A Routing Protocol for Anycast Messages

Dong Xuan, Weijia Jia, *Member*, *IEEE*, Wei Zhao, *Senior Member*, *IEEE*, and Hongwen Zhu

Abstract—An anycast packet is one that should be delivered to one member in a group of designated recipients. Using anycast services may considerably simplify some applications. Little work has been done on routing anycast packets. In this paper, we propose and analyze a routing protocol for anycast message. It is composed of two subprotocols: the routing table establishment subprotocol and the packet forwarding subprotocol. In the routing table establishment subprotocol, we propose four methods (SSP, MIN-D, SBT, and CBT) for enforcing an order among routers for the purpose of loop prevention. These methods differ from each other on information used to maintain orders, the impact on QoS, and the compatibility to the existing routing protocols. In the packet forwarding subprotocol, we propose a Weighted-Random Selection (WRS) approach for multiple path selection in order to balance network traffic. In particular, the fixed and adaptive methods are proposed to determine the weights. Both of them explicitly take into account the characteristics of distribution of anycast recipient group while the adaptive method uses the dynamic information of the anycast traffic as well. Correctness property of the protocol is formally proven. Extensive simulation is performed to evaluate our newly designed protocol. Performance data shows that the loop-prevention methods and the WRS approaches have great impact on the performance in terms of average end-to-end packet delay. In particular, the protocol using the SBT or CBT loop-prevention methods and the adaptive WRS approach performs very close to a dynamic optimal routing protocol in most cases.

Index Terms—Anycast message, multiple path routing, shortest path first, weight assignment.

1 Introduction

In this paper, we address routing problems for anycast messages in packet switching networks. An anycast message is the one that should be delivered to one member in a group of designated recipients [22]. The traditional unicast message is a special case of an anycast message in the sense that for the unicast message, the recipient group size is one.

Using anycast communication services may considerably simplify some applications. For example, multiple mirrored web sites can share a single anycast address, and users may simply send a request with the anycast address in order to obtain information (e.g., weather information and stock quotes, etc.) from multiple sites where information is provided. As another example of anycast service, a resolver will only need to send a query with a well-known Domain Name Service (DNS) anycast address whenever the resolver wants services from a domain name server. Without anycast service, a domain name resolver (client) has to be configured with the IP addresses of all the associated DNS servers.

Because more and more applications demand anycast services, in the latest version of IP specification, Ipv6, anycast has been defined as a standard service [7]. Several studies have been done on communication with anycast messages since the notion was introduced in [22]. Generally speaking, the problems pertaining to anycast can be divided into two classes: management methods at application layer for using anycast services, and procedures and protocols at network layer for routing and addressing anycast messages. In [14], it was determined that the anycast addresses are allocated from the unicast address space with any of the defined unicast address format. Recently, Deering and Hinden defined Subnet-Router as an anycast address [7] for all routers within a subnet prefix, taking from the unicast address space, an additional set of reserved anycast addresses within each subnet prefix has been defined and listed. In [1], the implication of an anycasting service supported at the application layer was explored. The Internet draft [3] addressed the issue relevant to notifying a client the address of a server that is initially accessed via an anycast address.

Little work has been done on routing anycast messages while extensive studies were carried out for routing unicast and multicast messages [2], [6], [8], [9]. Routing in a network is concerned with determining a path for a message to travel from its source node to its destination node. There are two classes of routing approaches: *single-path routing* and *multipath routing*. With single-path routing, the path of a given pair of source and destination hosts is unique. That is, the path for messages from the source to the destination is time invariant (unless, of course, the network topology changes). Single-path routing is simple and easy to implement, and has been widely used. For example, the most popular routing algorithm used in existing networks is the shortest path first (SPF) algorithm that selects a path that has the minimum value of an objective function. The objective function of a given path is usually defined by a numeric sum of the individual link parameters (e.g., number of hops, delay, bandwidth, etc.).

A problem associated with single-path routing is that it may overload the selected path and hence cause traffic congestion. The multipath routing approach addresses this

• *D. Xuan and W. Zhao are with the Department of Computer Science, Texas A&M University, College Station, TX, 77843-3112. E-mail: {dxuan, zhao}@cs.tamu.edu.*
• *W. Jia is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. E-mail: wjia@cs.cityu.edu.hk.*
• *H. Zhu is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200030, China. E-mail: hwzhu@info.sh.cn.*

Manuscript received 5 Sept. 1997; accepted 22 Oct. 1999.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 105620.

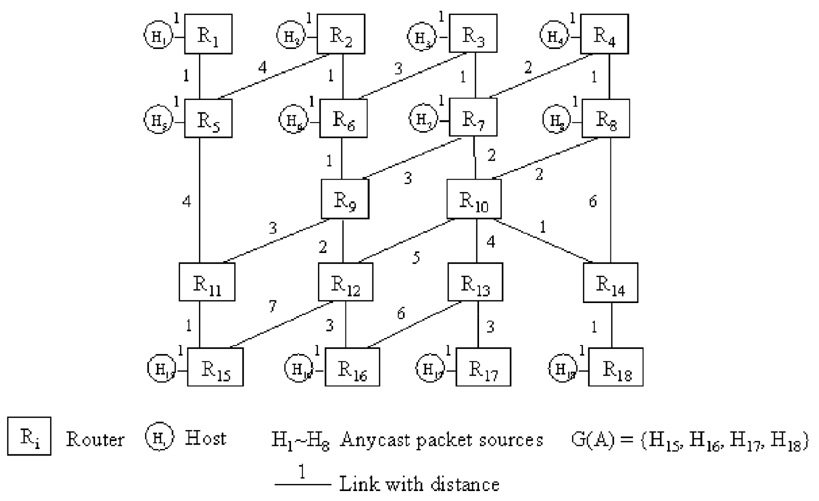

R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18
Router
Host
H1~H8 Anycast packet sources
G(A) = {H15, H16, H17, H18}
Link with distance

Fig. 1. Example network.

problem. Generally speaking, a multipath routing algorithm will split traffic into several different paths. Many analyses have shown that it tends to distribute traffic into multiple paths: both network throughput and message delay can be improved [5], [16], [17]. Solution techniques for multipath routing include heuristic methods [11] as well as optimal ones [4], [10], [13], [23]. Some ideas of these routing algorithms have been absorbed in recent versions of Internet protocols. For example, OSPF distributes traffic over paths that have (almost) equal length [19]. IGRP goes even further. It quantifies the notion of "almost equality" by introducing variance coefficient of path lengths and uses it in route selection [15]. While these earlier studies and projects have provided insight into the problem, they mostly addressed multipath routing for unicast messages.

Since the notion of anycast presumes that there are possibly multiple recipients with the same address, we would like to use multipath routing approach in our new designed protocol. There are two issues related to multipath routing. The first issue is loop prevention. The multipath routing inherently implies the risk for having a loop in a route. Care must be taken to prevent a loop. The second is how to make a selection among the multiple paths so that the better network performance, e.g., delay and throughput, can be effectively improved.

Our proposed protocol is composed of two subprotocols: the routing table establishment subprotocol and the packet forwarding subprotocol. The above two issues will be addressed in these two subprotocols.

In the routing table establishment subprotocol, an order is introduced among routers to make sure that only higher order router can send anycast messages to a lower order router. In this way, loop is prevented. We consider four methods (Shortest shortest Path, Minimum-Distance, Source-based Tree, and Core-based Tree) to enforce an order among routers. They differ from each other on information used to maintain the order and their consequent impact on QoS, network resource consuming, and the compatibility to the existing routing protocols.

In the packet forwarding subprotocol, we take a Weighted Random Selection (WRS) approach for multipath selection to balance network traffic. In particular, we consider two methods to assign weights: *fixed* and *adaptive* methods. While these two methods both take into account the characteristics of distribution of anycast recipient group, they differ from each other on whether the dynamic information of the anycast message traffic is used.

Our proposed protocol is simple, efficient, and compatible with most of the existing routing technologies. We formally prove the correctness property for our protocol. We performed extensive simulation study for our newly designed protocol. Our data shows that router ordering and weight assignment methods have great impact on the network performance in terms of average end-to-end packet delay. In particular, some methods we propose perform very close to a dynamic optimal protocol in most cases.

The rest of the paper is organized as follows: In Section 2, we define the models and notations that will be used in this paper. In Section 3, we present the primaries of routing protocols. In Sections 4 and 5, we describe our two subprotocols. In Section 6, we formally present a proof for the correctness property of our designed protocol, and discuss an extension that can be made to our work. In Section 7, a simulation model is introduced and the performance of our proposed protocol is evaluated. Finally, we summarize the paper with some concluding remarks in Section 8.

2 Models and Notations

We consider a network that consists of a number of nodes. A node can be either a router or a host. Nodes are connected by physical links along which packets can be transmitted. We assume that links are symmetric. Each link has an attribute called *distance*. The distance of a link is usually measured by delay, bandwidth, etc. As we will see, distance plays a critical role in route selection. See Fig. 1 for a sample network where the numerical values associated with links are their distances.

A packet is specified by addresses of its source and destination, and a source is a host. The destination for an anycast packet can be *any one* in a group of predefined hosts. Let A be an anycast (destination) address. We denote $G(A)$ to be the group of designated recipient hosts. That is, a packet with an anycast address A can be sent to any host in $G(A)$.

There are two classes of anycast traffic, i.e., the *independent* and *dependent* traffic. An anycast packet of the independent traffic can be sent to any host in the recipient group without any correlation with other packets. However, in the dependent traffic, anycast packets from a source are *dependent* on each other in terms of their destination. These dependent packets *form* a *flow*. That is, if the first packet from a flow of anycast packets is delivered to a member (say H) in the recipient group, all the consequent packets should also be delivered to the same member H. The flow is identified by an identification number in the header of packets.

A node (say R) is a *next hop*[1] of another node (say R') if R can receive a packet directly from R' without going through any other router. For example, in the network shown in Fig. 1, R_1 is a next hop of R_5 (and vice versa) while R_3 is not a next hop of R_5.

The sequence of routers via which a packet is transmitted from its source to destination forms a path. Formally, $P(X,Y)$ denotes a *path* from X to Y where X and Y are nodes. Sometimes, we would like to list explicitly the sequence of nodes in a path. For example, as shown in Fig. 1, $< H_1, R_1, R_5, R_{11}, R_{15}, H_{15} >$ is a path from H_1 to H_{15}. We will use the terms "route" and "path" interchangeably.

$C(k, P(X,Y))$ denotes the *kth element* in path $P(X,Y)$. For example, for $P(H_1, H_{15}) =< H_1, R_1, R_5, R_{11}, R_{15}, H_{15} >$, $C(3, P(H_1, H_{15})) = R_5$.

$D(P(X,Y))$ denotes the *total distance* of links on path $P(X,Y)$. It is usually defined by a numeric sum of the individual link distances. Consider the network shown in Fig. 1. For $P(H_2, R_{15}) =< H_2, R_2, R_5, R_{11}, R_{15} >$, $D(P(H_2, R_{15})) = 10$.

A *shortest path* from X to Y is denoted as $P_{SP}(X,Y)$. By definition, for any $P(X,Y)$,

$$D(P_{SP}(X,Y)) \leq D(P(X,Y)) \quad (1)$$

In the network shown in Fig. 1,

$$P_{SP}(H_2, R_{15}) =< H_2, R_2, R_6, R_9, R_{11}, R_{15} > .$$

Let $G(A)$ be an anycast group and H is one of its member. If $P_{SP}(X,H)$ is the shortest among all the shortest paths from X to the members of the group, i.e., for any H' in $G(A)$,

$$D(P_{SP}(X,H)) \leq D(P_{SP}(X,H')), \quad (2)$$

then $P_{SP}(X,H)$ is called *the shortest-shortest path* from X to $G(A)$, and is denoted as $P_{SSP}(X, G(A))$.

