Distributed Admission Control for Anycast Flows with QoS Requirements

Dong Xuan
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
dxuan@cs.tamu.edu

Weijia Jia
Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
wjia@cs.cityu.edu.hk

Abstract

An anycast flow is a sequence of packets that can be sent to any one of the members in a group of designated recipients. Using anycast services can significantly simplify some applications. Little work has been done on providing QoS support to anycast flows. In this paper, we study Distributed Admission Control (DAC) procedure for anycast flows with QoS requirements. We focus on algorithms that perform destination selection, which is critical in anycast. Several algorithms are proposed. These algorithms differ from each other in their dependence on system status information. We also address the issue of resource reservation and re-trial control in the DAC procedure. We evaluate the proposed mechanisms by mathematical analysis and computer simulation. Performance data show that in terms of admission probabilities, DAC systems that are based on local status information can perform closely to those that utilize global and dynamic status information. We note that the latter is much more expensive and difficult to realize.

keywords Anycast Service, Admission Control, Destination Selection, and Resource Reservation

Contents

1 Introduction

In this paper, we study Distributed Admission Control (DAC) procedure for anycast flows 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. Traditional unicast flow is a special case of anycast flow in the sense that for the unicast flow, the recipient group size is one.

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 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 required QoS can be satisfied. Clearly, the admission control plays a critical role in meeting QoS requirement of flows.

Generally speaking, there are two categories of admission control mechanisms: *centralized* and *distributed* admission control.

- *Centralized Admission Control.* 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. Thus, while this mechanism may be sufficient for small networks, it does not scale adequately to be a solution for large systems.

- *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.

We adopt the mechanism of distributed admission control in this paper. A critical step in admitting an anycast flow is to select a destination among the members of an anycast group. A destination should be selected so that the QoS requirement of the requesting flow can be satisfied while not producing congestion in the network. We propose and analyze several algorithms that use different type and amount of status information in making destination selection. We evaluate the proposed distributed admission control mechanism by mathematical analysis and computer simulation. Performance data show that in terms of admission probabilities, DAC systems that are based on local status information can perform closely to those that utilize global and dynamic status information. We note that the latter is much more expensive and difficult to realize.

The rest of the paper is organized as follows. Previous work related to our study is discussed in Section 2. In Section 3, we introduce network and flow models used in this paper. Distributed admission control procedure is studied in Section 4. In Section 5, we evaluate the system performance in terms of admission probability and runtime overhead. We summarize the paper and discuss the future work in Section 6.

2 Previous Work

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 with any of the defined unicast address format. Recently, an additional set of reserved anycast addresses within each subnet prefix has been defined in [1]. Some subnet routers can be assigned the reserved anycast address that represent for all routers within a subnet prefix. In [3], the implication of an anycasting service supported at the application layer was explored. A framework for scalable global IP anycast was proposed in [4]. We have developed an anycast routing protocol and studied its integration with other routing approaches [5, 6]. While these previous studies were innovative and aimed at improving performance in terms of end-to-end delay or bandwidth consumption, there has been no report on providing QoS guarantees to anycast flows.

Admission control is a mean to provide QoS guarantees and has been studied extensively [7, 8, 9, 10, 11]. Generally speaking, there are two classes of admission control mechanisms: centralized admission control and distributed admission control. For instance, NetEx [10] adopts centralized admission mechanism to provide end-to-end delay guarantees in a LAN environment. The centralized admission mechanism is easy to implement. Tenet scheme II [11] 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 [12] that has been proposed for signaling and resource reservation in the Integrated Services architecture. RSVP has been widely adopted in distributed admission control system. All these previous studies focus on QoS support to unicast or multicast communications. No study has been reported on admission control to support anycast flows with QoS requirements.

3 Models

In this section, we describe network and flow models and define terminology that will be used in the following sections.

Network. 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. Each link has attribute called link *capacity* or *bandwidth*. Link capacity is consumed by anycast active flows. *Remaining Capacity* or *Available Bandwidth* of link l, (AB_l), at a time moment t, is defined as the part of link *capacity* on a link which has not been consumed by any traffic at time t. This part of link capacity is available for the new coming flows. As we will see, this parameter will play an important role in our admission control process.

Anycast Flows. An *anycast flow* is a sequence of packets that can be sent to *any one* of the members in a group of designated recipients. Let A be an anycast (destination) address. We denote $G(A)$ to be the group of designated recipients. That is, an anycast flow with anycast address A can be sent to any host in $G(A)$.

We assume that anycast flows have a requirement on sequencing. That is, once the first packet in an anycast flow is delivered to a member (say d) in the recipient group, all the consequent packets of the anycast flow should also be delivered to the same member d. In other words, once the destination of the first packet in the flow is determined, the destination of all the consequent packets in the same flow is fixed. Hence, the *destination of a flow* is referred to the recipient node where all the packets in the flow are sent to.

In this paper, we also assume that anycast flows have certain bandwidth requirement. It is the function of admission control to admit a flow if its bandwidth requirement can be met. For the sake of simplicity, in this paper, we may say "anycast flow" without explicitly stating attributive clause "QoS requirements" when the context is clear.

Route. The sequence of routers via which an anycast flow is transmitted from its source to destination forms a path or a route. We assume that to one source, there is a fixed path to each member in an anycast group. This kind of paths can be obtained via the existing routing protocols [13, 14]. The length of a path is usually defined as the number of hops (i.e. nodes) on the path. As we will see, path length will play a critical role in our distributed admission control procedure. In the remaining of the paper, we will use terms "route" and "path" interchangeably.

To simplify our discussion, we assume that the network has no faults, and between any pair of nodes, there exists at least one functioning path, and the path can be obtained via the existing routing protocols. However, our approach can be extended to deal with the situation when this assumption does not hold.

4 Distributed Admission Control

4.1 Overview

In this section we study admission control procedure for anycast flows with QoS requirements. We have the following objectives in mind:

- Scalability: admission control must be scalable as the number of flows increases and the size of the network expands. Ideally, the overhead of admission control should be independent of the number of flows in the system and the size of the network.
- Effectiveness: A good admission control algorithm should maximize the bandwidth utilization to the possible extent. Thus, an ideal system should have high probability of admitting a new flow if resources are available.
- Compatibility: For practical purposes, the proposed admission control mechanism should be compatible with current industrial practice. We intend to avoid modifying the underline communication infrastructure, such as changing the architecture of routers, revising packet format, etc.

While all these objectives are important and critical, they may conflict with each other. For example, in order to improve scalability and compatibility, the admission probability may be compromised.

