

Quickest Paths For Different Network Router Mechanisms

Nageswara S.V. Rao, William C. Grimmell
Computer Science and Mathematics Division
Oak Ridge National Laboratory
P.O. Box 2008, Bldg 6010
Oak Ridge, Tennessee 37831-6355

Sridhar Radhakrishnan, Young C. Bang
School of Computer Science
University of Oklahoma
Norman, Oklahoma 73019

DATE PUBLISHED — June 2000

Research sponsored by the
Engineering Research Program
Office of Basic Energy Sciences
U.S. Department of Energy
and
Defense Advanced Research Projects Agency

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
managed by
UT-BATTELLE, LLC
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-00OR22725

blank page

Contents

List of Figures	iv
Acknowledgements	v
Abstract	vi
1 Introduction	1
2 Computation of Quickest Paths	4
3 Conclusions	8
Appendix	8

List of Figures

1	Arrival and departure timing diagrams at an intermediate node for various modes.	3
2	Example 1.1	4

Acknowledgements

This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC, and Defense Advanced Research Projects Agency under MIPR No. K153.

Abstract

The quickest path problem deals with the transmission of a message of size σ from a source to a destination with the minimum end-to-end delay over a network with bandwidth and delay constraints on the links. We consider four basic modes and two variations for the message delivery at the nodes reflecting the mechanisms such as circuit switching, Internet protocol, and their combinations. For each of first three modes, we present $O(m^2 + mn \log n)$ time algorithm to compute the quickest path for a given message size σ . For the last mode, the quickest path can be computed in $O(m + n \log n)$ time.

Keywords and Phrases: Routing algorithms, quality of service, router modes.

1 Introduction

We consider a network represented by a graph $G = (V, E)$ with n nodes and m edges or links. Each link $e = (i, j) \in E$ has a *bandwidth* or *capacity* $B(e) \geq 0$ and a *link-delay* $D(e) \geq 0$. A message is sent as a continuous stream along the edge e at a constant flow rate denoted by $f_e \leq B(e)$. A message of length σ units can be sent along the edge e at flow rate f_e in $\sigma/f_e + D(e)$ time. The flow rates can be different in different edges, and the message can be delayed at the nodes.

Consider a *simple path* P from $s = v_0$ to $d = v_k$ given by $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$, where $(v_j, v_{j+1}) \in E$, for $j = 0, 1, \dots, (k-1)$, and all v_0, v_1, \dots, v_k are distinct. The delay experienced by a message sent via P depends on the *message forwarding mechanism* used at the intermediate nodes. For a node v on P , let $B_{\text{in}}(v)$ and $B_{\text{out}}(v)$ be the bandwidths of incoming and outgoing edges, respectively, and $f_{\text{in}}(v)$ and $f_{\text{out}}(v)$ be the flow rates of incoming and outgoing messages. We consider the following four basic modes and two variations. The timing diagrams of arrival and departure processes of a message at an intermediate node v are shown for all modes in Figure 1. The *delay* of P is $D(P) = \sum_{j=0}^{k-1} D(e_j)$, where $e_j = (v_j, v_{j+1})$.

- I. **Circuit Switching:** In mode I, the message is sent at a constant rate from s to d with no buffering at intermediate nodes. The *bandwidth of type I* of P is $B^I(P) = \min_{j=0,1,\dots,k-1} B(e_j)$. The *end-to-end delay* in mode I of path P in transmitting a message of size σ is $T^I(P) = \sigma/B^I(P) + D(P)$. Thus, $f_e = B^I(P)$ for all e on P , and $f_{\text{in}}(v) = f_{\text{out}}(v) = B^I(P)$ for all v on P except the end nodes.
- II. **Earliest Departure:** A message received at an intermediate node v is sent out at the rate equal to the minimum of the incoming rate and outgoing bandwidth, i.e. $f_{\text{out}}(v) = \min\{f_{\text{in}}(v), B_{\text{out}}(v)\}$. If the outgoing bandwidth is smaller than the incoming rate, then the message is suitably buffered and sent at a lower rate without any delay. In mode IIa, the retransmission at a lower bandwidth starts only after the entire message is received completely at v under the condition $f_{\text{in}}(v) > B_{\text{out}}(v)$; but, the retransmission is without delay under the condition $f_{\text{in}}(v) \leq B_{\text{out}}(v)$.
- III. **Full Outgoing Bandwidth:** In this mode the flow in any edge is equal to its bandwidth, i.e. $f_e = B(e)$ for all e on P . A message received at an intermediate node v is retransmitted as follows: (a) if $f_{\text{in}}(v) = B_{\text{in}}(v) \geq B_{\text{out}}(v)$, the message is transmitted at the rate $B_{\text{out}}(v)$ starting immediately, and (b) if $f_{\text{in}}(v) = B_{\text{in}}(v) < B_{\text{out}}(v)$, the message is buffered and suitably delayed (so that the rate can be boosted) to be sent at the rate of $B_{\text{out}}(v)$ as soon as possible. Mode IIIa is same as III, except when the outgoing bandwidth is higher than incoming flow rate, in which case the message is completely buffered at v before it is sent out at the rate of $B_{\text{out}}(v)$.
- IV. **Store-Forward:** A message sent along an edge (u, v) will be received in its entirety at v before it is sent from v such that $f_e = B_e$ for all e on P . The

