

# Integrated Routing Algorithms for Anycast Messages

Weijia Jia, City University of Hong Kong

Dong Xuan and Wei Zhao, Texas A&M University

## ABSTRACT

Use of anycast service can considerably simplify many communication applications. Two approaches can be used for routing anycast packets. Single-path routing always uses the same path for anycast packets from a source, while multipath routing splits anycast traffic into several different paths. The former is simple and easy to implement, while the latter may potentially reduce congestion, improving delay and throughput performance. However, multipath routing may require additional cost to maintain multipath information. To take advantage of both approaches and overcome their shortcomings, we develop a set of integrated routing algorithms that can adaptively select a subset of routers in the network to carry out multipath routing for anycast messages. We demonstrate that systems using our integrated routing algorithms perform substantially better than those where either the single-path or multipath approach is used alone.

## ANYCAST

Anycast is a new communication service defined in IPv6. An anycast message is one that should be delivered to one member in a group of designated recipients [1]. As stated in [2], anycast service provides "a stateless best effort delivery of an anycast datagram to at least one host; and preferably only one host, which serves the anycast address."

Using anycast communication services may considerably simplify some applications. For example, anycast queries can be used to locate an appropriate server from a group of available ones. This will improve efficiency over multicast queries, as proposed in Service Location Protocol (SLP) [3]. In particular, multiple mirrored Web sites can share a single anycast address with which users could simply send a request in order to obtain information (weather information, stock quotes, etc.).

Several studies have been done on communication with anycast messages since the notion was introduced [2]. In [4] it was determined that the anycast addresses are allocated from the unicast address space with unicast address formats. Recently, Deering and Hinden defined Subnet-Router as an anycast address [1] for all routers

within a subnet prefix, taken from the unicast address space, an additional set of reserved anycast addresses within each subnet prefix has been defined and listed. In [5], the implication of an anycasting service supported at the application layer was explored.

## SINGLE- AND MULTIPATH ROUTING

Routing a packet in a network is concerned with determining a path for the packet to travel from its source to its destination. There are two classes of routing approaches: single-path routing (SPR) and multipath routing (MPR). With SPR, the path used for transmitting packets from source to destination is unique (unless, of course, the network topology changes). SPR, such as Shortest Path First (SPF) routing, is simple and easy to implement, and has been widely used in the Internet.

A problem associated with SPR is that it may overload the selected path and hence cause traffic congestion. The MPR algorithm intends to split traffic into several different paths. Many analyses have shown that multipath routing algorithms can increase network throughput and decrease message delay. A recent survey on MPR algorithms can be found in [6]. Naturally, one tends to believe that MPR is suitable for anycast packets. Using MPR allows us to take advantage of the anycast semantics and hence improve network performance. A preliminary study has been reported on using MPR for anycast packets [7].

However, there are some problems associated with the MPR approach. For example, a router with MPR requires additional storage in order to maintain information of multiple paths it will use. The storage requirement can become very large. When the memory size of a router is limited, using MPR may actually increase packet delay at the router due to the time taken to swap in/out interfaces in the routing table in the process of establishing the routing trees.

## INTEGRATED ROUTING

To solve the dilemma, we propose and evaluate integrated routing algorithms for anycast packets. With our routing algorithms, a subset of (but not all) routers are designated to execute MPR to reduce time overhead and router cost. In the selection of routers to execute MPR, we explicitly take into account the distribution of sources

and recipient groups, and explore the connectivity properties between sources and receivers. As a result, our integrated routing algorithms perform much better than those proposed in previous studies in which either SPR [2] or MPR [7] is used at all routers.

## MODELS AND NOTATIONS

### THE NETWORK

The network we consider consists of a number of nodes. Nodes are connected by physical links along which packets can be transmitted. Each link has an attribute called distance. The distance of a link is usually measured by quantities such as delay and bandwidth. Figure 1 shows a sample network where the numerical values associated with links are their distances.

Routers in the network cooperatively decide a path for a packet and transmit the packet along the path.  $P(X, Y)$  denotes a path from  $X$  to  $Y$  where  $X$  and  $Y$  are nodes. We will use the terms route and path interchangeably. A shortest path from  $X$  to  $Y$  is the path with the smallest distance value, which is usually denoted  $P_{SP}(X, Y)$ . In the network shown in Fig. 1,  $P_{SP}(H_2, R_6) = \langle H_2, R_2, R_3, R_5, R_6 \rangle$ .

