



Title	Hop-by-hop routing in wireless mesh networks with bandwidth guarantees
Author(s)	Hou, R; Lui, KS; Baker, F; Li, J
Citation	IEEE Transactions on Mobile Computing, 2012, v. 11 n. 2, p. 264-277
Issued Date	2012
URL	http://hdl.handle.net/10722/155717
Rights	IEEE Transactions on Mobile Computing. Copyright © IEEE

Hop-by-Hop Routing in Wireless Mesh Networks with Bandwidth Guarantees

Ronghui Hou, *Member, IEEE*, King-Shan Lui, *Senior Member, IEEE*,
Fred Baker, and Jiandong Li, *Senior Member, IEEE*

Abstract—Wireless Mesh Network (WMN) has become an important edge network to provide Internet access to remote areas and wireless connections in a metropolitan scale. In this paper, we study the problem of identifying the maximum available bandwidth path, a fundamental issue in supporting quality-of-service in WMNs. Due to interference among links, bandwidth, a well-known bottleneck metric in wired networks, is neither concave nor additive in wireless networks. We propose a new path weight which captures the available path bandwidth information. We formally prove that our hop-by-hop routing protocol based on the new path weight satisfies the consistency and loop-freeness requirements. The consistency property guarantees that each node makes a proper packet forwarding decision, so that a data packet does traverse over the intended path. Our extensive simulation experiments also show that our proposed path weight outperforms existing path metrics in identifying high-throughput paths.

Index Terms—Wireless mesh networks, QoS routing, proactive hop-by-hop routing, distributed algorithm.

1 INTRODUCTION

A wireless mesh network (WMN) consists of a large number of wireless nodes. The nodes form a wireless overlay to cover the service area while a few nodes are wired to the Internet. As part of the Internet, WMN has to support diversified multimedia applications for its users. It is essential to provide efficient Quality-of-Service (QoS) support in this kind of networks [1]. Seeking the path with the maximum available bandwidth is one of the fundamental issues for supporting QoS in the wireless mesh networks. The available path bandwidth is defined as *the maximum additional rate a flow can push before saturating its path* [2]. Therefore, if the traffic rate of a new flow on a path is no greater than the available bandwidth of this path, accepting the new traffic will not violate the bandwidth guaranteed of the existing flows. This paper focuses on the problem of identifying the maximum available bandwidth path from a source to a destination, which is also called the *Maximum Bandwidth Problem* (MBP). MBP is a subproblem of the *Bandwidth-Constrained Routing Problem* (BCRP), the problem of identifying a path with at least a given amount of available bandwidth [3]. In the literatures, *maximum available bandwidth path* is also called *widest path*. In this paper, we use these two terms interchangeably.

Finding the widest path between the source and the destination in wireless networks is very challenging due to

the wireless transmission interference. Generally speaking, there are two types of interference: *interflow interference* and *intraflow interference* [2], [4]. Interflow interference refers to the situation that the resource available for a flow is affected by the presence of other flows. In other words, the interflow interference affects the amount of residual channel resources on each link that can be allocated for a new flow. The work in [5] gives how to estimate the available bandwidth (residual channel resources) of each link. It means that if the link has to carry another 1-hop flow without violating the bandwidth guarantees of existing flows, the rate of this flow can be at most the available bandwidth of the link. On the other hand, intraflow interference refers to the scenario where when a data packet is being transmitted on a link along a path, some link along the path has to remain idle to avoid conflict. Intraflow interference complicates the process of developing hop-by-hop routing protocol for finding widest paths. Considering intraflow interference, the works in [2] and [6] present a formula to compute the available bandwidth of a path with the knowledge of the available bandwidth on individual links of the path. Unfortunately, finding widest path in a hop-by-hop manner is still not solved. The unique structure of the path bandwidth computation formula introduces two challenges described below:

1. Some nodes may not find the widest path if only the available bandwidth is used as the routing metric.
2. Even though a source identifies a widest path to a destination, intermediate nodes on the widest path may not make a consistent packet forwarding decisions by using the traditional destination-based hop-by-hop packet forwarding mechanism.

For example, in Fig. 1, according to the formula in [2] and [6] (will be described in detailed later), the upper path from v to d has a larger available bandwidth than the lower path from v to d . Nevertheless, by the formula in [2] and [6], the

• R. Hou and J. Li are with the State Key Lab of Integrated Service Networks, Xidian University, PO BOX 102 710071, Xi'an, China. E-mail: rhhou@xidian.edu.cn, jdli@mail.xidian.edu.cn.

• K.-S. Lui is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: kslui@eee.hku.hk.

• F. Baker is with the Cisco Research Center, Cisco Systems, Inc., 170 West Tasman Dr., San Jose, CA 95134. E-mail: fred@cisco.com.

Manuscript received 28 Dec. 2009; revised 24 Sept. 2010; accepted 31 Dec. 2010; published online 2 Feb. 2011.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-12-0557. Digital Object Identifier no. 10.1109/TMC.2011.25.

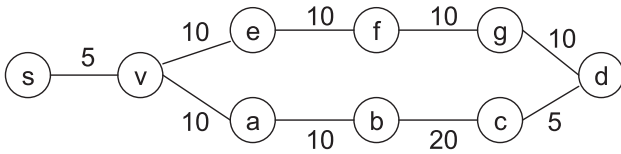


Fig. 1. An example of network topology.

lower path from s to d is better in terms of available bandwidth. According to the traditional distance vector protocol, node v just advertises the upper path information to its neighbors, so that node s cannot obtain the widest path from itself to d . Even s identifies the lower path to d which has the larger available bandwidth, the problem is not solved. When node v receives the data packet from s , it will forward the packet to e but not to a by using the traditional destination-based hop-by-hop routing, since the upper path from v to d has the larger available bandwidth. That is, the data packet actually does not traverse on the widest path from s to d .

In fact, the above two challenges mean that a correct routing protocol should satisfy the *optimality requirement* and *consistency requirement*. The key for designing such routing protocol is to develop an isotonic routing metric. Interested readers can refer to [7] and [8] for the detailed discussion.

In this work, we study how to perform routing in the 802.11-based WMNs and make the following contributions.

- We propose a new path weight that captures the concept of available bandwidth. We give the mechanism to compare two paths based on the new path weight. We formally prove that the proposed path weight is left-isotonic.
- We describe how to construct the routing table and distance table, and we develop a hop-by-hop packet forwarding scheme. We formally prove that our routing protocol satisfies the *optimality* and *consistency* requirements.
- Finally, we implement our routing protocol based on the DSDV protocol in the NS2 simulator. The extensive simulation experiments demonstrate that our routing protocol outperforms the existing routing protocols for finding the maximum available bandwidth paths.

The rest of the paper is organized as follows: After describing the related works in Section 2, we explain how to compute the available bandwidth on a path in Section 3. Section 4 describes our hop-by-hop routing protocol in details, and Section 5 presents our extensive simulation results. We finally conclude our paper in Section 6.

2 RELATED WORKS

To identify the widest path, many researchers develop new path weights, and the path with the minimum/maximum weight is assumed to be the maximum available bandwidth path. In [9] and [10], the expected transmission count (ETX) metric was proposed. The ETX of a link is the predicted number of data transmissions required to send a packet over that link, which is estimated by proactively sending a

dedicated link probe packet periodically. The ETX of a path is the sum of the ETX metrics of all links on this path. It is the earliest link metric developed and many other metrics are extended from it [11]. ETT [12] is an improved version of ETX that also considers the effect of packet size and raw data rate on the links because of the use of multiple channels. In this paper, we consider the single-channel wireless mesh networks, and assume that the raw data rates of all the links are the same, as well as all the packets are of the same size. In this case, ETT is the same as ETX. Several other metrics, such as iAWARE [13], IRU [14], and CATT [15], are all extended from ETT. iAWARE is the ETT metric adjusted based on the number of the interference links and the existing traffic load on the interference links. IRU is the ETT metric weighted with the number of the interference links, while CATT extends IRU by considering the effect of packet size and raw data rate on the links because of the use of multiple channels.

Some existing QoS routing protocols operate with the knowledge of the available bandwidth of each link [2], [4], [6], [16], [17], [18], [19]. These works study how to compute the available bandwidth of a path based on the available bandwidth of each link on this path. Liu and Liao [17] give a new link metric which is the available bandwidth of the link divided by the number of interference links of this link. The path bandwidth is thus defined as the minimum value of the new metrics of all the links on this path. In the mechanism described in [18], the available bandwidth of a path is the minimum bandwidth among the links on the path divided by 2, 3, or 4, depended on the number of hops on the path. Such formula cannot reflect the exact path bandwidth. The path selection processes in [4], [19], [20], [21], and [22] assume the bandwidth requirement of a connection request is known. The metric proposed in [4] is based on the bandwidth requirement of a certain request. The protocol in [19] checks the local available bandwidth of each node to determine whether it can satisfy the bandwidth requirement. Some works [20], [21], [22] consider the TDMA-based MAC model and discuss how to assign the available time slots on each link for a new flow in order to satisfy the bandwidth requirement of the new flow.

Former studies [2], [6], [16], [23], [24], [25], [26] discuss how to estimate the available bandwidth of a given path. They all apply the *clique-based path bandwidth computation* method. Zhai and Fang [23], Jia et al. [24], Kordialam and Nandagopal [25] give the formula to compute the exact available bandwidth of a path, which cannot be solved in polynomial-time, because the problem is NP-complete in nature [23], [26]. Even though we can find the available bandwidth of a given path, it is not easy to identify a schedule that achieves that bandwidth since the scheduling problem is also NP-complete [22]. In other words, finding the available bandwidth on any kind of MAC model is NP-complete [3]. The works in [2] and [6] developed another formula to approximately compute the available bandwidth of a path. We will show that the bandwidth calculated by this formula can be easily achieved. In other words, we can find a simple scheduling mechanism to achieve the bandwidth calculated by the formula in [2] and [6]. In this

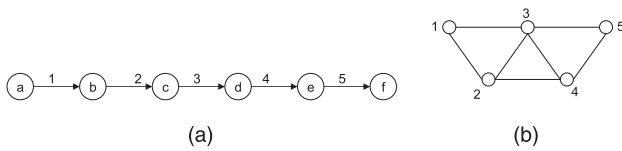


Fig. 2. Illustration for interference model. (a) The original graph, (b) The conflict graph.

work, we will apply the mechanism in [2] and [6] to estimate the available bandwidth of a given path. Although a formula is developed in [2] and [6], the authors did not provide a packet forwarding mechanism to assure that the data packet traverses over the estimated widest path from the source to the destination. Our main goal is to develop a practical routing protocol that allows packets to go through the estimated widest path.

