INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2006; **16**: 317–336
Published online 20 February 2006 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/nem.596

Architecture for dynamic and fair distribution of bandwidth

Vasil Hnatyshin[1,*] and Adarshpal S. Sethi[2]

[1]*Department of Computer Science, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA.*
[2]*Department of Computer Science and Information Sciences, University of Delaware, Newark, DE, 19716, USA.*

SUMMARY

The problem of fair distribution of available bandwidth among traffic flows or aggregates remains an essential issue in computer networks. This paper introduces a novel approach, called the E**x**act **B**andwidth **D**istribution **S**cheme (X-BDS), for dynamic and fair distribution of available bandwidth among individual flows. In this approach, the edge routers keep per-flow information, while the core routers maintain the aggregate flow requirements. The X-BDS approach employs a distributed message exchange protocol for providing network feedback and for distributing aggregate flow requirements among the nodes in the network. Based on the obtained feedback, the edge routers employ the X-BDS resource management unit to dynamically distribute available bandwidth among individual flows. The X-BDS admission control and resource management units are responsible for fair resource allocation that supports minimum bandwidth guarantees of individual flows. This paper evaluates the Bandwidth Distribution Scheme through simulation and shows that the X-BDS is capable of supporting per-flow bandwidth guarantees in a dynamically changing network environment.

1. INTRODUCTION

The problem of fair distribution of available bandwidth among traffic flows or aggregates remains an essential issue in computer networks. This issue gained even more importance as the bandwidth-demanding multimedia flows became one of the most commonly transferred types of traffic. To provide satisfactory quality of service to the end-user, multimedia traffic usually requires considerable amounts of bandwidth. However, bandwidth resources are very scarce and such requirements often cannot be met. Traditionally, when a new traffic flow wants to enter the network, the edge router either admits a new flow without any verification or performs an admission control check to ensure that the network is able to support resource requirements of the new flow. In a network with no unused bandwidth, admission of a new flow may lead to congestion. As a result, multiple flows might fail to acquire sufficient resources and thus will not provide satisfactory quality of service to the end-users. On the other hand, in the absence of available resources, denying access for a new flow may be too conservative if the flows that are already admitted into the network can adjust or degrade their resource consumption to accommodate a new flow. We argue that it is more profitable to have multiple flows transmitting data at the smallest possible rates that satisfy the end-user QoS requirements than to have a single flow allocated the same amount of bandwidth all to itself. Current networks can benefit from an intelligent mechanism that dynamically adjusts bandwidth distribution of already admitted flows in the network to accommodate a new flow. This paper examines such a mechanism called the E**x**act **B**andwidth **D**istribution **S**cheme (X-BDS). (We call our scheme 'exact' because the X-BDS scheme relies on the exact and not estimated

*Correspondence to: Vasil Hnatyshin, Department of Computer Science, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA.
†E-mail: hnatyshin@rowan.edu

Received 12 August 2005
Accepted 7 September 2005

values of the flow requirements. An alternate and different Bandwidth Distribution Scheme that estimates the flow requirements was presented in Reference 1) The primary goals of the X-BDS approach are to provide minimum bandwidth guarantees to individual flows, to share available bandwidth fairly, and to dynamically adjust per-flow resource allocation when needed.

The X-BDS approach is designed to operate within a single network domain. In such an X-BDS domain, the edge nodes maintain per-flow information and fairly (re)distribute network resources (e.g., bandwidth) among individual flows according to the flow requirements and resource availability. In the X-BDS approach, the flow requirements consist of the minimum and maximum amounts of bandwidth requested by the flow. This information is negotiated ahead of time for each X-BDS flow. The information about resource availability consists of:

- the total bandwidth allocated for X-BDS traffic on the link;
- the total arrival rate of the X-BDS traffic on the link; and
- the aggregated resource requirements of the X-BDS traffic (e.g., the sums of the minimum and the maximum amounts of bandwidth).