1. The term "next hop" usually means a router. Here, we use it in a generalized sense. In this paper, a next hop can be either a router or a host. This simplifies our discussion of the problems.

We make two assumptions when we describe our routing protocol and analyze its properties. We assume that the network has *no faults*, and between any pair of nodes, there exists at least one functioning path. Furthermore, we assume that routers share all of the routing information of the whole network. These two assumptions simplify the discussion of routing algorithms we are going to introduce. In Section 6, we will discuss how to extend our protocol to the systems where these assumptions may not hold.

3 Preliminary of Routing Protocols

Routers in the network cooperatively decide a path for a packet and transmit the packet along the path. This network function is called *routing.* A *routing protocol* defines the operations performed by the routers in order to realize the routing function. Design and realization of such a protocol are, nevertheless, a challenging task. There are three primary objectives:

1. *Quality of services.* Any protocol should provide the best possible services to the users. Here, we are interested in the end-to-end average-delay of a packet (average delay in short). One of our protocol's objectives is to reduce average delay.
2. *Overhead.* Executing any protocol will always consume some network resource, such as router memory for storing routing information, and network bandwidth for transferring control message. A good routing protocol should reduce these overheads and avoid over-consuming resources.
3. *Compatibility.* The newly designed protocol should be compatible to the existing ones. The major modification, such as adding a field to the packet header to existing protocols, should be avoided if possible.

While all these objectives are important, they may conflict with each other. For example, reducing the runtime overhead may compromise the network performance. One has to make a good balance when designing the routing protocols.

The primary data structure that routers use to make routing decisions is a *routing table*. Each router may have a number of routing tables. A routing table consists of a list of *entries*. An entry in a routing table usually contains fields of destination address, next hop, distance, etc. The *next hop field* defines the next hop (i.e., a router or a host) where the packet should be sent. The *distance field* contains the value of the total distance of a path that leads to the destination address. For an incoming packet, a router uses its routing table to determine which next hop to transmit the packet.

Thus, a routing protocol can be decomposed into two subprotocols:

1. *Routing Table Establishment Subprotocol.* This subprotocol is responsible for collecting and exchanging information in regard to the network topology and resource availability. It then uses this collected information to construct or update routing entries in routing tables which are used in packets forwarding.

2. *Packet Forwarding Subprotocol.* The subprotocol is used to forward packets at run time. Upon a packet arrival, a router extracts the destination information from the packet, and looks up its routing table to find the matching entry or entries to decide the next hop to which the packet should be forwarded. Once the next hop of a packet is determined, the packet will be transported to a proper output port where the packet will be transmitted into the associated output link, which, in turn, connects to the next hop.

In unicast and multicast, once the entries in the routing table are established, forwarding packets is trivial. Thus, many studies on unicast and multicast routing protocols focus on routing table establishment. In anycast, while the establishment of routing table remains important, forwarding anycast packets is also critical. This is because in the routing table there are likely multiple entries that can be used to forward one packet. As such, how to make a selection among the matching entries becomes an interesting and challenging task if we want to avoid congestion and improve QoS.

Next, we introduce our design and analyses of these two subprotocols.

4 Routing Table Establishment Subprotocol

The objective of the Routing Table Establishment Subprotocol is to establish a routing table for anycast packets. This is achieved by performing the following three tasks in sequence:

Task A. To generate candidate entries used for constructing/updating the routing table. This is done based on the collected network status information.

Task B. To determine eligibility of candidate entries. We will see that not all the candidate entries are eligible due to the loop problem. Some criterion must be established to eliminate those entries that are not eligible.

Task C. To construct or update routing tables with the eligible entries, which will be used by the Packet Forwarding Subprotocol.

In the following subsections, we will discuss each of these tasks in detail.

4.1 Task A: Generating Candidate Entries

Generally speaking, this task involves two steps: 1) to collect information over the network, such as route availability and the group memberships, etc., and 2) to generate candidate entries used for constructing/updating routing tables.

Much work has been done for collecting route availability and group membership information in the domains of unicast and multicast. These existing results can be applied to anycast. In order to simplify the discussion, we assume that routers can invoke some existing protocols to achieve these functions. For example, a host can use an enhanced version of IGMP to advertise to routers from which it wants to receive an anycast packet for a specific address (in other words, it wants to join an anycast group). Each router can then advertise its reachability of an anycast recipient with its neighbors by the routing updates. The updates are the same as those in unicast and multicast domains.

Once the information on network status and group memberships is collected, candidate entries used to construct/update routing tables can be generated. In a router (say R), for an anycast address A, and its group $G(A)$, the candidate entries are generated as follows: for each host H in $G(A)$,

Step 1. Compute the path with the shortest distance from R to H.

Step 2. Construct an entry for the path computed in Step 1. The entry may contain fields such as anycast address, distance, next hop, etc.

Consider the example. In Fig. 2, router R_7 computes the shortest paths for each host in $G(A)$. The shortest path from R_7 *to* H_{15} is $< R_7, R_9, R_{11}, R_{15}, H_{15} >$, the distance of the path is 8. Thus, the corresponding entry has a destination field of (A, H_{15}), a distance field of 8, and a next hop field of R_9.

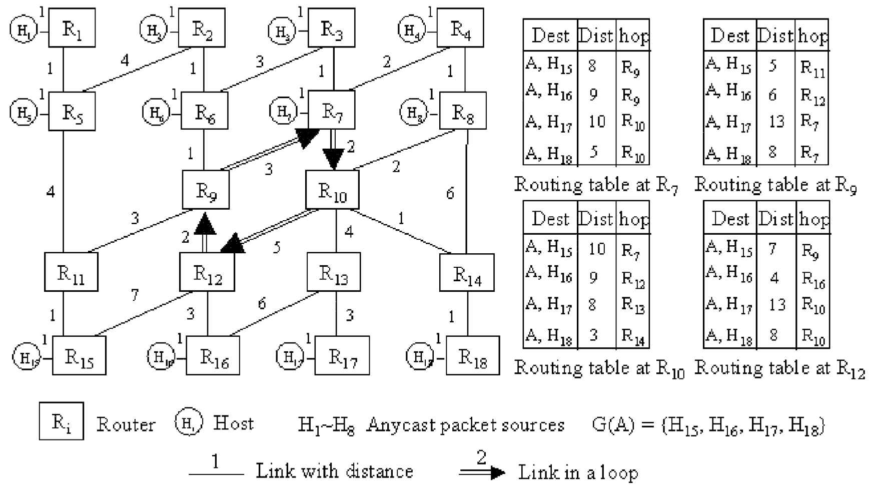
4.2 Task B: Determination of Eligible Entries

If we were dealing with a unicast problem, a candidate entry would be the one directly used in the routing table to route packets from the router to the corresponding host. In anycast communication, however, the case is not as simple as in unicast. Anycast inherently implies that multiple paths may exist for a packet. The following example shows that there is a risk for having loops in a route due to usage of multiple paths.

Consider the example in Fig. 2 again. Assume that H_7 delivers an anycast packet to R_7. In the routing table at R_7, there are four candidate entries corresponding to shortest paths to the recipients of $G(A)$. R_7 may select the entry that leads the path to H_{18}. Thus, R_7 forwards the anycast packet to R_{10}. Similarly, R_{10} may forward the packet to R_{12}, and R_{12} to R_9. When the packet arrives, R_9 may forward it back to R_7 if it selects the entry that leads the path to H_{17}. Thus, a loop (R_7, R_{10}, R_{12}, R_9, R_7) occurs as indicated by double arrow lines in Fig. 2.

Clearly, the loop is due to poor coordination among the routers. To prevent a loop, we introduce an order among routers in the network. Only a router with higher order can transmit a packet to a lower-order one. The noncyclic property of the order guarantees the loop-free routing. We say that a candidate entry is eligible only if the current router has a higher order than the next hop defined in this entry. The route resulting from the eligible entries is guaranteed to be loop-free.

As a matter of fact, the eligibility of a candidate entry depends on what order is used. We will introduce four different orders and consequently propose four methods to determine the eligibility of a candidate entry. The methods we propose differ on 1) information used to establish and maintain the order and 2) topology formed by the routes for forwarding anycast packets. Consequently, they impact



Routing table at R_7

Dest	Dist	hop
A, H_{15}	8	R_9
A, H_{16}	9	R_9
A, H_{17}	10	R_{10}
A, H_{18}	5	R_{10}

Routing table at R_9

Dest	Dist	hop
A, H_{15}	5	R_{11}
A, H_{16}	6	R_{12}
A, H_{17}	13	R_7
A, H_{18}	8	R_7

Routing table at R_{10}

Dest	Dist	hop
A, H_{15}	10	R_7
A, H_{16}	9	R_{12}
A, H_{17}	8	R_{13}
A, H_{18}	3	R_{14}

Routing table at R_{12}

Dest	Dist	hop
A, H_{15}	7	R_9
A, H_{16}	4	R_{16}
A, H_{17}	13	R_{10}
A, H_{18}	8	R_{10}

Fig. 2. Example network with routing table at some routers.

differently on Quality-of-Service, run-time overhead, and compatibility with existing protocols.

4.2.1 The Shortest-Shortest Path Method

The first method we introduce establishes an order that is based on the value of distance from a router to members in the anycast group. Recall that $P_{SSP}(X, G(A))$ represents the shortest-shortest path from X to group $G(A)$. That is, if X wants to send a packet to group $G(A)$, $P_{SSP}(X, G(A))$ is the shortest way to reach (some member of) the group. We define an order between two adjacent nodes as follows:

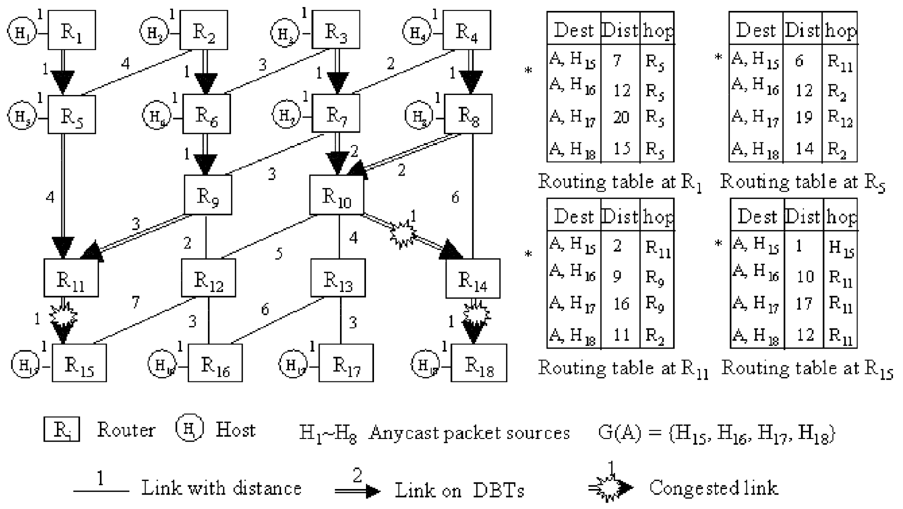
Definition 1 (SSP order). *Let $G(A)$ be an anycast group. For two adjacent routers X and Y, $X >_{SSP} Y$ if Y is the next hop of X on $P_{SSP}(X, G(A))$.*

This definition implies that if Y is a next hop of the X and Y is on the shortest-shortest path from X to $G(A)$, then $X >_{SSP} Y$. Using this order, we have an eligibility criterion as follows:

Definition 2 (Eligibility criterion based on the SSP order). *For a given anycast group $G(A)$ and a router R, a candidate routing entry is eligible if $R >_{SSP} R'$ where R' is specified in the next hop field of the entry.*

