Efficient Distributed Admission Control for Anycast Flows *

Weijia Jia, Wanqing Tu and Lidong Lin
Department of Computer Engineering and Information Technology
City University of Hong Kong, Kowloon, Hong Kong, P. R. China
E-mail: itjia@cityu.edu.hk

Abstract

Anycasting becomes an important research topic recently especially for the replicated servers where availability and fault-tolerance are critical for quality of service. With anycasting, applications can request the "nearest" server for provision of a desired services. In this paper, we study efficient Admission Control for Anycast (ACA) flows. We focus on algorithms that perform smart destination selection and efficient route establishment weighing at the probe message round trip time (RTT). Taking advantages of anycasting, our ACA algorithms differ from other approach in their dependence on system status information. Performance data obtained through simulation show that our ACA system performs efficiently in terms of admission delay, overhead and admission probability.

Key Words: Admission Control; Anycast Service and Flow; Destination Selection; Weight Assignment.

1 Introduction

Because more and more applications demand anycast services, in the latest version of IPv6, anycast has been defined as a standard service [1]. The problems pertaining to anycast can be divided in 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 [2], it was determined that anycast addresses are allocated from the unicast address space. An additional set of reserved anycast addresses within each subnet prefix has been also defined in [1]. Some subnet routers can be assigned the reserved anycast address that represent for all routers within a subnet prefix. A framework for scalable global IP anycast was proposed in [3]. Early work using anycast for locating the nearby replicated server has been proposed in [4]. In [5, 6], the implication of an anycasting service supported at the application layer was explored through server push and probe approach. Anycast has been partially defined as private network-network interface (PNNI) standardization [7]. In [8], QoS routing for anycast communications are discussed in the context of DiffServ networks in which server selection and resource reservation are attained in an aggregated fashion rather than being driven by individual client demand. We have developed some anycast routing protocols and studied their integration with other routing approaches in [9, 10].

In this paper, we study an efficient distributed Admission Control procedure for Anycast flows (ACA) with QoS requirement in the computer network. An anycast flow is a sequence of packets that can be sent to any one of the members in a group of designated recipients.An anycast flow will target at any server in a replicated server group. Once the connection is established, the flow becomes a unicast flow. Therefore, the anycast connection setup plays the critical role.

There are many applications that need communication service in the form of anycast flow, e.g., e-transaction, e-banking, down-loading, up-loading, etc. Using anycast communication services may considerably simplify these applications. Multiple mirrored (replicated) servers of the service providers, such as e-commerce companies, banks, and web-based information providers, can share a single anycast address. Applications may simply send their information flows with the anycast address in order to upload or download information from or to these multiple sites.

Different from datagram communication, *flow-oriented* communication has to go through an admission control process in which application makes a request, with certain QoS requirement, to the network for establishing a flow between a source and an anycast group. An anycast flow can be admitted (or we say, the flow can be established) only if sufficient network resources are available so that the QoS requirement can be satisfied. Clearly, the admission control plays a critical role in meeting QoS requirements of flows.

Admission control has been studied extensively for uni-

---

*The work is partially supported by Research grant council (RGC) Hong Kong, SAR China under grant Nos CityU 1055/00E (9040687) and CityU 1039/02E (9040596) and CityU strategic grant nos. 7001355 and 7001446.

cast flows [11, 12, 13]. Generally speaking, there are two categories of admission control mechanisms: *centralized* and *distributed* admission control.

- *Centralized Admission Control.* NetEx [14] adopts centralized admission mechanism to provide end-to-end delay guarantees in a LAN environment. This approach requires a centralized agency to perform admission control for the entire system. The main advantage of this scheme is its simplicity and easy implementation. But, it has vulnerability of single failure - if the centralized agency goes down, the whole system gets crashed. Another problem is the scalability. For a large network, the centralized agency could become a bottleneck.
- *Distributed Admission Control.* Distributed admission control mechanism can overcome the scalability problem raised in the centralized admission control. In this mechanism, the admission decisions are made by individual nodes, rather than by a centralized agency. In this way, we may achieve better scalability. The challenge here is to let nodes effectively and efficiently coordinate with each other in order to make correct admission decisions and hence achieve high admission probabilities. Tenet scheme II [15] uses distributed admission control mechanism to achieve scalability, but it needs a signaling protocol for multi-party communication to realize the required functionality. A related technology to distributed admission control is the well-known RSVP Protocol [16] that has been proposed for signaling and resource reservation in the Integrated Services architecture.