To achieve scalability, we adopt Distributed Admission Control (DAC) mechanism. By this mechanism, the admission decisions are made by individual nodes, rather than by a centralized agency. Because of this, the overhead caused by admission control is distributed, hence the distributed admission control is less sensitive to the system parameters such as the number of the active flows, or the size of the network, compared with the centralized admission control mechanism.

However, a distributed admission control procedure may not attain good performance in terms of admission probabilities. As we know, to achieve high admission probabilities, nodes who make admission decisions need closely coordinate with others by sharing the global static and dynamic information. The communication overhead among the geographically distributed nodes prevents this kind of information sharing. Hence, generally speaking, the effectiveness in terms of admission probabilities may be compromised when distributed admission control is utilized.

To address this problem, in the design of our distributed admission control procedure, we consider to randomly distribute anycast flows along available paths in the network in order to reduce the possible congestion and hence increase the chance that an anycast flow is admitted.

4.2 Distributed Admission Control Procedure

As mentioned above, in this paper, we adopt Distribution Admission Control (DAC) mechanism. In DAC, 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). In this paper, we assume that the source routers that receive anycast flow requests are AC-routers.

Once a new anycst flow establishment request arrives, the following procedure is invoked at the correspondent AC-router for the purpose of admission 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 Admission Control Procedure

The above procedure consists of three main steps: destination selection, resource reservation, and retrial control.

The step of destination selection will determine which destination the packets of the requesting flow should be sent to. A good selection will bring 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 continue 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. In the following subsections, we will elaborate our ideas for addressing these issues.

4.3 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 randomized 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:

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

In this paper, we will study two categories of weight assignment algorithms: *unbiased* algorithm and *biased* algorithm. They differ from each other in the sense of whether treating destinations equally or not in weight assignment.

4.3.1 Unbiased Weight Assignment Algorithm

The basic idea of this algorithm is that all the members in an anycast group will have equal probabilities to be selected as a destination of a new incoming flow. To force such an even distribution, the weights associated to individual members must be the same, that is, for $i = 1, 2, ..., K$,

$$W_i = 1/K. \tag{2}$$

Since with this algorithm the anycast flows will be evenly distributed among members of an anycast group, we call this algorithm as Even Distribution algorithm (ED in short). This algorithm is simple. It uses no system status information except the number of members in the anycast group.

However, in general, members in an anycast group, i.e., potential destinations of the anycast flow, are distributed at different locations. There may likely be significant differences among the destinations in the sense of their topological and traffic characteristics, such as distance and load of paths. Ideally, these differences should be taken into account in destination selection. In the following section, we will propose another category of algorithms, so called biased algorithms, which do consider the above differences, and try to perform selection with certain "discrimination" among destinations.

4.3.2 Biased Weight Assignment Algorithms

In this category of algorithms, the members in an anycast group will be treated with certain discrimination in destination selection. Once realized properly, this approach may effectively distribute traffic over different parts of the network, and potentially improve admission probability. To fully take advantage of this effect, we must carefully utilize information available to AC-routers in weight assignment. We propose to use the following information, which can be collected by AC-routers (with different costs, of course), in weight assignment.

- Route Distance Information: For one source, there is one route to a member in an anycast group, which is assumed to be fixed in Section 3. The length of the route, is easy to obtain via the current routing protocols [13, 14]. The differences of route distances reflect the different resource consumption by an anycast flow. Intuitively, the flow to destinations with shorter distances will consume less bandwidth. Hence a smart destination selection algorithm should discriminate destinations with longer distances.

- Local Admission History Information: It is the log that records the successfulness in selecting individual destinations in admission control. Recall that a destination selection is successful only if there is sufficient resource along the path that leads to the destination. Obviously, this kind of information is readily available at the AC-router. Its collection does not cost much at all. However, while this information does reflect network dynamic status, it is not very accurately.

- Route Bandwidth Information: The (minimum available) bandwidth of a route represents the minimum available bandwidth of links along this route. This kind of information reflects network dynamic status. However, it is difficult to get it without extending some of current signaling protocols, such as RSVP.

In the following, we will consider two algorithms use the above information in order to make a proper weight assignment. While both of our weight assignment algorithms use route distance information, the first one uses flow admission history information, and the second one uses route bandwidth information. 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.

Weight Assignment based on Route Distance and Local Admission History Here, we propose a destination selection algorithm based on route distance and local admission history information. Firstly, we will introduce a heuristic weight assignment method that takes into account distance information, we then extend it to include information on local admission history.

As mentioned earlier, a flow with short route distance will consume less resources. Hence a smart destination selection algorithm should prefer destinations with short route distances. Based on the above consideration, a rule of thumb is that weight associated with a destination should be inversely proportional to the distances of routes. That is, for $i = 1, 2, ..., K$,

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

where D_i is the value of the distance of route leading to destination i. To satisfy (1), the weights should be properly normalized, i.e. for $i = 1, 2, ..., K$,

$$W_i = \frac{1/D_i}{\sum_{j=1}^{K} 1/D_j}. \tag{4}$$

Now, we are ready to incorporate the information on admission history for dynamic weight assignment. Recall that 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:

$$H =< h_1, h_2, ..., h_K >, \tag{5}$$

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 defined as follows. At the time of initialization,

$$h_i = 0. \tag{6}$$

Every time when destination i is selected, the following update is made on h_i:

$$h_i = \begin{cases} 0, & \text{if resource reservation returns SUCCESS;} \\ h_i + 1, & \text{otherwise.} \end{cases} \tag{7}$$

That is, the value of h_i records the number of the continual failures in the most recent admission history. For example, $h_i = 3$ implies that for the last three times when destination i was selected in admission control process, there was insufficient bandwidth and resource reservation returned FAILURE.

With the information of list H, the weights at an AC-router are assigned as follows. At the system initialization, the weights are assigned in accordance to (4). Every time when a destination selection is about to be made, weights are updated. The basic idea of updating is to reduce the weights of those destinations whose records in list H are not 0, and to increase weights of those whose records in list H are 0. In particular, the weights are updated via the following three steps:

1. Calculating the total amount of adjustable weights:

$$AW = \sum_{i=1}^{K} W_i * (1 - \alpha^{h_i}). \tag{8}$$

   This step sums up the adjustable weights to AW. α is a parameter used to adjust the impact of local admission history on weight assignment. It will be discussed later.