Thus, with this definition of eligibility, packets will be transmitted along the shortest-shortest path. Consider Fig. 3, where we have marked entries that are eligible according to this definition. Assume that H_1 delivers an anycast packet to R_1. Following the defined eligible entry, R_1 will forward the packet to R_5. Similarly, R_5 will forward the packet to R_{11}, and R_{11} to R_{15}. Finally, R_{15} should transmit the packet to H_{15}. Several remarks can be made for this method:

1. This method is simple. It only uses the distance information from a router to the recipients in the



Routing table at R_1

	Dest	Dist	hop
*	A, H_{15}	7	R_5
	A, H_{16}	12	R_5
	A, H_{17}	20	R_5
	A, H_{18}	15	R_5

Routing table at R_5

	Dest	Dist	hop
*	A, H_{15}	6	R_{11}
	A, H_{16}	12	R_2
	A, H_{17}	19	R_{12}
	A, H_{18}	14	R_2

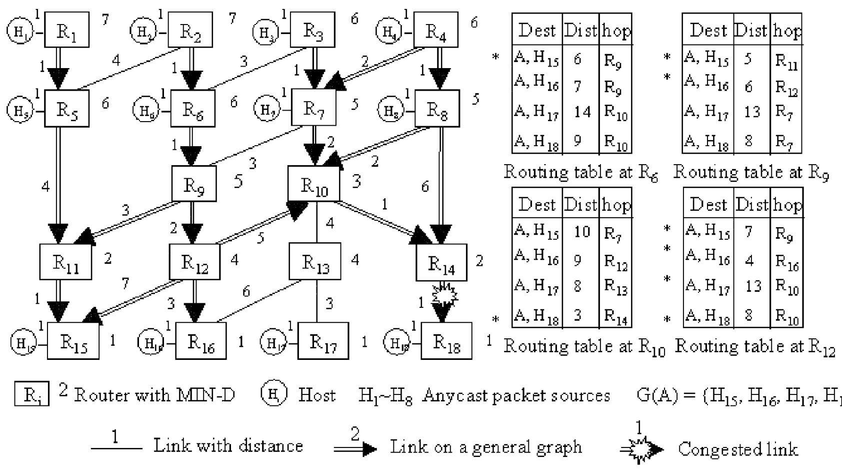
Routing table at R_{11}

	Dest	Dist	hop
*	A, H_{15}	2	R_{11}
	A, H_{16}	9	R_9
	A, H_{17}	16	R_9
	A, H_{18}	11	R_2

Routing table at R_{15}

	Dest	Dist	hop
*	A, H_{15}	1	H_{15}
	A, H_{16}	10	R_{11}
	A, H_{17}	17	R_{11}
	A, H_{18}	12	R_{11}

Fig. 3. Example network with SSP method.


Router with MIN-D
Host
H_1~H_8 Anycast packet sources
$G(A) = \{H_{15}, H_{16}, H_{17}, H_{18}\}$
Link with distance
Link on a general graph
Congested link

Routing table at R_6

	Dest	Dist	hop
*	A, H_{15}	6	R_9
	A, H_{16}	7	R_9
	A, H_{17}	14	R_{10}
	A, H_{18}	9	R_{10}

Routing table at R_9

	Dest	Dist	hop
*	A, H_{15}	5	R_{11}
*	A, H_{16}	6	R_{12}
	A, H_{17}	13	R_7
	A, H_{18}	8	R_7

Routing table at R_{10}

	Dest	Dist	hop
	A, H_{15}	10	R_7
	A, H_{16}	9	R_{12}
	A, H_{17}	8	R_{13}
*	A, H_{18}	3	R_{14}

Routing table at R_{12}

	Dest	Dist	hop
*	A, H_{15}	7	R_9
*	A, H_{16}	4	R_{16}
*	A, H_{17}	13	R_{10}
*	A, H_{18}	8	R_{10}

Fig. 4. Example network with MIN-D method.

group to determine the eligibility of candidate entries for the router.

2. In general, the routes via which anycast packets are routed form a topology that consists of a number of trees. Each member in the recipient group is the root of a tree. The sources are the leaves of the trees. These trees are, thus, called *Destination-Based Trees* (DBT, in short). Fig. 3 illustrates the topology with the SSP method. Given that the path between a leave and the root is unique and uncyclic, the property of loop-free routing is guaranteed.
3. However, when routing along a destination-based tree, anycast traffic can be easily congested on links that are close to the root (i.e., the destination). In Fig. 3, the routers connected by double-arrow-line links compose two DBT trees. The link from R_{11} to R_{15}, from R_{10} to R_{14} and from R_{14} to R_{18} may be overloaded, while the links from R_{12} to R_{16} and from R_{13} to R_{17}, which also lead to the recipients of $G(A)$ have no traffic. Hence, this method may result in a poor delay and throughput performance (as we will see in Section 7).

4.2.2 The Minimum Distance Method

The second method we are going to introduce will relax the ordering condition introduced with the SSP method, and hence, we may improve the potential overloading situation. We would like to introduce the following notation first:

Definition 3 (Function $\min\_\mathrm{d}()$). *Let A be an anycast address, X be a router in the network. $\min\_\mathrm{d}(X, A)$ is the minimum distance from X to any member of the recipient group with anycast address A. That is,*

$$\min\_\mathrm{d}(X, A) = \min_{H \in G(A)} (D(P_{SP}(X, H))).$$

Specifically, if X is a member of $G(A)$, then $\min\_\mathrm{d}(X, A) = 0$.

Let's consider some examples of using $\min\_\mathrm{d}()$. In Fig. 4, the shortest paths from R_6, R_9, R_{10} and R_{12} to the members of $G(A)$ and their distances are listed in their routing tables. According to the definition, we have:

$$\min\_\mathrm{d}(R_6, A) = 6, \min\_\mathrm{d}(R_9, A) = 5, \min\_\mathrm{d}(R_{10}, A) = 3, \text{ and } \min\_\mathrm{d}(R_{12}, A) = 4.$$

These values are indicated in italic font besides the routers in Fig. 4.

Based on the above notation, we define the following order:

Definition 4 (MIN-D order). *Let $G(A)$ be an anycast group. For two adjacent routers X and Y, $X >_{MIN\text{-}D} Y$ if $\min\_\mathrm{d}(X, A) > \min\_\mathrm{d}(Y, A)$. In the case of*

$$\min\_\mathrm{d}(X, A) = \min\_\mathrm{d}(Y, A),$$

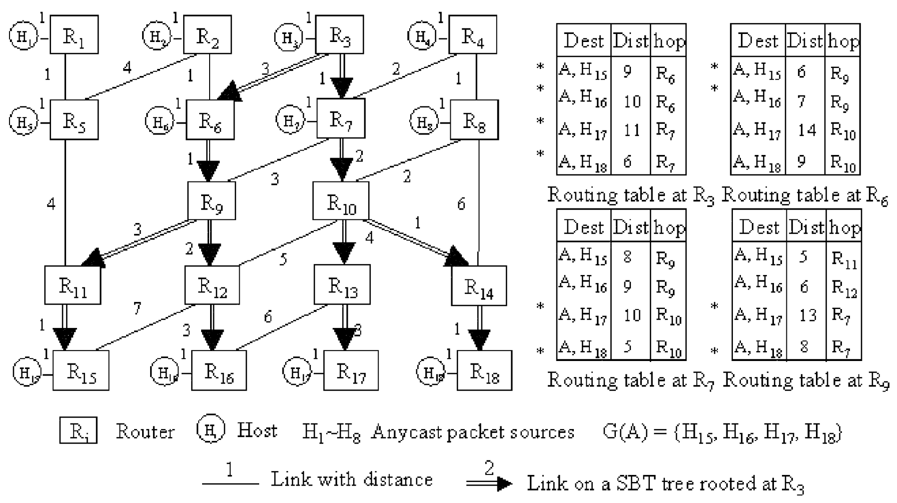
the tie is broken arbitrarily.

Informally, this definition implies that X has a higher order than Y (i.e., $X >_{MIN\text{-}D} Y$) if Y is closer to recipients in the anycast group than X. The following is an eligibility criterion based on this order.

Definition 5 (Eligibility criterion based on the MIN-D order). *On router R, a candidate routing entry for anycast group $G(A)$ is* eligible *if $R >_{MIN\text{-}D} R'$ where R' is specified in the next hop field of the entry.*

Several remarks can be made for this method:

1. The MIN-D method uses more information than the SSP method. To determine the eligibility on router R, this method requires distance information (e.g., $\min\_\mathrm{d}()$) for router R itself, as well as the next hops of R.
2. The topology formed by the routes for forwarding anycast packets is a general graph, instead of the destination-based trees as produced by the SSP method. Fig. 4 illustrates the topology with the MIN-D method. Routing over a general graph may potentially result in looped paths. Fortunately, we can show that loop-free property still holds for the



R_i Router
H_i Host
H_1~H_8 Anycast packet sources
$G(A) = \{H_{15}, H_{16}, H_{17}, H_{18}\}$
Link with distance
Link on a SBT tree rooted at R_3

	Dest	Dist	hop
*	A, H_{15}	9	R_6
*	A, H_{16}	10	R_6
*	A, H_{17}	11	R_7
*	A, H_{18}	6	R_7

Routing table at R_3

	Dest	Dist	hop
*	A, H_{15}	6	R_9
*	A, H_{16}	7	R_9
	A, H_{17}	14	R_{10}
	A, H_{18}	9	R_{10}

Routing table at R_6

	Dest	Dist	hop
	A, H_{15}	8	R_9
	A, H_{16}	9	R_9
*	A, H_{17}	10	R_{10}
*	A, H_{18}	5	R_{10}

Routing table at R_7

	Dest	Dist	hop
	A, H_{15}	5	R_{11}
	A, H_{16}	6	R_{12}
*	A, H_{17}	13	R_7
*	A, H_{18}	8	R_7

Routing table at R_9

Fig. 5. Example network with the SBT method.

MIN-D method. The formal proof is given in the Appendix.

3. The MIN-D method allows more candidate entries to be eligible than the SSP method does.[2] Consequently, more paths are available for a router to route anycact packets, in comparison with the SSP method. We expect this may lead improvement on delay and throughput performance (as we will see in Section 7).
4. In spite of the anticipated performance improvement, the load over links may still not be well balanced. A router with a smaller $\min\_d()$ value tends to be a sink to absorb more traffic than others, potentially becoming a bottleneck. Consider the network in Fig. 4, router R_{14} has its $\min\_d()$ equal to 2. Traffic from $R_3, R_4, R_7,$ and R_8 tends to concentrate in R_{14} and overloads the link from R_{14} to R_{18}. In the meantime, the link from R_{13} to R_{17} that leads to another recipient H_{17} may not have any traffic at all.

4.2.3 The Source-Based Tree Method

As we have discussed, route topology impacts the quality-of-service. The SSP method produces a route topology of destination-based trees that can easily cause traffic congestion. The MIN-D method results in a general graph. While better than the destination-based trees, it may still have performance bottleneck. What is the ideal route topology? An ideal topology should ensure (and allow a simple proof of) loop-free routing and at the same time it should maximize the number of paths for dispatching a packet from a source to destinations. In this and the next subsections, we will study two methods that meet these requirements.

Let us consider a source-based tree (SBT). For a source and an anycast group, the SBT is constructed as follows: the source is the root of the tree, the members of the anycast group are the leaves. The path from the source (i.e., the root) to a member (i.e., a leave) is the shortest path between the two. For two adjacent nodes (say X and Y) on the tree, X is the *father* of Y if X is closer to the root than Y. Y is then a *son* of X. We can define an order based on SBT and consequently develop the eligibility criterion.