We adopt the mechanism of distributed admission control in this paper [19]. A critical step in admitting an anycast flow is to select a destination among the members of an anycast (replicated) server group. A destination should be selected so that (soft) QoS requirement of the requesting flow can be satisfied with high possibility while not producing congestion in the network. We propose and analyze several algorithms that use different types and amount of status information in making destination selection. The ACA system is performed by the admission control (AC) routers (such as edge or border routers), thus, not all the network routers need to follow up the admission control mechanism. The contribution of the paper is two-fold (1) AC-routers use the local information to decide the admission of the flows. We have adapted a simple control mechanism by using novel weight assignments to the selection of the destinations to enhance the admission scalability with simple probe round trip time (RTT); (2) Based on the destination selection, we employ the adaptive retrial and parallel connection mechanisms that enable the system to increase the admission probability and to reduce the admission delay. We evaluate the proposed distributed admission control mechanism by extensive simulations. Performance data show that in terms of admission probabilities and efficiency, ACA systems that are based on local status information can perform closely to those that utilize global and dynamic status information such as available bandwidth, etc. We note that the latter is much more expensive to realize.

2 Admission Control for Anycast Flow

In this section, we will discuss the design of distributed Admission Control mechanism for Anycast (ACA) flows. Several weight assignment algorithms will be studied for destination selection, and we will discuss the issues of efficient resource reservation and route connection control in the next section.

2.1 The ACA algorithm

In ACA, admission decisions for anycast flow establishment requests are made by some routers at different locations. For the sake of simplicity, in the remaining of the paper, we call the routers who make admission decisions as Admission-Control routers (AC-router in short).

Once a new anycst flow establishment request arrives, the following procedure is invoked at the correspondent AC-router for the purpose of admission control as shown in Fig. 1. The shown procedure consists of three main steps: destination selection, resource reservation, and retrial control.

```
1. REPEAT
        1.1. Select a destination in the anycast group for the
                 requesting flow;
        1.2. Reserve resource for the requesting flow to the
                 selected destination;
        1.3. If resource reservation is SUCCESSFUL, then
                 the flow is admitted and return;
        1.4. Keep_going = retrial_control();
   UNTIL not Keep_going;
2. The flow is rejected and return.
```

Figure 1. The Distributed ACA

The step of destination selection will determine which destination the packets of the requesting flow should be sent to. A good selection will bring a better chance for the flow to be admitted. Once a selection is made, resource reservation will take place to check if there is sufficient bandwidth available on each link of the path leading to the selected destination. If yes, resource is reserved and the flow is admitted. Otherwise, a retrial control scheme will be invoked to determine whether an alternative destination should be tried. The flow is rejected unless the retrial control decides to stop the procedure.

While this procedure is simple, there are two challenge issues: (1) how to properly make destination selection, given the limited local information which an AC-router has; and (2) how to efficiently perform resource reservation for path establishment. In this section, we will elaborate our ideas for addressing the first issue and the second issue will be discussed in the next section.

2.2 Destination Selection

Destination selection is a critical and special problem in admission control for *anycast* flows. The uniqueness of destination(s) in unicast and multicast eliminates this problem in their admission controls. The semantics of anycast defines that an anycast flow can be sent to any member in an anycast group. Nevertheless, we explore how to take advantage of this feature to improve admission probability.

As mentioned before, the communication overhead prohibits AC-routers from obtaining complete global system status information. Thus, a difficult task is how to select an appropriate destination at absence of global system information.

In this paper, we adopt a simple approach for destination selection. Specifically, for an anycast group, an AC-router keeps a list of weights, each corresponds to a destination in the anycast group. Without loss of generality, let us assume that there are K members in an anycast group $G(A)$. Their weights are denoted as $W_1, W_2, ..., W_K$, respectively. The weight of a destination represents the probability that the destination is to be selected. Thus, a member with higher weight value will have higher probability to be selected than those with lower weight values. The assignment of weights will be discussed in the next two sub-subsections. Nevertheless, any assignment is subject to the following constraint:

$$\sum_{i=1}^{K} W_i = 1. \tag{1}$$

2.2.1 Weight Assignment Algorithms

Our previous work [19] introduces Weight Assignment based on Route Distance, Local Admission History, and Available Bandwidth information (called ED, WD/D+H and WD/D+B respectively). They are briefly described as follows:

1. ED: A simple destination selection approach based on route distance is proposed. A flow with short route distance will consume less resources. Hence a better destination selection algorithm should prefer destinations with short route distances. The weight associated with a destination should be inversely proportional to the distances of routes. The weights should be properly normalized, i.e. for $i = 1, 2, ..., K$, $W_i = \frac{1/D_i}{\sum_{j=1}^{K} 1/D_j}$.