### ANYCAST PACKETS

Let  $A$  be an anycast address. We denote  $G(A)$  to be the group of designated recipient hosts. That is, a packet with anycast address  $A$  can be sent to any host in  $G(A)$ .  $G_S(A)$  is the group of source hosts (i.e., those that may send packets with anycast address  $A$ ). Note that the size of  $G_S(A)$ , denoted  $|G_S(A)|$ , can be very large.

### ROUTER ARCHITECTURE

A router typically has an equal number of input and output interfaces (i.e., ports), which in turn connect to input and output links, respectively. Each router has a number of routing tables. An entry in a routing table usually consists of fields for destination address, next hop, distance, and so on. See Fig. 2 for the architecture of a router.

For an incoming packet, a router locates an entry in a routing table such that the destination address of the packet matches that of the entry. Once the next hop is determined, the packet will be transported to a proper output link, which in turn connects to the next hop. It is possible that

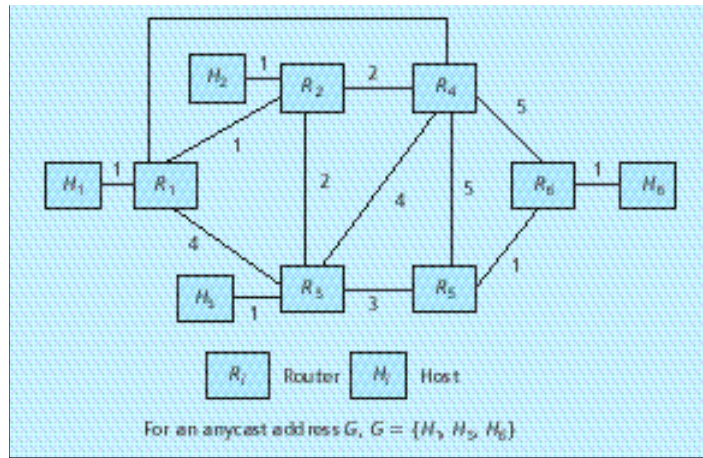


Figure 1. A sample network.

a packet may have to be queued at the output interface because the transmission of previous packet(s) is not yet finished.

Many new multigigabit routers have been studied and implemented [8]. The new router pushes the routing tables down into each forwarding engine. Since we know the memory at each forwarding engine is limited, we assume that there is sufficient secondary storage available to store the routing information for fast gigabit routers. Entries in the routing table may have to be swapped in and out when needed.

## INTEGRATED ROUTING ALGORITHMS

### AN OVERVIEW OF ALGORITHMS

Recall that two approaches can be taken for routing anycast packets: SPR and MPR. We take an integrated approach. For a given network, we propose to select a subset of routers to carry out MPR. The rest of the routers execute SPR. With proper selection of MPR routers, we can take advantage of both approaches and overcome their shortcomings. To achieve this objective, our algorithms have to address the following issues:

- Router assignment. We should properly determine which routers are to perform MPR and which SPR. The assignment must

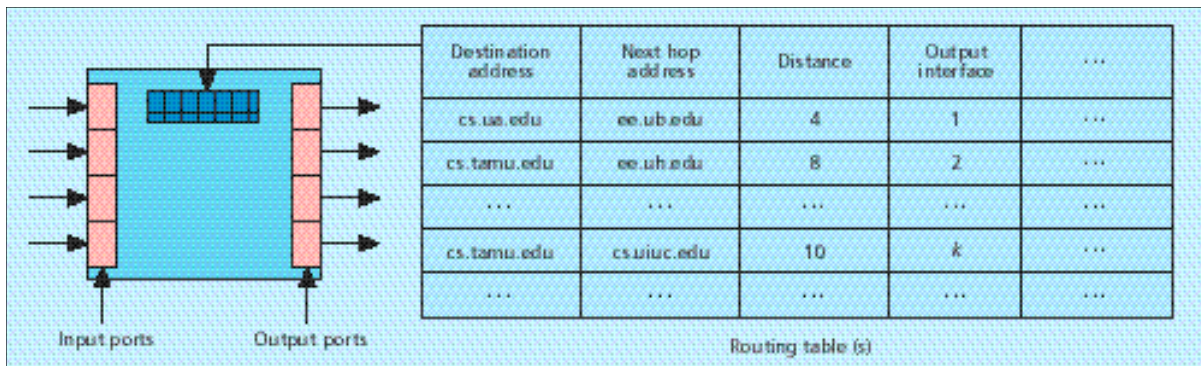


Figure 2. Router architecture.