Definition 6 (SBT order). *Let S be a source, G(A) be an anycast group, and SBT be the corresponding source-based tree. For two adjacent routers X and Y on T, $X >_{SBT} Y$ if X is the father of Y.*

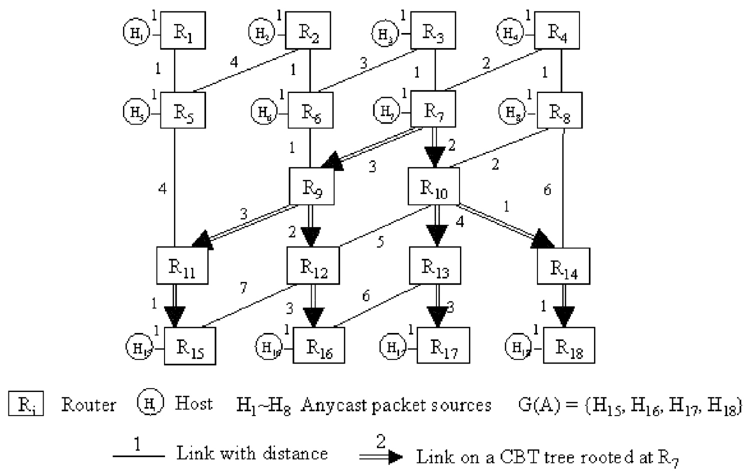
The following is the eligibility criterion based on this order.

Definition 7 (Eligibility criterion based on the SBT order). *On router R, a candidate entry for anycast group G(A) is* eligible *if $R >_{SBT} R'$ where R' is specified in the next hop field of the entry.*

Several remarks can be made for this method:

1. The SBT method uses more information than the SSP method and MIN-D method. This method requires the network topology information in addition to the distance information.
2. The topology formed by the routes for forwarding anycast packets are Source-Based Trees. Each source is the root of a tree. The members in the recipient group are the leaves of the trees. Fig. 5 illustrates the topology with the SBT method. Given that the path between a leave and the root is unique and uncyclic, the property of loop-free routing is guaranteed.
3. Since every SBT tree covers all the members in the recipient group, the SBT method allows more eligible entries than the SSP and MIN-D methods. Consequently, more paths are available for a router to route anycast packets. Furthermore, the resultant paths are all the shortest paths from the source to the recipients, while with the MIN-D method, the

2. This is due to the fact that the relation $>_{MIN\text{-}D}$ contains the relation $>_{SSP}$. That is, if $X >_{SSP} Y$, then $X >_{MIN\text{-}D} Y$, but not necessarily vice versa.



R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13 R14 R15 R16 R17 R18
Router
Host
H1~H8 Anycast packet sources
G(A) = {H15, H16, H17, H18}
Link with distance
Link on a CBT tree rooted at R7

Fig. 6. Example network with CBT.

resultant paths are not guaranteed to be so. We expect this may lead improvement on delay and throughput performance (as we will see in Section 7).

4. In spite of the anticipated performance improvement, with the SBT method, a router has to differentiate eligible entries for individual sources. Accordingly, the router has to build up different sets of anycast routing entries for different sources. Consider an anycast group $G(A)$ with M recipients, and N sources. For one router, the number of anycast routing entries with the SBT method is $O(NM)$, while the numbers of routing entries with the SSP method and MIN-D method are only *one* and *O(M)*, respectively. This means that the SBT method requires much larger storage than the other two methods. Based on this consideration, we will next consider a method that uses the similar idea of the SBT method, but manages to reduce the usage of memory space.

4.2.4 The Core-Based Tree Method

As stated above, this method is similar to the SBT method. The difference is that instead of using multiple trees (i.e., source-based trees), this method only uses a single common tree to determine the eligible entries.

Let us consider the core-based tree (CBT in short). There is a predesignated router to be a *core*. The core can be the center of the network, or the one recipients of G(A), or any others.[3] For an anycast group, the CBT is constructed as follows: the core is the root of the tree, the members of the anycast group are the leaves. The path from the core (i.e., the root) to a member (i.e., a leave) is the shortest path between the two. For two adjacent nodes (say X and Y) on the tree, X is the *father* of Y if X is closer to the root than Y. Y is then a *son* of X. Consider Fig. 6, where there is a core-based tree rooted at R_7. R_9 is the father of R_{11}, and R_{11} is a son of R_9. We can define an order based on CBT and consequently develop the eligibility criterion.

3. For the sake of space limitation, we will not discuss the issue of how to determine the core router of core-based tree; the interested reader can refer the paper [24].

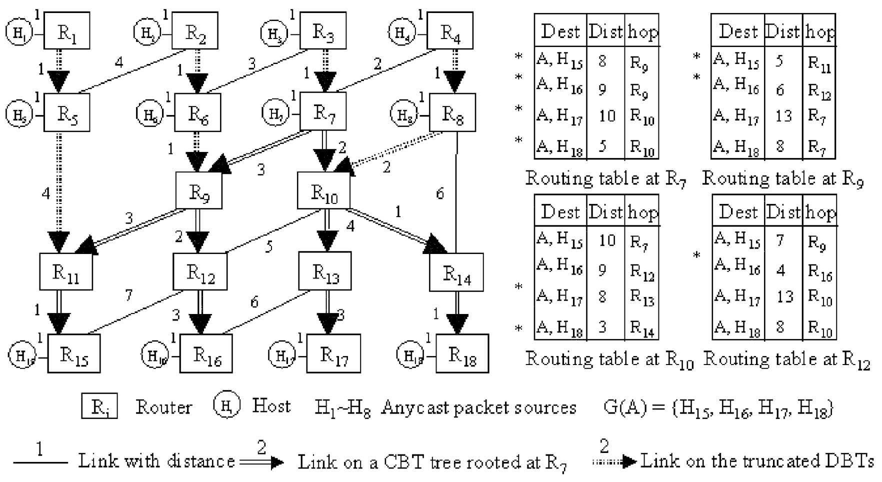
Definition 8 (CBT order). *Let $G(A)$ be an anycast group, and CBT be the corresponding core-based tree. For two adjacent routers X and Y on T, $X >_{CBT} Y$ if X is the father of Y.*

Informally, this definition implies that if X and Y are two adjacent nodes on the shortest paths from the core to members of $G(A)$ and X is closer to the core than Y is, then $X >_{CBT} Y$. Since Definition 8 specifies an order based on CBT, we call it a CBT order. Note that some routers are not on the core-based tree, but anycast packets may still have to go through them. For these off-CBT routers, the order has to be established by other criterion (e.g., SSP order). Formally, the eligibility criterion associated with the core-based tree is as follows:

Definition 9. (Eligibility criterion based on the CBT order and the SSP order). *For an on-CBT router R, a candidate entry for anycast group $G(A)$ is* eligible *if $R >_{CBT} R'$ where R' is specified in the next hop field of the entry. For an off-CBT router R, a candidate entry for anycast group $G(A)$ is eligible if $R >_{SSP} R'$ where R' is specified in the next hop field of the entry.*

Several remarks can be made for this method:

1. The topology formed by the routes with the CBT method is a core-based tree plus the multiple truncated destination-based trees. The members in the recipient group are the leaves of the CBT tree. Fig. 7 illustrates the topology with the CBT method. In fact, the topology consists of two components: one is the core-based tree shown in Fig. 6, and another is the portions of the destination-based trees in Fig. 3 truncated by the core-based tree. Since there will be no loop on the core-based tree and the truncated destination-based trees, the property of loop-free routing is guaranteed.
2. Since the CBT tree covers each member in the recipient group, the CBT method results in more


Router
Host
H_1~H_8 Anycast packet sources
$G(A) = \{H_{15}, H_{16}, H_{17}, H_{18}\}$
Link with distance
Link on a CBT tree rooted at R_7
Link on the truncated DBTs

Routing table at R_7

	Dest	Dist	hop
*	A, H_{15}	8	R_9
*	A, H_{16}	9	R_9
*	A, H_{17}	10	R_{10}
*	A, H_{18}	5	R_{10}

Routing table at R_9

	Dest	Dist	hop
*	A, H_{15}	5	R_{11}
*	A, H_{16}	6	R_{12}
	A, H_{17}	13	R_7
	A, H_{18}	8	R_7

Routing table at R_{10}

	Dest	Dist	hop
	A, H_{15}	10	R_7
	A, H_{16}	9	R_{12}
*	A, H_{17}	8	R_{13}
*	A, H_{18}	3	R_{14}

Routing table at R_{12}

	Dest	Dist	hop
	A, H_{15}	7	R_9
*	A, H_{16}	4	R_{16}
	A, H_{17}	13	R_{10}
	A, H_{18}	8	R_{10}

Fig. 7. Example network with CBT method.

eligible entries than the SSP and the MIN-D method. Consequently, more paths are available for a router to route anycast packets, in comparison with the SSP method and MIN-D method.

3. However, since the routes may not be the shortest paths from the source to the recipients of anycast groups, we can imagine that the performance of average delay with the CBT method may be not as good as that with the SBT method. This consideration will be confirmed by our simulation result.
4. With this method, since the eligibility of candidate entries only depends on a single tree, the router need not to construct routing entries for individual sources. For one anycast group with M recipients and N sources, with the CBT method, the off-CBT routers only need to keep *one* entry, the on-CBT routers need to keep $O(M)$ entries. Hence, the CBT method needs less storage than the SBT method.

In this subsection, we have proposed four methods to determine eligible routing entries. In the following subsection, we will discuss how to use the eligible routing entries determined by these methods to construct anycast routing tables.

4.3 Construction of Anycast Routing Table Using the Eligible Entries

Once eligible entries are determined, constructing an anycast routing table is straightforward. For anycast, we may have to add some additional fields to the traditional entries. For example, an additional field is needed to store the weight information. This field will be used in multi-entry selection (which will be discussed in Section 5.2). Another field (i.e., the source field) is needed as well in the case that the SBT method is used.

It is recommended that the entries for the same anycast groups are stored successively in the anycast routing table. In this way, locating entries in the anycast routing table in packet forwarding can be simplified.

We have discussed the three tasks that the routing table establishment subprotocol has to perform. Note that these tasks should be carried whenever the network status, such as reachablity of the nodes and the membership of anycast groups, are changed. In this way, the resultant routing table will be consistent with the real network topology.

5 Packet Forwarding Subprotocol

The objective of the Packet Forwarding Subprotocol is to forward a packet when it arrives at a router. This is achieved by performing the following three tasks in sequence:

Task A. To locate the routing entry (entries) for incoming anycast packet

Task B. To select one entry among the located entries.

Task C. To forward the anycast packet according to the fields in the selected entry.

We will discuss each task in following subsections.

5.1 Task A: Locating of Anycast Routing Entries

Two steps are taken to locate routing entries by a router when it receive a packet:

Step 1. Extract the routing information from the header of the packet, such as the destination, source, and flow identification.

Step 2. Search in the routing table (or its cache) for the entries that match the extracted routing information from the packet.

The work in Step 1 is straightforward. The extracted information will be used in Step 2 to locate the matched entries. The way to locate the routing entries may vary depending on whether the packet is an independent one or one in a flow. The routing entries for a flow may be stored in a cache for routing the successive packets from the same flow. Thus, for the dependent packets, the router may directly search the cache instead of the routing table.

5.2 Task B: Selection of Multiple Entries

Once the multiple routing entries are located, the next task is how to select one entry to forward the anycast packet. To avoid congestion and to obtain good QoS, we propose a weighted random selection approach.

5.2.1 Weighted-Random Selection

The basic idea of the weighted random selection approach is that first each matched entry is assigned a weight, then one entry is chosen randomly based on weights. The entry with higher weight value has higher probability to be selected than those with lower weight values. This approach may effectively distribute the traffic over different routes, and potentially improves the delay and throughput performance. To fully take advantage of this load sharing effect, we must carefully assign the weights to the matched entries.