QoS support in multihop wireless networks has been studied from the cross-layer design perspectives. Zhang and Zhang [1] give a comprehensive review for the current study on the cross-layer paradigm for QoS support in multihop wireless networks. Contrary to the cross-layer mechanism, our protocol performs over the practical 802.11 MAC protocol, and so our routing protocol can be easily incorporated in the current wireless devices.

3 PRELIMINARIES

In this section, we give the overview of the clique-based method for computing the available path bandwidth.

Lots of the existing works [2], [6], [23], [24], [25], [26], [27], [28] apply the *link conflict graph* (or *conflict graph* for short) to reflect the interference relationship between links. A link in the wireless network becomes a node in the link conflict graph. If two links in the wireless network interfere with each other, we put a link between the corresponding nodes in the link conflict graph. We use an example in [23] to illustrate the link conflict graph. Fig. 2a shows a five-link chain topology. The numbers on the links are the ids of the links. The link conflict graph of the network is shown in Fig. 2b. Links 1 and 2 interfere with each other since node b cannot send and receive simultaneously. Links 1 and 3 interfere with each other since the signal from c is strong enough to interfere the reception at b . Therefore, there are links between 1 and 2 as well as 1 and 3 in the conflict graph. Assume that links 1 and 4 do not interfere because the signal from d cannot affect b in successfully receiving the signal from a . Then, there is no link between 1 and 4 in Fig. 2b.

An *interference clique* is the set of links which interfere with each other. In the conflict graph, the corresponding nodes of these links form a complete subgraph. In Fig. 2b, $\{1, 2\}$, $\{1, 3\}$, $\{1, 2, 3\}$, and $\{3, 4, 5\}$ are interference cliques. A *maximal interference clique* is a complete subgraph that is not contained in any other complete subgraph. For instance, $\{1, 2, 3\}$ and $\{3, 4, 5\}$ are maximal cliques while $\{1, 2\}$ and $\{1, 3\}$ are not maximal cliques. In this work, we consider single-channel single-rate wireless networks, and so the original capacity of each link is the same, denoted by C . Denote $\{Q_1, \dots, Q_K\}$ as the maximal interference clique set of the network. The work [25] introduces the following lemma.

Lemma 1. Denote \mathbf{f} as a link flow vector, where $f(e)$ is the aggregate data rate of the flow on link e . If \mathbf{f} does not satisfy the following inequalities

$$\sum_{e \in Q_k} f(e) \leq C, \quad \forall k, \quad (1)$$

then \mathbf{f} is not schedulable.

Lemma 1 gives the method to compute the theoretical available bandwidth of a path. Given a path $p = \langle v_1, v_2, \dots, v_h \rangle$, we first find the set of the maximal cliques $\{S_1, S_2, \dots, S_M\}$ such that $S_m \cap p \neq \emptyset$ for all $m = 1, \dots, M$. Denote $f_{\text{sum},m}$ as the total *current* data rate of the flows on all the links of the maximal clique S_m and $|S_m \cap p| = k_m$. Equation (1) implies that the maximum *additional* data rate r on path p should satisfy the condition that $k_m r \leq C - f_{\text{sum},m}$ for all $m = 1, \dots, M$. The rationale behind this constraint is that the aggregate additional data rates on all links in the maximal clique S_m should be less than $C - f_{\text{sum},m}$ in order to avoid conflict. By finding all the maximal cliques, the maximum available bandwidth of path p can be found. However, finding all maximal cliques is NP-complete [23], [26]. Moreover, it is difficult to find a scheduling mechanism to achieve the maximum available bandwidth. In the following, we describe another mechanism to approximately compute the maximum available bandwidth of a path, and there exists a simple scheduling to achieve the estimated bandwidth.

Given a path $p = \langle v_1, v_2, \dots, v_h \rangle$, based on the current flows on each link in the network, denote $B(e)$ as the available bandwidth of link e . It means that if a new connection only needs to go through link e , e can send at most $B(e)$ Kbits amount of information in a second without affecting existing flows. The work in [5] described how to obtain $B(e)$, and the following discussion assumes $B(e)$ is known. Note that the bit error rate of a link is considered in the link estimator, and thus the available bandwidth of each link becomes the *expected available link bandwidth* [23]. Denote \mathbf{Q}_p as the set of the maximal cliques containing *only* the links on p . Generally speaking, if two links on a path interfere with each other, all the links between them along the path conflict with each other [23]. This implies that it is easy to find \mathbf{Q}_p for path p . The available bandwidth of path p is estimated as follows [5], [6]:

$$B(p) = \min_{\mathbf{q} \in \mathbf{Q}_p} C_{\mathbf{q}}; \quad C_{\mathbf{q}} = \frac{1}{\sum_{l \in \mathbf{q}} \frac{1}{B(l)}}. \quad (2)$$

The rationale behind the formula is: transmissions on the links in a clique cannot be concurrent but occur in a serial manner. Thus, the time it takes $\sum_{l \in \mathbf{q}} \frac{1}{B(l)}$ for 1 Mbit data to traverse all the links in the clique \mathbf{q} . $C_{\mathbf{q}}$ is thus the bandwidth available over the clique \mathbf{q} . The available bandwidth of the path is the bandwidth of the bottleneck clique. We refer readers to the references for further explanation.

Example 1. Let $B(1)$, $B(2)$, $B(3)$, and $B(4)$ of the network in Fig. 2a be 50, 100, 25, and 20 Mbps, respectively, as in the example provided in [2]. There are two maximal cliques on path $\langle a, b, c, d, e \rangle$ and they are $\{1, 2, 3\}$

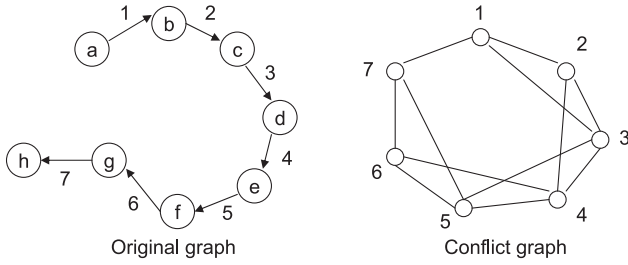


Fig. 3. An example in [23].

and $\{2, 3, 4\}$. $C_{\{1,2,3\}} = (\frac{1}{50} + \frac{1}{100} + \frac{1}{25})^{-1} = \frac{100}{7}$ and $C_{\{2,3,4\}} = (\frac{1}{100} + \frac{1}{25} + \frac{1}{20})^{-1} = 10$. The estimated available bandwidth of path $\langle a, b, c, d, e \rangle$ is $\min\{\frac{100}{7}, 10\} = 10$ Mbps.

We are going to show that we can find a simple scheduling mechanism to achieve $B(p)$ computed by (2). First, it is not difficult to find all the maximal cliques \mathbf{Q}_p containing only the links on path p . Let $B(e)$ be the transmission data rate of link e . It takes $\sum_{e \in \mathbf{q}} \frac{1}{B(e)}$ time for all links in a clique \mathbf{q} to sequentially transmit 1 unit of data. We let all the links on path p , which do not interfere with each other, to transmit concurrently. The total time for each link on p to transmit 1 unit of data is $\max_{\mathbf{q} \in \mathbf{Q}_p} \{\sum_{e \in \mathbf{q}} \frac{1}{B(e)}\}$. We use an example to show the conclusion. Following [23], if two links on path p interfere with each other, all links between them (include both the links) interfere with each other. Without loss of generality, assume \mathbf{Q}_p contains two maximal cliques $\mathbf{q}_1 = \{(v_0, v_1), (v_1, v_2), (v_2, v_3)\}$ and $\mathbf{q}_2 = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5)\}$ for path $p = \langle v_0, v_1, \dots, v_5 \rangle$. Let $e_i = (v_i, v_{i+1})$, where $0 \leq i \leq 4$. In this example, two cliques contain the different number of links, in order to consider a generic scenario. If the total time for each link in \mathbf{q}_1 to transmit 1 unit of data is larger than that in \mathbf{q}_2 , the time for link e_0 to transmit 1 unit of data is larger than the total time for e_3 and e_4 to transmit 1 unit of data. Since link e_0 does not interfere with e_3 or e_4 , when link e_0 transmits, either e_3 or e_4 can transmit. Thus, when e_0 completes transmitting 1 unit of data, both e_3 and e_4 have completed transmitting 1 unit of data. Therefore, the total time for each link on p to transmit 1 unit of data is the total time for \mathbf{q}_1 to transmit 1 unit of data. The above discussion implies that the path bandwidth calculated by (2) can be achieved. In other words, (2) gives an underestimation for the available path bandwidth.

The size of a maximal clique depends on how many links interfere with each other, which depends on the interference model adopted in the network. Due to the popularity of the 802.11 technology, we develop our work based on this MAC protocol. Both the two-way handshake DATA/ACK and the four-way handshake RTS/CTS/DATA/ACK of 802.11 require the receiver of a data packet to send an ACK back to the sender of the data packet. Therefore, for a packet transmission to be successful, both the sender and the receiver should not be interfered by other nodes. a is interfered by another node b if a is within the *interference range* of b . In other words, the transmissions on links (u, v) and (s, d) are successful at the same time if and only if both s and d are outside the interference ranges of u and v . This model is referred as the *bidirectional transmission model* [23] and the *Transmitter-Receiver Conflict*

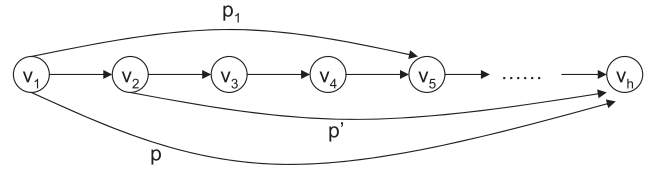


Fig. 4. Path bandwidth computation in a hop-by-hop manner.

Avoidance (TRCA) interference model [25] in the literatures, and is adopted by many existing works [3], [4], [27], [28]. (Fig. 2b is NOT constructed based on the TRCA model. We will come back to this later in this section.) Following [6], we define the transmission range of a node to be *one hop*, while the interference range to be r hops. To simplify our discussion, we set $r = 2$ [6]. It is worth noting that our results can be extended to any value of r . Moreover, our mechanism also works, after extension, on other commonly used interference models, such as the *protocol model* [28], in which as long as the receiver is free from interference, the transmission is regarded as successful.

Applying the hop count to approximate the distance will introduce some error for computing the estimated available path bandwidth. An example in [23] illustrates this situation. In Fig. 3, if node a is in the interference range of g , then link 1 interferes with 7. Assume that each link has the same available bandwidth B , the available bandwidth of this path is actually $\frac{2}{3}B$, while it is computed as $\frac{1}{3}B$ by using (2). Jia et al. [23] calls p a *detour route*, and other paths are called *direct routes*. Similar to [6], [18], [22], and [23], we do not consider detour routes when computing the available path bandwidth.