2. WD/D+H: The local admission history is a log that records the successfulness of selecting individual destinations in admission control. Formally, at an AC-router, this information is represented as a list: $H =< h_1, h_2, ..., h_K >$ where K is the number of members in $G(A)$; h_i corresponds to the admission history of destination i. The value of h_i is used to adjust the weight of W_i.

3. WD/D+B: Assume that the bandwidth usage of links on routes to all destinations are available. Let r be the route from the source to destination i. Route Bandwidth B_i is defined as:

$$B_i = \min_{l \in r}(AB_l), \tag{2}$$

where AB_l is the available bandwidth of link l. Hence B_i represents the minimum available bandwidth of links along route r to destination i. Intuitively, the destination, whose route has more route bandwidth, should have higher possibility to be selected.

4. Our solution: *Weight Assignment based on Round Trip Time of Probe Messages.* We propose to use the following information, which can be collected by AC-routers.Here is our another contribution using Round Trip-Time (RTT) of probe messages to assign the weights for the destinations. We design the algorithms in this way so that we can investigate how different status information impacts differently on the network performance in terms of admission probability, overhead, and compatibility.

Probe packet mechanism has been studied by many well-known systems [6, 17]. In these systems, either the probes are used to fetch server load information ([6]) or used to record the paths/nodes' availability that the probes traversed on a hop by hop basis ([18]). In this section, we propose a destination selection algorithm based on Round Trip Time (RTT) of light weight probe messages for collecting path and destination's dynamic loading information implicitly. By lightweight probe message we mean that a probe message is a dummy packet and only its RTT information is used just like a ping message. Although acquiring this information requires some effort of AC-routers and causes some load to the network, as we will show later that this approach is indeed cost-effective. There are two ways of using probe RTT: *periodically (proactive) probe* and *on-demand probe*. In the former approach, the AC-routers, by some interval I and probe the members in $G(A)$ and in the later approach, an AC-router, upon reception of a request, sends the probe to $G(A)$ as will be detailed later.

As mentioned earlier, a smart destination selection algorithm should not only prefer the destinations with short route distances but also that with quick response. Here we propose an efficient destination selection algorithm using lightweight probe messages (probes for short) to attain the (implicit) path information (such as delay and loading etc.) based on their RTT time. Using probe RTT as the weight assignment for deciding the destinations is very useful as RTT will bring the following up-to-date (heuristic) information: (1) Delay from the destination probed to the AC-router; (2) Availability of the probe path to the destination; (3)Availability of the destination (ie., the server load).

With the information, as we will demonstrate in the simulation section that the RTT approach indeed selects a better destination for the service. RTT destination selection approach can be roughly described as follows: an AC-router (periodically or on demand) sends the probes (ping) to the members in $G(A)$. The members, upon reception the probe message should send an ack message to the AC-router. A member may choose to ignore the probe message if it is busy. Once the acknowledgement of a member is received by the AC-router, the RTT time of the probe message is used to assign the weight for the member as shown in formula (5) below. If no acknowledgement is received from a specific destination i, then the timeout interval can be used as the particular RTT instead. The following notations will be used by the algorithm for an AC-router:

(1) I: The timeout interval for RTT, an ack message to probe p received after I from member i is considered lost.

(2) RTT_i: The record stores the round trip time value of current probe for destination i.

(3) $SRTT_i$: exponential average of RTT. Considering dynamic estimate of variability in estimating RTT_i, $SRTT_i$, initialized to I, is averaged as:

$$SRTT_i(n) = \beta SRTT_i(n-1) + (1-\beta)RTT_i(n). \quad (3)$$

where $RTT_i(n)$ is the current measurement of RTT and $0 < \beta < 1$. We would like to give the greater weight to more recent instance RTT. Thus the small value of β is assigned, the greater the weight given to the more recent observation of RTT. In our experiments, we use $\beta = 0.1$. Based on the above consideration, the weight associated with a destination should be inversely proportional to $SRTT_i$. That is, for $i = 1, 2, ..., K$,

$$W_i \sim 1/SRTT_i(n), \quad (4)$$

where $SRTT_i(n)$ is the n^{th} round trip time received from destination i at the n^{th} probe. Thus the normalized weights are defined as (for $i = 1, 2, ..., K$):

$$W_i = \frac{1/SRTT_i(n)}{\sum_{j=1}^{K} 1/SRTT_j(n)}. \quad (5)$$

Recall that $SRTT_i$ is the log that records the smoothing round trip time from the selecting individual destinations. At an AC-router, the information is represented as a list:

$$SRTT =< SRTT_1, SRTT_2, ..., SRTT_K > \quad (6)$$

where K is the number of members in $G(A)$.