Using anycast communication services may considerably simplify some applications. For example, anycast queries can be used to locate an appropriate server from a group of available ones. This will improve efficiency over multicast queries, as proposed in SLP.

take into account the population distribution of both the source and destination groups. The impact of topology and traffic should also be considered.

- Routing interface selection. For a routing table in an MPR router, there are multiple entries for a given anycast address. Accordingly, there may be multiple interfaces via which the destinations with the anycast address can be reached. When an anycast packet arrives, the router has to select one of the interfaces and then route the packet accordingly. In the selection we must ensure that anycast packets are effectively transmitted and distributed over different routes so that congestion is avoided.
- Loop prevention. A loop in a route implies wasting network resources and worsening packet delay. Our routing algorithm should produce loop-free routes for anycast messages. There are various approaches to dealing with loop elimination in a network. An easy way is to let all routers perform SPR; however, this method would not be able to distribute the traffic over multiple routes. This is not desirable for reducing average delay and increasing throughput. The second approach is to use a time-to-live (TTL) field for a packet to prevent looping, as done in SLP. The third approach is to use the shared tree approach in which the traffic is transmitted on a specific tree. Packets may split in different paths but will “follow” the tree to the destinations. Thus, there is no loop.

We will focus on the first two issues in this section. The integrated algorithm will apply the above standard techniques in dealing with loop prevention.

#### ROUTER ASSIGNMENT

The objective here is to assign a number (say,  $k$ ) of routers to perform MPR and let the rest execute SPR. We first introduce two simple baseline algorithms:

None\_MPR: This algorithm selects no router to perform MPR.

All\_MPR: This algorithm selects all the routers to perform MPR.

The above baseline algorithms are simple, but the delay performance may be not good. We propose two load-based assignment algorithms. The basic idea is to assign heavily loaded routers to perform MPR. To realize this idea, we will have to address two issues:

- How to measure the load of a router
- How to assign routers based on given load information

#### Static Load Based Assignment ( $L_S$ )

Static Load of a Router — For anycast address  $A$  and source  $S$ , there may be multiple shortest paths to  $G(A)$ ; one shortest path leads to one member in  $G(A)$ . Among these paths, let  $Q_{SP}(S, A)$  be the absolute shortest path from  $S$  to the members in  $G(A)$ . We define a status function for router  $R$  as follows: If  $Q_{SP}(S, A)$  passes through  $R$ ,  $F(R, S, A)$  is one; otherwise,  $F(R, S, A)$  is zero.

Now, the number of SP paths that pass

through router  $R$ , denoted  $L_d(R)$ , is defined as the summation of  $F(R, S, A)$  over  $S$ . That is,  $L_d(R)$  measures the total number of the absolute shortest paths that pass through router  $R$ . In the rest of this discussion, without loss of generality, we assume that  $n$  routers in the network are indexed in decreasing order of loads. That is,  $L_d(R_1) \geq L_d(R_2) \geq \dots \geq L_d(R_{n-1}) \geq L_d(R_n) \geq 0$ . We say a router is eligible if it has more than one interface that leads to the members in the recipient group. Obviously, we should not assign a router to perform MPR unless it is eligible. After we eliminate, say,  $m$  routers that are not eligible and re-index the routers, we have  $L_d(R_1) \geq L_d(R_2) \geq \dots \geq L_d(R_{n-m}) \geq 0$ .

We should consider the MPR routing algorithms for the remaining  $n - m$  routers. Which routers perform MPR depend on a heavy load threshold, as we discuss next.

Router Assignment Based on Static Load — We propose two router assignment algorithms that utilize static load information:

$Tk\_MPR$  assigns  $k$  ( $0 < k \leq n - m$ ) heaviest loaded routers (i.e.,  $R_1, R_2, \dots, R_k$ ) to execute MPR where  $k$  is a given parameter.

$TB\_MPR$  does the assignment based on a threshold  $b$ , that is, router  $R$  with  $L_d(R) > b$  is assigned to execute MPR.