Both the conflict graphs in Figs. 2b and 3 assume $r = 1$, which is not the TRCA interference model we are using in this paper. In Fig. 2a, under the TRCA model, when a sends data to b , d is not allowed to transmit since it is in the interference range of b . This means that links 1 and 4 interfere with each other under the TRCA interference model. Then, each maximal clique contains four consecutive links. Based on the link bandwidth values in Example 1, if we apply the TRCA interference model, the estimated available bandwidth of path $\langle a, b, c, d, e \rangle$ is $(\frac{1}{50} + \frac{1}{100} + \frac{1}{25} + \frac{1}{20})^{-1} = \frac{25}{3}$, which is less than the available bandwidth calculated in Example 1. Given a path $p = \langle v_1, v_2, \dots, v_h \rangle$, let $B(k)$ be the estimated available bandwidth on the link between v_k and v_{k+1} . Under the TRCA interference model, the formula for estimating the available bandwidth of path p is as follows [6]:

$$B(p) = \min_{1 \leq k \leq h-4} C_k, \quad (3)$$

$$C_k = \left(\frac{1}{B(k)} + \frac{1}{B(k+1)} + \frac{1}{B(k+2)} + \frac{1}{B(k+3)} \right)^{-1}.$$

Given path $p = \langle v_1, v_2, \dots, v_h \rangle$, let $p' = \langle v_2, \dots, v_h \rangle$ and $p_1 = \langle v_1, v_2, v_3, v_4, v_5 \rangle$, as illustrated in Fig. 4. We can easily verify that $B(p) = \min\{B(p_1), B(p')\}$. This formula allows the estimated path bandwidth to be computed in a hop-by-hop manner. Although the works in [2], [6], and [16] apply this mechanism to compute the path bandwidth, no work has been found to propose an efficient path selection mechanism which satisfies the optimality requirement. That is, no existing protocol can provide the performance

guarantee for finding the maximum available bandwidth path by using (3).

In the following discussion, we assume that the interference range for each node is the same, which is modeled as 2-hop count. In practice, the interference range for different node may be different. The following discussion focuses on introducing a new way to design a routing metric with the isotonic property. We will not mention the case that the interference ranges are different for different nodes due to the space limitation. Actually, our protocol can be easily extended for this case.

4 QoS ROUTING PROTOCOL

In this section, we first present our path selection mechanism. It is based on the distance-vector mechanism. We give the necessary and sufficient condition to determine whether a path is not worthwhile to be advertised. We then describe our new isotonic path weight. We show that the routing protocol based on this new path weight satisfies the optimality requirement [7], [8]. Afterward, we present our hop-by-hop packet forwarding mechanism which satisfies the consistency requirement. We apply (3) to estimate the available bandwidth of a path. To simplify our discussion, in the rest of our paper, we use "available bandwidth" instead of "estimated available bandwidth" when the context is clear. On the other hand, "widest path" refers to the path that has the maximum estimated available bandwidth.

4.1 Path Selection

We would like to develop a distance-vector based mechanism. In the traditional distance-vector mechanism, a node only has to advertise the information of its own best path to its neighbors. Each neighbor can then identify its own best path. In Section 1, we mentioned that if a node only advertises the widest path from its own perspective, its neighbors may not be able to find the widest path. To illustrate, consider the network in Fig. 1 where the number of each link is the available bandwidth on the link.

Example 2. Based on (3), $B(\langle v, e, f, g, d \rangle)$ (upper path) = 2.5 and $B(\langle v, a, b, c, d \rangle)$ (lower path) = $\frac{20}{9} = 2.22$. On the other hand, $B(\langle s, v, e, f, g, d \rangle)$ (upper path) = 2 and $B(\langle s, v, a, b, c, d \rangle)$ (lower path) = $\frac{20}{9} = 2.22$. It can be observed that while the upper path is a better path for v , it does *not* lie on widest path of s . Thus, if v does not advertise the lower path to s , s would not be able to identify its own widest path. It violates the *optimality* requirement of a routing protocol. Even s could identify the widest path, when s sends the packet to v , and v sends out the packet on its own widest path $\langle v, e, f, g, d \rangle$, the packet still cannot traverse on the widest path of s . It violates the *consistency* requirement.

In order to assure that the widest path from each node to a destination can be identified, a trivial way is to advertise all the possible paths to a destination. This is definitely too expensive. On the other hand, as long as we advertise every path which is a subpath of a widest path (e.g., $\langle v, a, b, c, d \rangle$ is a subpath of the widest path of $\langle s, v, a, b, c, d \rangle$), we allow every node to identify its own widest path. Thus, to reduce the overhead, we should not advertise those paths that

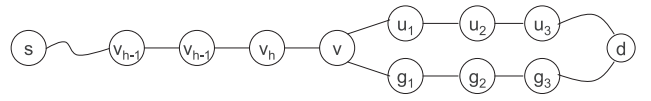


Fig. 5. Illustration for path comparison.

would not be a subpath of any widest path. In this section, we study the sufficient and necessary condition for a node to determine whether a path must not be the subpath of any maximum bandwidth path.

We first introduce some notations. The bandwidth of the link from a to b is $B(a, b)$. Given a path $p = \langle v_1, v_2, \dots, v_h \rangle$, let $WB(p) = B(p)$, $FB(p) = B(v_1, v_2)$, $TB(p) = WB(\langle v_1, v_2, v_3 \rangle)$, and $HB(p) = WB(\langle v_1, v_2, v_3, v_4 \rangle)$. In other words, $WB(p)$ is the bandwidth of the whole path, $FB(p)$ is the bandwidth on the first link, $TB(p)$ is the bandwidth of the subpath composed of the first two links, and $HB(p)$ is the bandwidth of the subpath composed of the first three links. We further denote the concatenation of paths p_1 and p_2 as $p_1 \oplus p_2$. Lemma 2 gives the sufficient condition to determine a path is not a subpath of any widest path.

Lemma 2. Suppose that p_1 and p_2 are two paths from v to d . If $WB(p_1) \geq WB(p_2)$, $HB(p_1) \geq HB(p_2)$, $TB(p_1) \geq TB(p_2)$, and $FB(p_1) \geq FB(p_2)$, then $WB(p \oplus p_1) \geq WB(p \oplus p_2)$ for any path p that ends at v .

Proof. Let $p_1 = \langle v, u_1, \dots, u_n, d \rangle$, $p_2 = \langle v, g_1, \dots, g_m, d \rangle$, and $p = \langle s, v_1, \dots, v_h, v \rangle$ as illustrated in Fig. 5. Let $p_{1,1} = \langle v_{h-2}, v_{h-1}, v_h, v, u_1 \rangle$, $p_{1,2} = \langle v_{h-1}, v_h, v, u_1, u_2 \rangle$, $p_{1,3} = \langle v_h, v, u_1, u_2, u_3 \rangle$. Recall that the bandwidth of a path is the bandwidth of the bottleneck clique, and each clique consists of four links. According to (3)

$$WB(p \oplus p_1) = \min\{WB(p), WB(p_{1,1}), WB(p_{1,2}), WB(p_{1,3}), WB(p_1)\}. \quad (4)$$

Similarly, let $p_{2,1} = \langle v_{h-2}, v_{h-1}, v_h, v, g_1 \rangle$, $p_{2,2} = \langle v_{h-1}, v_h, v, g_1, g_2 \rangle$, $p_{2,3} = \langle v_h, v, g_1, g_2, g_3 \rangle$. We have

$$WB(p \oplus p_2) = \min\{WB(p), WB(p_{2,1}), WB(p_{2,2}), WB(p_{2,3}), WB(p_2)\}. \quad (5)$$

$p_{1,1}$ and $p_{2,1}$ differ only on the last link. $FB(p_1) \geq FB(p_2)$ implies $B(v, u_1) \geq B(v, g_1)$, and so $WB(p_{1,1}) \geq WB(p_{2,1})$.

$$\begin{aligned} TB(p_1) &\geq TB(p_2) \\ \Rightarrow \left(\frac{1}{B(v, u_1)} + \frac{1}{B(u_1, u_2)} \right)^{-1} &\geq \left(\frac{1}{B(v, g_1)} + \frac{1}{B(g_1, g_2)} \right)^{-1} \\ \Rightarrow \frac{1}{B(v, g_1)} + \frac{1}{B(g_1, g_2)} &\geq \frac{1}{B(v, u_1)} + \frac{1}{B(u_1, u_2)} \\ \Rightarrow \frac{1}{B(v_{h-1}, v_h)} + \frac{1}{B(v_h, v)} + \frac{1}{B(v, g_1)} + \frac{1}{B(g_1, g_2)} &\geq \\ \frac{1}{B(v_{h-1}, v_h)} + \frac{1}{B(v_h, v)} + \frac{1}{B(v, u_1)} + \frac{1}{B(u_1, u_2)} & \\ \Rightarrow \left(\frac{1}{B(v_{h-1}, v_h)} + \frac{1}{B(v_h, v)} + \frac{1}{B(v, u_1)} + \frac{1}{B(u_1, u_2)} \right)^{-1} &\geq \\ \left(\frac{1}{B(v_{h-1}, v_h)} + \frac{1}{B(v_h, v)} + \frac{1}{B(v, g_1)} + \frac{1}{B(g_1, g_2)} \right)^{-1} & \\ \Rightarrow WB(p_{1,2}) &\geq WB(p_{2,2}). \end{aligned} \quad (6)$$

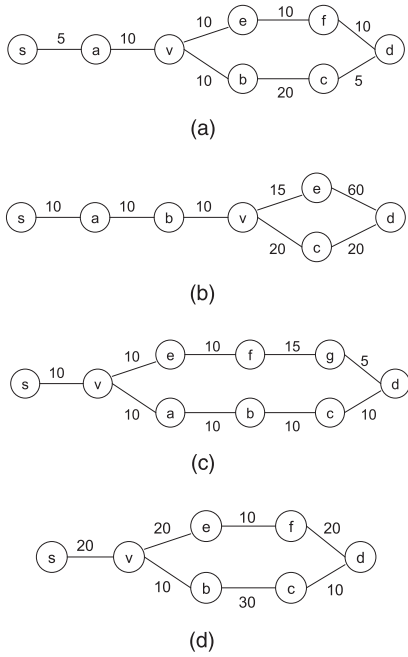


Fig. 6. Examples of network topologies.

Based on a similar argument, $HB(p_1) \geq HB(p_2) \Rightarrow WB(p_{1,3}) \geq WB(p_{2,3})$. Since $WB(p_1) \geq WB(p_2)$, $WB(p_{1,1}) \geq WB(p_{2,1})$, $WB(p_{1,2}) \geq WB(p_{2,2})$, and $WB(p_{1,3}) \geq WB(p_{2,3})$, we have $WB(p \oplus p_1) \geq WB(p \oplus p_2)$. \square

Definition 1. We call the condition $WB(p_1) \geq WB(p_2)$, $FB(p_1) \geq FB(p_2)$, $TB(p_1) \geq TB(p_2)$, and $HB(p_1) \geq HB(p_2)$ the *pruning condition*.