bandwidth of type IV of P is $B^{IV}(P) = \frac{1}{\sum_{j=0}^{k-1} \frac{1}{B(e_j)}}$. The end-to-end delay in mode

IV of path P in transmitting a message of size σ is $T^{IV}(P) = \sigma/B^{IV}(P) + D(P)$.

A path of mode i is referred to as i -path, and a path with the least end-to-end delay among all i -paths is referred to as the i -quickest path, for $i = I, II, IIa, III, IIIa, IV$. Let P_σ^i denote the i -quickest path in mode i for the message size σ .

The various modes abstract different mechanisms used in the data networks. Mode I corresponds to the classical circuit switching which involves no buffering, and mode IV is the other extreme wherein the message is buffered in its entirety at every intermediate node. The telephone networks belong to mode I, and the IP (Internet Protocol) computer networks belong to mode IV. With the advent of ATM (Asynchronous Transmission Mode) and active network technologies, combinations of these modes are being employed, which is the motivation for the other modes. In mode II the message is circuit switched if there is sufficient outgoing bandwidth, and is sent at a reduced flow level otherwise. Such reduction of flow in circuit switching is sometimes difficult, and the message may have to be received in full before the retransmission as in mode IIa. In modes III and IIIa, the main focus is to avoid the fragmentation of the bandwidth of an edge, and hence the entire bandwidth of each edge is utilized during the message transmission along the edge.

A II-path can be converted into I-path by utilizing the flow rate corresponding to the edge with the minimum bandwidth. A I-path can be converted into II-path by suitably increasing the flow rate starting from s and repeatedly moving along P until the next edge with a lower bandwidth is reached and then lowering the flow. During these flow reductions, the end-to-end delays of the path P remain the same. In general for any path P , we have the following inequalities on the end-to-end delays of various modes:

$$T^I(P) = T^{II}(P) \leq T^{III}(P) \leq T^{IIIa} \leq T^{IV}(P)$$

and

$$T^{II}(P) \leq T^{IIa}(P) \leq T^{IV}(P).$$

Example 1.1: Consider the network in Figure 2(a) which consists of two paths P_1 and P_2 from s to d . The first number of the pair next to the link in Figure 2 is the link's bandwidth and the second number is the delay. The end-to-end delays of paths under various modes are given in the following table.

Path	Mode I	Mode II	Mode IIa	Mode III	Mode IIIa	Mode IV
P_1	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{4/5} + 5$	$\frac{\sigma}{1} + 5$	$\frac{\sigma}{4/5} + 5$	$\frac{\sigma}{2/3} + 5$
P_2	$\frac{\sigma}{4} + 6$	$\frac{\sigma}{4} + 6$	$\frac{\sigma}{20/7} + 6$	$\frac{\sigma}{20/6} + 6$	$\frac{\sigma}{20/7} + 6$	$\frac{\sigma}{5/2} + 6$