The resource availability information is maintained and dynamically distributed by the core routers. The X-BDS relies on the idea of pushing per-flow information to the network edges while the network core maintains only minimal amount of information and keeps the packet processing simple. This idea is not new and has been examined before [1–6]. However, the primary contribution of this paper is a novel approach to aggregating flow requirements and a new distributed network feedback protocol that allows the edge nodes to discover network changes and dynamically adjust bandwidth distribution to satisfy minimum bandwidth requirements of individual X-BDS flows.

Even though we present the X-BDS approach as a method for dynamic bandwidth distribution among individual flows, the same idea could be extended to traffic classes or aggregates (e.g., Differentiated Services [2], traffic aggregates, Multi Protocol Label Switching [7] tunnels). However, the issue of applying the X-BDS idea to traffic aggregates is outside the scope of this paper and will not be discussed further.

The rest of the paper is organized as follows. The next section introduces the X-BDS architecture along with related specifications and definitions. The protocol for distribution of the aggregate flow requirements is described in the third section. The fourth section presents definitions of fairness and the resource management mechanism. In the fifth section we introduce implementation details, while the sixth section presents performance evaluation of the X-BDS approach. Finally, the seventh section provides related work overview and discussion and the eighth section presents the conclusions.

2. THE X-BDS ARCHITECTURE

2.1 Overview of the X-BDS Architecture

Figure 1 presents the X-BDS architecture, which consists of three components: a set of specifications and definitions, *the Requested Bandwidth Range* (RBR) *Distribution and Feedback* (RDF) *protocol*, and the resource management mechanism. The specifications and definitions consist of the network architecture, which defines the working environment of the X-BDS, and the flow requirements definition, which outlines the user expectations of traffic treatment.

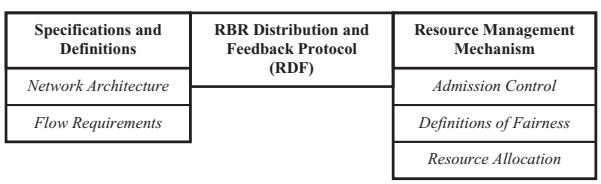


Figure 1. The X-BDS architecture

DOI: 10.1002/nem

The RBR RDF protocol is the glue that holds the X-BDS approach together. The RDF protocol is a distributed explicit message exchange protocol that provides feedback about the changes of network characteristics. More specifically, the RDF protocol distributes aggregate flow requirements among the routers in the network and generates explicit congestion notifications when needed.

The X-BDS resource management mechanism relies on the network feedback provided by the RDF protocol to perform per-flow admission control and to fairly distribute available bandwidth among the flows. The X-BDS resource management mechanism consists of the admission control that determines if a new flow can be admitted into the network without violating existing guarantees, the definitions of fairness which specify what it means for the resource distribution to be fair, and per-flow resource allocation that dynamically distributes available bandwidth among the flows in a fair manner.

2.2 Network Architecture

The X-BDS approach uses a network architecture similar to that of the Differentiated Services model [2], where the Internet is viewed as a network consisting of routers grouped into independent network domains as shown in Figure 2. A cluster of interconnected routers that are governed by the same administrator is called a network domain. Each network domain contains two types of nodes: the edge or boundary routers and the core routers. Traffic enters a network domain through the edge nodes called ingress routers, travels through the core routers to reach the network boundary, and exits the domain through the edge nodes called egress routers. This paper examines the X-BDS approach within the confines of a single network domain.

In a network domain, the X-BDS core routers do not perform per-flow management. Instead each core router maintains information about the edge routers that send traffic through its links and distributes the aggregate flow information in the network. These additional X-BDS responsibilities of the core router require only a small amount of extra space for data maintenance (e.g., proportional to the number of edge routers that send traffic via this core router) and a limited amount of processing power for dealing with and forwarding control packets used for distribution of the aggregate flow requirements. In particular, the processing and storage requirements in the core routers are not proportional to the number of flows passing through the router.