Lemma 2 says that the pruning condition is a *sufficient* condition for v to determine p_2 is not worthwhile to be advertised because $p \oplus p_1$ must be better than $p \oplus p_2$ for every p , which implies p_2 can never be a subpath of a widest path. We say p_1 *prunes* p_2 if $WB(p \oplus p_1) \geq WB(p \oplus p_2)$ for every p . The pruning condition contains four subconditions. To show that each of the subconditions is also necessary for v to conclude p_2 should not be advertised, we use examples to illustrate that if either one of the subconditions $WB(p_1) \geq WB(p_2)$, $HB(p_1) \geq HB(p_2)$, $TB(p_1) \geq TB(p_2)$, or $FB(p_1) \geq FB(p_2)$ does not hold, $WB(p \oplus p_1) \not\geq WB(p \oplus p_2)$ may not hold even the other three are satisfied. In Figs. 1, 6a, 6b, 6c, and 6d, there are two paths from v to d . Denote the upper path as p_1 and the lower path as p_2 . We further denote the unique path from s to v as p .

Fig. 1 shows subcondition $HB(p_1) \geq HB(p_2)$ is necessary. $WB(p_1) = \frac{5}{2} > WB(p_2) = \frac{20}{9}$, $TB(p_1) = TB(p_2) = 5$, and $FB(p_1) = FB(p_2) = 10$, but $HB(p_1) = \frac{10}{3} < HB(p_2) = 4$. It turns out that $WB(p \oplus p_1) = 2 < WB(p \oplus p_2) = \frac{20}{9}$. In this situation, if v just advertises p_1 , s will not find the widest path from its own perspective.

Fig. 6a shows subcondition $TB(p_1) \geq TB(p_2)$ is necessary. $WB(p_1) = HB(p_1) = \frac{10}{3} > WB(p_2) = HB(p_2) = \frac{20}{7}$ and $FB(p_1) = FB(p_2) = 10$, but $TB(p_1) = 5 < TB(p_2) = \frac{20}{3}$ and $WB(p \oplus p_1) = 2 < WB(p \oplus p_2) = \frac{20}{9}$.

In Fig. 6b, since p_1 and p_2 only have two hop counts, respectively, we have $WB(p_1) = HB(p_1) = TB(p_1) = 12$ and

$WB(p_2) = HB(p_2) = TB(p_2) = 10$. In this topology, we have $WB(p_1) \geq WB(p_2)$, $HB(p_1) \geq HB(p_2)$, and $TB(p_1) \geq TB(p_2)$, but $FB(p_1) = 15 < FB(p_2) = 20$. Also, $WB(p \oplus p_1) = \frac{30}{11} < WB(p \oplus p_2) = \frac{20}{7}$.

In Fig. 6c, we have $HB(p_1) = \frac{15}{4} \geq HB(p_2) = \frac{10}{3}$, $TB(p_1) = TB(p_2) = 5$, and $FB(p_1) = FB(p_2) = 10$, but $WB(p_1) = \frac{15}{7} < WB(p_2) = \frac{5}{2}$. We then obtain $WB(p \oplus p_1) = \frac{15}{7} < WB(p \oplus p_2) = \frac{5}{2}$.

Note that it is possible that $WB(p \oplus p_1) \geq WB(p \oplus p_2)$ but some of the subconditions do not hold. For example, in Fig. 6d, by (3), $WB(p_1) = HB(p_1) = 5$, $WB(p_2) = HB(p_2) = \frac{30}{7}$, $TB(p_1) = \frac{20}{3}$, $TB(p_2) = \frac{15}{2}$, $FB(p_1) = 20$, and $FB(p_2) = 10$. We have $WB(p_1) \geq WB(p_2)$, $HB(p_1) \geq HB(p_2)$, and $FB(p_1) \geq FB(p_2)$, but $TB(p_1) < TB(p_2)$. However, $WB(p \oplus p_1) = 4 > WB(p \oplus p_2) = \frac{60}{17}$. In other words, the pruning condition is *not* a necessary condition for $WB(p \oplus p_1) \geq WB(p \oplus p_2)$ for every p , p_1 , and p_2 . However, we are interested in whether v can determine a path should not be advertised based on the information. From the examples in Fig. 6, v cannot tell for sure whether p_1 prunes p_2 if any one of the subconditions does not hold. By allowing p_2 to be advertised as long as either one of the subconditions does not hold, it is possible that v advertises unnecessary path information but it is guaranteed that v advertises *every* path that is a subpath of a widest path.

4.2 Isotonic Path Weight

In this section, we introduce our new isotonic path weight, while the next section describes how we use the path weight to construct routing tables. The isotonicity property of a path weight is the necessary and sufficient condition for developing a routing protocol satisfying the optimality and consistency requirements. We first describe the definition of isotonicity introduced in [7] and [8].

Definition 2. *Left-isotonicity* The quadruplet (S, \oplus, w, \succeq) is left-isotonic if $w(a) \succeq w(b)$ implies $w(c \oplus a) \succeq w(c \oplus b)$, for all $a, b, c \in S$, where S is a set of paths, \oplus is the path concatenation operation, w is a function which maps a path to a weight, and \succeq is the order relation.

Given two paths p_1 and p_2 from a node s to d , assume that p_1 is better than p_2 by comparing their weights. If the path weight used is left-isotonic, Definition 2 tells us that, given any path p' from a node v to s , $p' \oplus p_1$ must be better than $p' \oplus p_2$. Available bandwidth of a path alone is *not* left-isotonic, and it is illustrated in Example 2 in Section 4.1. Now, we present the proposed left-isotonic path weight, called *composite available bandwidth* (CAB), as follows:

Definition 3. Given a path p , the composite available bandwidth of p , denoted by $\vec{w}(p)$, is $(\omega_1(p), \omega_2(p), \omega_3(p), \omega_4(p))$ where $\omega_1(p) = WB(p)$, $\omega_2(p) = HB(p)$, $\omega_3(p) = TB(p)$, and $\omega_4(p) = FB(p)$. $\vec{w}(p_1) \succeq \vec{w}(p_2)$ iff $\omega_1(p_1) \geq \omega_1(p_2)$, $\omega_2(p_1) \geq \omega_2(p_2)$, $\omega_3(p_1) \geq \omega_3(p_2)$, and $\omega_4(p_1) \geq \omega_4(p_2)$.

Definition 4. Given two paths p_1 and p_2 , if $\vec{w}(p_1) \succeq \vec{w}(p_2)$, we call p_1 dominates p_2 . If we cannot find a path dominating p_1 , we call p_1 a nondominated path.

Theorem 1. Composite available bandwidth is left-isotonic.

TABLE 1
The Distance Table of Node a in Fig. 6b

Destination	Neighbor	Non-dominated path			
		$\bar{\omega}(p)$	NF(p)	NS(p)	NT(p)
d	b	$(\frac{60}{11}, \frac{60}{11}, 6, 10)$	v	e	d
	b	$(5, 5, \frac{20}{3}, 10)$	v	c	d

Proof. Let $p_1 = \langle v, u_1, u_2, \dots, u_n, d \rangle$ and $p_2 = \langle v, g_1, g_2, \dots, g_m, d \rangle$ such that $\bar{\omega}(p_1) \succeq \bar{\omega}(p_2)$. Let $p_3 = \langle s, v_1, \dots, v_h, v \rangle$ from s to v . Denote $p = p_3 \oplus p_1$ and $p' = p_3 \oplus p_2$. We are going to show that $\bar{\omega}(p) \succeq \bar{\omega}(p')$.

By Lemma 2, we have $\omega_1(p) \geq \omega_1(p')$ for any p_3 .

If p_3 has three or more links, both paths p and p' share the same first three links. We thus have $\omega_2(p) = \omega_2(p')$, $\omega_3(p) = \omega_3(p')$, and $\omega_4(p) = \omega_4(p')$. In this case, we have $\bar{\omega}(p) \succeq \bar{\omega}(p')$.

If p_3 consists of only one link, we have $\omega_4(p) = \omega_4(p')$. On the other hand, $\frac{1}{\omega_3(p)} = \frac{1}{B(s,v)} + \frac{1}{\omega_4(p_1)}$ and $\frac{1}{\omega_3(p')} = \frac{1}{B(s,v)} + \frac{1}{\omega_4(p_2)}$. Since $\omega_4(p_1) \geq \omega_4(p_2)$, we have $\omega_3(p) \geq \omega_3(p')$. We also have $\frac{1}{\omega_2(p)} = \frac{1}{B(s,v)} + \frac{1}{\omega_3(p_1)}$, $\frac{1}{\omega_2(p')} = \frac{1}{B(s,v)} + \frac{1}{\omega_3(p_2)}$. Since $\omega_3(p_1) \geq \omega_3(p_2)$, it holds that $\omega_2(p) \geq \omega_2(p')$. We thus prove that $\bar{\omega}(p) \succeq \bar{\omega}(p')$ when p_3 is an one-hop path.

With a similar argument, we can also prove $\omega(p) \geq \omega(p')$ when p_3 consists of two links. \square

Refer back to Example 2, the CAB of path $p_1 = \langle v, e, f, g, d \rangle$ is $(\frac{5}{3}, \frac{10}{3}, 5, 10)$ and the CAB of path $p_2 = \langle v, a, b, c, d \rangle$ is $(\frac{20}{9}, 4, 5, 10)$. Neither $\bar{\omega}(p_1) \succeq \bar{\omega}(p_2)$ nor $\bar{\omega}(p_2) \succeq \bar{\omega}(p_1)$. Both paths are nondominated paths.

4.3 Table Construction and Optimality

The isotonicity property of the proposed path weight allows us to develop a routing protocol that can identify the maximum bandwidth path from each node to each destination. In particular, it tells us whether a path is worthwhile to be advertised, meaning whether a path is a potential subpath of a widest path. In our routing protocol, if a node finds a new nondominated path, it will advertise this path information to its neighbors. We call the packet carrying the path information the *route packet*. For each nondominated path p from s to d , s advertises the tuple $(s, d, \text{NF}(p), \text{NS}(p), \text{NT}(p), \bar{\omega}(p))$ to its neighbors in a route packet. NF(p), NS(p), and NT(p) are the next hop, the second next hop, and the third next hop on p from s , respectively. Based on the information contained in a route packet, each node knows the information about the first four hops of a path identified. This information is necessary for consistent routing, which will be discussed in details later.

