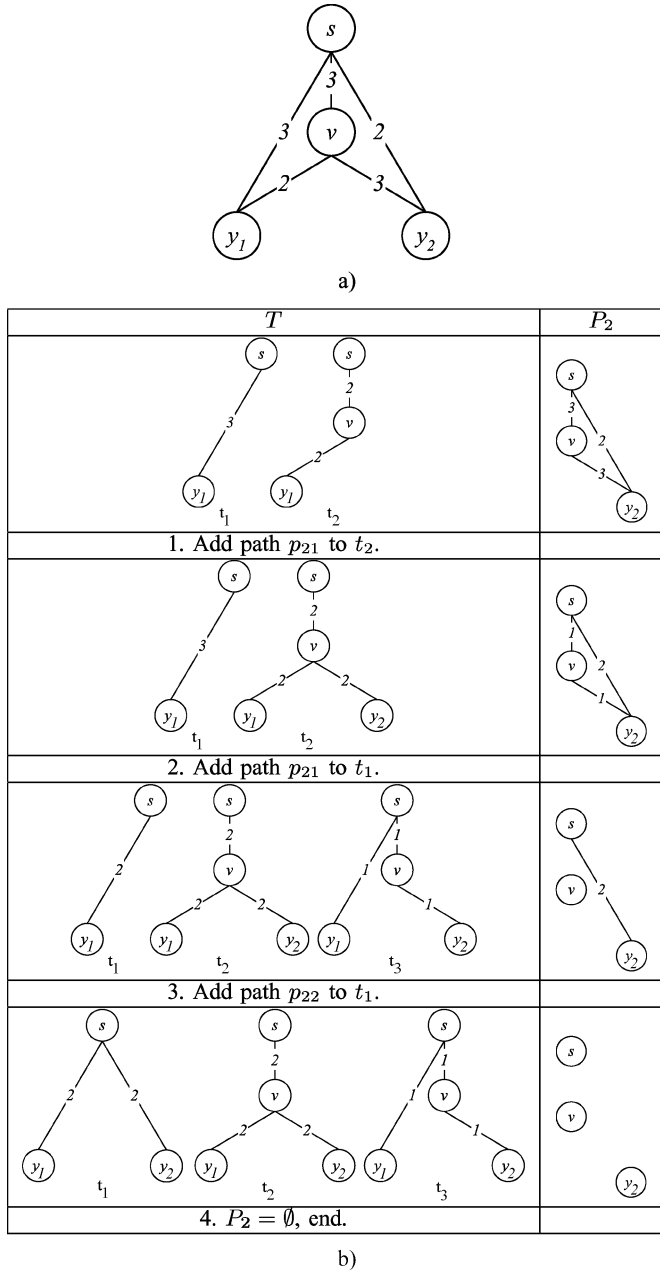


Fig. 10. Demonstration of the multipath extension algorithm. a) Network with multiple paths from s to y_1 and y_2 . b) Procedures of the multipath extension algorithm.

Algorithm 4 Low-Cost Delay-Constrained Tree Extension

- Step 1) Initially, let $T = \emptyset$ and $l = 1$;
- Step 2) Find a minimum-cost tree t_l spanning all the nodes in the given network G ;
- Step 3) Check the delay leading to each destination y in tree t_l . If the delay violates the delay constraint, replace the path in t_l by the shortest path from s to y .
- Step 4) Let $T = T \cup \{t_l\}$. If T meets all bandwidth requirements (i.e.,

$w(T : s \rightarrow y_i) \geq b_i, \forall i \in \{1, 2, \dots, m\}$), then end;

- Step 5) Subtract $w(t_l)$ from $w(e)$ for any link $e \in t_l$ (i.e., $w(e) = w(e) - w(t_l), \forall e \in t_l$) to form a residual network G' and remove the links with zero residual bandwidth from t_l . Let the new partial tree rooted at the source s be t'_l ;
- Step 6) Let $l = l + 1$. Add destinations with insufficient bandwidth to the partial tree t'_l by the nearest destination first (NDF) algorithm to form a new tree t_l [39]. Goto Step 3).