Weight assignments for multipath routing have been studied in the literature. For example, an optimal weight assignment was proposed in [13] by which the average delay of messages can be minimized. We will take a heuristic approach. Our idea is to have a simple parameterized weight assignment formula that can cover a wide range of heuristics that one may like to use in practice.

Without loss of generality, let us assume that there are K eligible entries in a routing table for a given anycast address. Let these entries be indexed by $1, 2, \ldots, K$; and their associated weights be denoted as $W_1, W_2, \ldots, W_K$, respectively.

From the information provided in the entries of a routing table, how should the weight of an entry be assigned? A rule of thumb is that the weight should be inversely proportional to the distance of the route that leads to a member in the recipient group. That is, for $i = 1, 2, \ldots, K$,

$$W_i \propto 1/D_i, \tag{3}$$

where D_i is the value in the distance field of entry i. To normalize W_i to be a probability such that $\Sigma_{i=1}^{K} W_i = 1$, we should assign W_i as follows: for $i = 1, 2, \ldots, K$,

$$W_i = \frac{\left(\frac{1}{D_i}\right)}{\Sigma_{j=1}^{K}\left(\frac{1}{D_j}\right)}. \tag{4}$$

However, we may further generalize (4) by raising the factor $1/D_i$ to a power of r where r is a nonnegative real number. That is, we propose, for $i = 1, 2, \ldots, K$,

$$W_i = \frac{\left(\frac{1}{D_i}\right)^r}{\Sigma_{j=1}^{K}\left(\frac{1}{D_j}\right)^r}. \tag{5}$$

We would like to argue that (5) presents a wide spectrum of heuristics. We illustrate this by several special cases. First, if $r = 0$, then (5) becomes, for $i = 1, 2, \ldots, K$,

$$W_i = 1/K. \tag{6}$$

This implies that all the eligible entries get the equal chance to be selected if $r = 0$.

Now consider the case when r approaches infinity. We have the following observations. Assume that $D_j < D_i$ for $i = 1, 2, \ldots, K$ and $i \neq j$. That is, entry j will route the anycast packets along the SSP route. We can rewrite $1/D_i$ as β_i/D_j, where $0 < \beta_i < 1$. Note that $\lim_{r\to\infty} \beta_i^r = 0$. Substituting $1/D_i = \beta_i/D_j$ into (5), we obtain the weight for entry j as follows:

$$W_j = \lim_{r\to\infty} \frac{\left(\frac{1}{D_j}\right)^r}{\Sigma_{i=1}^{K}\left(\frac{1}{D_i}\right)^r} = \lim_{r\to\infty} \frac{\left(\frac{1}{D_j}\right)^r}{\left(\frac{1}{D_j}\right)^r\left(1 + \Sigma_{i=1,i\neq j}^{K}\beta_i^r\right)} = 1. \tag{7}$$

For $i = 1, 2, \ldots, K$ and $i \neq j$,

$$W_i = \lim_{r\to\infty} \frac{\left(\frac{1}{D_i}\right)^r}{\Sigma_{h=1}^{K}\left(\frac{1}{D_h}\right)^r} = \lim_{r\to\infty} \frac{\left(\frac{1}{D_j}\right)^r \beta_i^r}{\left(\frac{1}{D_j}\right)^r\left(1 + \Sigma_{i=h,h\neq j}^{K}\beta_h^r\right)} = 0. \tag{8}$$

That is, when r approaches infinity, our weight assignment algorithm will result in only SSP being selected.

5.2.2 The Methods to Determine the Value of r

We have shown that the weight assignment is sensitive to the value of r. In this subsection, we would like to discuss the methods for determining the value of r. We propose two methods: the *fixed method* and the *adaptive method.* They are different on whether being adaptive to the dynamics of traffic in the network. For the fixed method, the value of r is fixed for all the routers throughout the whole session. For the adaptive method, the value of r is adaptive to the change of network traffic.

Let us consider the following two cases in order to develop heuristics to the adaptive r:

Case 1. If the traffic is light, there is no congestion at the links in a router. As we know, the transmission delay and queuing delay at the links are two major components of the total delay suffered by a packet. In this case, the queuing delay is very small, the transmission delay plays the key role in the total delay. Thus, since the SSP (say, it is associated with entry j) costs the smallest transmission delay, W_j should get very large value, compared with the weights of other routes. In the extreme case, we may like to have W_j to be *one* and all other $W_i (i \neq j)$ to be *zero*. From (7) and (8), we know that in this case, r should be infinity. This implies that when the traffic is light, the value of r should be large.

Case 2. If the traffic is very heavy, the queuing delay will become large, compared with the transmission delay suffered by a packet. We would like to distribute traffic to all the routes to reduce the queuing delay. In other words, the weights for all the routes should be almost equal. As shown in (6), when $r = 0$, all the weights are the same. Considering the fact that the variance of the weights is a function of r, we conclude that when the traffic is heavy, the value of r should be small.

The above two cases suggest that the value of r should be inversely proportional to the traffic load. Thus, we propose the following heuristic for selecting r:

$$r = \exp(\mu/\lambda) - 1, \tag{9}$$

where λ is the arrival rate of packets, μ is a parameter related the capacity of links of the router.

5.3 Task C: Forwarding Anycast Packets

Once a routing entry is selected, the next task is to forward the packet. This task is straightforward. Based on the selected entry, the router can determine the next hop, and the corresponding output port. The packet is then sent to the output port for transmission.

Additional work has to be done for the first packet in a flow. Once the packet is forwarded, the selected routing entry should be marked and stored in a cache for future usage in forwarding the successive packets in the same flow. The entries in the cache will be expired for a pre-defined time.

6 Discussion

6.1 Correctness of the Routing Protocol

The semantics of an anycast packet require that one copy of an anycast packet be delivered to one of the members in the recipient group if the network has no fault and is connected. We say that a routing protocol is *correct* if the semantics requirement of anycast can be satisfied.

The proof for correctness of the routing protocol seems to be trivial. One would argue that if the network has no faults, all the routers and hosts are connected. Hence, there is always at least one path from a source to a member in the recipient group. The routing table establishment subprotocol can always construct an entry, which corresponds to the path. While correctness is indeed obvious for the routing table establishment subprotocol when it uses the SSP method to determine the eligible entry, it may not be so for the subprotocol when the MIN-D, SBT, or CBT method is used. This is because in order to eliminate loops, certain entries have to be marked ineligible. As a result, the correspondent paths will not be used to deliver packets even if the links along the paths are physically functioning. We have to show that after eliminating the ineligible entries, there is still at least one path that connects a source to a member host in the recipient group.

We will prove the correct property of our protocol, respectively conditioned on a particular method (SSPF, MIN_D, SBT, or CBT) is used to determine the eligible entry. The basic idea is, however, the same. We will prove that once a router receives a packet, it can always find at least one next hop[4] to forward the packet. The loop-free property then guarantees that the packet will eventually arrive at a recipient. We need the following lemmas to establish the correct property of our protocol.

Lemma 1. *Let A be an anycast address. For any router R, there is X, where X is either a router or a host, such that $R >_{SSP} X$.*

Lemma 2. *Let A be an anycast address. For any router R, there is X, where X is either a router or a host, such that $R >_{MIN\text{-}D} X$.*

Lemma 3. *Let S be a source host, A be an anycast address, and X be a router or host. If there is R such that $R >_{SBT} X$, then there is Y (Y is either a router or host) such that $X >_{SBT} Y$ unless X itself is in $G(A)$.*

Lemma 4. *Let A be an anycast address, and CBT be a core-based tree for anycast group $G(A)$. If R is on the CBT, then there is Y (Y is either a router or host) such that $R >_{CBT} Y$ unless R itself is in $G(A)$.*

4. Clearly, the next hop cannot be the one from which the current router receives the packet.

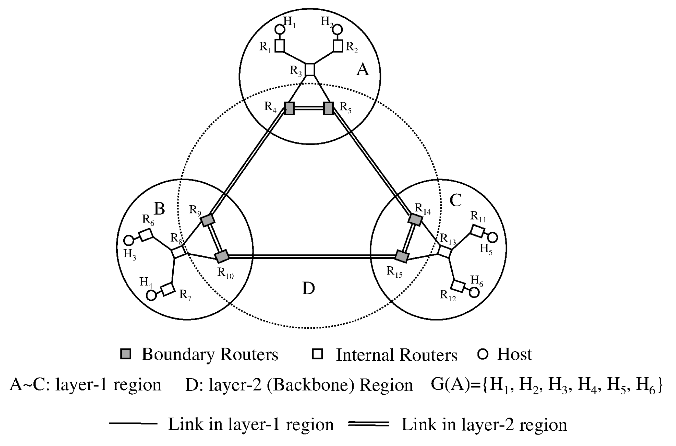


Fig. 8. An example hierarchical network.

The above lemmas describe the availability of next hops to which a router can forward a packet by following the SSP, MIN-D, SBT and CBT order. The proofs of these lemmas are presented in Appendix B. Now, we prove the correctness property of our routing protocol.

Theorem 1 (Correctness of the protocol). *If the network has no fault, then with our routing protocol, any anycast packet from a source will eventually be delivered to one of the members in the recipient group.*

Proof. First, consider the case of the SSP method. Let us say a router R' receives a packet from node R (R can be the source). Router R' can receive a packet from router R only if

$$R >_{SSP} R'. \tag{10}$$

By Lemma 1, there is Y such that Y is a next hop of R' and

$$R' >_{SSP} Y. \tag{11}$$

Equation (11) means that the entry with Y being the next hop is eligible. That is, router R' can route an incoming packet to Y. Given that any path taken by our protocol is loop free, this packet will eventually be delivered.

Similarly, for the cases of the MIN-D and SBT methods, the proofs can be obtained.

Now, we consider the case of the CBT method. Assume that router R receives an anycast packet. Two subcases need to be considered.

Subcase 1. R is a router on the core-based tree. Lemma 4 guarantees that either R is a terminal leave on the tree or there must be at least one eligible entry with a next hop (say X). Then, R can forward the packet to X.

Subcase 2. R is a router not on the core-based tree. According to Lemma 1, there must be one eligible entry with the next hop (say Y). Then R can forward the packet to Y.

Given that any path taken by our protocol is loop free, this packet will eventually be delivered. □

6.2 Extension to Hierarchical Routing

The protocol we described above can be extended to hierarchical networks. A hierarchical network consists of multiple layers. Routers at each layer are partitioned into regions. Several regions at a layer are interconnected by routers in a region at a higher layer. See Fig. 8 for a two-layer hierarchical network. At the low layer (layer-1), there are regions A, B, and C. Each consists of a number of routers, among which, some are *boundary* routers, and the rest are *internal* ones. The boundary routers of the layer-1 region are connected by a high layer (layer-2) region (D). Each boundary router belongs to one layer-1 region as well as the layer-2 region.

Under the hierarchical network model, it is reasonable to assume that every router in a region is able to obtain *complete* routing information of its own region, but a router may not have such information for other regions. Because of this, our protocol described earlier needs to be extended to handle this kind of situation.

One idea is to use the reverse-path routing. This method was proposed to extend tree-based routing algorithms of multicast to the hierarchical network [20]. It works in this way: In the hierarchical network, an internal router has only the distance information from itself to the out region routers, rather than from the out region routers to itself. Because of this, an internal router may use the former to approximate the latter. In this way, the proper infrastructure (e.g., trees) can be built as requested by our protocol. While this method is straightforward and easy to understand, in the presence of asymmetry links, the reversed-path routing may lead to less efficient routes. Also, it does not take advantage of the semantics of anycast messages.