For this case, we have

$$T^{III}(P) < T^{IIa}(P) = T^{IIIa}(P)$$

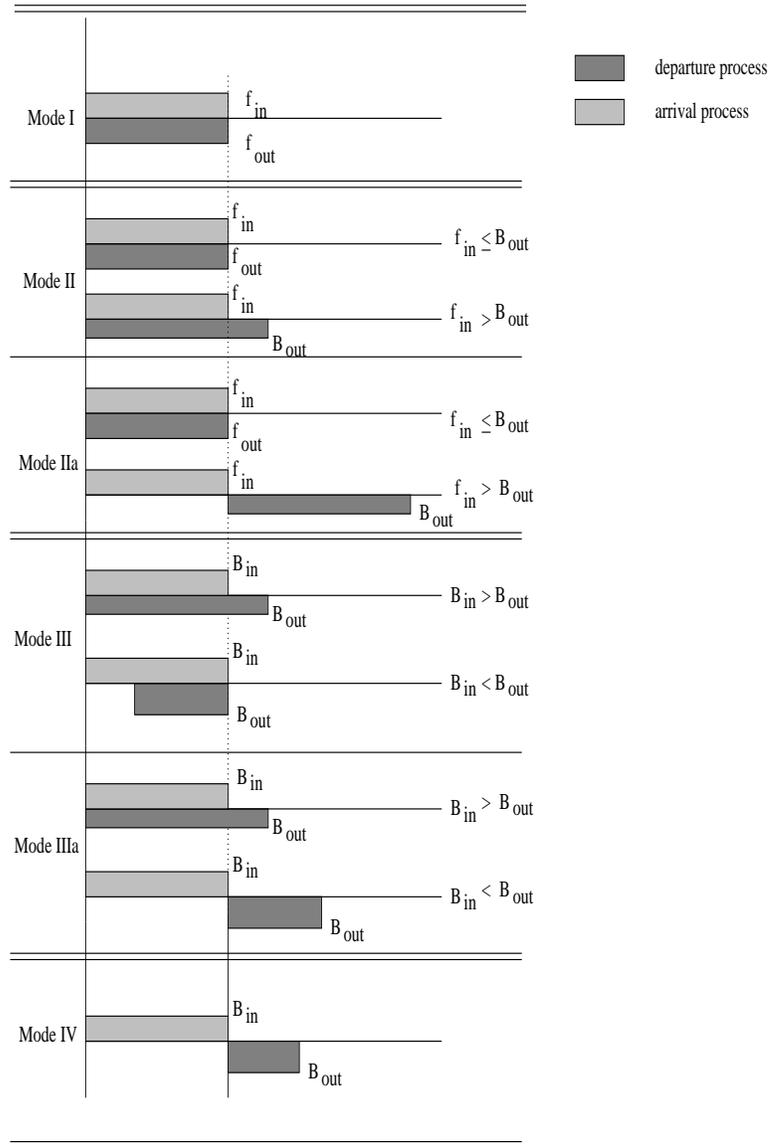


Figure 1: Arrival and departure timing diagrams at an intermediate node for various modes.

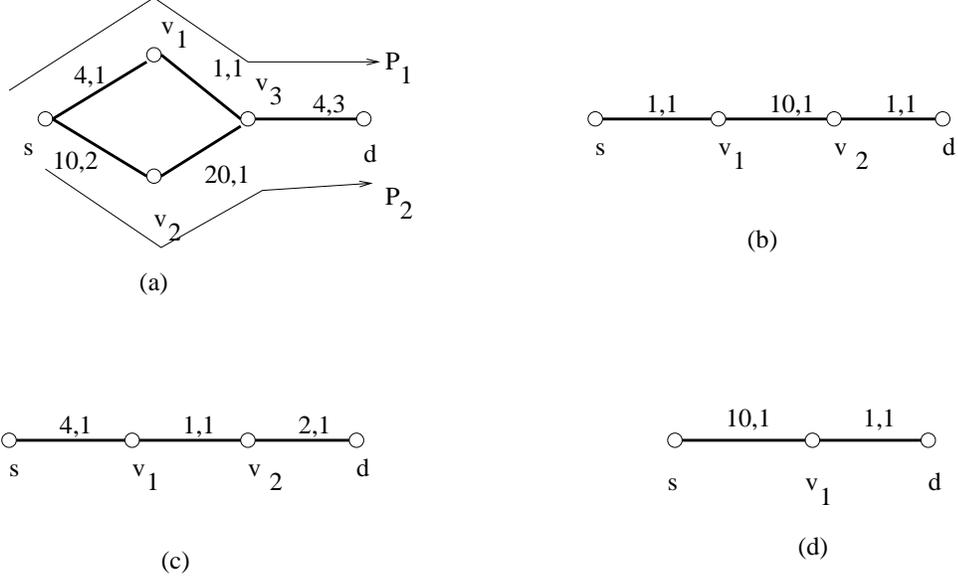


Figure 2: Illustrative networks for Example 1.1

for $P = P_1, P_2$. In general, however, the end-to-end delays in modes IIa and III (or IIIa) do not obey ordering as illustrated below. For the network in Figure 2(b), we have $T^{IIa}(P) = \sigma + 3$, $T^{III}(P) = \frac{\sigma}{10/19} + 3$, and $T^{IIIa}(P) = \frac{\sigma}{1/2} + 3$. Hence, we have

$$T^{IIa}(P) < T^{III}(P) < T^{IIIa}(P).$$

For the network in Figure 2(c), we have $T^{IIa}(P) = \frac{\sigma}{4/5} + 3$, $T^{III}(P) = \frac{\sigma}{1} + 3$, and $T^{IIIa}(P) = \frac{\sigma}{2/3} + 3$. Hence, we have