2. Updating weights:

$$W_i^{'} = \begin{cases} W_i * \alpha^{h_i}, & \text{if } h_i \neq 0; \\ W_i + AW/M, & \text{otherwise,} \end{cases} \tag{9}$$

   where M is the total number of destinations whose records in list H are 0. This step is to reduce the weights of those destinations whose records in list H are not 0, and increase weights of the destinations whose records in list H are 0. Note that the amount of weights to be reduced is determined by parameter α. If α is 0, the local admission history has the maximum impact on weight assignment. At the other extreme, if α is 1, no impact will the local admission history have.

3. Normalizing weights:

$$W_i = \frac{W_i^{'}}{\sum_{j=1}^{K} W_j^{'}}, \tag{10}$$

   where $W_i^{'}$ is given in (9). This step guarantees that the weight assignment satisfies (1).

Given that this weight assignment algorithm uses both the route distance and local admission history, we call this algorithm as Weighted Distribution with route Distance and local admission History information (WD/D+H in short). We expect that its admission probability will be higher than that of Even Distribution (ED) algorithm.

However, the local admission history is coarse and may not accurately reflect the network dynamic status. In the following subsection, we propose an algorithm using on-route bandwidth information, which is better than the local history.

Weighted Assignment based on Route Distance and Available Bandwidth Here, we 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:

$$B_i = \min_{l \in r}(AB_l), \tag{11}$$

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. Thus, we can extend (4) as follows:

$$W_i = \frac{B_i/D_i}{\sum_{i=1}^{K} B_i/D_i}, \tag{12}$$

where B_i and D_i are route bandwidth and route distance of the route from the source to destination i.

Since this weight assignment algorithm uses the route distance and route bandwidth information, we call it as Weighted Distribution based on route Distance and available Bandwidth algorithm (WD/D+B in short).

We expect that WD/D+B will achieve better performance than the previous one. The cost is that we have to know the route bandwidth information. To obtain this kind of information, we have to extend some of current signaling protocols. For example, if RSVP is adopted, we have to extend it to let *RESV* message carry this kind of information back to AC-routers.

In brief, we have introduced three destination selection algorithms. They differ from each other on using system information: the ED algorithm uses no system status information except the number of members in the anycast group, WD/D+H algorithm uses the routes distances and local flow admission history information while WD/D+B depends on the routes distances and bandwidth information.

4.4 Resource Reservation

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 to the selected destination for the requesting flow.

We assumed that the routes from the source to destinations in an anycast group are fixed. 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 DAC procedure. Otherwise,
- Task 2: reserve bandwidth on the each of links along the route, and report SUCCESS by the DAC procedure.

The resourse reservation can be made by the standard RSVP protocol [12]. 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.

4.5 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 Section 5, We will evaluate the sensitivity of R in terms of the admission probabilities. We will show that a few re-trials can significantly improve the system performance.

In this section, we have discussed the design of a distributed admission control mechanism for anycast flows. We have presented three weight assignment algorithms used for destination selection, and discussed the issues in resource reservation and retrial control. In the next section, we will evaluate the admission probability performance of our proposed mechanism.

5 Experimental Evaluation

In this section, we evaluate the performance of the systems that use our distributed admission control procedures with different destination selection algorithms and retrial control schemes. We will first describe the experimental model and then report performance results.

5.1 Experimental Model

- Network: The network considered in our experiments is MCI ISP backbone network. Figure 2 shows the topology of the network. There are 19 nodes that are interconnected by links. Link bandwidth capacity is assumed to be 100 Mbits, and 20 percent of link bandwidth is reserved for anycast flows. Every node is a router and has one host attached.

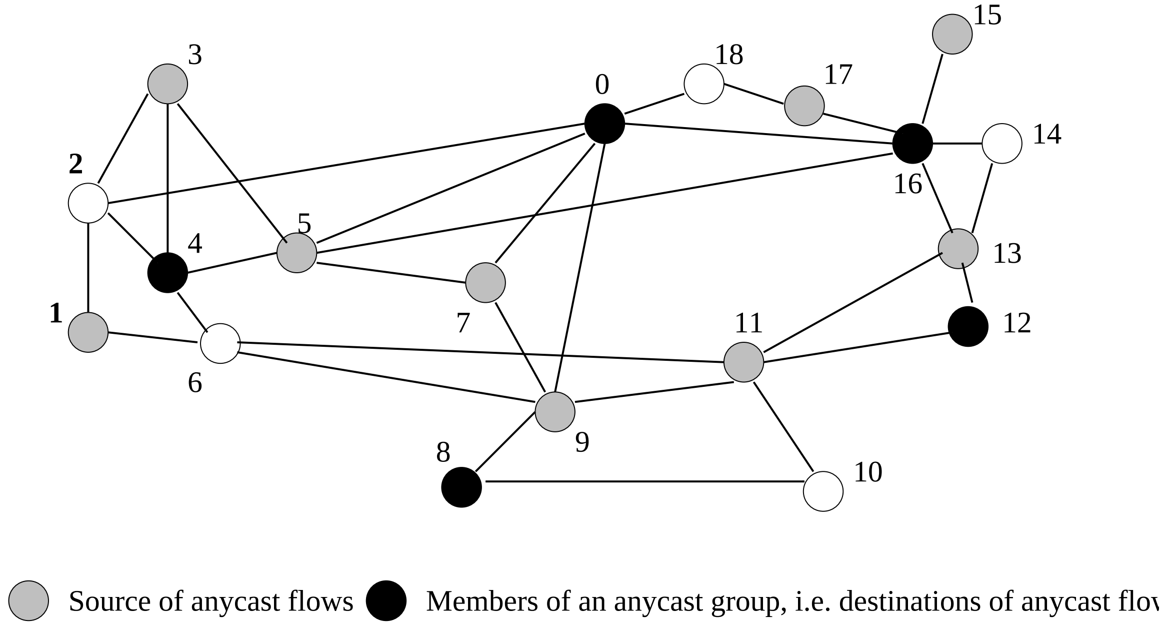

Figure 2: The Experimental Network

- Traffic Model: We assume that requests for anycast flow establishment form a Poisson process with rate λ, while flow lifetimes are exponentially distributed with an average of 180 seconds. Each flow has bandwidth requirement of 64000 bps. Sources of anycast flows are chosen randomly among those hosts that attach the routers with the odd identification numbers. 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.