Each node keeps two tables: *distance table* and *routing table*. Node s puts all the nondominated paths advertised by its neighbors in its *distance table*. It keeps all the nondominated paths found by s itself in its *routing table*. When s receives an advertisement $(u, d, \text{NF}(p), \text{NS}(p), \text{NT}(p), \bar{\omega}(p))$ from u which represents a nondominated path p from u to d , s removes all the locally recorded paths from u to d which are dominated by p . Denote p' as the path from s to d which is one-hop extended from p . s computes the CAB of p' as follows:

TABLE 2
The Routing Table of Node a in Fig. 6b

Destination	QoS to Destination				
	$\bar{\omega}(p)$	NF(p)	NS(p)	NT(p)	NU(p)
d	$(\frac{60}{17}, \frac{15}{4}, 5, 10)$	b	v	e	d
	$(\frac{24}{7}, \frac{24}{5}, 5, 10)$	b	v	c	d

$$\begin{cases} \frac{1}{\omega_1(p')} = \min \left\{ \frac{1}{B(s,u)} + \frac{1}{\omega_2(p)}, \omega_1(p) \right\}, \\ \frac{1}{\omega_2(p')} = \frac{1}{B(s,u)} + \frac{1}{\omega_3(p)}, \\ \frac{1}{\omega_3(p')} = \frac{1}{B(s,u)} + \frac{1}{\omega_4(p)}, \\ \omega_4(p') = B(s,u). \end{cases} \quad (7)$$

By comparing $\bar{\omega}(p')$ with the CABs of the paths from s to d in the routing table, s can determine whether p' is a nondominated path and remove the paths that are dominated by p' . If p' is a nondominated path, s generates an advertisement $(s, d, u, \text{NF}(p), \text{NS}(p), \bar{\omega}(p'))$.

Table 1 illustrates the distance table of node a in Fig. 6b. Based on its distance table, a knows that there are two nondominated paths from b to destination d . Path $\langle b, v, e, d \rangle$ has a CAB of $(\frac{60}{11}, \frac{60}{11}, 6, 10)$ and path $\langle b, v, c, d \rangle$'s CAB is $(5, 5, \frac{20}{3}, 10)$. Based on the two nondominated paths from b to d , a finds two nondominated paths from itself to d and puts the information in the routing table. Table 2 illustrates the routing table of a . NU(p) denotes the fourth next hop on p . For each path p , the source keeps the subpath of the first four hops on p . NF(p) is the neighbor that sent the nondominated path to a . NS(p), NT(p), and NU(p) are the NF(p'), NS(p'), and NT(p'), respectively, where p' is the nondominated path used to construct p .

To ease our discussion, we use the tuple $(s, d, \text{NF}(p), \text{NS}(p), \text{NT}(p), \text{NU}(p), \bar{\omega}(p))$ to identify an entry in the routing table of node s , where p is a nondominated path from s to d . The implementation of the above discussion is illustrated in *Procedure QoS_Update*. Now, we would like to illustrate the process using the simple network in Fig. 6b. Suppose that each node computes the maximum bandwidth path from itself to d . Initially, the distance table and the routing table of each node are empty. In the first step, e finds a one-hop path from itself to d with the CAB $(60, 60, 60, 60)$. It will add an entry $(e, d, d, d, d, d, (60, 60, 60, 60))$ in its routing table and generate an advertisement $\mathcal{R}_e = (e, d, d, d, d, (60, 60, 60, 60))$. When node v receives \mathcal{R}_e , it includes the path information contained in \mathcal{R}_e in its distance table. By using (7), it obtains a new path with the CAB $(12, 12, 12, 15)$. It then adds an entry $(v, d, e, d, d, d, (12, 12, 12, 15))$ in its routing table and generates an advertisement $\mathcal{R}_{v,1} = (v, d, e, d, d, (12, 12, 12, 15))$. With the similar method, node c will add an entry $(c, d, d, d, d, d, (20, 20, 20, 20))$ in its routing table and generate an advertisement $\mathcal{R}_c = (c, d, d, d, d, (20, 20, 20, 20))$. Based on \mathcal{R}_e , node v gets another new path with $(10, 10, 10, 20)$. Since this path is also a nondominated path, node v will add another entry $(v, d, c, d, d, d, (10, 10, 10, 20))$ in its routing table and generate another advertisement $\mathcal{R}_{v,2} = (v, d, c, d, d, d, (10, 10, 10, 20))$. With the same reason,

based on $\mathcal{R}_{v,1}$ and $\mathcal{R}_{v,2}$, node b will obtain two nondominated paths with the CABs $(\frac{60}{11}, \frac{60}{11}, 6, 10)$ and $(5, 5, \frac{20}{3}, 10)$, respectively. b will advertise two route packets $\mathcal{R}_{b,1} = (b, d, v, e, d, (\frac{60}{11}, \frac{60}{11}, 6, 10))$ and $\mathcal{R}_{b,2} = (b, d, v, c, d, (5, 5, \frac{20}{3}, 10))$. After a receiving these two packets, it will keep the path information in its distance table, as shown in Table 1. Then, a will get two nondominated paths which are recorded in its routing table, as shown in Table 2. Node a also generates two advertisements $\mathcal{R}_{a,1} = (a, d, b, v, e, (\frac{60}{17}, \frac{15}{4}, 5, 10))$ and $\mathcal{R}_{a,2} = (a, d, b, v, c, (\frac{24}{7}, \frac{24}{5}, 5, 10))$.

Finally, node s will get two paths with the CABs $(\frac{30}{11}, \frac{10}{3}, 5, 10)$ and $(\frac{20}{7}, \frac{10}{3}, 5, 10)$. As $(\frac{20}{7}, \frac{10}{3}, 5, 10) \succeq (\frac{30}{11}, \frac{10}{3}, 5, 10)$, we can see that the latter path going through $\langle s, a, b, v, c \rangle$ is better than the first path going through $\langle s, a, b, v, e \rangle$. That is, there is only one entry $(s, d, a, b, v, c, (\frac{20}{7}, \frac{10}{3}, 5, 10))$ in the routing table of node s .

Theorem 2. *Our routing protocol satisfies the optimality requirement.*

Proof. We now prove that each node v_1 must find the maximum bandwidth path to destination v_n , denoted by $\langle v_1, v_2, \dots, v_n \rangle$. Suppose that the widest path between v_1 and v_n is unique, we now prove that each on-path node v_i must advertise the information of the subpath $\langle v_i, \dots, v_n \rangle$ to v_{i-1} , where $i = 2, \dots, n-1$, by induction.

As the basic step, since v_{n-1} is a direct neighbor of v_n , it must advertise the information of path $\langle v_{n-1}, v_n \rangle$ to v_{n-2} .

For the inductive step, assume that v_k advertises the information of path $\langle v_k, \dots, v_n \rangle$ to v_{k-1} . If v_{k-1} does not advertise path $p_1 = \langle v_{k-1}, v_k, \dots, v_n \rangle$ to v_{k-2} , there must exist a path $p_2 = \langle v_{k-1}, g_1, g_2, \dots, g_m, v_n \rangle$ which dominates p_1 . We thus have $\vec{\omega}(p_2) \succ \vec{\omega}(p_1)$. Denote $p = \langle v_1, \dots, v_{k-1} \rangle$. By Theorem 1 and Definition 2, we have $\vec{\omega}(p \oplus p_2) \succeq \vec{\omega}(p \oplus p_1)$. This means $p \oplus p_2$ has larger available bandwidth than path $\langle v_1, \dots, v_n \rangle$, which implies that $\langle v_1, \dots, v_n \rangle$ is not the maximum bandwidth path, which leads to contradiction. \square

We have proved that our routing protocol satisfies the optimality requirement, meaning a node can definitely identify a widest path to every destination through advertisement from its neighbors. However, it is not sufficient to ensure a packet does traverse over the widest path. We need a *consistent* hop-by-hop packet forwarding mechanism to send a packet along the intended route of the sender. The consistency property also ensures loop-free routing [7].

Procedure QoS_Update of Node s

/*

s receives advertisement $(u, d, \text{NF}(p), \text{NS}(p), \text{NT}(p), \vec{\omega}(p))$

/

```

1: for each path  $p_1$  from  $u$  to  $d$  in the distance table of  $s$  do
2:   if  $\vec{\omega}(p) \succ \vec{\omega}(p_1)$  then
3:     Remove  $p_1$  from the distance table
4:    $p' \leftarrow \langle s, u \rangle \oplus p$ 
5:   Calculate  $\vec{\omega}(p')$  using (7)
6: for each path  $p_2$  from  $s$  to  $d$  in the routing table of  $s$  do
7:   if  $\vec{\omega}(p') \succ \vec{\omega}(p_2)$  then
8:     Remove  $p_2$  from the routing table
9:   else
10:    if  $\vec{\omega}(p_2) \succeq \vec{\omega}(p')$  then

```

```

11:    return

```

```

12: Add  $(s, d, u, \text{NF}(p), \text{NS}(p), \text{NT}(p), \vec{\omega}(p'))$  in the routing table

```

```

13: Advertise  $(s, d, u, \text{NF}(p), \text{NS}(p), \vec{\omega}(p'))$ 

```

4.4 Packet Forwarding and Consistency

Suppose that node s wants to transmit traffic to d along the widest path $p = \langle s, v_1, \dots, v_n, d \rangle$. Then, each node v_i on this path should make the consistent decision so that the traffic does travel along p . However, as mentioned earlier in Example 2, the widest path from v_i to d may not be a subpath on p . If v_i selects the next hop according to its widest path to d , the traffic may not be sent along the best path from s to d . In this section, we present the consistent hop-by-hop packet forwarding mechanism.

In a traditional hop-by-hop routing protocol, a packet carries the destination of the packet, and when a node receives a packet, it looks up the next hop by the destination only. In our mechanism, apart from the destination, a packet also carries a *Routing Field* which specifies the next four hops the packet should traverse. When a node receives this packet, it identifies the path based on the information in the Routing Field. It updates the Routing Field and sends it to the next hop.

For example, assume that node s in Fig. 6b wants to send a packet to d . In the previous section, we know that there is one entry $(s, d, a, b, v, c, (\frac{20}{7}, \frac{10}{3}, 5, 10))$ in the routing table of s . By looking up the routing table, the Routing Field $\langle a, b, v, c \rangle$ will be put in the packet. The packet is sent to the next hop a . When a receives the data packet from s , it knows that the packet should traverse over subpath $\langle a, b, v, c \rangle$. Thus, it locates the path p where $\text{NF}(p) = b$ and $\text{NS}(p) = v$ and $\text{NT}(p) = c$ in its routing table. Table 2 shows that the next four hop of the path going through $\langle a, b, v, c \rangle$ is $\text{NU}(p) = d$. Then, it updates the Routing Field to $\langle b, v, c, d \rangle$ and sends it to b . We can see that the data packet does traverse over the widest path from s to d in this example.