$$T^{III}(P) < T^{IIa}(P) < T^{IIIa}(P).$$

For the network in Figure 2(d), we have $T^{IIa}(P) = \frac{\sigma}{10/11} + 2$, $T^{III}(P) = \frac{\sigma}{1} + 2$, and $T^{IIIa}(P) = \frac{\sigma}{1} + 2$. Hence, we have

$$T^{III}(P) = T^{IIIa}(P) < T^{IIa}(P). \quad \square$$

Mode I has been studied under the title of *quickest path problem* by Chen and Chin [1], Rosen *et al.* [5], and Rao and Batsell [3]. For any message size σ , I-quickest path can be computed in $O(m^2 + mn \log n)$ time. There seems to be no direct way of extending the previous algorithms for I-quickest paths to handle the other modes. In this paper, we show that II-, IIa-, III- and IIIa-quickest paths can be computed with the same time complexity. It is easy to see that IV-quickest path can be computed in $O(m + n \log n)$ time, using the Dijkstra's shortest path algorithm by utilizing $r/B(e) + D(e)$ as the cost of edge e .

2 Computation of Quickest Paths

We now present an algorithm, QuickII(σ), to compute the end-to-end delay of the II-quickest path for a given message size σ . An alternative implementation of this

```

algorithm Quick-II( $\sigma$ )
1.  $A \leftarrow \emptyset$ ;
2. for each vertex  $v$  in  $V - \{s\}$  do
3.   for  $b \in SBW = \{b_1, b_2, \dots, b_c\}$  do
4.      $TE[v][b] \leftarrow \infty$ ;
5.      $A \leftarrow A \cup \{(v, b)\}$ ;
6.     if  $v \in \text{Adj}(s)$  then  $TE[v][B(s, v)] \leftarrow \sigma/B(s, v) + D(s, v)$ ;
7.   while  $A \neq \emptyset$  do
8.     choose  $(v, b) \in A$  such that  $TE[v][b]$  is minimum;
9.     for each  $w$  in  $\text{Adj}(v)$  do
10.      if  $b \leq B(v, w)$  then
11.         $TE[w][b] \leftarrow \min\{TE[w][b], TE[v][b] + D(v, w)\}$ ;
12.      else
13.         $TE[w][B(v, w)]$ 
         $\leftarrow \min\{TE[w][B(v, w)],$ 
         $TE[v][b] - \sigma/b + D(v, w) + \sigma/B(v, w)\}$ ;
14.      $A \leftarrow A - \{(v, b)\}$ ;
15. return  $\left( \min_{b \in SBW} \{TE[d][b]\} \right)$ ;

```

Algorithm II. Algorithm for computing II-quickest path.

algorithm and its correctness proof are presented in Appendix (the algorithm in the appendix and its proof highlight different properties compared to the ones presented in this section). The path P_σ^{II} itself can be constructed by suitably maintaining the predecessor pointers as in the case of Dijkstra's algorithm [2]. The quickest paths for the other modes (except mode IV) can be computed using minor variations of this algorithm.

Let b_1, b_2, \dots, b_c denote the distinct values of the bandwidths $B(e)$, $e \in E$. Each node v is represented by an array $TE[v][.]$ such that $TE[v][b]$, for $b \in SBW$, where $SBW = \{b_1, b_2, \dots, b_c\}$, whenever finite, is the time at which the trailing edge of the message reaches v at a flow rate b via some path from s to v (note that the flow rate at nodes in between s and v must be at least b in this mode). Since the message leading edge is not delayed at any intermediate node, the flow rate once reduced will stay at this value or be further reduced subsequently. As a result, if the message is received at a flow rate of b at d , then $T^{II}(P) = \sigma/b + D(P)$.

The outline of algorithm is as follows. On the initial entry into the main loop (lines 8-14), the set A contains all pairs (v, b) for all $v \in V - \{s\}$ and $b \in SBW$. The algorithm has $(n - 1)c$ iterations of the main loop, and in each iteration a node v and bandwidth b are selected such that the path from s to v , denoted by P_v and represented by the current $T[v][b]$, has the least end-to-end delay of any path currently corresponding to a $(v, b) \in A$ (line 8). Let $\text{Adj}(v)$ denote the set of all w such that

$(v, w) \in E$. Once v is selected, every vertex $w \in \text{Adj}(v)$, is examined to see if the extension of P_v to w , obtained by appending (v, w) to it, results in a smaller end-to-end delay than a comparable current value of $T[w][b']$ (lines 9-13). In particular, if the bandwidth of (v, w) is higher than or equal to b , then a flow rate of b is used along (v, w) (line 11), and a lower flow rate of $B(v, w)$ is used otherwise (line 13). In either case the extension of the path from v to w can result in a flow into w that is no more than b . Once all $w \in \text{Adj}(v)$ are examined, (v, b) is removed from A and is not considered further at line 8.