There are two ways of sending the multicast probe messages: *periodical (proactive)* and *on-demand*:

1. Proactive probe: Each AC-router multicasts (or point-to-point send) a probe (like a ping) message to the destinations in group $G(A)$ within a predefined interval I. With the information in list $SRTT$, an AC-router may update the weights every time interval I in accordance with (5). Thus the algorithm is called as Weighted Distribution-proactive with RTT (WD/P+RTT in short).

2. On-demand probe: An AC-router, upon reception of a request, multicasts the probe to $G(A)$ and expects the ack from the destinations. The RTT of the probe is used by the AC-riouter to assign the weights similar to above case and the algorithm is called as Weighted Distribution-On-demand with RTT (WD/O+RTT in short).

The advantage of WD/P+RTT scheme is that once a request arrives at an AC-router, the AC-router can immediately select the best destination using the most recent weighting. In this way, the admission time is short. However, the drawback of this approach is that if every AC-router periodically multicats the probe messages to $G(A)$ (say in every 500 ms), then this approach will introduce many (ping) messages to the network. But the WD/O+RTT scheme is just in opposite: a probe message is sent only upon the arrival a request. This approach introduce much less load to the network, however, the admission delay may be longer as compared with the former. In this paper, we adopt the on-demand approach.

In brief, we have introduced four destination selection algorithms. They differ from each other in the sense of using different degree of information: the ED algorithm uses no system status information except the number of members in an anycast group; WD/D+H algorithm uses the route distances and local flow admission history information; WD/D+B depends on the routes distances and bandwidth information while WD/O+RTT only employs simple probe message's RTT to reflect the dynamic network/member loading instantaneous information. It can be seen that WD/D+B approach requires other protocol(s) (such as RSVP) to provide the bandwidth information. The overhead is apparently much higher than a RTT of probe/ack message. The probe/ack message can be easily transformed into TTL field, thus the WD/O+RTT approach can be easily implemented on top of TCP/IP.

Although the probe/ack approach itself is not new, however, it is novel to incorporate its RTT into the weight assignment for destination selection. Assume that the number of hops and initial bandwidth between the sender and receivers are known initially, thus the minimum round trip time is known also. Thus probe RTT may reflect (implicitly) the dynamic available bandwidth on the link/path, delay and the loading situation for the destinations. We expect that WD/O+RTT may perform very close to WD/D+B in terms of QoS but the overhead is much less in terms of compatibility and scalability for current Internet deployment.

3 Efficient Route Connection Algorithms

Recall that in our distributed admission procedure, once a destination is determined, the next step is to reserve resource along the route from the source (represented by the AC-router) to the selected destination for the requesting flow. The connection must be setup for the flow transmission. As we mentioned before, the destination selected cannot guarantee that the route selected can be successfully setup eventually. In this section, we discuss efficient QoS route set-up algorithm that reduce the admission delay and try to maximize the admission probability.

3.1 Resource Reservation

We assumed that the routes from the source to destinations in an anycast group are predefined (such as the shortest path). Hence, the resource reservation involves only two simple tasks:

- Task 1: check if there is sufficient bandwidth available on all the links of the route. If there is not, FAILURE will be returned by the ACA procedure. Otherwise,
- Task 2: reserve bandwidth on the each of links along the route, and report SUCCESS by the ACA procedure.

The resource reservation can be made by the standard RSVP protocol [16]. That is, to check the availability of link bandwidth along the route, we can use some polling and reserving messages, such as the *RESV* and *PATH* messages available with RSVP.

3.2 Retrial Control

Once resource reservation returns SUCCESS, the route with enough bandwidth is setup, the newly requesting flow will be admitted, it will be offered to the route which leads to the selected destination.

However, if the route cannot be established, a decision must be made if an alternative destination need to be tried. There is a trade-off here. On one hand, to improve admission probability, more destinations should be tried. On the other hand, more destinations tried, more overhead (e.g., the communication overhead due to resource reservation messages) will be introduced.