Algorithm 4 takes advantage of the constructed partial tree and new branches after each iteration, increasing the end-to-end bandwidth of some clients. Every new tree formed this way can be a layer tree. The cost of the NDF algorithm is $O(m|V|^2)$ [39]. Since there are at most B_0 trees to be constructed, the complexity of the algorithm is $O(mB_0|V|^2)$.

After constructing the multicast trees, if the users have heterogeneous bandwidth requirements, the video can be encoded into layers and transmitted along the multiple trees according to Algorithm 5. Otherwise, the video is multiplexed along the trees and reassembled at the clients.

Algorithm 5 Bandwidth Assignment of Layers

- Step 1) Group the trees spanning the same set of users;
- Step 2) Arrange these groups according to decreasing number of users covered. Note that for our tree construction, the previous set of users is the superset of the latter;
- Step 3) The aggregate bandwidth of the first tree-group is the bandwidth of the base layer. The aggregate bandwidth of the second group is the bandwidth of the enhancement layer 1, and so on. \otimes

In our simulations, we typically construct B_0 multicast trees in order to meet the bandwidth requirement. Clearly, the number of video layers is no more than B_0 . If there are trees covering the same set of users, we merge them to serve the same video layer such that the number of video layers is reduced.

C. Illustrative Simulation Results

In this section, we show some illustrative numerical results to compare the two heuristics and the single-tree approach based on the low-cost delay-constrained tree in terms of bandwidth achieved, total tree cost and the success rate (i.e., the probability of meeting all the requests' requirements). We also use GT-ITM to generate routers and links on a 10000×10000 plane. The

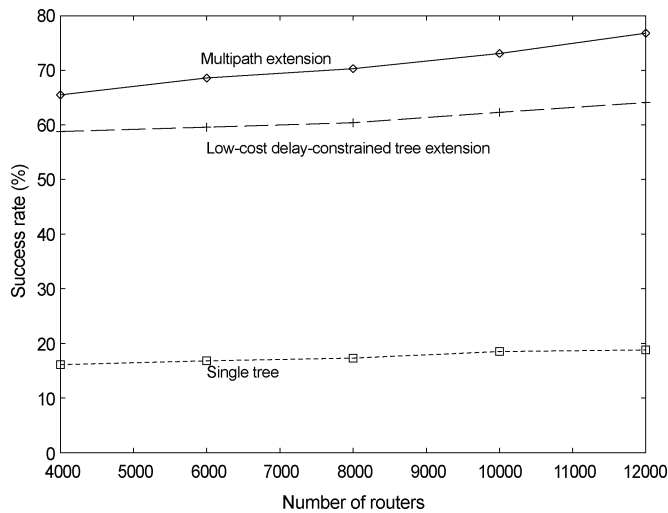


Fig. 11. Success rate versus the number of routers for the heuristics studied.

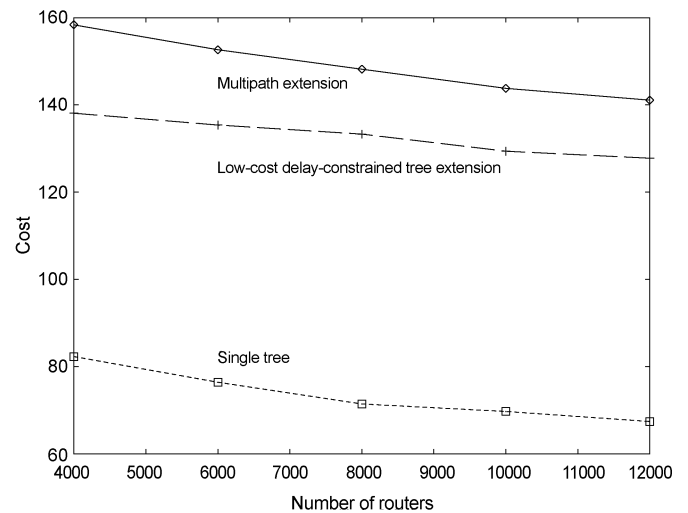


Fig. 13. Tree cost achieved by the two heuristics as compared with the single-tree approach.