We shall now introduce some preliminaries to be used in the correctness proof of Quick-II. For path $P = \{s, v_1, v_2, \dots, v_p, v_{p+1}\}$, let $T_{P,\sigma}[v_i]$ denote the end-to-end delay of the subpath from s to v_i for message size σ . Let $\widehat{TE}[v][b]$ denote the value of $TE[v][b]$ when (v, b) is chosen in line 8. For any path $P = \{s, v_1, v_2, \dots, v_p, v_{p+1}\}$, we have $T_{P,\sigma}[v_i] \leq T_{P,\sigma}[v_j]$ for $i < j$. It is easily shown that the sequence of $\widehat{TE}[v][b]$ values generated by various choices of (v, b) in line 8 is non-decreasing, and the sequence of $TE[v][b]$ for every (v, b) generated in lines 4,6,11 and 13 is non-increasing. Note that each finite $\widehat{TE}[v][b]$ computed by Quick-II corresponds to the end-to-end delay of some II-path from s to v . This observation implies the following result.

Lemma 2.1 *If $P^* = \{s, v_1, v_2, \dots, v_p, v_{p+1}\}$ is a quickest II-path from s to v_{p+1} with flow of f_i into v_i , then throughout the execution of Quick-II, we have $TE[v_i][f_i] \geq T_{P,\sigma}[v_i]$ for all $i = 1, 2, \dots, p + 1$.*

Theorem 2.1 *Algorithm Quick-II(σ) computes the end-to-end delay of II-quickest path for transmitting a message of size σ from s to every node in $O(m^2 + mn \log n)$ time.*

Proof: If there is no path from s to d , the $TE[d][b]$ will not be reduced from ∞ for any b . If there is a path, let $P^* = \{v_0, v_1, v_2, \dots, v_p, v_{p+1}\}$ be a quickest II-path for message size σ , where $s = v_0$ and $v_{p+1} = d$. Let f_i be the incoming flow at v_i corresponding to P . We will now show that for all $i = 1, 2, \dots, p + 1$, we have $\widehat{TE}[v_i][f_i] = T_{P,\sigma}[v_i]$. This condition implies that $TE[v_i][f_i] = T_{P,\sigma}[v_i]$ at the termination of the algorithm, since $TE[v_i][f_i] \geq T_{P,\sigma}[v_i]$ throughout the execution of the algorithm.

From the above observations and Lemma 2.1, for any $1 \leq i < j \leq p + 1$, we have

$$\widehat{TE}[v_j][f_j] \geq T_{P,\sigma}[v_j] \geq T_{P,\sigma}[v_i]$$

If (v_j, f_j) is selected before (v_i, f_i) , then by the monotonicity of \widehat{TE} values, we have $\widehat{TE}[v_j][f_j] \leq \widehat{TE}[v_i][f_i]$. Then, if in addition the condition $\widehat{TE}[v_i][f_i] = T_{P,\sigma}[v_i]$, is satisfied, we have $TE[v_j][f_j] = T_{P,\sigma}[v_j]$.

As the basis of an induction note that $\widehat{TE}[v_1][f_1] = T_{P,\sigma}[v_1]$ as initialized in line 6. For the inductive hypothesis, assume $\widehat{TE}[v_i][f_i] = T_{P,\sigma}[v_i]$. Then if (v_{i+1}, f_{i+1}) is chosen in line 8 before (v_i, f_i) , then from the previous paragraph $\widehat{TE}[v_{i+1}][f_{i+1}] = T_{P,\sigma}[v_{i+1}]$. If (v_{i+1}, f_{i+1}) is chosen after (v_i, f_i) , then $TE[v_{i+1}][f_{i+1}]$ is set to $T_{P,\sigma}[v_{i+1}]$ in lines 11-13, where it will be when (v_{i+1}, f_{i+1}) is subsequently chosen in line 8. Then by induction $\widehat{TE}[v_{p+1}][f_{p+1}] = T_{P,\sigma}[v_{p+1}]$, which is the delay of II-quickest path.

This algorithm essentially consists of at most $c \leq m$ instances of Dijkstra's algorithm, each corresponding to $b \in SBW$ algorithm with interleaved steps. The time complexity follows directly from that of Dijkstra's algorithm [2]. \square

The algorithm for mode IIa is obtained by replacing the line 13 of Quick-II by the following line.

13'. $TE[w][B(v, w)] \leftarrow \min\{TE[w][B(v, w)], TE[v][b_v] + D(v, w) + \sigma/B(v, w)\};$

As noted above for any path P , $T^I(P) = T^{II}(P)$, and therefore, Quick-II can be used to determine I-quickest paths. However, the trailing edge arrival times at the intermediate nodes along such I-paths are not given by $TE[v][b]$ computed by Quick-II.