Another idea on extending our protocol to hierarchical networks is to let boundary routers act as *proxies*. To illustrate this idea, we will first consider the networks where each layer-1 region has exactly one boundary router. We will discuss how to extend it to the general case. Assume that we are using the CBT method to determine the entry eligibility. The question is how to build the core-based tree(s). We propose to construct multiple core-based trees, one for one region, instead of one CBT for the whole network. There are two types of core-based trees:

Type 1. The CBT in layer-1 region. Each region has a core-based tree in its own region. The boundary router acts as a proxy for the out-region members in the anycast receipt group. That is, this regional CBT stops at the boundary of the region.

Type 2. The CBT in layer-2 region. The backbone region, i.e., the layer-2 region, is also a core-based tree. The boundary routers act as proxies for the anycast group members of their layer-1 regions. The core-based tree also stops at the boundary routers.

Obviously, starting from the root of the CBT in the layer-2 region, we still have a complete tree that covers all the members in the anycast receipt group. Consequently, each node can apply the CBT method to determine the entry eligibility and use the Packet Forwarding Subprotocol, described in Section 5, to forward the anycast packets.

What should we do if a layer-1 region has more than one boundary router? One method is to partition the region into subregions so that each subregion has exactly one boundary router. Then we can establish a CBT in each of these subregions. The other parts of the protocol should then work as before. We see that this method takes advantage of anycast: if an anycast packet from one region, via the backbone, gets into a subregion of another region, then the other subregions will not receive this packet. Note that this is exactly what the anycast semantics defines: an anycast packet should be received by one of the members in the receipt group. Extension with other eligibility determination methods can be similarly carried out.

Several problems remain to be addressed: how to partition a region so that the delay-throughput performance can be optimized? Another issue is how to perform intra- and interregional group management. Due to the space limitation of this paper, we will not further discuss these issues here. A systematic treatment on these issues and a comprehensive discussion on the anycast protocols for hierarchical networks will be given in a forthcoming paper.

7 Performance Evaluation

7.1 Simulation Model

In this section, we will discuss performance results of the routing protocol we introduced in this paper. We consider an ARPANET [21]. There are 47 nodes that are interconnected by 92 links. Every node is a router and has one host attached. We assume that the size of an anycast group is five. Link bandwidth capacity is assumed to be 10Mbits per second. The distance value of each link is normalized to be one.

To obtain the performance data, we use a discrete event model to simulate the network. The program is written in the C programming language and runs in a SUN SPARC workstation 20. The delay of a packet (in seconds) at a router is defined as the summation of the routing delay, the queuing delay, and the transmission delay. The end-to-end delay of a packet is the sum of the delays at all the routers through which the packet passes. For a given session, we are interested in the average end-to-end delay of all anycast packets.

A triple $<\alpha,\beta,\gamma>$ is used to represent the parameters of the system we simulate. In particular, α represents the method used to determine eligibility of routing entries. That is,

$$\alpha \in \{\mathrm{SSP}, \text{MIN-D}, \mathrm{SBT}, \mathrm{CBT}\}.$$

β indicates the kind of weight assignment algorithm we use. Hence,

$$\beta \in \{\text{fixed}, \text{adaptive}\}.$$

Finally,

$$\gamma \in \{\text{independent}, \text{flow}\}$$

which denotes the type of anycast traffic in the simulated system. The anycast packets arrive as a Poisson process.

For example, <SSP, fixed, flow> represents a system in which the SSP method is used to determine the eligibility of routing entries, the fixed method is used to assign weights to the entries, and anycast traffic in the system is not

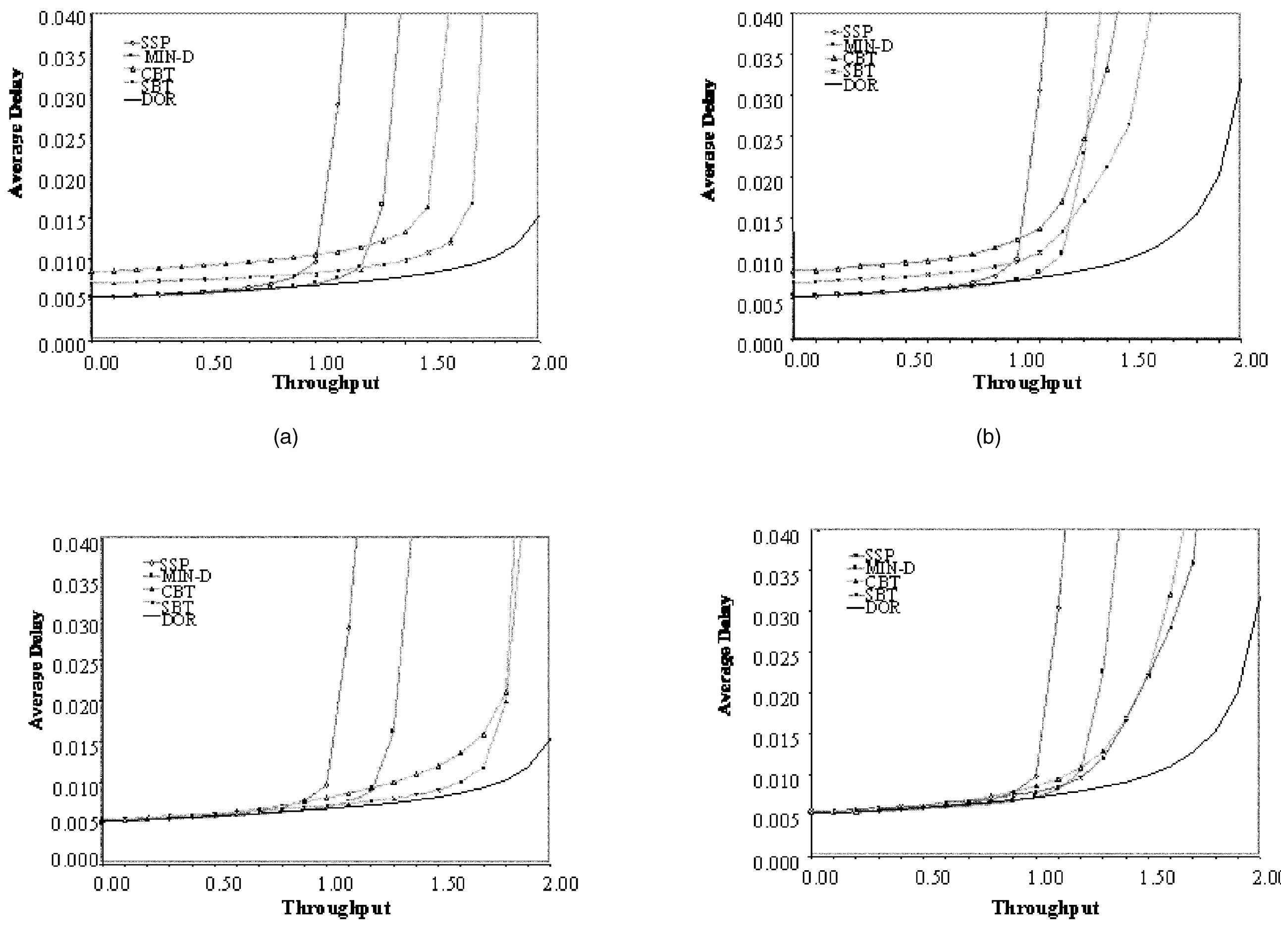


Fig. 9. The delay performance. (a) $<^*, \mathrm{Fixed}, \mathrm{Nonflow}>$, (b) $<^*, \mathrm{Fixed}, \mathrm{Flow}>$, (c) $<^*, \mathrm{Adaptive}, \mathrm{Nonflow}>$, (d) $<^*, \mathrm{Adaptive}, \mathrm{Flow}>$.

independent but consists of flows. To simplify our discussion, we use symbol "$*$" to denote the value of a parameter if the discussion covers all the values of that parameter. For example, <SSP, adaptive, * > means systems that uses the SSP method for determination of routing entry eligibility, the adaptive method for weight assignment. The traffic type can be either independent or flow.

In the simulation, when the fixed method is used to assign the weight, r is set to be *one*. The motivation for doing so is as follows: we are mostly interested in the heavily loaded situation, where the proper control mechanism should be invoked in order to ensure reasonable delay throughput performance. In the Appendix, we show that under the assumption of heavy loading, an approximate analysis reveals that the proper weight assignment, as defined in Section 5.2, should be of the following form, for $i = 1, 2, \ldots, K$,

$$W_i = \frac{\left(\frac{1}{D_i}\right)}{\Sigma_{h=1}^{K}\left(\frac{1}{D_h}\right)}. \tag{12}$$

This is a special case of (5) when $r = 1$.

We will also consider a baseline protocol for comparison. The baseline protocol is *dynamic optimal routing protocol* (DOR). For each packet, this protocol will select a path that results in the minimum end-to-end delay for the packet. The DOR protocol achieves this by assuming that, without any cost, the complete dynamic system state information (e.g., queue length at each interface and current arrivals, etc.) is available and uses it in making runtime decisions on how to route a packet. Obviously, the performance of this protocol is ideal but it is not realistic and cannot be implemented in practice.

7.2 Performance Observations

The simulation results are shown in Fig. 9. We observe that a combination of some methods we proposed in this paper (e.g., <SBT, adaptive, * > and <CBT, adaptive, * >) results in delay performance that is close to that of DOR under most throughput conditions. This demonstrates the effectiveness of these methods and justifies our approach. In particular, we note that the CBT method requires much less storage than SBT, and hence, is scalable to large systems. As a result, for practical usage, <CBT, adaptive, * > is recommended. Next, we would like to make some remarks on the sensitivities of various methods.

1. *The delay is sensitive to the method used for determining eligibility of routing entries.*

   a. When the system throughput is light, the average delay is small, as expected. Nevertheless, we note that the delay of <SSP, *, * > is lower than that of <MIN-D, *, * >, <CBT, *, * > and <SBT, *, * >, and is close to that of the system running DOR. This fact indicates that in the light throughput situation, it may be unnecessary to distribute anycast traffic over different paths.
   b. When the system throughput increases, the average delay increases. Furthermore, the performance difference due to the different methods used to determine entry eligibility becomes clearly visible. We observe that the delay of <SSP, *, * > increases rapidly. In each graph in Fig. 9, when Throughput is more than 1.20, the delay of <SSP, *, * > is beyond 0.040, while the delay of <MIN-D, *, * >, <CBT, *, * > and <SBT, *, * > stays at a relatively low value. The poor performance of the SSP method is due to the fact that it does not distribute traffic among different paths. As a result, the path, which the SSP method uses, quickly becomes overloaded when the overall throughput becomes large.
   c. Under the medium and heavy throughput conditions, the delay of <CBT, *, * > and <SBT, *, * > are lower than that of <MIN-D, * , * >. For example, in Fig. 9a, when Throughput is 1.25, the SBT and CBT methods result in delay values of 0.010 and 0.007, respectively, while MIN-D results in a delay of 0.016.
   d. In most loading situations (except the extremely heavily loaded case, i.e, throughput is around 1.5), the delay of the systems using the SBT and CBT methods is close to that of DOR. This indicates the effectiveness of these methods due to their ability to distribute traffic over different paths.

2. *The delay is sensitive to the weight assignment method.*

   a. In particular, the systems using the adaptive weight assignment method always lead to better performance. For example, in the low or medium throughput conditions, the delay of <CBT, adaptive, * > is very close to that of the system running the DOR protocol (Fig. 9c), while the delay of <CBT, fixed , * > is much higher than that of the system running DOR protocol (Fig. 9a). A similar observation can be made for <SBT, fixed, * > vis <SBT, adaptive, * >. The good performance of < *, adaptive, * > is due to its ability to use the dynamic traffic information.
   b. However, the sensitivity on the weight assignment methods seem to vary, depending on the method used for determining routing entry eligibility. In particular, the delay of the system running the protocol using SSP and MIN-D methods seems to be not much sensitive to the weight assignment method at all. From Fig. 9, we can see that the delay of <SSP, fixed, * > and <SSP, adaptive, * > are almost equal under any throughput condition. So do <MIN-D, fixed, * > and <MIN-D, adaptive, * >.