- Evaluated Systems: A 2-tuple $< A, R >$ is used to represent the systems which run our proposed DAC with different destination selection algorithms and retrial control schemes. In particular, A represents the destination selection algorithm. That is,

$$A \in \{ED, WD/D + H, WD/D + B\}. \tag{13}$$

  R indicates the maximum number of retrials that are allowed. For example, $< ED, 2 >$ represents a system in which Even Distribution (ED) algorithm is used, and the maximum number of retrials is 2. In other words, up to 2 destinations can be tried 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, we will consider two baseline systems.

  - SP System: In this baseline system, the admission control procedure will always pick the destination which has the shortest distance from the source router for each incoming flow. In this system, anycast flows from the same source will be always sent to the same destination. Thus, anycast traffic is not be distributed. Congestion is more likely to occur. We expect the system running our DAC procedure will out-perform this system. Since the destination with the shortest path is always selected, we call this system as Shortest-Path (SP) system.

- GDI System: In this baseline system, the admission control procedure is assumed to have the perfect Global Dynamic Information (GDI) on network status, including the active flows and their usage of bandwidth on each link in the network. It is further assumed that this system is allowed to use any path from a source to a destination and is not subject to the limitation of the fixed path between them. During the admission control, this system does an exhaustive search for all the available paths to all the destinations. As long as there is a path with sufficient bandwidth, the admission control procedure will set up the flow with the qualified path. Obviously, its performance is ideal, but it is not realistic, and it is difficult, if not impossible, to implement such system in practice.

- Performance Metrics:
  - First, we are interested in Admission Probability (AP). It is defined as the probability that an anycast flow is admitted in a stable system. Higher the AP value, the better the performance.
  - The second metric we use is the average number of retrials. Recall that in our systems, we allow up to R times of retrials. Given that we use RSVP-like resource reservation, R is directly proportional to the overhead in terms of resource reservation messages and admission delay. Thus, we would like to have the value of this metric to be as small as possible.
- Evaluation Methods: Both mathematical analysis and computer simulation are used in obtaining performance data. Specifically, we found that the admission probability of systems we consider can be analyzed by queuing theory and fixed point method. Appendix A provides a brief description of the analytical method we used. For all the systems we consider, we also use computer simulation to obtain performance data. Our simulator is Mesquite CSIM, which a process-oriented, general purpose simulation toolkit written in C. The toolkit allows programmers to create and implement process-oriented, discrete-event simulation models. All of our computer simulation run in a SUN Ultra-SPARC workstation.

5.2 Performance Results and Observations

In this sub-section, we report performance results and make observations. Due to the limitation of space, 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.

5.2.1 Sensitivity of Admission Probability

Data on sensitivity of admission probability are given in Figures 3, 4, 5. With these figures, we can assess how AP is sensitive to the arrival rate and maximum number of re-trials (R) in systems $< ED, * >$, $< WD/D + H, * >$, and $< WD/D + B, * >$.

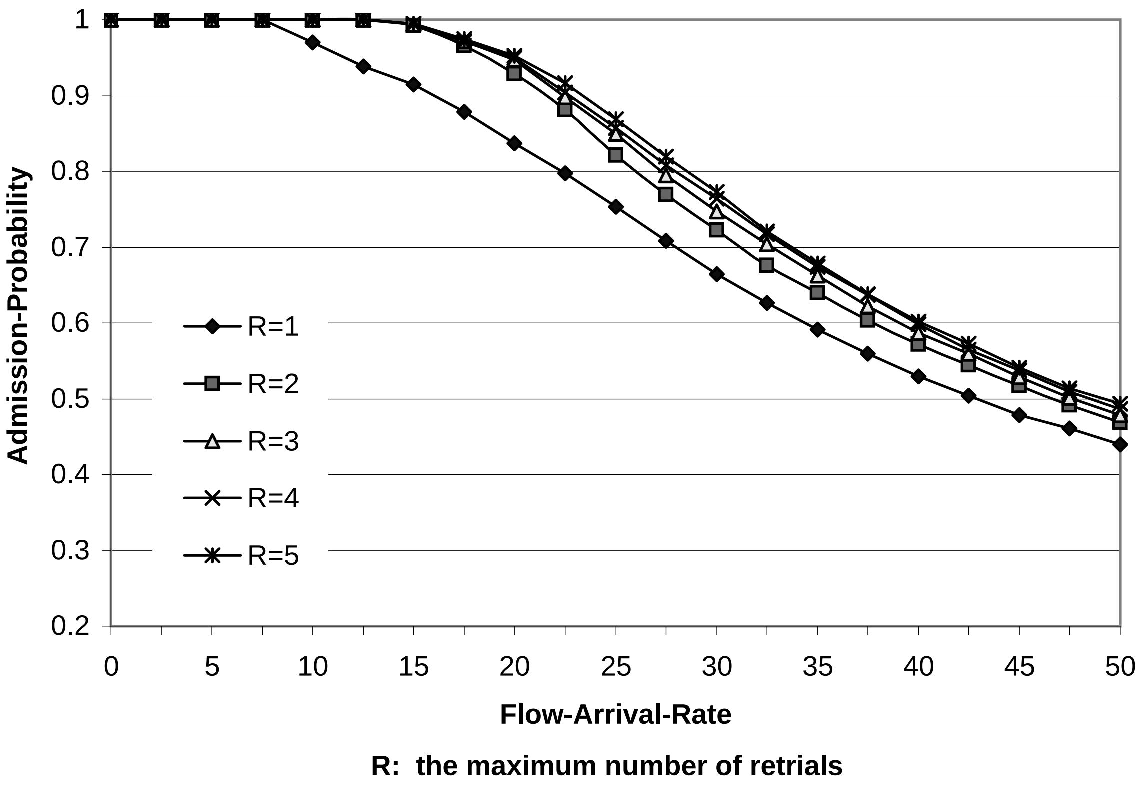

Figure 3: Admission Probability of Systems $< ED, * >$

From Figures 3, 4, 5, we have the following observations:

1. Admission probability is sensitive to the maximum number of retrials (R) in all the systems considered. The value of AP increases as R increases in each system. The reason is obvious: A large R value implies that a large number of retrials are allowed. Consequently, the system has more chance to find a destination with a route having sufficient bandwidth.

2. Furthermore, improvement of admission probability is significant when R increases from 1 to 2. But, further increasing the value of R doesn't result in the same improvement. As matter of fact, the improvement becomes almost invisible when R approaches to 5 which is the upper limit. This observation suggests that our randomized destination selection is effective, in spite of its simplicity. We do not need to exhaustively search all the paths.