The algorithm for mode I, namely algorithm Quick-I, is obtained by replacing the lines 10-13 of Quick-II by the following.

if $b \leq B(v, w)$ **then**

$TE[w][b] \leftarrow \min\{TE[w][b], TE[v][b] + D(v, w)\};$

Algorithm Quick-I can be viewed as a variation of the algorithm of [5], which is based on c instances of Dijkstra's algorithm, executed separately. Let $G(b) = (V, E(b))$ denote the subnetwork where $e \in E(b)$ if and only if $B(e) \geq b$. Let a $s - d$ path in $G(b)$ denote the shortest delay path based *only* on the link-delays. The $s - d$ paths in [5] are independently computed in each $G(b)$ for each $b = b_1, b_2, \dots, b_c$. Then, I-quickest path is selected to be the one with lowest end-to-end delay among the c paths. The algorithm Quick-I "intermingles" the path computations in $G(b)$'s by expanding a path from v to w only if $B(v, w) \geq b$

The algorithm for mode III is obtained by replacing the lines 10-13 of Quick-II by the following lines.

10'. **if** $b \leq B(v, w)$ **then**

11'. $TE[w][B(v, w)] \leftarrow \min\{TE[w][B(v, w)], TE[v][b] + D(v, w)\};$

12'. **else**

13'. $TE[w][B(v, w)]$

$\leftarrow \min\{TE[w][B(v, w)],$
 $TE[v][b] - \sigma/b + D(v, w) + \sigma/B(v, w)\};$

The algorithm for mode IIIa is obtained by replacing the line 11' of Quick-III by the following line.

11'. $TE[w][B(v, w)] \leftarrow \min\{TE[w][B(v, w)], TE[v][b] + \sigma/B(v, w) + D(v, w)\};$

The correctness proof and the time complexity of the corresponding algorithms, namely algorithm Quick- i for $i = I, IIa, III, IIIa$, can be established with minor modifications to Theorem 2.1.

3 Conclusions

We presented four modes and two of their variations for the quickest path problem that reflect mechanisms such as circuit switching, Internet protocol, and their combinations. We presented a basic algorithm, whose variations compute the quickest paths in the first three modes; for the last mode, Dijkstra's algorithm computes the quickest path.

Future research directions include the computation of path-tables for all modes. For mode I, the path-table is of size $O(m)$ and can be easily computed [1, 5, 3], and such results can be very useful in the other modes. The other research directions include multiple paths (as in [4, 8] for mode I), multicast trees [7], and incorporation of reliabilities [6] for various modes described in this paper.

Appendix

We now present an alternative to the algorithm Quick-II and its variants. While the overall algorithm is essentially the same as the one presented in Section 2, it provides different details, and its proof method exploits different properties.

For initialization, consider distinct $u, v \in V$. If (u, v) is an edge e then define $DE(u, v) = D(e)$, and if not $DE(u, v) = \infty$. Similarly, if (u, v) is an edge e then define $BE(u, v) = B(e)$, and if not $BE(u, v) = 0$. Each node v is represented by an array $TE[v][.]$ such that $TE[v][b]$, for $b = b_1, b_2, \dots, b_c$, is the time at which the trailing edge of the message reaches v at a flow rate b .

The outline of algorithm is as follows. The algorithm has at most $(n - 1)c$ iterations, and in each iteration a node v with bandwidth b , for some $b_i = b$, is selected such that the path from s to v , denoted by P_v , has the least end-to-end delay (line 7). Once v is selected, every vertex $w \in V - A$, is examined to see if the path P_v can be extended to w to result in a smaller end-to-end delay in lines 12-21. In particular, if the bandwidth of (v, w) is higher than b , then flow rate of b is used along (v, w) (line 13), and a lower flow rate of $B(v, w)$ is used otherwise (line 18). In either case the extension of the path from v to w can result in a flow into w that is no more than b . Once the end-to-end delay of v is known for all $b = b_1, b_2, \dots, b_c$, it is added to A and is not considered further (line 9). For the sake of this algorithm the infinities are handled according to the following rules: $1/0 = \infty$, $\infty + \infty = \infty$, and $\infty \leq \infty$.