      On the other hand, the delay performance of the systems running the protocol using CBT and SBT methods (especially CBT methods) are very sensitive to the weight assignment method. The observation can be explained by the fact that there are more eligible entries to chose in the case that either the SBT or CBT method is used. Furthermore, since the core based tree has more traffic concentration than the source based tree, the adaptive method is especially useful to the CBT method.

3. *The delay is also sensitive to the traffic type.* It is obvious that the delay of < *, *, independent> is usually lower than (or equal to) the one of < *, *, flow>. The reason is that in the latter case, the protocol can only do multipath routing on the first packet of one flow and the successive packets can only be routed along the same path chosen with the first packet. This also can be used to explain why the system running the DOR protocol also performs worse when traffic is not independent but consists of flows.

8 Conclusion

In this paper, we have addressed the issues related to routing anycast packets. The anycast protocol is consisted of two subprotocols: the routing table establishment subprotocol and the packet forwarding subprotocol. For the former, we propose four methods (SSP, MIN-D, SBT, and CBT methods) with which an order among routers is introduced for the purpose loop-free routing. Performance evaluation shows that the systems that use the SBT or CBT method perform much better than the one using the SSP or MIN-D method.

In the packet forwarding subprotocol, we proposed to use random weight assignment methods to deal with the issue of selecting a routing entry from multiple eligible ones. Two weight assignment methods are considered: fixed and adaptive methods. We find that the system using the adaptive weight assignment method outperforms the one with the fixed method. We also note that some combination of the methods introduced in this paper (e.g., <CBT, adaptive, * >) results in delay performance very close to the dynamic optimal routing protocol (DOR). While the DOR is theoretically ideal, it cannot be realized in practice. It is used to provide a baseline for the purpose of performance comparison.

It should be pointed out that our solutions are practical. They are simple and compatible to the existing ones and use comparable amount of information with those protocols that are currently used to route other types (unicast and multicast) of packets. We are currently studying how to extend our methods to deal with other QoS requirements, e.g., hard real-time and fault-tolerant ones.

APPENDIX A
PROOF OF LOOP FREE ROUTING FOR THE MIN-D METHOD

Theorem 2. *The routes resulting from the eligible entries determined by the MIN-D method is guaranteed to be loop free.*

Proof. Let us assume that with the MIN-D method, there is a loop in a route. Let the loop be as follows:

$$R_n \rightarrow R_{n+1} \rightarrow R_{n+2} \ldots \rightarrow R_{n+m} \rightarrow R_n. \tag{13}$$

With the MIN-D method, for any router R_i in the loop, we have:

$$R_i >_{MIN\text{-}D} R_{i+1}. \tag{14}$$

By the definition of $>_{MIN-D}$, we have the following inequalities

$$\min\_\mathrm{d}(R_i, A) > \min\_\mathrm{d}(R_{i+1}, A). \tag{15}$$

That is,

$$\min\_\mathrm{d}(R_n, A) > \min\_\mathrm{d}(R_{n+1}, A) > \ldots > \min\_\mathrm{d}(R_{n+m}, A) \\ > \min\_\mathrm{d}(R_n, A). \tag{16}$$

This is a contradiction. □

APPENDIX B
PROOFS OF LEMMAS 1, 2, 3, AND 4

Lemma 1. *Let A be an anycast address. For any router R, there is X, where X is either a router or a host, such that $R >_{SSP} X$.*

Proof. Because for H in $G(A)$ there is a path (say, $P_{SP}(R,H)$) between R and H, there must be a node X (either a router or a host) following R in $P_{SP}(R,H)$. By the definition of $>_{SSP}$, we have $R >_{SSP} X$. □

Lemma 2. *Let A be an anycast address. For any router R, there is X, where X is either a router or a host, such that $R >_{MIN\text{-}D} X$.*

Proof. Our proof of this lemma is divided into two cases.

Case 1. There is an H such that H is in $G(A)$ and H is directly connected to router R. If so,

$$\min\_\mathrm{d}(R, A) = \min\_\mathrm{d}_{H\in G(A)}(D(P_{SP}(R,H))) > 0 \\ = \min\_\mathrm{d}(H, A). \tag{17}$$

Thus, $R >_{MIN\text{-}D} H$.

Case 2. There is no H such that H is in $G(A)$) and H is directly connected to router R. Assume that R has N next hops, namely $R_1, R_2, \ldots, R_N$. Assume that the lemma is not true. That is, there is no $R_i (i = 1, 2, \ldots, N)$ such that $R >_{MIN\text{-}D} R_i$. This implies, for $i = 1, 2, \ldots, N$,

$$\min\_\mathrm{d}(R, A) = \min_{H\in G(A)}(D(P_{SP}(R,H))) \le \min\_\mathrm{d}(R_i, A) \\ = \min_{H\in G(A)}(D(P_{SP}(R_i,H))). \tag{18}$$

Thus, none of R_i is in $P_{SP}(R,H)$ for some H. Because $R_1, R_2, \ldots$, and R_N are the only next hop routers of R. $P_{SP}(R,H)$ must be a direct path from R to H without going through any other router. That is, H is directly connected to router R. This is a contradiction. □

Lemma 3. *Let S be a source host, A be an anycast address, and X be a router or host. If there is R such that $R >_{SBT} X$, then there is Y (Y is either a router or host) such that $X >_{SBT} Y$ unless X itself is in $G(A)$.*

Proof. Because there is R such that $R >_{SBT} X$, X must be in $P_{SP}(S,H)$ for some H in $G(A)$. Then, if X is the last element in $P_{SP}(S,H)$ (i.e., $X = H$), we are done. Otherwise, let X be the kth element of $P_{SP}(S,H)$. Because X is not the last element in $P_{SP}(S,H)$, there must be Y such that $Y = C(k+1, P_{SP}(S,H))$. By the definition of $>_{SBT}$, we have $X >_{SBT} Y$. □

Lemma 4. *Let A be an anycast address, and CBT be the core-based tree for anycast group $G(A)$. If R is on the CBT, then there is Y (Y is either a router or host) such that $R >_{CBT} Y$ unless R itself is in $G(A)$.*

Proof. Assume the CBT is rooted at C. Because R is on the CBT, R must be in $P_{SP}(C,H)$ for some H in $G(A)$. Then if R is the last element in $P_{SP}(C,H)$ (i.e., $R = H$), we are done. Otherwise, let R be the kth element of $P_{SP}(C,H)$. Because X is not the last element in $P_{SP}(C,H)$, there must be Y such that $Y = C(k+1, P_{SP}(C,H))$. By the definition of $>_{CBT}$, we have $X >_{CBT} Y$. □

APPENDIX C
An Approximate Analysis of Weighted Random Assignment in Heavy Load Situation

In this appendix, we will carry out an approximate analysis in order to derive the weight assignment formulas under the heavy load situation.

Consider a router. Assume that for an anycast address, there are K eligible entries in (one of) its routing table(s). Let these K entries be indexed by $1, 2, \ldots, K$. Let the values in the distance fields of these entries be $D_1, D_2, \ldots, D_K$. The weights of these entries are $W_1, W_2, \ldots, W_K$, respectively. Note that each entry represents a route for packets with the anycast address.

Let us assume that packets arrive at the router with the anycast address as a Poisson process with rate λ. Further, the transmission of the packets along link i (i.e., the one corresponding to entry i) is modeled as an M/M/1 system with the arrival rate being $\lambda_i = \lambda^* W_i$ and service rate $\mu_i = 1/D_i$. We note that these assumptions may not hold in practice but we use them here to approximately analyze the formulas of weight assignments given in Section 7.1. Our simulation study in Section 7 shows that the systems with our weight assignment method do work effectively.

With the approximation assumptions, the average delay of all the packets with an anycast address is given by [12]:

$$\overline{\mathrm{d}} = f(W_1, W_2, \ldots, W_K) = \Sigma_{j=1}^{K} W_j \frac{1/\mu_j}{1\text{-}\lambda W_j/\mu_j}, \tag{19}$$

where $W_1, W_2, \ldots,$ and W_K are subject to the following constraints

$$g(W_1, W_2, \ldots, W_K) = \Sigma_{j=1\ldots,K} W_j = 1. \tag{20}$$

We would like to find assignments to $W_1, W_2, \ldots, W_K$ such that $\overline{\mathrm{d}}$ is minimized. We take two steps to get such assignments: at the first step, we use the Lagrange multipliers method [18] to obtain assignments, which will result in local optimum of $\overline{\mathrm{d}}$. Secondly, we verify whether the resultant assignments indeed minimize $\overline{\mathrm{d}}$.

Step 1. Derivation of assignments to $W_1, W_2, \ldots, W_K$, that will result in local optimum of $\overline{\mathrm{d}}$.

According to the Lagrange multipliers method [18], to obtain a local optimum value of $\overline{\mathrm{d}}$, weights must satisfy the following Lagrange multipler equations: for $j = 1, \ldots, K$,

$$\frac{\partial f}{\partial W_j} = \alpha \frac{\partial g}{\partial W_j} \tag{21}$$

and

$$g(W_1, W_2, \ldots, W_K) = \Sigma_{j=1} W_j = 1. \tag{22}$$

Let us solve $W_1, W_2, \ldots, W_K$ from the system of equations (21) and (22). For $j = 1, \ldots, K,$ we have, from (19)

$$\frac{\partial f}{\partial W_j} = \frac{1/\mu_j}{(1 - \lambda W_j/\mu_j)^2}, \tag{23}$$

and

$$\frac{\partial g}{\partial W_j} = 1. \tag{24}$$

Then, substituting (23) and (24) into (21), we have

$$\alpha = \frac{1/\mu_1}{(1 - \lambda W_1/\mu_1)^2} = \frac{1/\mu_2}{(1 - \lambda W_2/\mu_2)^2} = \ldots = \frac{1/\mu_K}{(1 - \lambda W_K/\mu_K)^2}. \tag{25}$$

That is, for any $1 < i, j < k$,

$$\frac{1/\mu_i}{(1 - \lambda W_i/\mu_i)^2} = \frac{1/\mu_j}{(1 - \lambda W_j/\mu_j)^2}. \tag{26}$$

We assume that the system is stable, i.e., $\lambda W_i/\mu_i < 1,$ for any $1 < i < K$. Solving W_i from (26), we have

$$W_i = \frac{\mu_i - \sqrt{\mu_i \mu_j}}{\lambda} + \sqrt{\frac{\mu_i}{\mu_j}} W_j. \tag{27}$$

Substituting (27) into (22) and solving W_j, we have

$$W_j = \frac{1 - \Sigma_{i=1}^K \mu_j/\lambda + \sqrt{\mu_j}\ \Sigma_{i=1}^K \sqrt{\mu_i}/\lambda}{\Sigma_{i=1}^K \sqrt{\mu_i}/\sqrt{\mu_j}}. \tag{28}$$

Step 2. Verification that the weight assignment (28) indeed minimizes $\overline{\mathrm{d}}$.