The value of  $b$  can reflect the fact that router  $R$  may get congested at its output links when  $L_d(R) > b$ . To determine the value of  $b$ , the maximum transmission unit (MTU) of anycast and the path bandwidth are taken into account. The MTU is expressed in kilobits. For a router, let  $B_j$  be the bandwidth of the narrowest link on the absolute shortest path via interface  $j$  to  $G(A)$ . It can be expressed on an inverse scale, as the division of 10 million by the data rate of the link expressed in kilobits per second; this can be understood as the number of milliseconds necessary to transfer a 10 kb message. For example, for 10 Mb/s Ethernet, the transmission delay for a 10 kb message is about 1 ms; thus,  $B_j = 1$ .

For router  $R$ ,  $y = L_d(R)$  paths overlap at  $R$  and exit at the absolute shortest path through interface  $j$ . Congestion may occur at  $R$  when packets from  $y$  paths arrive at  $j$  simultaneously. In this situation, the delay could be as large as  $x = y * MTU * B_j / 10$  ms. Define  $m_d$  as the maximum delay allowance (in milliseconds) for anycast packets by the system (which can be set by applications or system administrators). Obviously,  $x > m_d$  violates the delay allowance. Thus, the threshold of heavy load can be defined as  $b_j = 10 * m_d / (MTU * B_j)$ . For example, let  $m_d$  be 1 ms for router  $R$  and  $B_j = 0.1$  (i.e., 100 Mb/s data rate link) and  $MTU = 10$  kb; then  $b_j = 10$  approximately.

#### Dynamic Load Based Router Assignment ( $L_D$ )

Apparently, static load approach  $L_S$  cannot reflect the dynamic load of traffic. We now propose another MPR router selection algorithm based on dynamic load on a router. The idea of  $L_D$  is simple. Initially, all routers use SPR for anycast traffic. Like unicast, the absolute shortest path is used. The routers will adaptively select alternative shortest paths

		Routing-interface-selection	
		$W_S$	$W_D$
Router assignment	$L_S$	$L_S W_S$	$L_S W_D$
	$L_D$	$L_D W_S$	$L_D W_D$

Table 1. Integrated routing algorithms.

for splitting the anycast traffic if overall traffic becomes heavy for the absolute shortest path.

**Dynamic Load of a Router** — Recall that in a routing table, an entry defines a mapping from an output interface to the destination. Without loss of generality, let us assume that there are  $K$  interfaces in a routing table for a given anycast address, and the interfaces are indexed by  $1, 2, \dots, K$ . Let  $\text{Dis}(i)$  ( $i = 1, 2, \dots, K$ ) be the distance which reflects the minimum cost (or number of hops) when interface  $i$  is used to route an anycast packet to the destination. Let the interface of the absolute shortest path be indexed as  $j$ ,  $Q_j$  the queue attached to the interface  $j$ , and  $|Q_j|$  the number of packets (measured in 10 kb) queued in  $Q_j$ . Then the delay  $T_j$  of the packet buffered in  $Q_j$  at interface  $j$  can be estimated as  $B_j * (|Q_j| + \text{Dis}(j))$ . Note that  $B_j$  and  $\text{Dis}(j)$  are static and can be calculated together with establishment of a routing table, while  $|Q_j|$  is dynamic.

**Router Assignment Based on Dynamic Load** — Here is our heuristic approach based on dynamic load to decide when a router should do SPR and when MPR. Consider router  $R$ , for its interface  $j$ :

Initially, a router (say  $R$ ) performs SPR. That is, all anycast traffic is routed through interface  $j$ . At runtime,  $R$  may switch between SPR and MPR, depending on dynamic load. Assume that at a time,  $R$  is performing SPR. Now, if traffic becomes heavy such that  $T_j = B_j * (|Q_j| + \text{Dis}(j))$  becomes larger than  $B_i * \text{Dis}(i)$  ( $i$  is the index of any one interface of  $R$ ), then  $R$  switches to MPR. When  $R$  is performing MPR, it is possible that the traffic will turn light such that for every interface ( $i$ ), except interface  $j$ ,  $B_j * (|Q_n| + \text{Dis}(j)) < B_i * \text{Dis}(i)$ ,  $R$  will switch back to SPR. Here  $\alpha$  is an adjustment parameter between 0 and 1.