3. Different system has different sensitivity to R in terms of admission probability. Especially, systems with lower admission probabilities tend to be more sensitive to the value of R. For example, the AP value of $< ED, * >$ is more sensitive to R than that of $< WD/D+B, * >$. This observation can be explained: the purpose of a retrial is to correct the mistake made in the previous selection. The systems with lower admission probabilities make more mistakes, hence retrial provides better improvement.

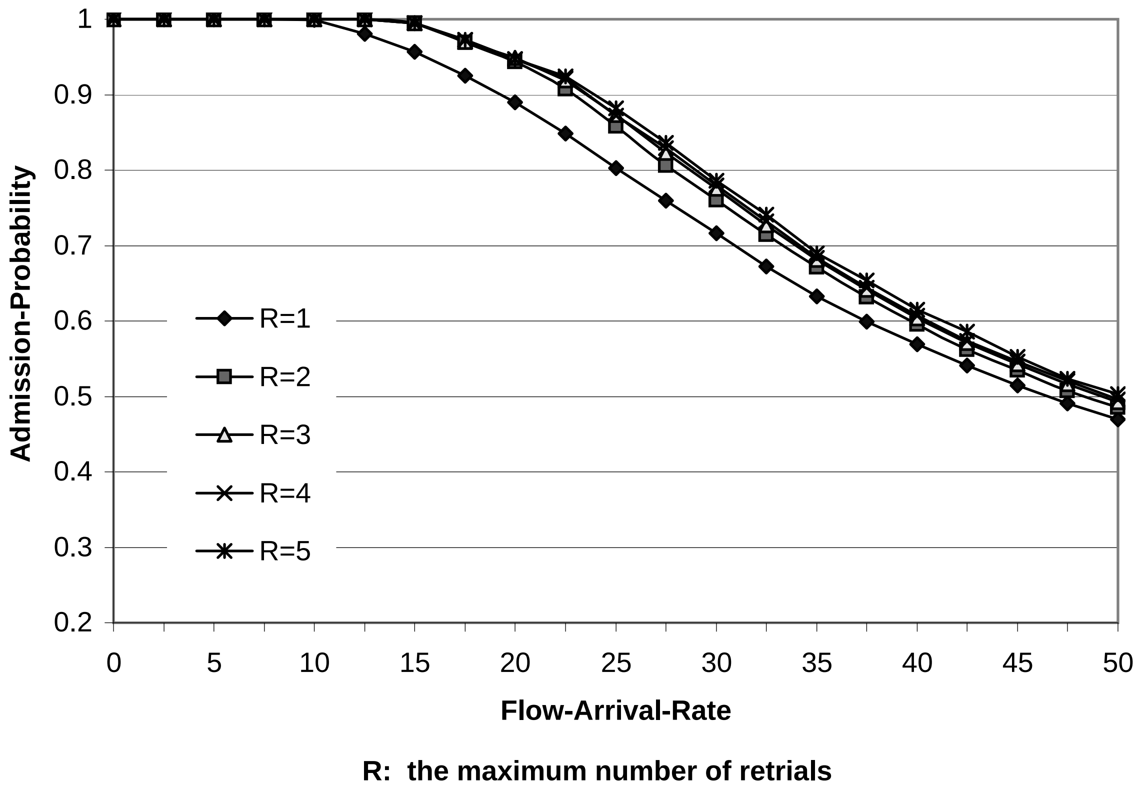

Figure 4: Admission Probability of Systems $< WD/D + H, * >$

5.2.2 Performance Comparison of Different Systems

Figure 6 shows the admission probabilities of systems $< ED, 2 >$, $< WD/D + H, 2 >$, and $< WD/D + B, 2 >$ with two baseline systems: SP and GDI. Figure 7 shows the average number of retrials of system $< ED, 2 >$, $< WD/D + H, 2 >$, and $< WD/D + B, 2 >$ as a function of flow arrival rates.

From Figure 6 and Figure 7, we can make the following observations:

1. As expected, system GDI outperforms all other systems, and SP is the worst in term of admission probabilities in all cases of arrival rates, except in the cases of very low arrival rates, where all systems perform equally. The reason that GDI is the best relies on the fact that GDI uses the perfect global status information so that it can always find a destination as long as there is a path with sufficient bandwidth to the destination. The relatively poor performance of SP supports our argument that it is extremely important to properly distribute anycast traffic in the network.

2. All of our three systems (namely, $< ED, 2 >$, $< WD/D + H, 2 >$, and $< WD/D + B, 2 >$) outperform SP and are close to GDI. This is particularly true for systems $< WD/D + H, 2 >$ and $< WD/D + B, 2 >$. Among our three systems, we can notice that in terms of AP, $< WD/D + H, 2 >$ outperforms $< ED, 2 >$, and so does