We use a simple counter-based re-trial control scheme. Initially, a counter c is set to $zero$. Each time when a destination is tried, c is increased by one. The re-trial control returns true if c is less than a pre-defined parameter R. R is the total number of re-trials allowed. In the simulation, we have found that a few re-trials can significantly improve the admission probability for the flows. Admission process through retrial control may increase the admission probability for the flows, however, the retrial process is executed in sequential that also increases the admission delay and reduce the system response time. As we will demonstrate in next subsection that this delay can be shorten through parallel multi-path connections to different destinations.

4 Experimental Evaluation

In this section, we evaluate the performance of the systems that use our distributed admission control procedures with different destination selection algorithms.

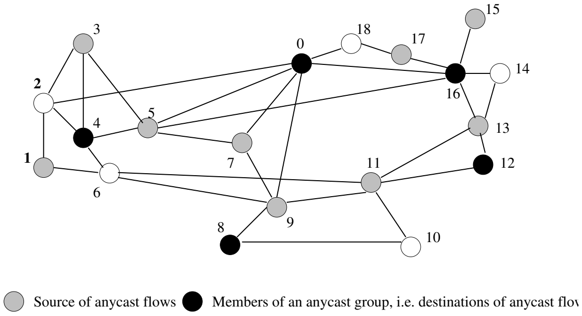


Figure 2. The Experimental Network

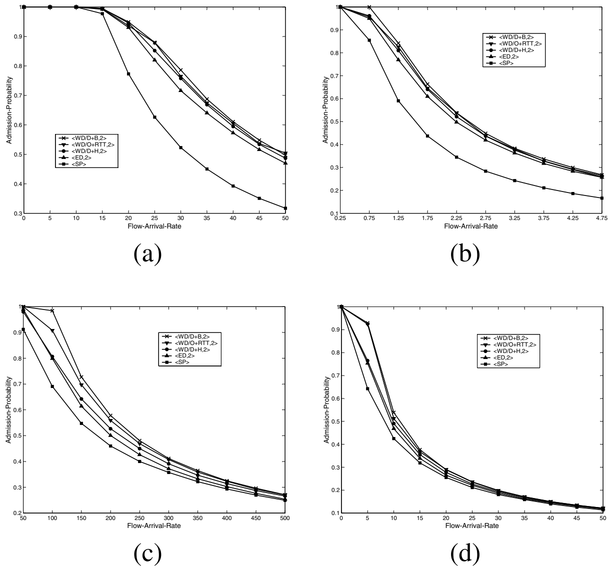
4.1 Experimental Model

Network: Two types of networks are considered in our experiments. The first one is MCI ISP backbone (called MCI) network. Figure 2 shows the topology of the network. There are 19 nodes that are interconnected by links. The second type of networks randomly generated from 1000 random nodes with a general-purpose random topology generator BRITE [20]. BRITE is specially designed to generate Internet-link network topologies that exhibit the power-law for the distribution of node degrees [21]. That is, the number of nodes with a given degree d is proportional to d^{μ}, where μ is a constant. Such a network can be generated in two steps. First, n nodes are randomly placed in a rectangular area (this process is called node placement). Then, a bi-directional link is added according to $RouterWaxman$ model; that is, nodes are incrementally added to the network. For each pair of nodes, a distance is chosen in $(0, L)$ from a uniform random distribution. Edges are introduced between pairs of nodes u, v with a probability that depends on the distance between them. The edge probability is given by

$$P(\{u,v\}) = \delta exp\frac{-d(u,v)}{L\gamma}$$

where $d(u,v)$ is the distance from node u to v , L is the maximum distance between the two nodes; γ and δ are parameters in the range $(0,1]$. Larger values of δ result in graphs with higher edge densities, while small values of γ increase the density of short edges relative to longer ones. In our experiment, we set $\delta = 0.15$ and $\gamma = 0.2$.