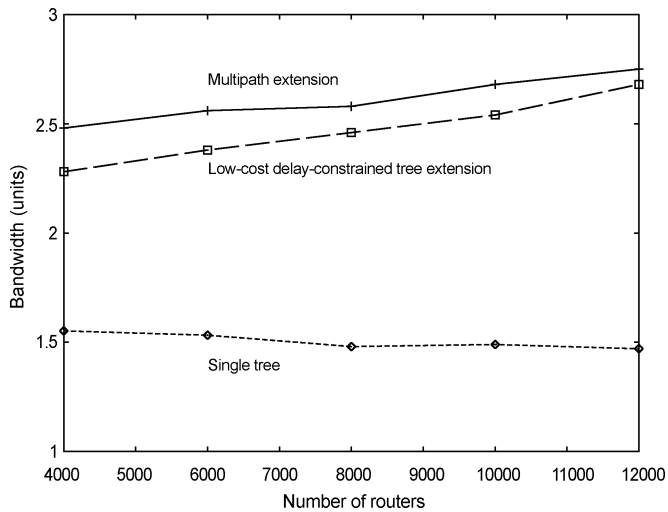


Fig. 12. Average bandwidth achieved versus the number of routers for the two multicast heuristics and the single-tree approach.

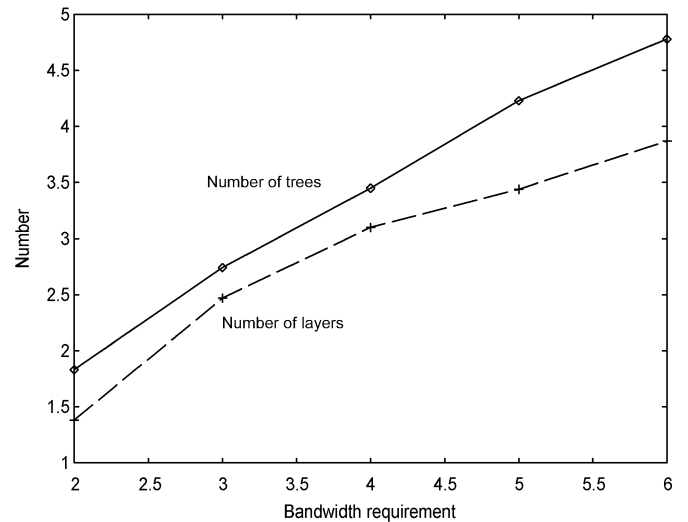


Fig. 14. The difference of the average number of trees constructed and video layers.

delay and bandwidth definition of a link are defined as aforementioned. The cost of the links is a uniform distribution in (0,1). The source and m users are randomly chosen to form a multicast group. We have used $m = 40$, a delay constraint of $D = 100$ ms and the bandwidth requirements of the users are 3 units.

Fig. 11 plots the corresponding success rate of the three algorithms to meet the bandwidth and delay requirements in different network sizes represented by the number of routers in the plane. The multipath extension is better than the aggregate trees. Both of them achieve substantially higher success rate than the single-tree approach, which fails to meet the bandwidth requirements in 80% of the cases considered.

In Fig. 12, we plot the average bandwidth achieved by the three tree construction algorithms versus network size. Clearly, our heuristics achieve much larger bandwidth than the single-tree approach. The bandwidth obtained by the multipath extension is close to the bandwidth requirement and increases with the number of routers because of higher connectivity.

We show in Fig. 13 the average cost of the aggregated trees of the three algorithms. Clearly, the cost of the single-tree approach is the lowest. As expected, the cost of the aggregated trees is lower than that of the multipath extension.

After the multicast trees are constructed, those who cover the same users are merged together to transmit the same video layer so that the number of layers can be reduced. We vary the bandwidth requirement of the users from two to six to see the reduction between the number of trees and layers. Fig. 14 shows the average number of trees constructed and video layers in the above simulations. Clearly, the number of layers is reduced significantly due to the merge of trees. When the number of trees increases, the trees are more likely to span the same users and to be merged. The difference is, hence, enlarged.