#### ROUTING INTERFACE SELECTION

When a router is performing MPR, it must select one interface to transmit a packet from multiple available ones. We do so by weighted random selection. Each available interface is assigned a weight. The weight of an interface represents the probability that the interface is selected. To fully take advantage of MPR, we must carefully assign the weights to the interfaces.

Weight assignments for MPR have been studied in the literature. For example, an optimal weight assignment was proposed in [9] by which the average delay of messages can be minimized. Our idea is to have a simple parameterized weight assignment formula that can cover a wide range of heuristics.

we first consider a weight assignment method based on static information ( $W_S$ ). As assumed earlier, there are  $K$  interfaces in a routing table for a given anycast address. These interfaces are indexed by  $1, 2, \dots, K$ . Let their associated weights be denoted  $W_1, W_2, \dots, W_K$ , respectively. How should the weight of an interface 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  $i = 1, 2, \dots, K$ ,  $W_i \propto 1/D_i$ , where  $D_i = B_i * \text{Dis}(i)$ . We normalize  $W_i$  to be a probability such that  $\sum_{i=1}^K W_i = 1$ ; thus, for  $i = 1, 2, \dots, K$ , we have  $W_i = (1/D_i) / \sum_{j=1}^K (1/D_j)$ .

However, the formula can be generalized by raising the factor  $1/D_i$  to a power of  $r$  where  $r$  is a nonnegative real number. That is, we propose, for  $i = 1, 2, \dots, K$  such that  $W_i = (1/D_i)^r / \sum_{j=1}^K (1/D_j)^r$  where  $r$  is used to measure the degree of variance of distances in comparison with the average value. The detailed descriptions and analysis of this weight random selection method can be found in [7].

Now, let us extend the above method to the dynamic case ( $W_D$ ). We propose to use the simple heuristic method as proposed in  $W_S$ , but changing the definition of  $D_i$ . We define  $D_i = T_i = B_i * (|Q_i| + \text{Dis}(i))$  where  $T_i$  is the estimated delay of interface  $i$ , as discussed earlier.

#### INTEGRATED ROUTING ALGORITHMS

With the router assignment and interface selection approaches, this section proposes integration algorithms of  $L_S/L_D$  and  $W_S/W_D$ , respectively. Four integrated routing algorithms are generated, as shown in Table 1.

#### PERFORMANCE EVALUATION

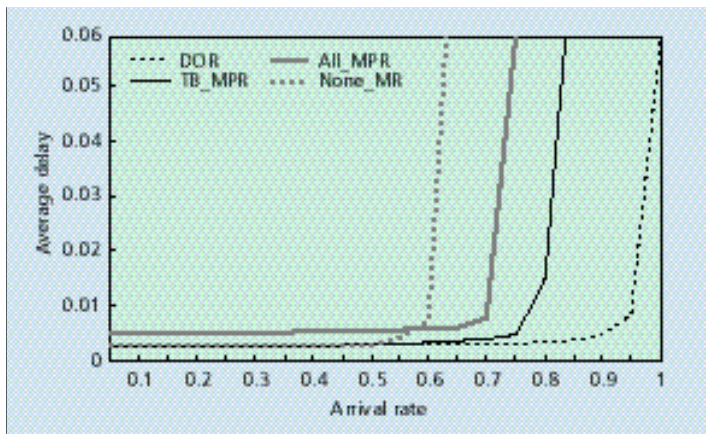
We measure the network performance in terms of average end-to-end delay. In order to obtain a lower bound on the average delay, we have also considered a dynamic optimal routing (DOR) algorithm. For each packet, this algorithm will select a path that results in the minimum end-to-end delay for the packet. The DOR algorithm achieves this by assuming that the complete dynamic system state information (e.g., queue length at each interface, current and future arrivals, etc.) is available without any delay or cost. Obviously, the performance of this algorithm is ideal but it is not realistic and cannot be implemented in practice.

#### SIMULATION MODEL

We simulated anycast in an Advanced Research Projects Agency (ARPA) network. During the simulation, 20,000 anycast packets are randomly generated as a Poisson process at arrival rate  $\lambda$ . Six routers are randomly selected as the anycast receipt group and another eight nodes as traffic sources.