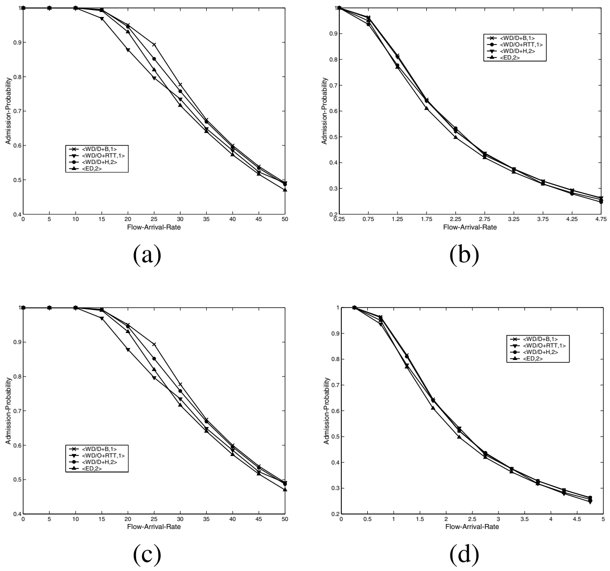
In MCI network, link bandwidth capacity is set to be 100 Mbps. In the second type of (random) networks, the link bandwidth capacity is assigned a value chosen in the range of [50Mbps,100Mbps] on a uniform random distribution. In both cases, 20 percent of link bandwidth is reserved for anycast flows. Every node is a router and has one host attached. All networks in our simulations are assumed as the *guaranteed-rate scheduling* networks, i.e., once a connection is set up, certain amount of bandwidth are reserved for the flow and the end-to-end delay requirement of the flow transmission can be guaranteed. Thus we do not model the queuing behavior for the routers within the networks. Our simulator is Mesquite CSIM C [22], a process-oriented, general purpose simulation toolkit. All the simulations run in 10 SUN Ultra-SPARC workstations.



(a) (b)

(c) (d)

Figure 3. AP of Different Systems under Retrial=2. (a) and (b) are with MCI network under Rate=64Kbps and 1.2Mbps accordingly. (c) and (d) are with 1000 Nodes Random-Generated Top including Five Groups under Rate=64Kbps and 1.2Mbps accordingly.



(a) (b)

(c) (d)

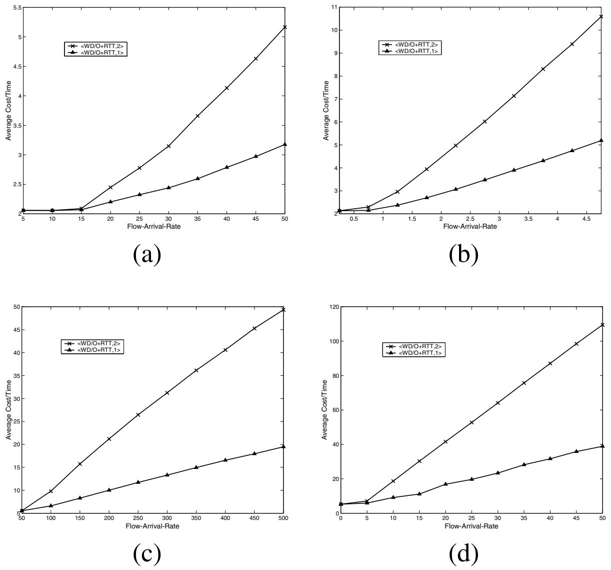
Figure 4. AP of different systems. (a) and (b) are with MCI network under Rate=64Kbps and 1.2Mbps accordingly. (c) and (d) are with 1000 Nodes Random-Generated Top including Five Groups under Rate=64Kbps and 1.2Mbps accordingly.

Traffic Model: We assume that requests for anycast flow establishment form a Poisson process with rate λ, while flow lifetimes are exponentially distributed with average of 180 seconds. Two flows with the bandwidth requirement of 64kbps and 1.2Mbps, representing audio and standard MPEG-1 video streams, are used in the experiments. Sources of anycast flows are chosen randomly among those hosts that attach the routers with the odd identification numbers. In the small network, there is an anycast group that consists of 5 members. They are those hosts which attach to router $0, 4, 8, 12$, and 16 , respectively. For the large random networks (1000 nodes), we carried out experiments with 10 random topologies with the deployment: The flow requestors are chosen randomly among those nodes with the odd identification numbers and send the flow requests to five anycast groups, each group contains five members, that are randomly selected among the nodes with even ids.

Evaluated Systems: Recall that a 2-tuple (A, R) is used to represent the systems which run our proposed ACA with different destination selection algorithms and retrial control schemes. In particular, A represents the destination selection algorithm. That is,

$$A \in \{SP, ED, WD/D{+}H, WD/D{+}B, WD/O{+}RTT\}. \quad (7)$$

R indicates the maximum number of retrials that are allowed. For example,(ED, 2) represents a system in which



(a) (b)

(c) (d)

Figure 5. Cost of System (WD/O+RTT,*). (a) and (b) are with MCI network under Rate=64Kbps and 1.2Mbps accordingly. (c) and (d) are with 1000 Nodes Random-Generated Top including Five Groups under Rate=64Kbps and 1.2Mbps accordingly.