Our results show that the multiple-tree heuristics achieve substantially better performance than the single-tree approach. Multipath extension performs better in terms of success rate and bandwidth achieved, while low-cost delay-constrained tree

extension achieves lower cost. Therefore, there is a tradeoff between high bandwidth and low cost in this multicast setting.

V. CONCLUSION

In order to offer high video quality, a certain minimum end-to-end bandwidth has to be provided. In bandwidth-limited networks, it is difficult to guarantee such a bandwidth with a single path. To meet the bandwidth requirement, a multipath approach can be used, where the video data is transmitted over different paths and reassembled at the end-host(s). Such multipath algorithms have to be designed with low network delay or cost.

The multipath problem can be formulated as finding multiple paths to meet a certain aggregate bandwidth requirement with minimum delay. For the point-to-point case, under the general case where bandwidth and delay of the links are arbitrary, we present an efficient heuristic for the problem based on max-flow and shortest-path algorithms. The complexity of the algorithm is only $O(|V|^3)$, where $|V|$ is the number of nodes. The heuristic is shown to offer sufficient bandwidth with much higher probability as compared with the single-path approaches such as the shortest path and shortest-feasible path. The delay of our heuristic is also close to that of the shortest path.

Given a set of path-lengths, we also propose a video scheduling algorithm at the server and show that it achieves the theoretical minimum delay (i.e., no other algorithms can achieve a lower delay than ours). In the algorithm, the source partitions the video into segments and transmits different segments along different paths in a manner such that the end-host can play back the video at the earliest time with continuity. Our results show that such algorithm is able to reduce the delay of the multipath scheme to be close to that of the shortest-path algorithm. With this algorithm, we show that the multipath problem has an exact and efficient solution when each link in the network is of unit bandwidth.

To meet the bandwidth requirements of clients in multicast networks, we consider constructing multiple trees with low aggregated cost and delay constraint. We propose two heuristics to address the problem, one of which is a multicast extension of our point-to-point multipath algorithm, while the other is based on the NDF algorithm. Simulation results show that both heuristics, as compared with the single-tree approach, achieve higher bandwidth with some increase in cost.