In our packet forwarding mechanism, each intermediate node determines the fourth next hop but not the next hop as in the traditional mechanism. Our packet forwarding mechanism still requires each intermediate node to make route decision based on its routing table. Besides, only the information of the first few hops of a path is kept in the routing table in each node and the routing field in a packet. Therefore, our mechanism possesses the same characteristics of a hop-by-hop packet routing mechanism [7], and is a distributed packet forwarding scheme.

Corollary 1. *Two paths p_1 and p_2 from s to d traverse the same next three hops v_1, v_2, v_3 . If $\text{WB}(p_1) \geq \text{WB}(p_2)$, $\vec{\omega}(p_1) \succeq \vec{\omega}(p_2)$.*

If two paths p_1 and p_2 traverse the same next three hops, we have $\text{HB}(p_1) = \text{HB}(p_2)$, $\text{TB}(p_1) = \text{TB}(p_2)$, and $\text{FB}(p_1) = \text{FB}(p_2)$. If $\text{WB}(p_1) \geq \text{WB}(p_2)$, we have $\vec{\omega}(p_1) \succeq \vec{\omega}(p_2)$ by Definition 3.

Theorem 3. *Our routing protocol satisfies the consistency requirement.*

Proof. Assume that node v_1 wants to transmit a data packet over path $\langle v_1, v_2, \dots, v_n \rangle$. In our packet forwarding mechanism, the Routing Field of each data packet

specifies the next four hops. When v_j receives a data packet from v_{j-1} , if it can update the Routing Field of the packet to $\langle v_{j+1}, v_{j+2}, v_{j+3}, v_{j+4} \rangle$, where $2 \leq j \leq n-4$, our hop-by-hop packet forwarding scheme satisfies the consistency requirement. We prove by induction.

In the basic step, according to the routing table of v_1 , v_1 knows the next four hops on path $\langle v_1, \dots, v_n \rangle$. Therefore, it sets the Routing Field of the data packet to $\langle v_1, v_2, v_3, v_4 \rangle$.

For the inductive step, assume that v_j receives the data packet from v_{j-1} , and the Route Field of this data packet specifies that this packet should be transmitted on the subpath $\langle v_j, v_{j+1}, v_{j+2}, v_{j+3} \rangle$, where $j = 2, \dots, n-4$. According to the proof of Theorem 2, if v_1 obtains the path $\langle v_1, \dots, v_n \rangle$, each node v_j has obtained and advertised the path $\langle v_j, \dots, v_n \rangle$, where $2 \leq j < n$. In other words, v_j must consider $\langle v_j, \dots, v_n \rangle$ as a nondominated path. By Corollary 1, there exists only one nondominated path from v_j to v_n going through the subpath $\langle v_j, v_{j+1}, v_{j+2}, v_{j+3} \rangle$. This implies that the path recorded in the routing table that goes through $\langle v_j, v_{j+1}, v_{j+2}, v_{j+3} \rangle$ must be the path $\langle v_j, \dots, v_n \rangle$. Since the routing table of v_j also keeps the next four hops on path $\langle v_j, \dots, v_n \rangle$, v_j will update the Routing Field of the data packet to $\langle v_{j+1}, v_{j+2}, v_{j+3}, v_{j+4} \rangle$. This theorem is thus proved. \square

We can see that the space complexity and the advertisement complexity of our routing protocol are directly related to the number of nondominated paths from each node to each destination. Denote \mathcal{A} as the average number of the neighbors of each node. Since there is only one nondominated path going through the same first three links, the maximum number of nondominated paths from each node to a destination is $\mathcal{O}(\mathcal{A}^3)$. Therefore, our mechanism is a polynomial-time routing algorithm for computing the maximum throughput path.

Note that the consistency discussed in the above assumes that each node has the accurate state information about its neighbors. Route update may also cause inconsistency, as discussed later. However, such inconsistency is independent on which routing metric or what kind of the packet forwarding mechanism is applied, while it is completely due to the delay of the route update propagation. Therefore, such inconsistency exists in all distributed routing protocols.

4.5 Route Update

After the network accepts a new flow or releases an existing connection, the local available bandwidth of each node will change, and thus the widest path from a source to a destination may be different. When the change of the local available bandwidth of a node is larger than a threshold (say 10 percent), the node will advertise the new information to its neighbors. After receiving the new bandwidth information, the available bandwidth of a path to a destination may be changed. Although the node is static, the network state information changes very often. Therefore, our routing protocol applies the route update mechanism in DSDV [29]. Based on DSDV, each routing entry is tagged with a sequence number which is originated by the destination, so that nodes can quickly distinguish

stale routes from the new ones. Each node periodically transmits updates and transmits updates immediately when significant new route information is available. Given two route entries from a source to a destination, the source always selects the one the larger sequence number, which is newer, to be kept in the routing table. Only if two entries have the same sequence number, our path comparison is used to determine which path should be kept.

Due to the delay of the route update propagation, it is possible that route information kept in some nodes is inconsistent. For instance, the widest path kept in the routing table may not be the widest anymore. Routing loops may occur as well. The situations are referred as inconsistency due to transient route updates, which is different from the definition used in [7]. In [7] and this paper, we consider whether packets can be routed on the computed widest path when the routing tables are stable. How to avoid loops when routing tables change is an important but difficult problem, and is outside the scope of this paper. We refer readers to [29] for the techniques to reduce route update inconsistencies in the distance-vector protocol which can be applied in our mechanism as well.

5 PERFORMANCE EVALUATION

In this section, we conduct the simulation experiments under NS2 [30] to investigate the performance of our routing protocol for finding the maximum available bandwidth path. We compare our proposed path weight, *Composite Available Bandwidth*, with some existing path weights.

5.1 Routing Metrics

The earliest metric proposed for finding the maximum available bandwidth path is ETX [10]. The ETX metric of each link l is defined as $ETX_l = \frac{1}{p_l}$, where p_l denotes the packet loss probability on link l at the MAC layer. p_l is estimated by proactively broadcasting the dedicated link probe packets. Couto et al. [10] give the details on how to calculate p_l . In our simulation, we completely follow the instructions presented in [10] to compute p_l . As we consider single-channel networks in this work, we would not compare with metrics that are developed for the multi-channel situation, such as ETT [12]. Another metric we compare is the Interference-aware Resource Usage (IRU) proposed in [14], which is defined as $IRU_l = ETX \times |N_l|$, where N_l consists of the neighbors whose transmission interfere with the transmission on link l . Because we assume all data packets have the same size and all the links have the same raw data rate, the performance of IRU is the same as the performance of the CATT metric proposed in [15].

5.2 Simulation Settings

Unless otherwise stated, the simulation experiment setup is as follows: The MAC layer protocol is IEEE 802.11 with RTS/CTS. The radio transmission range and the carrier-sensing range (interference range) are 250 and 550 m, respectively. The bandwidth of the wireless channel is 1 Mbps. All the traffics are CBR flows with the packet size of 1,000 Bytes. The bit error rate of each channel is zero.

In order to simulate different link available bandwidths, we generate some background traffic which takes up the

capacities of the links by randomly deploying some one-hop flows in the network. The data rates of the one-hop flows follow the uniform distribution $U(1, 20)$ Kbps. After accepting all these one-hop flows, the available bandwidth of each link is different. Each destination then initiates the path computation process to compute the best paths from all the other nodes to itself in the network. When a node receives a connection to a destination, it has the widest path to the destination kept in its routing table. We then randomly select a pair of nodes which are not direct neighbors. A CBR traffic is then established between this pair of nodes. This traffic is called a new flow or a multihop flow to differentiate with the existing background one-hop flows.

When the traffic rate of the multihop flow is larger than the actual available bandwidth of the best path, accepting the new flow will violate the bandwidth guarantees of the existing flows. In our simulation, in order to reserve enough bandwidth resources for the existing flows, we always let an existing flow have a higher priority to use a link that a node always transmits the higher priority packet before a lower priority one. We set the buffer size of each node to be 50 packets.

To understand whether the priority mechanism works, we study the throughput of the existing flows before and after a new flow is introduced. For example, in one instance of the simulation, we randomly deploy 200 one-hop flows in the network, where there are around 400 links in total. The total throughput of these one-hop flows is 4.1882 Mbps. We then select a pair of nodes that are farthest apart in terms of hop count in the network. We apply our algorithm to find the widest path between this node pair, and push a flow of 300 Kbps, which is much larger than the available bandwidth, on this path. We measured the total throughput of the existing flows again and it is 4.1730 Mbps, while the throughput of the multihop flow is 62.385 Kbps. We can see that the new flow does not take up the capacity meant to be allocated for the existing flows. It means that we can almost fairly measure the actual throughput of the best paths found by the different algorithms under the condition that the bandwidth guarantee of the existing flows is not violated.

5.3 Simulation Results

In our simulation experiments, the random network topology was generated by the “setdest” tool provided in the NS2 simulator. We define the distance between two nodes as the minimum hop-count between them. For each possible node pair distance in a network, we randomly select some node pairs. For each node pair, our protocol (CAB), IRU, ETX, and the minimum hop count may find different paths between the node pair. Our protocol can also give an estimation for the available bandwidth of its own widest path. We then establish a new flow on the paths found by the algorithms, one at a time, to measure the throughput of the paths. The new flow has a data rate much larger than the available bandwidth of our widest path, so that we can obtain the maximum throughput supported by the path without violating the bandwidth guaranteed for the existing flows. We compare the throughput of the paths found by the different protocols to evaluate the performances of the different protocols for finding the maximum available bandwidth path. Denote B_{CAB} , B_{MPC} , B_{ETX} , B_{IRU}

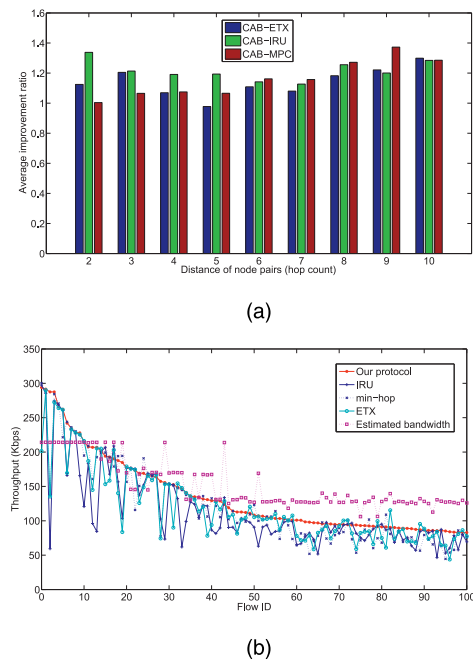


Fig. 7. 100-node in 1,450m*1,450m. (a) Average improvement ratios, (b) Throughput of flows.

as the average throughput of the paths found by applying the CAB, minimum hop count, ETX, and IRU metrics, respectively. $\frac{B_{CAB}}{B_{MPC}}$, $\frac{B_{CAB}}{B_{ETX}}$, and $\frac{B_{CAB}}{B_{IRU}}$ are called the improvement ratios of our new metric (CAB) with the minimum hop count, ETX, and IRU, respectively. The larger the improvement ratio, the better our new metric.