Lemma A.1. *Let $P^* = \{s, v_1, v_2, \dots, v_p\}$ denote a II-quickest path from s to v_p for the message size σ with flow f_p^* at v_p (i. e. $f_{in}(v_p) = f_p^*$). For any II-path P from s to v_p with flow f_p at v_p , we have $T^{II}(P) = \sigma/f_p + D(P)$. If $f_p = f_p^*$, we have $D(P^*) \leq D(P)$.*

Proof: For any II-path P , the flow f_p at the last node v_p is the lowest of all flows at all nodes of P , which yields $T^{II}(P) = \sigma/f_p + D(P)$. Consider $f_p = f_p^*$. Since P^* is

algorithm Quick-II(σ)

1. $A \leftarrow \emptyset$; $CT[s] = 0$; $BT[s] = b_1$;
2. **for** $b = b_1, b_2, \dots, b_c$ **do**
3. $TE[s][b] = 0$; $BS[v] = \{b_1, b_2, \dots, b_c\}$
4. **for** each vertex v in $V - \{s\}$ **do**
5. **for** $b = b_1, b_2, \dots, b_c$ **do** $TE[v][b] \leftarrow \infty$;
6. **while** $A \neq V$ **do**
7. choose a vertex v in $V - A$ such that $CT[v]$ is minimum
8. $b_v \leftarrow BT[v]$;
9. **if** $CT[v] \geq TE[v][b]$ for all $b = b_1, b_2, \dots, b_c$ **then** add v to A ;
10. **else** $BS[v] \leftarrow BS[v] - \{b_v\}$; $CT[v] = TE[v][b]$;
- $BT[v] = b$ such that $T[v][b] = \min_{b' \in BS[v]} T[v][b']$;
11. **for** each $w \in V - A$ such that (v, w) in E **do**
12. **if** $b_v \leq BE(v, w)$ **then**
13. $TE[w][b_v] \leftarrow \min\{TE[w][b_v], TE[v][b_v] + DE(v, w)\}$;
14. **if** $TE[w][b_v] < CT[w]$ **then**
15. $CT[w] \leftarrow TE[w][b_v]$;
16. $BT[w] \leftarrow b_v$;
17. **else**
18. $TE[w][BE(v, w)]$
 $\leftarrow \min\{TE[w][BE(v, w)],$
 $TE[v][b_v] - \sigma/b_v + DE(v, w) + \sigma/BE(v, w)\}$;
19. **if** $TE[w][BE(v, w)] < CT[w]$ **then**
20. $CT[w] \leftarrow TE[w][BE(v, w)]$;
21. $BT[w] \leftarrow BE(v, w)$;
22. **return** $\left(\min_b \{TE[d][b]\} \right)$;

Algorithm II. Algorithm for computing II-quickest path.

II-quickest path, $\sigma/f_p + D(P^*) \leq \sigma/f_p + D(P)$, which yields $D(P^*) \leq D(P)$. \square

Theorem A. 1. *Algorithm Quick-II computes the end-to-end delay of II-quickest path for a message of size σ from s to every node in $O(m^2 + mn \log n)$ time.*

Proof: Let the iteration of Quick-II in which v with $BT[v] = b_v$ is selected in line 7 be denoted by (v, b_v) . We establish the correctness of Quick-II through induction on iterations by asserting that in iteration (v, b_v) , $CT[v] = TE[v][b_v]$ is the lowest end-to-end delay among all paths from s to v with flow b_v at v .

For the induction basis, consider the first iteration (s, b_1) . Claim is true since the message is already at s and $TE[s][b_1] = 0$.

For the induction step, consider the iteration (v, b_v) , and assume that the claim is true in *every* prior iteration. Consider that $TE[v][b_v]$, that corresponds to P_v , is not the minimum end-to-end delay among all II-paths from s to v with flow b_v at v . Then, let $P^* = \{s, v_1, \dots, v_p, v\}$ be a II-quickest path from s to v with flow b_v at v . Let $f_i^* = f_{in}(v_i)$, for $i = 1, \dots, p$. Then for each v_i , the pair (v_i, f_i^*) has been chosen in a previous iteration as will be shown now. For if this was not the case, for some k , v_k would be the first node on P^* while moving from s to v for which (v_i, f_i^*) was not previously chosen. Now consider the subpath P_k^* of P^* from s to v_k . Since P^* is II-path, $f_k^* \geq b_v$, and by Lemma A.1 we have $D(P_k^*) < D(P^*) \leq D(P_v)$. Then, by the inductive hypothesis and Lemma A.1, we have

$$TE[k][f_k^*] = \sigma/f_k^* + D(P_k^*) < \sigma/b_v + D(P^*) \leq \sigma/b_v + D(P_v) \leq TE[v][b_v],$$

which is a contradiction, since, if it were true, (k, f_k^*) would have been chosen at this iteration instead of (v, b_v) .