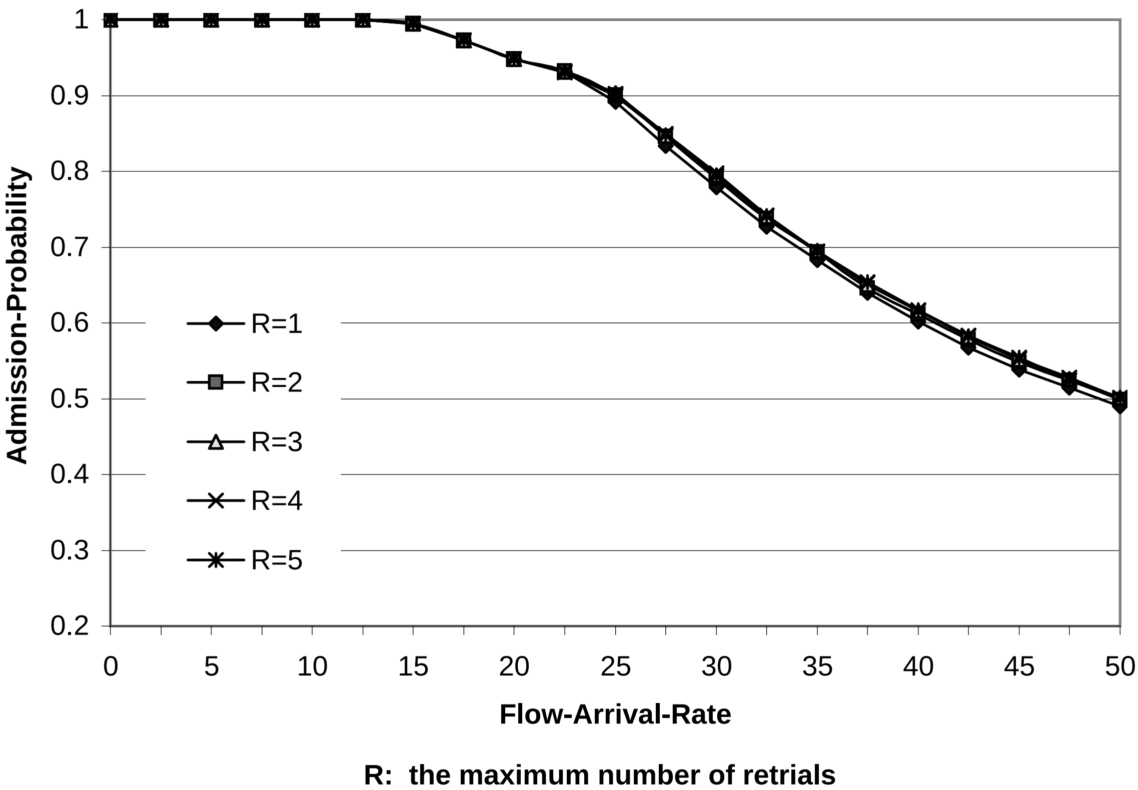

Figure 5: Admission Probability of Systems $< WD/D + B, * >$

$< WD/D+B, 2 >$. This observation implies that our randomized destination selection can work very well, especially proper available knowledge is utilized.

3. Further more, from Figure 7, we find that in terms of the average number of retrials, $< ED, 2 >$ is the worst, which means that it introduces large overhead in comparison with other two. $< WD/D + B, 2 >$ is the best among the three due to its access to the route bandwidth information. However, as we mentioned early, obtaining such information may cause compatibility problem. Hence, in practice one may still prefer system $< WD/D + H, 2 >$.

In any case, the above observations justify our randomized distribution approach and indicate that some of our proposed systems (e.g., $< WD/D + H, 2 >$) are effective in terms of admitting anycast flows at very reasonable overhead and maintaining compatibility with existing network protocols.

6 Final Remarks

We have studied Distributed Admission Control (DAC) procedure for anycast flows with QoS requirements. To the best of our knowledge, this is the first study that addresses the issue on provide QoS support to anycast flows.

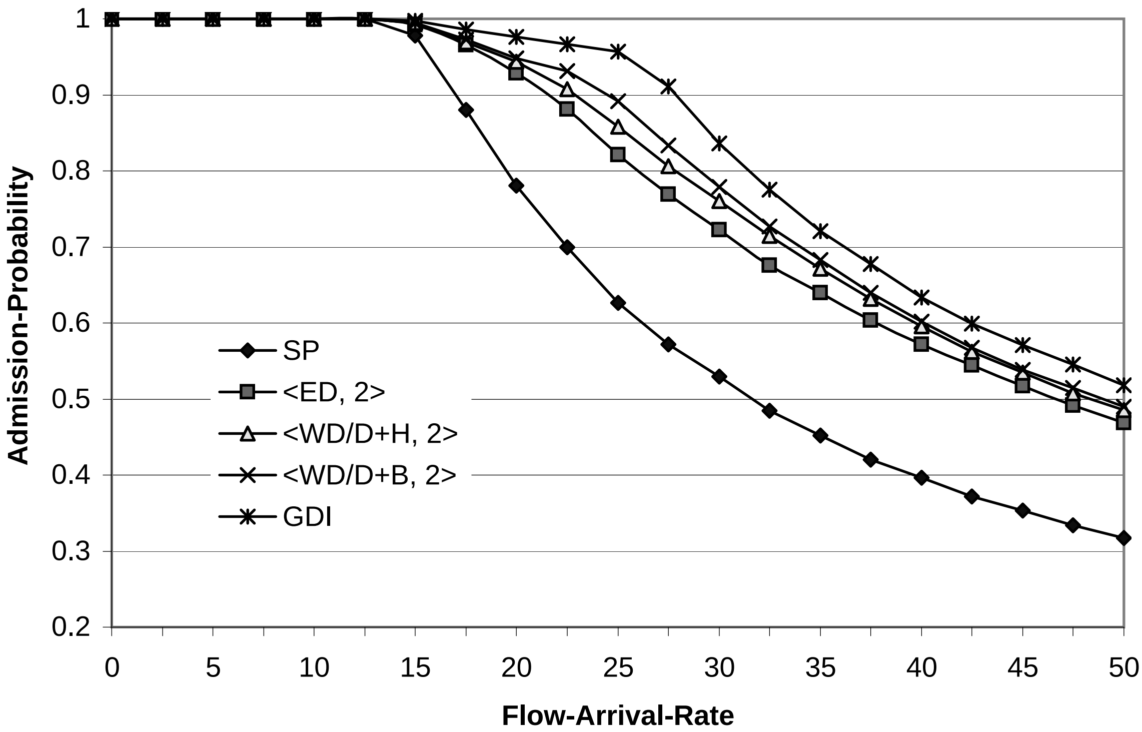

Figure 6: Admission Probability of Different Systems

In our distributed admission control procedure, we focus on algorithms that perform destination selection for anyast flows. We design three algorithms: Even Distribution (ED), Weighted Distribution with route Distance and local admission History information (WD/D+H), and Weighted Distribution with route Distance and available Bandwidth information (WD/D+B) algorithms. These algorithms differ from each other on their dependence on system status information. We address the issues related to resource reservation to ensure the compatibility with existing protocols, and we use re-trial control to make a balance between admission probability and overhead.

We evaluate the proposed mechanisms by mathematical analysis and computer simulation. Performance data have demonstrated that in terms of admission probabilities, our heuristic DAC can perform close to those that utilize global and dynamic status information of network, which is much more expensive and difficult to realize.

The QoS requirement we considered in this paper is limited to bandwidth requirement. While bandwidth is an extremely important QoS aspect, our approach can be easily extended to deal with other QoS requirements, such as end-to-end delay requirement. In the networks with rate-based schedulers, such as weighted-fair-queuing (WFQ) [18, 19], virtual clock (VC) [20], etc., delay requirement can be directly mapped to bandwidth requirement. In the networks with priority-driven schedulers, we can follow the approach presented in [21, 22] to transform delay requirement into an upper bound on bandwidth that need to be reserved, hence converting the delay related QoS to bandwidth-based QoS.