REFERENCES

- [1] D. Saporilla and K. W. Ross, "Optimal streaming of layered video," in *Proc. IEEE INFOCOM*, Mar. 2000, pp. 737–746.
- [2] H. Suzuki and F. A. Tobagi, "Fast bandwidth reservation scheme with multilink and multipath routing in ATM networks," in *Proc. IEEE INFOCOM*, Mar. 1992, pp. 2233–2240.
- [3] J. Chen and S.-H. G. Chan, "Multipath routing for video unicast over bandwidth-limited networks," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 25–29, 2001, pp. 1963–1997.
- [4] R. G. Ogier, V. Rutenburg, and N. Shacham, "Distributed algorithms for computing shortest pairs of disjoint paths," *IEEE Trans. Inform. Theory*, vol. 39, pp. 443–455, Mar. 1993.
- [5] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. COM-31, pp. 343–351, 1983.
- [6] S. Khuller, B. Raghavachari, and N. Young, "Balancing the minimum spanning and shortest path trees," in *Proc. ACM/SIAM SODA*, 1993, pp. 243–250.
- [7] Q. Zhu, M. Parsa, and J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting," in *Proc. IEEE INFOCOM*, Los Alamitos, CA, 1995, pp. 377–385.
- [8] M. Parsa, Q. Zhu, and J. Garcia-Luna-Aceves, "An iterative algorithm for delay-constrained minimum-cost multicasting," *IEEE Trans. Commun.*, vol. 6, pp. 461–474, Aug. 1998.
- [9] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 983–1001, Aug. 1997.
- [10] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming video over the Internet: approaches and directions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 282–300, Mar. 2001.
- [11] T.-W. A. Lee, S.-H. G. Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Optimal allocation of packet-level and byte-level FEC in video multicasting over wired and wireless networks," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Nov. 25–29, 2001, pp. 1994–1998.
- [12] —, "Allocation of layer bandwidth and fec for video multicast over wired and wireless networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1059–1070, Dec. 2002.
- [13] R. Krishnan and J. Silvester, "Choice of allocation granularity in multipath source routing schemes," in *Proc. IEEE INFOCOM*, vol. 1, 1993, pp. 322–329.
- [14] N. Gogate, D.-M. Chung, S. Panwar, and Y. Wang, "Supporting image and video applications in a multihop radio environment using path diversity and multiple description coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 777–792, Sept. 2002.
- [15] L. Zhang, Z. Zhao, Y. Shu, L. Wang, and O. Yang, "Load balancing of multipath source routing in ad hoc networks," in *Proc. IEEE Int. Conf. Communications*, vol. 5, 2002, pp. 3197–3201.
- [16] S. Vutukury and J. Garcia-Luna-Aceves, "A simple approximation to minimum-delay routing," *Comput. Commun. Rev.*, vol. 5, no. 1, pp. 227–238, Feb. 1999.
- [17] J. Garcia-Luna-Aceves, S. Vutukury, and W. T. Zaymen, "A practical approach to minimizing delays in internet routing," in *Proc. IEEE Int. Conf. Communications*, Piscataway, NJ, 1999, pp. 479–483.
- [18] S. Vutukury and J. Garcia-Luna-Aceves, "A distributed algorithm for multipath computation," in *Proc. Global Telecommunications Conf.*, 1999, pp. 1689–1693.
- [19] B. Ramesh, *Survivable Networks: Algorithms for Diverse Routing*. Norwell, MA: Kluwer, 1999.
- [20] Z. Wang and J. Crowcroft, "Bandwidth-delay based routing algorithms," in *Proc. GLOBECOM*, Nov. 1995, pp. 2129–2133.
- [21] W. Zhao, T. Seth, M. Kim, and M. Willebeek-LeMair, "Optimal bandwidth/delay tradeoff for feasible-region-based scalable multimedia scheduling," in *Proc. IEEE INFOCOM*, New York, Mar. 1998, pp. 1131–1138.
- [22] T. Kormaz and M. Krunz, "A randomized algorithm for finding a path subject to multiple QoS constraints," in *Proc. IEEE GLOBECOM*, 1999, pp. 1694–1698.
- [23] Z. Wang and J. Crowcroft, "QoS routing for supporting multimedia applications," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1188–1234, Sept. 1996.
- [24] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 885–896, Dec. 1999.
- [25] Y. Lee, Y. Seok, Y. Choi, and C. Kim, "A constrained multipath traffic engineering scheme for MPLS networks," in *Proc. IEEE Int. Conf. Communications*, New York, May 2002, pp. 2431–2436.
- [26] K.-C. Leung and V. O. K. Li, "Flow assignment and packet scheduling for multipath routing," *IEEE/KICS J. Commun. Networks*, vol. 5, pp. 230–239, Sept. 2003.
- [27] K. Carlberg and J. Crowcroft, "Building shared trees using a one-to-many joining mechanism," *Comput. Commun. Rev.*, vol. 27, no. 1, pp. 5–11, Jan. 1997.
- [28] K.-S. Lui, K. Nahrstedt, and S. Chen, "Hierarchical QoS routing in delay-bandwidth sensitive network," in *Proc. IEEE LCN 2000*, Nov. 2000, pp. 579–588.
- [29] G. Xue, "Provably good approximation to minimum cost delay-constrained multicast trees," in *Proc. IEEE Computer Communications and Networks*, Boston, MA, Oct. 1999, pp. 610–614.
- [30] W. M. Moh and B. Nguyen, "An optimal QoS-guaranteed multicast routing algorithm with dynamic membership support," in *Proc. IEEE Int. Conf. Communications*, Piscataway, NJ, 1999, pp. 727–732.