The delay of a packet at a router is defined as the summation of the routing delay, the queuing delay, and the transmission delay. In addition, if the routing interface has been swapped out, there will be an interface reestablishment delay. The end-to-end delay of a packet is the sum of the delays at all the routers through which the packet passes. We say that a system

Weight assignments for MPR have been studied in the literature. For example, an optimal weight assignment has been proposed by which the average delay of messages can be minimized. Our idea is to have a simple parameterized weight assignment formula that can cover a wide range of heuristics.



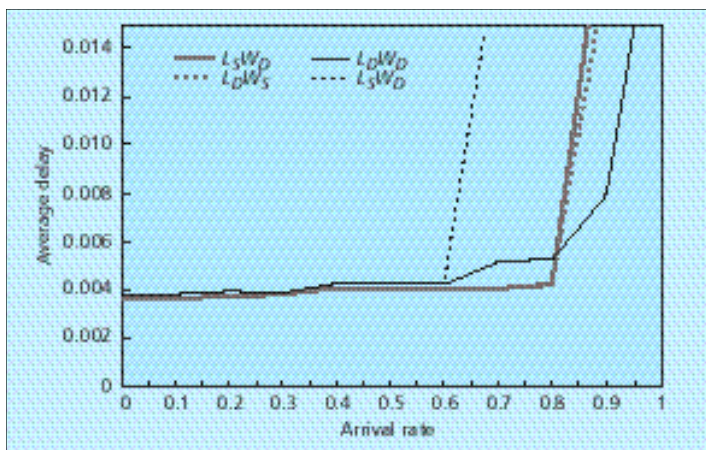
■ Figure 3. A comparison of basic algorithms.

saturates at  $\rho = \rho^*$  if any increase of  $\rho$  value beyond  $\rho^*$  will cause the system to become unstable. We call  $\rho^*$  the saturation rate.

#### PERFORMANCE COMPARISONS FOR BASIC ALGORITHMS

In this section, we compare the performance of DOR, All\_MPR, TB\_MPR, and None\_MPR. Both All\_MPR and TB\_MPR use the static weighted approach ( $W_S$ ) to select routing interface. The results are shown in Fig. 3. We have the following observations:

- As expected, DOR achieves the best performance. This is simply because it uses much more state information, which is not available to the others.
- When the system load is light, the delays of the None\_MPR algorithms are close to DOR. But when the system load increases, we observe that the None\_MPR algorithm performs poorer than the other three, due to the fact that it does not distribute traffic from a source into several different paths.
- Under medium and heavy load conditions, the TB\_MPR algorithm performs better than the All\_MPR algorithm. For the system with the



■ Figure 4. A comparison of integrated algorithms.

1B\_MPR algorithm, saturation occurs at approximately  $\rho^* = 0.8$ , but with the All\_MPR algorithm, saturation occurs at around  $\rho^* = 0.7$ . All these facts clearly indicate the effectiveness of our integrated algorithm.

#### PERFORMANCE COMPARISONS FOR INTEGRATED ALGORITHMS

In this section we compare the performance of the integrated algorithms listed in Table 1. From Fig. 4 we have the following observations:

- Algorithm  $L_S W_S$  takes no dynamic information for anycast traffic; routers get congested when traffic becomes heavy ( $\rho^* = 0.6$ ).
- Both algorithms  $L_S W_D$  and  $L_D W_S$  perform closely (saturation rate  $\rho^* = 0.8$ ).  $L_S W_D$  performs slightly better than  $L_D W_S$ . This indicates that  $L_S W_D$  takes dynamic weight assignment into consideration, which is better than fixed weight assignment even when the dynamic queuing length is considered.
- Algorithm  $L_D W_D$  fully takes advantage of dynamic load balance for anycast traffic and network availability. It performs well, especially in heavy load condition (saturation rate  $\rho^* = 0.89$ ), compared to  $L_S W_D$  and  $L_D W_S$  algorithms.

#### FUTURE EXTENSIONS

##### NETWORKS WITH FAULTS

Faults may permanently partition the network into several subnetworks. In this case, our algorithms will still work as long as there is at least one alive member of  $G(A)$  in a subnetwork. The sources can send anycast packets with anycast address  $A$  if the routing tables of the routers in the subnetwork have been modified accordingly [10]. Sometimes, a network may have a transient fault. Packets may be lost sporadically. To deal with transient faults, one may consider transmitting more than one copy of an anycast packet in the network.