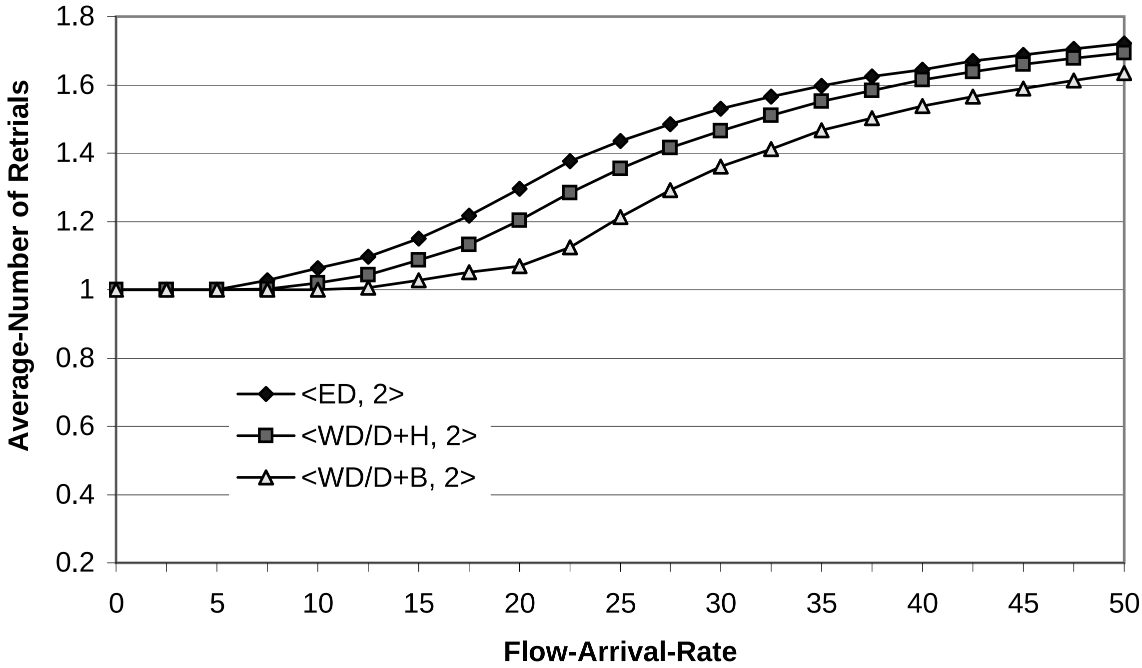

Figure 7: Average Number of Retrials

A Derivation of Admission Probability

In this appendix, we study analytical method for computing admission probability. To simplify the discussion, we will focus on systems $< ED, 1 >$ and SP. The method presented here can be extended to other systems (under certain approximation assumptions). We will not discuss the extension here due to the space limitation.

A.1 Problem Definition

We consider that a network that consists of N nodes and LL links. Let link l ($l = 1, 2, \ldots, LL$) have capacity C_l. Let $G(A)$ be an anycast group with K members. Denote S to be the set of sources of anycast flows. Assume that all the anycast flows have the same bandwidth requirement, say b. Let $(s, G(A))$ denote the stream of flow requests from source s to anycast group $G(A)$. The arrivals of flow requests from stream $(s, G(A))$ form a Poisson process, with mean rate λ_s.

Let $RR(s, G(A))$ be the set of routes for stream $(s, G(A))$[1]. If a flow request in stream (s, G(A)) is offered to route $r \in RR(s, G(A)$, it may be blocked and rejected if there is

[1]In this paper, we assume that routes from the sources of anycast flows to the anycast group are fixed. Hence, in the DAC procedure, if the destination is selected, the route for a new coming anycast flow is determined accordingly.

insufficient bandwidth available on any link of route r. If the flow is admitted, the bandwidth on each link is simultaneously held for the duration of the flow. The holding period (i.e., the life time) of a flow in stream $(s, G(A))$ is assumed to be exponentially distributed with mean $1/\mu_s$. Hence the demanded traffic intensity from s to $G(A)$ is $\rho_s = \lambda_s/\mu_s$.

Let the portion of anycast traffic from source s in set S offered to route r in $RR(s, G(A))$ be $\rho_{s,r}$. In system $< ED, 1 >$, the anycast traffic from s is uniformly distributed among the routes in $RR(s, G(A))$, i.e., $\rho_{s,r}$ is given by ρ_s/K, for any r in $RR(s, G(A))$. In system SP, all the flows are offered to the route with the shortest path in $RR(s, G(A))$. Thus,

$$\rho_{s,r} = \begin{cases} \rho_s, & \text{if r is the shortest path in RR(s, G(A));} \\ \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

We use $L_{s,r}$ to denote the rejection probability of anycast flows from source s to $G(A)$ given that they are offered to route $r \in RR(s, G(A))$. Then, the admission probability of the network can be expressed as follows:

$$AP = \frac{\sum_s \sum_{r \in RR(s,G(A))} \rho_{s,r} * (1 - L_{s,r})}{\sum_s \sum_{r \in RR(s,G(A))} \rho_{s,r}} \tag{15}$$

With (15), once $RR(s, G(A))$ and $\rho_{s,r}$ for each s in S are given, AP can be obtained if $L_{s,r}$ can be computed. Hence, the problem of analyzing AP is reduced to that of analyzing the rejection probabilities of individual routes.

Let B_l denote the rejection probability of link l. This is the probability of a flow request being rejected because of insufficient bandwidth on link l. Recall that link l has capacity C_l and an anycast flow requires bandwidth b. Let v_l be the reduced load obtained after "thinning" of flow requests offered to link l [15]. Then, B_l is a function of b, v_l, and C_l. That is,

$$B_l = L(b, v_l, C_l). \tag{16}$$

Under the link independence assumption [15], the route rejection probability $L_{s,r}$ is now given by

$$L_{s,r} = 1 - \prod_{l \in r} (1 - B_l), \tag{17}$$

where $r \in RR(s, G(A))$. Thus, the problem of analyzing route rejection probability is now reduced to that of analyzing the link rejection probability.