5.4 Simulation Results for Scenario 1

We first deploy 100 nodes in a 1,450m*1,450m square (denoted by TOP1). There are about 400 bidirectional links in the network. We randomly select 100 links and deploy the existing one-hop flows on them. We define the distance of a node pair as the minimum hop count between them. We randomly select 20 node pairs such that each node pair has the same distance. In this topology, we consider the distance of node pair from 2 to 10, and there are totally 120 multihop flows. Fig. 7a shows the average improvement ratios of our metrics with the existing metrics as a function of the distance of node pair. We can observe that almost all of the improvement ratios are larger than 1, which implies that our metric works the best for finding the high throughput path. We randomly select 100 multihop flows and investigate the throughput of individual flow produced by the different protocols. Fig. 7b shows the simulation results of the flows which are sorted according to the throughput of our protocol. This figure also shows the gap between the practical throughput and the estimated available bandwidth.

We first analyze the differences among different protocols. We can observe that ETX and IRU do not work well in some cases. For instance, the practical throughputs of flow ID 3 delivered by ETX and IRU are much less than that of our metric. Without considering the bit error rate of each channel, the packet loss probability can reflect the traffic load on each link to a certain degree. However, the path ETX or IRU is simply computed by summing the ETXs or

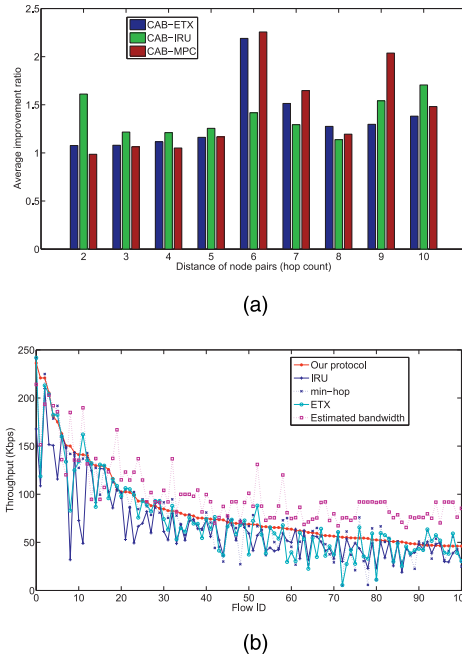


Fig. 8. 100-node in 1,450m*1,450m with exiting flows following $U(1,30)$ Kbps. (a) Average improvement ratios, (b) Throughput of flows.

IRUs of all the links on a path. Such calculation method causes ETX and IRU prefer the short path to the long path, such that ETX or IRU may select a low available bandwidth path. Although the practical throughput of the existing metric is higher than that of our metric for some particular flows, the difference is small. Therefore, our metric is relatively more efficient for finding the high-throughput path.

We now investigate why there is a difference between the practical throughput and the estimated available bandwidth. Fig. 7b shows that the practical throughput may be more than or less than the estimated one. First, according to [5] and the discussion in Section 3, our work develops an *underestimate* of the true available bandwidth. However, the theoretical studies do not take into account of packet overheads and collisions in the MAC layer, which reduce the actual throughput in a real network. For example, we have measured the actual throughput of a four-node network where the distance between neighbor nodes is the same as the transmission range. The theoretical throughput is 250 Kbps but the actual is only 200 Kbps. We believe networks of larger scale would experience even more serious collisions. Another factor that leads to the practical throughput is less than the theoretical throughput is the assumption on interference range. We assume 2-hop interference but situations like Fig. 3 can happen. The practical throughput is thus smaller than the estimated path bandwidth. Our simulation results show that our approach gives an overestimation for almost all of the flows with large hop-count distance. By (3), path bandwidth is independent on the hop-count distance of the path. However, the longer the path, the larger the collision probability. Thus, the hop-count distance affects the practical throughput of a path. That is why (3) is likely to overestimate the bandwidth of a path with large distance.

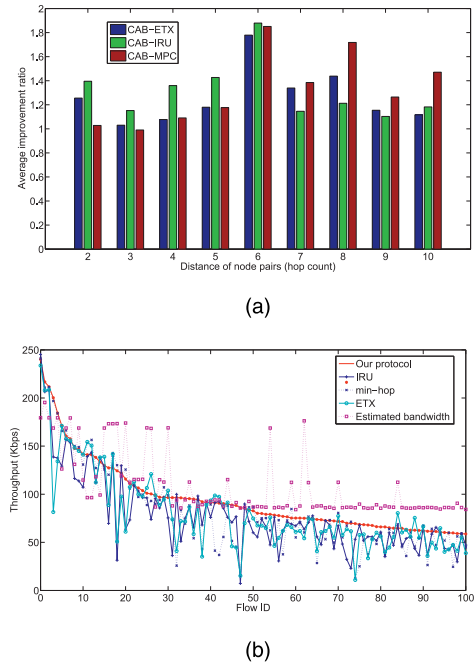


Fig. 9. 100-node in 1,450m*1,450m with 150 exiting flows. (a) Average improvement ratios, (b) Throughput of flows.

5.5 Simulation Results for Scenarios 2 and 3

As the performance of our routing protocol depends on the background traffic, we change the background traffic in TOP1 to evaluate the performance of the routing protocols. In scenario 2, we let the data rates of the existing flows follow $U(1,30)$ Kbps, and Fig. 8 shows the simulation experiments. In scenario 3, we let 150 links carry the existing flows, while the data rates of the existing flows still follow $U(1,20)$ Kbps, and Fig. 9 shows the simulation results. As the background traffic load in scenario 2 increases, the available bandwidth for each flow may be lower than that in scenario 1. Comparing Figs. 7b and 8b, we can observe that the average throughput of our protocol in scenario 2 is lower than that in scenario 1. From Fig. 8a, we can observe that the average improvement ratio of our protocol to the min-hop count is very high when the distance of node pair is 6, 7, and 9. As the min-hop count does not consider the traffic load on each link, it is probably that the min-hop path has very lower available bandwidth. Therefore, considering the current traffic load information is very important for finding the high-throughput path. Generally, Figs. 7a, 8a, and 9a show that our protocol works the best for finding the high-throughput path with the different background traffic loads.

5.6 Simulation Results for Scenarios 4 and 5

We now study the effect of network topology. We deploy 100 nodes in a 1,000 m*1,000 m square (denoted by TOP2) and deploy 200 nodes in a 2,000 m*2,000 m square (denoted by TOP3). We randomly select 100 and 230 links in TOP2 and TOP3, respectively, to carry background one-hop flows. Figs. 10 and 11 show the simulation results under both the network topologies. In TOP2, the maximum distance of node pair is 6, and the network is very dense. Fig. 10a shows that for a certain distance of node pair (say, 6), the

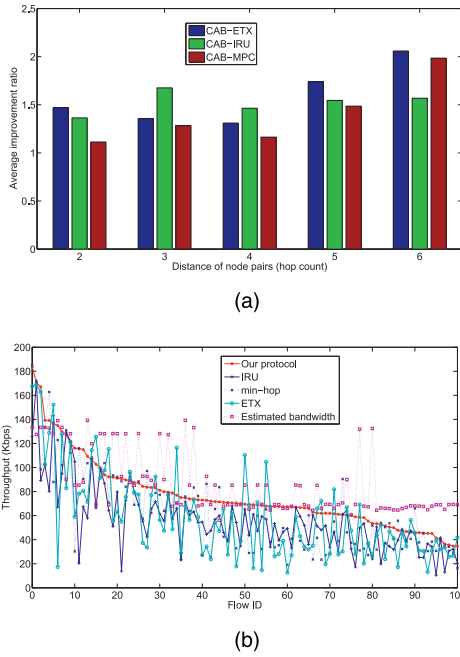


Fig. 10. 100-node in 1,000 m*1,000 m. (a) Average improvement ratios, (b) Throughput of flows.

average improvement ratio in TOP2 is probably higher than that in TOP1. The average improvement ratio also depends on the network topology. If there are many alternative paths between a node pair, there are lots of choices for our metric. On the other hand, if there is only one path between a node pair, we believe that any metric produces the same throughput. In network with larger node degree or larger number of nodes, there are many alternative paths between a node pair, so that the difference among the different routing metrics is more significant. That is why the performance improvement of our protocol in TOP2 and TOP3 is more significant than that in TOP1. Figs. 10b and 11b show the practical throughput of individual flow in TOP2 and TOP3, respectively. Both figures show that our approach probably overestimates the path bandwidth.

5.7 Simulation Results with Shadowing Model

In the previous simulation, we apply the two-ray ground propagation model, which is widely used in the existing works [2], [5], [6] for the long-range communication. We would like to use log-normal shadowing propagation model provided in NS2 to evaluate the performance of our protocol. The default transmission range in NS2 by applying shadowing model is about 20 m, and the work in [31] also applies the shadowing model for short-range communication. Our simulation experiments use the “threshold” tool in NS2 to calculate Rx Threshold (power threshold to correctly receive data) and CS Threshold (power threshold to sense transmission) so that the transmission range is 25 m while the carrier-sensing range is 55 m. The default value is used for other parameters. We deploy 100 nodes in a 145m*145m square. We randomly select 100 links to carry background flows.