##### DEPENDENT PACKETS

The semantics of anycast communication services implies that packets with an anycast address are relatively independent [2]. Sometimes, anycast packets from a source may be dependent in terms of their destination. To deal with this situation, we need to use the flow id in the packet header [1]. The router should register the flow id with the interface selected for the first packet in a flow for future usage in forwarding the subsequent packets in the same flow.

##### SYSTEMS WITH MULTIPLE GROUPS

With multiple anycast groups in the network, the interference of multigroup traffic will affect the traffic distribution, which cannot be overlooked. An approach is to develop an adaptive multipath routing algorithm that may distinguish traffic destined to individual groups. The algorithm must be "smart" in the sense that it should split the traffic before it overlaps with certain routers.

##### ANYCAST GROUP MANAGEMENT

According to the anycast message feature, the message is delivered to one (preferably the near-

est) member of the group. This feature allows membership management of an anycast group to be carried out with a great deal of flexibility.

## FINAL REMARKS

We have studied integrated routing algorithms for anycast messages. Our algorithm takes advantage of SPR, which is efficient and simple, and MPR, which can reduce potential congestion, and improve delay performance.

It should be pointed out that our algorithms are simple and practical, and can easily be implemented with current networking technology.

## REFERENCES

- [1] S. Deering and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification," RFC 2460, Dec. 1998.
- [2] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," RFC 1546, Nov. 1993.
- [3] E. Guttman et al., "Service Location Protocol," IETF Internet draft, draft-ietf-srvloc-protocol-v2-04.txt, Mar. 1998.
- [4] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC 1884, Dec. 1995.
- [5] S. Bhattacharjee et al., "Application-Layer Anycasting," Proc. IEEE INFOCOM '97, Apr. 1997.
- [6] E. Gustafsson and G. Karlsson, "A Literature Survey on Traffic Dispersion," IEEE Network, vol. 11, no. 2, Mar./Apr. 1997.
- [7] D. Xuan, W. Jia, and W. Zhao, "Routing Algorithms for Anycast Messages," Proc. ICPP '98, Aug. 1998.
- [8] C. Partridge et al., "A 50-Gb/s IP Router," IEEE/ACM Trans. Networking, vol. 6, no. 3, June 1998.
- [9] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE Trans. Commun., vol. COM-25, no. 1, Jan. 1977.
- [10] W. Jia et al., "An Efficient Fault-Tolerant Multicast Routing Protocol with Core-Based Tree Techniques," to appear, IEEE Trans. Parallel and Dist. Sys., Sept. 1999.

## BIOGRAPHIES

WEIJIA JIA (wjia@cs.cityu.edu.hk) received his B.Sc., M.Sc., and Ph.D. 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 of 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. He has published over 80 papers in journals, conferences, and books.

DONG XUAN (dxuan@cs.cityu.edu.hk) received his B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University (SJTU), China, in 1990 and 1993, respectively. He was on the Faculty of Electronic Engineering at SJTU from 1993 to 1997. In 1997 he worked as a research assistant and teaching at the City University of Hong Kong. He is presently a Ph.D. student in the Department of Computer Science, Texas A&M University. His main research interests are real-time computing and communication, computer network protocols, and distributed systems.

WEI ZHAO (zhao@cs.cityu.edu.hk) received his B.Sc. in physics from Shanxi Normal University, China, and his M.Sc. and Ph.D. 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. His current research interests include secured real-time computing and communication, distributed operating systems, database systems, and fault tolerant systems. He was an Editor of IEEE Transactions on Computers. He has served as guest editor for special issues on real-time operating systems for ACM Operating System Review and real-time middle ware 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 on Real-Time Systems. He was Program and General Chair of the IEEE Real-Time Technology and Applications Symposia in 1995 and 1996, respectively. He will be the Program and General Chairs of the IEEE Real-Time Systems Symposia in 1999 and 2000, respectively. In 2001 he will serve as general chair for the IEEE International Conference on Distributed Computing Systems. He has published over 130 papers in journals, conferences, and book chapters. He is an inventor of two U.S. patents. He and his students received the Best Paper Award at the IEEE International Conference on Distributed Computing Systems and at the IEEE National Aerospace and Electronics Conference.

Our algorithm takes advantage of SPR, which is efficient and simple, and MPR, which can reduce potential congestion, and improve delay performance.