A.2 Analyzing Link Rejection Probabilities

Fixed point method in conjunction with uniform asymptotic approximation can be used to compute link rejection probabilities [15, 16]. The fixed point equations are derived on the basis of the well-known link independence assumption [15]. Each route which uses link l contributes a load to link l, with a rate which is reduced by independent "thinning" by all other links in the route. With the link independence assumption, the reduced load is given by:

$$v_l = \sum_s \sum_{r \in RR(s,G(A))} \rho_{s,r} \prod_{m \in r - \{l\}} (1 - B_m). \tag{18}$$

Synthesizing (16) and (18), we have the following fixed point equations. For $l = 1, 2, \ldots, LL$,

$$B_l = L(v_l), \tag{19}$$

and

$$\begin{aligned} v_l &= \varphi(B_1, B_2, ..., B_{LL}) & (20) \\ &= \sum_s \sum_{r \in RR(s,G(A))} \rho_{s,r} \prod_{m \in r-\{l\}} (1 - B_m). & (21) \end{aligned}$$

Once L() in (19) is determined, B_l can be solved by using the fixed point method which works as follows: This method starts with initial values $v_l^{(0)}$. It then iterates with the following formulas: for $i = 0, 1, \ldots$

$$B_l^{(i+1)} = L(v_l^{(i)}), \tag{22}$$

and

$$v_l^{(i+1)} = \varphi(B_1^{(i+1)}, B_2^{(i+1)}, ..., B_{LL}^{(i+1)}), \tag{23}$$

where $l = 1, 2, \ldots, LL$. The iteration continues until a convergence criterion is satisfied.

The only question now is how to determine function L(). For this, we use the uniform asymptotic approximation (UAA) method [17] which we describe next. The UAA method relies on the following assumptions. For link l,

$$C_l \gg 1, \tag{24}$$

and

$$v_l = O(C_l), \tag{25}$$

where C_l is the capacity of link l, which is an integer. We define:

$$F_l(z) \equiv v_l * (z - 1) - C_l \log z; V_l(z) \equiv v_l * z. \tag{26}$$

Under these assumptions and definition, following the approach in [17], we can derive:

$$B_l = L(v_l) \approx \frac{e^{F_l(z^*)}}{M\sqrt{2\pi V_l(z^*)}}, \tag{27}$$

where

$$\begin{aligned} z_l^* &= \frac{C_l}{v_l}, & (28) \\ M &= \begin{cases} \frac{1}{2}\mathrm{Erfc}[\mathrm{sgn}(1 - z_l^*)\sqrt{-F_l(z_l^*)}] + \frac{e^{F_l(z_l^*)}}{\sqrt{2\pi}}[\frac{1}{\sqrt{V_l(z_l^*)}(1-z^*)} - \frac{\mathrm{sgn}(1-z_l^*)}{\sqrt{-2F_l(z_l^*)}}], & z_l^* \neq 1; \\ \frac{1}{2}\{1 + \frac{1}{\sqrt{2\pi V_l(1)}}[1 + \frac{v_l}{3V_l(1)}]\}, & \text{otherwise}. \end{cases} & (29) \end{aligned}$$

The complementary error function and the sign function are given by

$$\mathrm{Erfc}(y) = \frac{2}{\sqrt{\pi}} \int_y^\infty e^{-x^2} dx. \tag{30}$$

A.3 Comparison of Mathematical Analysis and Computer Simulation Data

As we mentioned above, several approximation assumptions in our mathematical analysis are made in order to simplify computation. Here, we compare performance data obtained by (approximated) mathematical analysis and computer simulation in order to validate the properness of assumptions made.

Method	λ=5.0	λ=20.0	λ=35.0	λ=50.0
Mathematical Analysis	1.000000	0.833933	0.584068	0.435654
Computer Simulation	1.000000	0.837443	0.591091	0.439993

Table 1: Comparison of Mathematical Analysis and Simulation Methods in $< ED, 1 >$

Method	λ=5.0	λ=20.0	λ=35.0	λ=50.0
Mathematical Analysis	1.000000	0.771044	0.444341	0.311417
Computer Simulation	1.000000	0.781039	0.451598	0.317420

Table 2: Comparison of Mathematical Analysis and Simulation Methods in SP

Table 1 and 2 show the admission probabilities of system $< ED, 1 >$ and SP respectively, as a function of flow arrival rates by both mathematical analysis and computer simulation. The network and traffic model are the same as ones described in Section 5.

From Table 1 and 2, we observe that the values of admission probability obtained by both mathematical analysis and computer simulation are almost identical. This justifies the use of the approximation assumptions in our mathematical analysis.

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] S. Bhattacharjee, M. H. Ammar, E. W. Zegura, V. Shah and Z. Fei, "Application-Layer Anycasting", *Proceedings of IEEE INFOCOM'97*, April 1997.

[4] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)", *Proceeding of SIGCOMM'00*, Aug. 2000.

[5] D. Xuan, W. Jia, W. Zhao, and H. Zhu, " A Routing Protocol for Anycast Messages", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, Nox. 6, June 2000.

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

[7] A. Dailianas and A. Bovopoulis, "Real-time admission control algorithms with delay and loss guarantees in ATM networks," In *Proc. of INFOCOM'94*, 1994.

[8] V. Firoiu, J. Kurose, and D. Towsley, "Efficient admission control for EDF schedulers," *Proceedings of Inforcom'97*, 1997.

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

[10] 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, to appear.

[11] R. Bettati, D. Ferrari, A. Gupta, W. Heffner, W. Howe, M. Moran, Q. Nguyen, and R. Yavatkar, "Connection establishment for multiparty real-time communication," *Proc. of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, New Hampshire, Apr. 1995.

[12] 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, September 1993.

[13] C. Hedrick, "Routing Information Protocol", *RFC 1058*, June 1988.

[14] J. Moy, "OSPF Version 2", *RFC1583*, March 1994.

[15] F. P. Kelly, "Blocking probabilities in large circuit switched networks", *Advanced in Applied Probability*, p473-505, 1986.

[16] D. Mitra, J. Morrison and K. Ramakrishnan, "ATM Network Design and Optimization: A Multirate Loss network Framework", *Proc. of Infocom'96*, 1996.

[17] D. Mitra and J. A. Morrison, "Erlang capacity and uniform approximations for shared unbuffered resources", *IEEE/ACM Trans. Networking*, P2:558-570, 1994.

[18] A. K. J. Parekh and R. G. Gallager, "A generalized processor sharing Approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, 1993.

[19] A. K. J. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple-node case," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, 1994.

[20] L. Zhang, "Virtual Clock: A new traffic control algorithm for packet switching networks," *ACM Transactions on Comput. Syst.*, Vol. 9. May. 1991.

[21] B. Choi, D. Xuan, C. Li, R. Bettati, W. Zhao, "Scalable QoS Guaranteed Communication Services for Real-Time Applications", *Proc. of IEEE ICDCS'00*, 2000.

[22] S. Wang, D. Xuan, R. Bettati and W. Zhao, "Providing Absolute Differentiated Services for Real-Time Applications in Static Priority Scheduling Networks", *Technique Report TR00-016, Dept. of Computer Science, Texas A&M Univ.*, July 2000.