With shadowing propagation model, the interference range cannot be simply represented by the distance. Fig. 12a

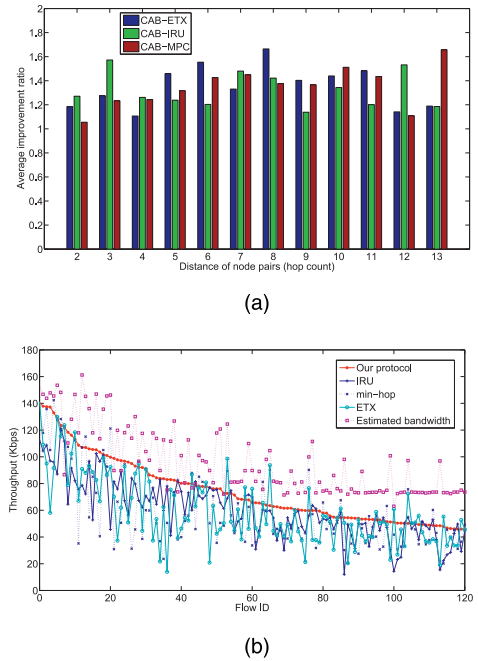


Fig. 11. 200-node in 2,000 m*2,000 m. (a) Average improvement ratios, (b) Throughput of flows.

shows the average improvement ratios for the different distance of node pair. Generally speaking, the simulation results show our protocol works better than the existing protocols. Therefore, our protocol works well under different propagation models. Fig. 12b shows the throughputs of 100 individual flows and the estimated path bandwidth calculated by our approach. With the shadowing model, a clique in Q_p of (2) is likely to contain three links but not four, such that (3) should underestimate the

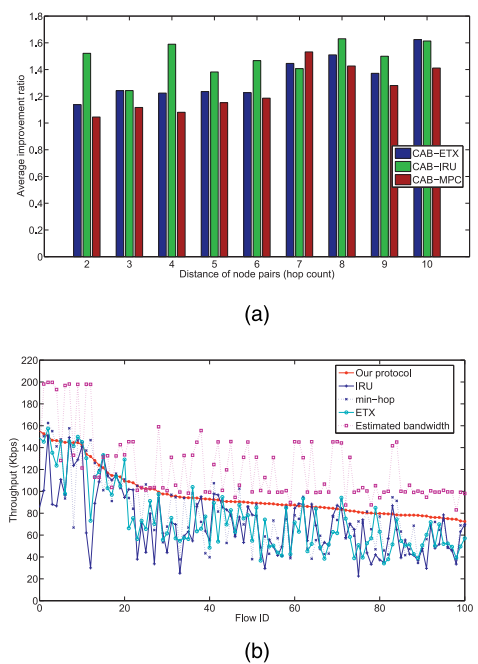


Fig. 12. 100-node in 145 m*145 m with shadowing model. (a) Average improvement ratios, (b) Throughput of flows.

path bandwidth from the theoretical perspective. However, with the effect of the collision caused by the 802.11 protocol, our simulation results show that (3) still gives the overestimate for most of the paths.

In conclusion, our extensive simulation results show that the proposed metric, CAB, performs better for finding the high-throughput path than the existing metrics. On one hand, our approach theoretically gives an underestimation for path bandwidth; on the other hand, our simulation results show that our approach probably gives an overestimation for path bandwidth in practice. This implies that if a source accepts a request with the bandwidth requirement larger than the maximum estimated end-to-end available bandwidth, probably, no path can support this connection, and so the source should reject this request in order to guarantee the available bandwidth allocated for the existing flows.

6 CONCLUSION

In this paper, we studied the maximum available bandwidth path problem, which is a fundamental issue to support quality-of-service in wireless mesh networks. The main contribution of our work is a new left-isotonic path weight which captures the available path bandwidth information. The left-isotonicity property of our proposed path weight facilitates us to develop a proactive hop-by-hop routing protocol, and we formally proved that our protocol satisfies the optimality and consistency requirements. Based on the available path bandwidth information, a source can immediately determine some infeasible connection requests with the high bandwidth requirement. We tested the performance of our protocol under different scenarios.

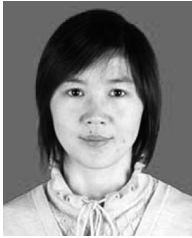
ACKNOWLEDGMENTS

This work was supported in part by the Cisco Research Initiative Award, the National Science Fund for Distinguished Young Scholars under Grant 60725105, the National Basic Research Program of China under Grant 2009CB320404, and the Program for Changjiang Scholars and Innovative Research Team in University.

REFERENCES

- [1] Q. Zhang and Y.-Q. Zhang, "Cross-Layer Design for QoS Support in Multihop Wireless Networks," *Proc. IEEE*, vol. 96, no. 1, pp. 234-244, Jan. 2008.
- [2] T. Salonidis, M. Garetto, A. Saha, and E. Knightly, "Identifying High Throughput Paths in 802.11 Mesh Networks: A Model-Based Approach," *Proc. IEEE Int'l Conf. Network Protocols (ICNP '07)*, pp. 21-30, Oct. 2007.
- [3] C.-Y. Chiu, Y.-L. Kuo, E. Wu, and G.-H. Chen, "Bandwidth-Constrained Routing Problem in Wireless Ad Hoc Networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 1, pp. 4-14, Jan. 2008.
- [4] J. Tang, G. Xue, and W. Zhang, "Interference-Aware Topology Control and QoS Routing in Multi-Channel Wireless Mesh Networks," *Proc. ACM MobiHoc*, pp. 68-77, May 2005.
- [5] Y. Yang and R. Kravets, "Contention-Aware Admission Control for Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 4, no. 4, pp. 363-377, Apr. 2009.
- [6] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing End-to-End Delay: A Novel Routing Metric for Multi-Radio Wireless Mesh Networks," *Proc. IEEE INFOCOM*, pp. 46-53, Apr. 2009.
- [7] Y. Yang and J. Wang, "Design Guidelines for Routing Metrics in Multihop Wireless Networks," *Proc. IEEE INFOCOM*, pp. 2288-2296, Apr. 2008.
- [8] J.L. Sobrinho, "Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet," *Proc. IEEE INFOCOM*, pp. 727-735, Apr. 2001.
- [9] R. Draves, J. Padhye, and B. Zill, "Comparison of Routing Metrics for Static Multi-Hop Wireless Networks," *Proc. ACM SIGCOMM*, pp. 133-144, Sept. 2004.
- [10] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing," *Proc. ACM MobiCom*, pp. 134-146, Sept. 2003.
- [11] M. Campista, D. Passos, P. Esposito, I. Moraes, C. Albuquerque, D. Saade, M. Rubinstein, L. Costa, and O. Duarte, "Routing Metrics and Protocols for Wireless Mesh Networks," *IEEE Network*, vol. 22, no. 1, pp. 6-12, Jan. 2002.
- [12] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," *Proc. ACM SIGCOMM*, pp. 114-128, Oct. 2004.
- [13] A.P. Subramanian, M.M. Buddkhot, and S. Miller, "Interference Aware Routing in Multi-Radio Wireless Mesh Networks," *Proc. Second IEEE Workshop Wireless Mesh Networks (WiMesh '06)*, pp. 55-63, Sept. 2006.
- [14] Y. Yang, J. Wang, and R. Kravets, "Designing Routing Metrics for Mesh Networks," *Proc. IEEE Workshop Wireless Mesh Networks (WiMesh '05)*, Sept. 2005.
- [15] M. Genetakis and V.A. Siris, "A Contention-Aware Routing Metric for Multi-Rate Multi-Radio Mesh Networks," *Proc. Fifth Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '08)*, pp. 242-250, 2008.
- [16] H. Li, Y. Cheng, and C. Zhou, "Multi-Hop Effective Bandwidth Based Routing in Multi-Radio Wireless Mesh Networks," *Proc. IEEE Global Telecomm. Conf. (GlobeCom '08)*, pp. 1-5, Nov. 2008.
- [17] T. Liu and W. Liao, "Interference-Aware QoS Routing for Multi-Rate Multi-Radio Multi-Channel IEEE 802.11 Wireless Mesh Networks," *IEEE Trans. Wireless Networks*, vol. 8, no. 1, pp. 166-175, Jan. 2009.
- [18] L. Chen and W.B. Heinzelman, "QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks," *IEEE J. Selected Areas in Comm.*, vol. 23, no. 3, pp. 561-572, Mar. 2005.
- [19] Q. Xue and A. Ganz, "Ad Hoc QoS On-Demand Routing (AQOR) in Mobile Ad Hoc Networks," *J. Parallel and Distributed Computing*, vol. 63, pp. 154-165, 2003.
- [20] W. Liao, Y. Tseng, and K. Shih, "A TDMA-Based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '02)*, pp. 3186-3190, Apr. 2002.
- [21] K. Shih, C. Chang, Y. Chen, and T. Chuang, "Dynamic Bandwidth Allocation for QoS Routing on TDMA-Based Mobile Ad Hoc Networks," *Computer Comm.*, vol. 29, pp. 1316-1329, 2006.
- [22] C. Zhu and M.S. Corson, "QoS Routing for Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM*, pp. 958-967, June 2002.
- [23] H. Zhai and Y. Fang, "Impact of Routing Metrics on Path Capacity in Multirate and Multihop Wireless Ad Hoc Networks," *Proc. 14th IEEE Int'l Conf. Network Protocols (ICNP '06)*, pp. 86-95, Nov. 2006.
- [24] Z. Jia, R. Gupta, J. Walrand, and P. Varaiya, "Bandwidth Guaranteed Routing for Ad-Hoc Networks with Interference Consideration," *Proc. IEEE Symp. Computers and Comm.*, pp. 3-9, 2005.
- [25] M. Kordialam and T. Nandagopal, "The Effect of Interference on the Capacity of Multi-Hop Wireless Networks," *Proc. IEEE Symp. Information Theory*, pp. 470-470, June 2004.
- [26] X. Cheng, P. Mohapatra, S.-J. Lee, and S. Banerjee, "MARIA: Interference-Aware Admission Control and QoS Routing in Wireless Mesh Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '08)*, pp. 2865-2870, May 2008.
- [27] M. Kordialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," *Proc. ACM MobiCom*, pp. 73-87, Aug. 2005.
- [28] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-Hop Wireless Network Performance," *Proc. ACM MobiCom*, pp. 66-80, 2003.
- [29] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM*, pp. 234-244, Aug. 1994.
- [30] The Network Simulator—ns2, <http://www.isi.edu/nsname/ns>, 2011.

[31] B. Liu, B. Otis, S. Challa, P. Aexon, C. Tung, and S. Jha, "On the Fading and Shadowing Effects for Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Mobile Adhoc and Sensor Systems (MASS '06)*, pp. 51-60, Oct. 2006.



Ronghui Hou received the BEng, MEng, and PhD degrees in communication engineering from Northwestern Polytechnical University in 2002, 2005, and 2007, respectively. She worked as a postdoctoral fellow in the Department of Electrical and Electronic Engineering at the University of Hong Kong between 2007 and 2009. Since December 2009, she has been with Xidian University, China, where she is currently an associate professor in the Department of

Telecommunication Engineering. Her research interests include network quality of service issues, routing algorithm design, and wireless networks. She is a member of the IEEE.



King-Shan Lui received the BEng and MPhil degrees in computer science from the Hong Kong University of Science and Technology. After receiving the PhD degree from the University of Illinois at Urbana-Champaign, she joined the Department of Electrical and Electronic Engineering of the University of Hong Kong. Her research interests include network protocol design and analysis, wireless mesh networks, and quality-of-service issues. She is a senior

member of the IEEE.



Fred Baker has been working in networking technology, including the Internet, since 1978. He is a fellow at Cisco Systems and participates in the IETF.



Jiandong Li received the bachelor's, master's, and PhD degrees in communications and electronic systems, respectively, in 1982, 1985, and 1991 from Xidian University. He has been with Xidian University since 1985, where he has been a professor since 1994 and the dean of the School of Telecommunication Engineering since 1997. He was a visiting professor in the School of Electrical and Computer Engineering at Cornell University between January 2002 and

January 2003. His current research interests include wireless communications, network protocol, and algorithm design. He is a senior member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**