Even Distribution (ED) algorithm is used, and the maximum number of retrial routes is 2. In other words, up to 2 destinations can be tried in for an anycast flow request. To simplify our discussion, we use symbol "*" to denote the value of R if the discussion covers all the values of R.

Baseline Systems: For the comparison purpose, a baseline system is used which is an admission control procedure that will always pick the destination which has the shortest distance from the source router for each incoming flow. In this system, anycast traffic is not distributed. We expect a system running our DAC procedure will out-perform this system. We call this system as Shortest-Path (SP) system.

Performance Metrics: First, we are interested in Admission Probability (AP). It is defined as the probability that an anycast flow is admitted in a given system. The higher the AP value, the better the performance.

The second metric we use is the average number of the hops visited for setting-up a path. Recall that in our systems, we allow up to R retries. Both R and k are directly proportional to the overhead in terms of resource reservation messages (network cost) and admission delay (user waiting time). Thus, we would like to have the overhead relatively small.

Denote N as the total generated flow number and M as the total successful established flow number during the simulation time,for the system (A,R), denote $n_{i,j}\,(i = 1,...,N, j = 1,...,R)$ as the number of hops visited for the i^{th} flow to establish a path when on j^{th} retrial. We define the Network $Cost$ and User Waiting $Time$ as follows:

$$Cost = Time = \frac{\sum_{i=1}^{N}\sum_{j=1}^{R} n_{i,j}}{M}. \tag{8}$$

4.2 Performance Results and Observations

In this subsection, we report performance results and make observations. We only present a limited number of cases here. However, we find that the conclusions we draw here generally hold for many other cases we have evaluated.

4.2.1 Sensitivity of Admission Probability

Comparisons of AP of the systems of re-trial=2 scheme in different networks: Figures 3 and 4 show the admission probabilities of systems SP, (ED, 2), (WD/D+H, 2), (WD/D+B, 2) and (WD/O+RTT) in different networks (MCI network and 1000 nodes random generated networks). From the figures, we can make the following observations:

Four systems (namely, (ED, 2), (WD/D+H,2), (WD/O+RTT,2), and (WD/D+B, 2)) outperform (SP). We also noticed that in terms of AP, (WD/D+H,2) outperforms (ED, 2), (WD/O+RTT,2) outperform (WD/D+H,2) and so does (WD/D+B, 2). This observation implies that our randomized destination selection can work very well, especially with available network information. (WD/D+B, 2) is the best among the five due to its access to the route bandwidth information. However, as we mentioned early, obtaining such information may cause compatibility problem. (WD/O+RTT,2) performs very close to that of (WD/D+B,2) and is compatible with current network setting. Hence, in practice, one may prefer (WD/O+RTT).

From the above discussion, we can see that system (WD/O+RTT) has better performance than other systems. An interesting observation is that systems (WD/O+RTT,P), (WD/O+RTT,2) do not improve (WD/O+RTT,1) very much and this indicated that RTT may bring sufficient information for an AC-router to make the selection. As shown in Figure 4, we can observe that (WD/O+RTT,1) is very close to (WD,2), (WD/D+H,2) and (WD/D+B,1) especially when the network is heavily loaded.

4.2.2 Comparison of Cost and Time

Figure 5 shows the sensitivity of Cost/Time of system (WD/O+RTT) under retail path setup in various networks. As we mentioned before, the performance comparisons use number of hops as the measurement for two reasons: in our simulation model, since all flows, once they are admitted, they will have the guaranteed end-to-end delay. Therefore, we count the average connection overhead (the cost and admission delay) as the average number of hops the connection setup procedure traversed until the connection is success. From the figure, it is observed that among all the systems, (WD/O+RTT, 1), with the better performances of AP but overall cost of (WD/O+RTT,1) is the minimal.