On the other hand, the edge routers are responsible for maintaining per-flow information and regulating data traffic that enters the network. Each edge router controls the flow of data traffic by dynamically adjusting bandwidth allocation of individual flows, which requires per-flow information stored at that edge router and aggregate information provided by the core routers. This network architecture follows the Internet philosophy of keeping the network core simple and moving all heavy-duty data maintenance and processing into the network edges. Furthermore, this network architecture does not require the X-BDS approach to be set up everywhere in the Internet at once. Instead, each network domain can choose to support the Bandwidth Distribution Scheme at its own discretion, which facilitates incremental deployment.

The proposed X-BDS approach is designed for a single network domain and works only with the X-BDS flows; e.g., the flows that adhere to all X-BDS requirements and processing. This paper defines a

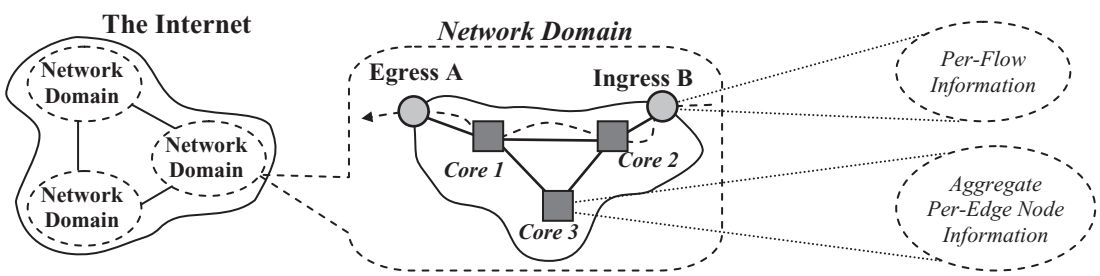


Figure 2. The X-BDS network architecture

DOI: 10.1002/nem

'flow' to be a sequence of packets that travel from a given source host to a given destination host. This definition of a flow, while different from the more conventional definition as a sequence of packets between individual source–destination applications (e.g., TCP or UDP streams), was chosen to simplify the presentation of the X-BDS scheme. The X-BDS architecture, as presented here, can be easily extended to apply to the conventional definition of a flow.

If a network domain decides to support the X-BDS, then a certain amount of resources (e.g., bandwidth) are allocated for the X-BDS traffic. The information about the allocated amount of X-BDS resources is readily available everywhere in the network. These X-BDS resources are fairly distributed among the X-BDS flows only. To ensure that non-X-BDS flows do not consume the X-BDS resources, these traffic types must be isolated. Isolation of the X-BDS flows from the rest of the traffic in the network is fairly simple and could be implemented as follows:

1. the edges routers identify the X-BDS based on the Service Level Agreement (SLA) established between the end-user and the network administrator;
2. the edge routers mark the packets that belong to the X-BDS flows (e.g., sets the value in the ToS field in the IP header, similarly to DiffServ architecture [2]);
3. the edge and core routers differentiate between X-BDS and non-X-BDS flows based on the packet marking and buffer these flows into separate queues; and
4. the edge and core routers apply a simple queuing mechanism such as Weighted Fair Queuing mechanism to the queues that contain X-BDS and non-X-BDS traffic.

2.3 The X-BDS Flow Requirements

This paper assumes that both the minimum and maximum transmission rates of the X-BDS flows are known ahead of time; e.g., are negotiated between the user or the application and the network administrator. To simplify the notation, we will use the term *flow* to mean an X-BDS flow (e.g., a flow that adheres to the X-BDS requirements and processing), the term *link capacity* or simply *capacity* to mean the X-BDS capacity (e.g., the link capacity explicitly allocated by the network administrator for the X-BDS traffic), and the term *bandwidth* to mean the X-BDS bandwidth (e.g., the bandwidth allocated for or consumed by the X-BDS traffic).