- [31] S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-aware multicast routing protocol," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2580–2592, Dec. 2000.
- [32] D. A. Tran, K. A. Hua, and T. T. Do, "A peer-to-peer architecture for media streaming," *IEEE J. Select. Areas Commun.*, vol. 22, pp. 121–133, Jan. 2004.
- [33] R. Cohen, L. Katzir, and D. Raz, "Scheduling algorithms for a cache pre-filling content distribution network," in *Proc. IEEE INFOCOM*, June 2002, pp. 940–949.
- [34] Z. Li and P. Mohapatra, "The impact of topology on overlay routing service," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.
- [35] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proc. IEEE SIGCOMM IMW*, San Francisco, CA, Nov. 2001, pp. 169–182.
- [36] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*. San Mateo, CA: Morgan Kaufmann, 2001.
- [37] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1998.
- [38] E. W. Zegura, K. Calvert, and S. Bhattacharjee. (1996) GT-ITM: Georgia Tech Internetwork topology models. [Online]. Available: <http://www.cc.gatech.edu/projects/gtitm/>
- [39] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner tree problems in graphs," *Mathematica Japonica*, vol. 24, no. 6, pp. 573–577, 1980.



Jiancong Chen (S'01) received the B.S. degree in computer science and in economics from Peking University, Beijing, China, in 1999 and the Ph.D. degree in computer science from Hong Kong University of Science and Technology, Hong Kong, China, in 2004.

His research interest includes wireless and mobile ad hoc networks, multimedia networking, and network protocols.



S.-H. Gary Chan (S'89–M'98–SM'03) received the B.S.E. degree (highest honor) in electrical engineering from Princeton University, Princeton, NJ, in 1993, and the M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration.

He is currently an Assistant Professor with the Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong, China, and an Adjunct Researcher with Microsoft

Research Asia, Beijing, China. He was a Visiting Assistant Professor in networking in the Department of Computer Science, University of California at Davis, from September 1998 to June 1999. From 1992 to 1993, he was a Research Intern at the NEC Research Institute, Princeton, NJ. His research interest includes multimedia networking, peer-to-peer multicast networks, high-speed and wireless communications networks, and Internet technologies and protocols.

Dr. Chan was the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence in 1993 at Princeton University. He is currently a Vice-Chair of the IEEE COMSOC Multimedia Communications Technical Committee, and is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He was a William and Leila Fellow at Stanford University from 1993 to 1994.



Victor O. K. Li (S'80–M'81–SM'86–F'92) was born in Hong Kong in 1954. He received B.S., M.S., E.E., and Sc.D. degrees in electrical engineering and computer Science from the Massachusetts Institute of Technology, Cambridge, in 1977, 1979, 1980, and 1981, respectively.

He joined the University of Southern California (USC), Los Angeles, in 1981, and became Professor of electrical engineering and Director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong,

Hong Kong, where he is Chair Professor of Information Engineering in the Department of Electrical and Electronic Engineering, and Managing Director of Versitech Ltd. (<http://www.versitech.com.hk/>), the technology transfer and commercial arm of the University. He also serves on various corporate boards. He was a Distinguished Lecturer at the University of California at San Diego, at the National Science Council of Taiwan, and at the California Polytechnic Institute. His research interests include high-speed communication networks, wireless networks, and Internet technologies and applications and he is sought by government, industry, and academic organizations. He has lectured and consulted extensively around the world.

Prof. Li has received numerous awards, including, most recently, the Outstanding Researcher Award of the University of Hong Kong, the KC Wong Education Foundation Lectureship, the Croucher Foundation Senior Research Fellowship, and the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China. He has Chaired the Computer Communications Technical Committee of the IEEE Communications Society (1987–1989), and the Los Angeles Chapter of the IEEE Information Theory Group (1983–1985). He cofounded the International Conference on Computer Communications and Networks (IC3N), and Chaired its Steering Committee (1992–1997). He also chaired various international workshops and conferences, including, most recently, IEEE INFOCOM 2004. He has served as an Editor of the IEEE NETWORK, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Wireless Communications Series, and Telecommunication Systems. He also Guest Edited Special Issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *Computer Networks*, and *ISDN Systems*, and KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKING. He is now serving as an Editor of *ACM/Kluwer Wireless Networks* and the *IEEE Communications Surveys and Tutorials*. He serves on various government committees, including the Hong Kong Information Infrastructure Advisory Committee, the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Program Committee, and the Engineering Panel of the Research Grants Council. He has also delivered keynote speeches at many international conferences. He is a Fellow of the HKIE and the IAE.