Since all (v_i, f_i^*) were chosen at previous iterations, (v_p, f_p^*) was chosen before (v, b_v) . In the iteration in which (v_p, f_p^*) was chosen $TE[v][b_v]$ was set to $\sigma/b_v + D(P^*)$ either in line 13 or 18, since at that iteration, $TE[v_p][f_p^*] = \sigma/f_p^* + D(P_p^*)$ by inductive hypothesis. Once set to this value, it is not replaced by a larger value, and hence $TE[v][b_v] = \sigma/b_v + D(P^*)$.

Note that at the termination of the algorithm every $v \in V$ is selected in line 7 for every bandwidth b exactly once. The II-path with the lowest end-to-end delay among paths with all flow rates is chosen in line 22. Then, the correctness of the algorithm follows by noting that II-quickest path must have a bandwidth equal to one of the b_i 's at v . The time complexity analysis of this algorithm is very similar to that in Theorem 2.1 \square

The algorithm for mode IIa is obtained by replacing the line 18 of Quick-II by the following line.

$$18'. \quad TE[w][BE(v, w)] \\ \leftarrow \min\{TE[w][BE(v, w)], TE[v][b_v] + DE(v, w) + \sigma/BE(v, w)\};$$

The algorithm for mode I, namely algorithm Quick-I, is obtained by replacing the lines 12-18 of Quick-II by the following.

$$\text{if } b_v \leq BE[v, w] \text{ then} \\ TE[w][b_v] \leftarrow \min\{TE[w][b_v], TE[v][b_v] + DE(v, w)\};$$

The algorithm for mode III is obtained by replacing the lines 12 through 22 of Quick-II by the following lines.

```

12.   if  $b_v \leq BE(v, w)$  then
13.      $TE[w][BE(v, w)] \leftarrow \min\{TE[w][BE(v, w)], TE[v][b_v] + DE(v, w)\};$ 
14.   else
15.      $TE[w][BE(v, w)]$ 
            $\leftarrow \min\{TE[w][BE(v, w)],$ 
            $TE[v][b_v] - \sigma/b_v + DE(v, w) + \sigma/BE(v, w)\};$ 
16.   if  $TE[w][BE(v, w)] < CT[w]$  then
17.      $CT[w] \leftarrow TE[w][BE(v, w)];$ 
18.      $BT[w] \leftarrow BE(v, w);$ 
19. return  $\left(\min_b\{TE[d][b]\}\right);$ 

```

The algorithm for mode IIIa is obtained by replacing the line 13 of Quick-III by the following line.

```

13'.    $TE[w][BE(v, w)]$ 
         $\leftarrow \min\{TE[w][BE(v, w)], TE[v][b_v] + \sigma/BE(v, w) + DE(v, w)\};$ 

```

References

- [1] Y. L. Chen and Y. H. Chin. The quickest path problem. *Computers and Operations Research*, 17(2):153–161, 1990.
- [2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Co., New York, 1990.
- [3] N. S. V. Rao and S. G. Batsell. Algorithm for minimum end-to-end delay paths. *IEEE Communications Letters*, 1(5):152–154, 1997.
- [4] N. S. V. Rao and S. G. Batsell. QoS routing via multiple paths using bandwidth reservation. In *IEEE INFOCOM98: The Conference on Computer Communications*, volume 1, pages 11–18. 1998.
- [5] J. B. Rosen, S. Z. Sun, and G. L. Xue. Algorithms for the quickest path problem and the enumeration of quickest paths. *Computers and Operations Research*, 18(6):579–584, 1991.
- [6] G. Xue. End-to-end data paths: Quickest or most reliable? *IEEE Communications Letters*, 2(6):156–158, 1998.
- [7] G. Xue and S. Sun. Optimal multicast trees in communication systems with channel capabilities and channel reliabilities. *IEEE Transactions on Communications*, 47(5):662–663, 1999.
- [8] G. Xue, S. Sun, and J. B. Rosen. Fast data transmission and maximal dynamic flow. *Information Processing Letters*, 66:127–132, 1998.

INTERNAL DISTRIBUTION

- | | | | |
|--------|----------------|-----|----------------------------|
| 1. | J. Barhen | 27. | T. Zacharia |
| 2. | T. Dunigan | 28. | Central Research Library |
| 3-13. | W. C. Grimmell | 29. | Document Reference Section |
| 14. | L. E. Parker | 30. | Laboratory Records - RC |
| 15-25. | N. S. V. Rao | 31. | ORNL Patent Office |
| 26. | W. Wing | | |

EXTERNAL DISTRIBUTION

32. Dr. Robert E. Price, Division of Material Science and Engineering, Office of Basic Energy Sciences, SC-131, U. S. Department of Energy, 19901 Germantown Road, Germantown, MD 20874-1290
- 33-34. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831
- 35-40. Prof. S. Radhakrishanan, School of Computer Science, 200 Felgar Street, Room 114, University of Oklahoma, Norman, Oklahoma 73019