In the X-BDS approach, the flow requirements are defined in the form of a range called the Requested Bandwidth Range (RBR). The RBR of flow f, RBR^f, consists of two values: a minimum rate, b^f, below which the flow cannot operate normally, and the maximum rate, B^f, that the flow can utilize.

$$\mathrm{RBR}^f = [b^f,\ B^f] \tag{1}$$

Link k is a *bottleneck link* for flow f traveling on path P if k limits transmission rate of f on P. Consider a core router's link k and a set of flows, F^k, that travel through it. The set F^k can be divided into two disjoint subsets: the subset, F^k_{B}, of flows that have link k as their bottleneck; and the subset, F^k_{NB}, that contain all the other flows. These subsets are called *bottleneck flows* and *non-bottleneck flows*, respectively.

$$F^k = F^k_{\mathrm{B}} \bigcup F^k_{\mathrm{NB}} \tag{2}$$

The *aggregate bottleneck RBR* and the *aggregate RBR* on link k are defined as follows:

$$b^k_{\mathrm{B}} = \sum_{f \in F^k_{\mathrm{B}}} b^f \quad B^k_{\mathrm{B}} = \sum_{f \in F^k_{\mathrm{B}}} B^f \tag{3}$$

$$b^k = \sum_{f \in F^k} b^f \quad B^k = \sum_{f \in F^k} B^f \tag{4}$$

The core routers maintain the aggregate RBR for each of their outgoing links and distribute this information along with the X-BDS link capacity among the routers in the network.

The X-BDS approach considers a simple definition where the flow's request consists of only two values: the minimum and the maximum requested bandwidth. Other research [8] defined the flow request to

DOI: 10.1002/nem

consist of a certain amount of bandwidth, utility function based on the received bandwidth, and priority. In X-BDS we made a simplified assumption that all flows have utility that corresponds to the linear utility function of reference 9: the flow's utility increases linearly as the amount of allocated bandwidth for the flow approaches the maximum requested rate. In addition, the X-BDS approach assumes that traffic is not explicitly prioritized. Instead, the importance or priority of a flow is determined by the requested bandwidth range. The flows with high importance or priority will request more bandwidth and/or will have a small difference between the maximum and the minimum requested rates; the small difference ensures little or no fluctuation of the flow's bandwidth allocation during resource redistribution. However, the X-BDS approach can easily be extended to include additional flow utility functions and explicit traffic priority. Such extension would require the X-BDS resource management mechanism to become similar to that of reference 9. However, this issue is outside the scope of this paper and will not be addressed further here.

3. THE RBR DISTRIBUTION AND FEEDBACK PROTOCOL

The RBR Distribution and Feedback (RDF) protocol is a distributed explicit message exchange protocol that governs information sharing between the routers in the X-BDS network. The RDF protocol operates as the 'glue' that holds the X-BDS architecture together by distributing information which enables the edge routers to perform admission control and fair per-flow bandwidth distribution. The communication between the X-BDS nodes in the network is implemented via an explicit exchange of control messages. Each control packet carries the necessary information and is marked as the X-BDS packet of the highest priority. The marking ensures that control packets are forwarded ahead of all the other X-BDS traffic and that they experience no loss and very limited queuing delay on the path to their destinations.

The protocol consists of three distinct and independent phases or parts: the path probing phase, the update phase, and the notification phase. The path probing phase discovers characteristics of a particular path. The edge routers initiate the RBR update phase to notify the core routers about the changes to the aggregate information due to flow activation or termination. In the event of congestion the core routers initiate the notification phase, which distributes aggregate flow information among the corresponding edge nodes and subsequently triggers redistribution of available bandwidth among the flows that contribute to congestion.

3.1 The Path Probing Phase

The edge routers initiate the path probing phase to discover characteristics of the not-yet-discovered path. Throughout the duration of the path probing phase, the edge routers periodically send the probe messages to collect aggregate information maintained at each visited core router. The frequency of the path probing depends on the expected frequency of the network changes. We expected the path probing frequency to be in the order of seconds. The duration of the path probing phase is equal to the time the corresponding route remains active. A route is considered to be active if there is at least one flow that travels on this route to reach its destination. Thus, the edge routers initiate the path probing phase when a new flow that travels over the not-yet-discovered path activates. The edge routers terminate the path probing of a particular route when there is no more traffic traveling on that route.