5 Conclusion

We have studied distributed Admission Control (ACA) procedure for Anycast flows. In our ACA procedure, we focus on algorithms that perform end-to-end destination selection for anyast flows based on RTT delay. We designed a novel destination selection WD/O+RTT algorithm based on probe message's RTT. As compared with other three algorithms, i.e., ED, WD/D+H, and WD/D+B, our algorithm achieves the best balance in term of cost and performance. It attains a very close admission probability as that of WD/D+B but does not rely on the support of other protocols to acquire bandwidth information. Compared with WD/D+H algorithm, the cost of WD/O+RTT is slightly higher due to the probe message. However, it achieves better AP than WD/D+H. Thus our destination selection ACA based on (WD/O+RTT) algorithm can indeed perform close to those that utilize global and dynamic status information, which is much more expensive to realize and difficult to deploy with the current best-effort Internet routing service. However, the QoS parameter we considered in this paper is limited to the available bandwidth. Future work includes the analysis to the statistical model of QoS parameters based on the information brought by RTT, particularly with the end-to-end delay bound and delay jitter derivations.

References

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

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

[3] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," In *Proc. of SIGCOMM 2000,* pp. 3-15.

[4] J. D. Guyton , M. F. Schwartz,"Locating nearby copies of replicated Internet servers," *Proc. SIGCOMM'95*, October 1995, pp.288-298.

[5] S. Bhattacharjee, M. H. Ammar, E. W. Zegura, V. Shah and Z. Fei, "Application-Layer Anycasting," In *Proc. of IEEE Infocom 1997,* pp. 1388-1396.

[6] E. W. Zegura, M. H. Ammar, Z. Fei, S. Bhattacharjee, Application-layer anycasting, *IEEE/ACM Transactions on Networking (TON)*, August 2000, Volume 8 Issue 4.

[7] ATM Forum, "Private Network-Network Interface Specification v.1.0", *Tech. rep. af-pnni-0055.000, ATM Forum*, Mar. 1996.

[8] F. Hao, E. W. Zegura , M. H. Ammar, "QoS routing for anycast communications: motivation and an architecture for DiffServ networks", *IEEE Communications Magazine*, June 2002, pp.48-56.

[9] D. Xuan, W. Jia, W. Zhao, and H. Zhu, " A Routing Protocol for Anycast Messages," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 6, June 2000. pp. 571-587.

[10] W. Jia, D. Xuan and W. Zhao, "Integrated Routing Algorithms for Anycast Messages", *IEEE Communication*, Janu. 2000.

[11] A. Dailianas and A. Bovopoulis, "Real-time Admission Control Algorithms with Delay and Loss Guarantees in ATM Networks," In *Proc. of Infocom 1994,* pp. 1065-1072.

[12] V. Firoiu, J. Kurose, and D. Towsley, "Efficient Admission Control for EDF Schedulers," In *Proc. of Infocom 1997,* pp. 310-317.

[13] J. Liebeherr, D.E. Wrege, and D. Ferrari, "Exact admission control in networks with bounded delay services," *IEEE/ACM Transactions on Networking*, 1996.

[14] A. Sahoo, B. Devalla, Y. Guan, R. Bettati, and W. Zhao, "Adaptive connection management for mission critical applications over ATM Networks," *International Journal of Parallel and Distributed Systems and Networks*, Special Issue On Network Architectures for End-to-end Quality-of-Service Support.

[15] R. Bettati, D. Ferrari, A. Gupta, W. Heffner, W. Howe, M. Moran, Q. Nguyen, and R. Yavatkar, "Connection Establishment for Multiparty Real-time Communication," In *Proc. of NOSSDAV 1995,* pp. 240-250.

[16] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP: A New Resource Reservation Protocol," *IEEE Networks Magazine*, vol. 31, No. 9, pp. 8-18, Sept. 1993.

[17] L. Breslau, E. Knightly, S. Schenker, I. Stoica and H. Zhang, "Endpoint Admission Control: Architecture Issues and Performance", In *Proc. of SIGCOMM 2000,* pp. 57-69.

[18] S. Chen and K. Nahrstedt, "Distributed Quality of Service Routing in High-Speed Networks Based on Selective Probing", In *Proc. of LCN 1998*.

[19] D. Xuan and W. Jia, "Distributed Admission Control for Anycast Flows with QoS Requirements," *The 21st IEEE International Conference on Distributed Computing Systems*, 2001, pp.292-300.

[20] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Boston University Representative Internet Topology Generator," http://cs-pub.bu.edu/brite/index.html, Mar. 2001.

[21] M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On power-law relationships of the internet topology. In *SIGCOMM* 1999, pp.251-262.

[22] Mesquite Software Inc. http://www.mesquite.com/products/csim19.htm