In this step, we would like to verify if the weight assignment satisfies the sufficient condition of minimizing. The sufficient condition is related to the following $(K+1) \times (K+1)$ matrix:

$$B_1 = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1K} & g^{(1)} \\ f_{21} & f_{22} & \cdots & f_{2K} & g^{(2)} \\ \cdots & & & & \\ f_{K1} & f_{K2} & \cdots & f_{KK} & g^{(K)} \\ g^{(1)} & g^{(2)} & \cdots & g^{(K)} & 0 \end{bmatrix}, \tag{29}$$

where

$$f_{ji} = \frac{\partial^2 f}{\partial W_j\, \partial W_i},\ g^{(j)} = \frac{\partial g}{\partial W_j}, i, j = 1, 2, \ldots, K.$$

Let Δ_1 denote the determinant of B_1. Furthermore, let $\Delta_2, \Delta_3, \ldots \Delta_{K-1}$ denote a set of principal minors of B_1, namely, the determinants of the principal submatrices $B_2, B_3, \ldots, B_{K-1},$ where B_n is obtained by deleting the first $n-1$ rows and the first $n-1$ columns of $B_1 (n = 2, 3, \ldots, K-1)$. All the partial derivatives used in $B_1, B_2, \ldots, B_{K-1}$ are evaluated at the assignment to $W_1, W_2, \ldots, W_K$ according to (28). Then, the sufficient condition for the assignment minimizing f (i.e., $\overline{\mathrm{d}}$) is as follows [18]:

$$\Delta_n < 0,\ \ n = 2, 3, \ldots, K-1. \tag{30}$$

Before we check if the condition (30) holds, we would like to derive the formula of f_{ji},[5] $j = 1, 2, 3, \ldots, K$. From (23), for $i, j = 1, 2, 3, \ldots, K$, we have

$$f_{ji} = \frac{\partial^2 f}{\partial W_j\, \partial W_i} = \partial\left(\frac{1/\mu_j}{(1 - \lambda W_j/\mu_j)^2}\right)/\partial W_i = \begin{cases} 0 & if\ \ i \neq j \\ \frac{2\lambda\mu_j}{(\mu_j - \lambda W_j)^3} & otherwise. \end{cases} \tag{31}$$

Let f_{ji}^* be the value of f_{ji} when f_{ji} is evaluated at the assignment (28). Substituting (28) into (31), for $i, j = 1, 2, 3, \ldots, K$, we have:

$$f_{ji}^* = \begin{cases} 0 & if\ \ i \neq j \\ \frac{\beta}{\sqrt{\mu_j}} & otherwise, \end{cases} \tag{32}$$

where

$$\beta = \frac{2\lambda\left(\Sigma_{i=1}^K \sqrt{\mu_i}\right)^3}{\left(\Sigma_{i=1}^K \mu_i - \lambda\right)^3}.$$

Since the system we consider is stable,

$$\Sigma_{i=1}^K \mu_i - \lambda > 0. \tag{33}$$

Substituting (33) into (32), for $j = 1, 2, \ldots, K$, we have,

$$\beta > 0, \quad f_{jj}^* > 0. \tag{34}$$

Having derived f_{ji} and $(i, j = 1, \ldots, K)$, let us check if (30) holds. When the partial derivatives in B_1 are evaluated at (28), according to (24) and (32), B_1 turns to be:

$$B_1 = \begin{bmatrix} \frac{\beta}{\sqrt{\mu_1}} & 0 & \cdots & 0 & 1 \\ 0 & \frac{\beta}{\sqrt{\mu_2}} & \cdots & 0 & 1 \\ \cdots & & & & \\ 0 & 0 & \cdots & \frac{\beta}{\sqrt{\mu_K}} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}. \tag{35}$$

5. We have already had (24) for $g^{(i)}$.

From (35), we can get B_{K-1} as follows:

$$B_{K-1} = \begin{bmatrix} \frac{\beta}{\sqrt{\mu_{K-1}}} & 0 & 1 \\ 0 & \frac{\beta}{\sqrt{\mu_K}} & 1 \\ 1 & 1 & 0 \end{bmatrix}. \quad (36)$$

Then, we have

$$\Delta_{K-1} = -\frac{\beta}{\sqrt{\mu_{K-1}}} - \frac{\beta}{\sqrt{\mu_K}} < 0. \quad (37)$$

From (35), for $n = 1, 2, \ldots, K-2$, it is easy to get:

$$\Delta_n = \frac{\beta}{\sqrt{\mu_n}} \Delta_{n+1} + (-1)^{(K-n+1)} \pi_n, \quad (38)$$

where π_n are the determinants of the submatrices A_n of $B_n (n = 1, 2, \ldots, K-2)$. A_n is obtained by deleting the first row and the last column of B_n. The value of π_n is as follows,

$$\pi_n = (-1)^{(K-n)} \Pi_{m=n+1}^{K} \frac{\beta}{\sqrt{\mu_m}}, n = 1, 2, \ldots, K-2. \quad (39)$$

Substituting (39) into (38), we can get,

$$\Delta_n = \frac{\beta}{\sqrt{\mu_n}} \Delta_{n+1} - \Pi_{m=n+1}^{K} \frac{\beta}{\sqrt{\mu_m}}, n = 1, 2, \ldots, K-2. \quad (40)$$

Based on (34), (37), and (40), we can conclude that:

$$\Delta_n < 0 \quad n = 1, 2 \ldots, K-1 \quad (41)$$

Thus, the sufficient condition for the assignment minimizing f (i.e., $\overline{\mathrm{d}}$) (30) holds.

Under the heavy loading condition, we can approximately assume

$$\lambda \approx \Pi_{i=1}^{K} \mu_i. \quad (42)$$

Substituting $\mu_i = 1/D_i$ and (42) into (28), we have a simplified expression for weight W_j in terms of $D_1, D_2, \ldots, D_K$:

$$W_j \approx \frac{1/D_j}{\Sigma_{i=1}^{K} 1/D_i}. \quad (43)$$

We note that (43) is a special case of (5) where $r = 1$.

References

[1] S. Bhattacharjee, M.H. Ammar, E.W. Zegura, V. Shah, and Z. Fei, "Application-Layer Anycasting," *Proc. IEEE INFOCOM '97,* Apr. 1997.

[2] A.J. Ballardie, P.F. Francis, and J. Crowcroft, "Core Based Trees," *Proc. ACM SIGCOM,* 1993.

[3] J. Bound., "Ipv6 Anycasting Service: Minimum Requirements for End Nodes," draft-bound-anycast-00.txt, June 1996.

[4] D.G. Cantor and M. Gerla, "Optimal Routing in a Packet-Switched Computer Network," *IEEE Trans. Computer,* vol. 23, no. 10, Oct. 1974.

[5] I. Cidon, R. Rom, "Multi-Path Routing Combined with Resource Reservation," *Proc. IEEE INFOCOM '97,* Apr. 1997.

[6] S. Deering, D.L. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei, "The PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Trans. Networking,* vol. 4, no. 2, pp. 153-162, Apr. 1996.

[7] S. Deering and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification," RFC 2460, Dec. 1998.

[8] S. Deering, "Multicast Routing in Internetworks and Extended LANs," *Proc. SIGCOMM,* Aug. 1988.

[9] A. Ephremides, "The Routing Problem in Computer Networks," *Comm. and Networks,* I. Blake and H. Poor, eds. New York: Springer Verlag, 1986.

[10] L. Fratta, M. Cerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design," *Networks,* vol. 3, 1973.

[11] H. Frank and W. Chou, "Routing in Computer Networks," *Networks,* vol. 1, 1971.

[12] D. Gross, C.M. Harris, *Fundamentals of Queueing Theory.* John Wiley and Sons, 1985.

[13] R.G. Gallager, "A Minimum Delay Routing Algorithms Using Distributed Computation," *IEEE Trans. Comm.,* vol. 25, no. 1, Jan. 1977.

[14] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," *RFC 1884,* Dec. 1995.

[15] C.L. Hedricks, "An Introduction to IGRP, Center for Computer and Information Services, Laboratory for Computer Science Research, Rutgers University, Aug. 22, 1991.

[16] A. Jean-Marie and L. Gun, "Parallel Queues with Resequencing," J. ACM, vol. 40, no. 5, Nov. 1993.

[17] A. Jean-Marie and Z. Liu, "A Stochastic Comparison for Queuing Models via Random Sums and Intervals," *J. Advanced Applied Probabilities,* no. 24, 1992.

[18] A.I. Khuri, *Advanced Calculus with Applications in Statistics.* John Wiley and Sons, 1993.

[19] J. Moy, "OSPF Version 2," RFC 1583, Mar. 1994.

[20] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Mar. 1994.

[21] M. Parsa and J.J. Garcia-Luna-Aceves, "A Protocol for Scalable Loop-Free Multicast Routing. Selected Areas in Communications," *IEEE J. Selected Areas in Comm.,* vol. 15, no. 3, Apr. 1997.

[22] C. Partridge, T. Mendez, and W. Milliken., "Host Anycasting Service," RFC 1546, Nov. 1993.

[23] M. Schwartz and C.K. Cheung, "The Gradient Projection Algorithm for Multiple Routing in Message-Switched Networks," *IEEE Trans. Comm.,* vol. 24, no. 4, Apr. 1976.

[24] D. Thaler and C. Ravishankar, "Distributed Center-Location Algorithms," *IEEE J. Selected Areas in Comm.,* vol. 15, no. 3, Apr. 1997.

Dong Xuan received the BS and MS degrees in electronic engineering from the Shanghai Jiao Tong University (SJTU), China, in 1990 and 1993, respectively. He was on the faculty of the Electronic Engineering Department at SJTU from 1993 to 1997. In 1997, he worked as a research/teaching assistant at the City University of Hong Kong. He is presently a PhD student at the Department of Computer Science, Texas A&M University. His main research interests are real-time computing and communication, computer networks protocols, and distributed systems.

Weijia Jia received the BSc, MSc, and PhD degrees in 1982, 1984, and 1993, respectively, all in computer science. In 1987, he worked at the Department of Computer Science, University of Ottawa, Canada, as a visiting researcher. In 1993, he joined the German National Research Center for Information Technology (GMD) in St. Augustin as a research fellow. Currently, he is an assistant professor at the Department of Computer Science, City University of Hong Kong. His research interests include computer network and systems (with emphasis on parallel and distributed systems), fault-tolerant multicast protocols, real-time communications, and next generation Internet. He is a member of the IEEE Computer Society.

Wei Zhao received the BSc degree in physics from Shanxi Normal University, China, and the MSc and PhD degrees in computer and information science from the University of Massachusetts at Amherst in 1983 and 1986, respectively. In 1990, he joined the Department of Computer Science at Texas A&M University, where he is currently a full professor and department head.

Dr. Zhao's current research interests include secured real-time computing and communication, distributed operating systems, database systems and fault tolerant systems. Dr. Zhao was an editor of the *IEEE Transactions on Computers*. He served as a guest editor for special issues on real-time operating systems for the *ACM Operating System Review* and real-time middleware for the *International Journal of Real-Time Systems*. Currently, he is an editor for the *International Journal of Real-Time Systems* and a member of the executive committee of the IEEE Technical Committee of Real-Time Systems. He has published more than 130 papers in journals, conferences, and book chapters. He is an inventor of two US patents. Dr. Zhao and his students received the Best Paper Awards in the IEEE International Conference on Distributed Computing Systems and in the IEEE National Aerospace and Electronics Conference. Dr. Zhao is a senior member of the IEEE.

Hongwen Zhu received the BS degree in electronic engineering from the Tsinghua University, China, in 1963. In 1981, he worked at the German National Research Cener for Information Technology (GMD) as a visiting researcher. In 1987, he joined the Shanghai Jiao Tong University, where he is currently a full professor in the Department of Electronic Engineering and director of the Modern Communication Technology Institute. Mr. Zhu's current research interests include information security, the Cooperative Supported Computer System (CSCW), and Internet communication. He is a fellow of the China Institute of Communication (CIC). Currently, he is serving as president of the Shanghai Information Technology (IT) Consultant Committee.