Initially, the control packet generated during the path probing phase is empty (e.g., it contains only the IP and the lower layer headers). However, as the packet traverses the network, each router on its forward path, including the edge router that generated this packet, inserts information such as the IP address of the packet's outgoing link as well as the X-BDS capacity and the aggregate RBR on that link. Once the path probing control packet reaches its destination (e.g., the corresponding egress router) it is forwarded without any modification back to the edge router that generated it. The routers on the reverse path do not update the content of the path probing control packet.

The edge routers maintain the information collected during the path probing phase in the following tables: the Path table, which contains information about each active path and a list of flows that travel

DOI: 10.1002/nem

on that path; and the Link table, which contains information about the links that are part of the active paths. In addition, the edge routers maintain the SLA table, which contains per-flow information. Figure 3 shows a simplified view of the data structures maintained at the edge nodes. Such design facilitates the storage of the necessary information only and fast access to it. The Link table, the Path table, and the SLA table are implemented as hash tables, with the hash key being the IP address of the outgoing link, the unique path ID, and the unique flow ID, respectively. This enables fast access to the required information in each of the tables. Furthermore, each link has only a single entry in the Link table, even if the link is a part of multiple paths. The Path table contain a linked list of pointers into the Link table. Thus, if a link belongs to multiple paths then multiple entries in the Path table will have pointers to the same entry in the Link table.

3.2 The Update Phase

The purpose of the update phase is to notify the core routers about the aggregate information changes. Generally, the edge routers initiate the update phase when a new flow is being admitted into the network. The edge routers terminate the update phase after corresponding flows deactivate. During the update phase the edge routers send two control packets: one upon activation and the other upon termination. These packets are called the CNG messages and they carry the change of flow(s) requirements on the corresponding path. An optimization of the update phase allows a single CNG message to carry the requirements change for multiple flows that travel on the same path. This optimization is described under 'BDS Implementation', below.

Upon CNG message arrival, the core routers accumulate flow(s) requirements within the aggregate information of a particular link. To store the aggregate information supplied by the CNG messages, the core routers maintain the *Core Links* table as shown in Figure 4. The CNG message terminates its progress upon arrival to its destination; e.g., the corresponding egress node.

3.3 The Notification Phase

In the event of congestion the core routers initiate the notification phase. The primary purpose of the notification phase is to eliminate network congestion by signaling the edge routers to throttle the

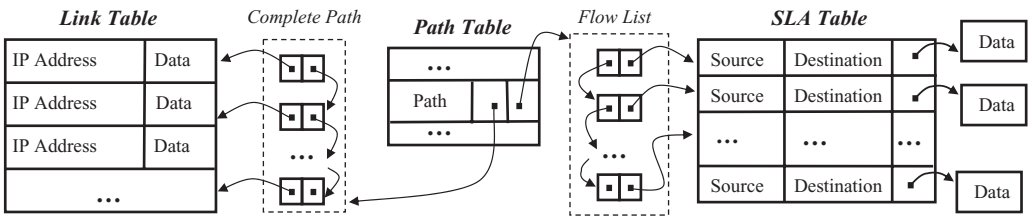


Figure 3. Data structures maintained at the X-BDS edge nodes

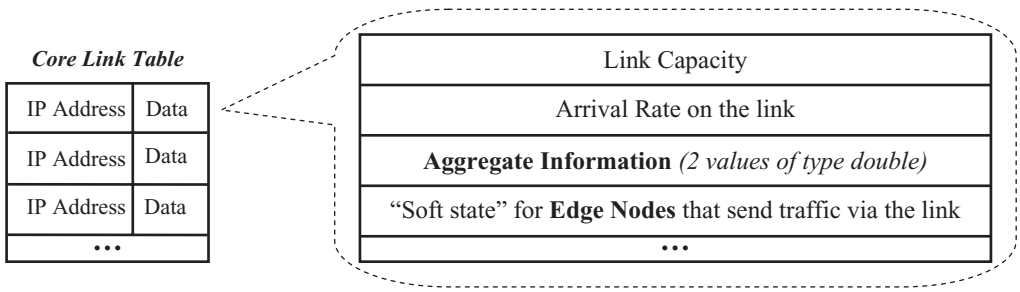


Figure 4. Data structures maintained at the X-BDS core nodes

DOI: 10.1002/nem

flows that contribute to congestion. The notification phase terminates when all the edge routers that contribute to congestion receive notification messages and adjust the bandwidth allocation of their flows accordingly.

When the core router discovers congestion on one of its links it consults the Core Links table to identify the edge routers to be notified about congestion. The core router sends a control packet called the CN message to each of the identified edge routers. The CN message carries the identity and aggregate information for the congested link. Alternatively, the CN message could be aggregated on the common part of the path to the edge routers [10,11]. Summary of the RDF protocol is presented in Table 1.

4. THE X-BDS RESOURCE MANAGEMENT MECHANISM

The X-BDS resource management mechanism relies on the information provided by the RDF protocol to fairly (re)allocate available bandwidth among individual X-BDS flows. At any point in time, the edge routers have the following information available: the RBR of individual flows (e.g., available in SLA and negotiated ahead of time), the aggregate RBR and the X-BDS link capacity (e.g. available in the Link and Path tables and provided by the RDF protocol). This information is sufficient to compute the fair resource allocation among individual X-BDS flows.

4.1 Definitions of Fairness

The X-BDS approach provides fair resource allocation and in this section we define what it means for the resource allocation to be fair. The section on 'The X-BDS Flow Requirements', above, defines the aggregate bottleneck RBR as the sum of the RBRs of the bottleneck flows on link k, and the aggregate RBR as the sum of the RBRs of all the flows that travel through link k. The total allocated rate of the non-bottleneck flows is called the *non-bottleneck rate* and is denoted as R^k_{NB}. The amount of bandwidth left for distribution among the bottleneck flows is the difference between the capacity of link k (e.g., the X-BDS capacity), C^k and the non-bottleneck rate, R^k_{NB}. This value is called the *bottleneck capacity*, C^k_{B}.

$$C^k_{\mathrm{B}} = C^k - \sum_{f \in F^k_{\mathrm{NB}}} R^f = C^k - R^k_{\mathrm{NB}} \tag{5}$$

Generally, the goal of fairness is to provide an amount of resources to a customer that is proportional to the price paid by the customer [9]. In the X-BDS approach, if customers are charged based on their desired RBRs, then the fairness goal is realized by allocating each bottleneck flow f on link k an amount of excess

Phase Name	Ingress Router	Core Router	Egress Router
Path Probing Phase	○ Initiate upon path activation. ○ Periodically generate control messages. ○ Update local data structure upon control message return. ○ Terminate the phase upon path becoming inactive.	Upon path probing control message arrival update the content of the message.	Return the control message back to the ingress node that generated it.
Update Phase *Notification Phase*	Generate the CNG message when the change of the aggregate flow information occurs. Adjust resource distribution upon CN message arrival.	Update the aggregate flow information if CNG message arrived. Generate CN message upon congestion occurrence.	Terminate the CNG message progress. Adjust resource distribution upon CN message arrival.

Table 1. Summary of the RDF protocol

DOI: 10.1002/nem

bandwidth proportional to the flow's weight w^f, which is a function of the flow's RBR and thus the price. The excess bandwidth is the resources left after each bottleneck flow is allocated its minimum rate. The fair share of flow f through bottleneck link k is then

$$\mathrm{FS}_f^k = b^f + (C_{\mathrm{B}}^k - b_{\mathrm{B}}^k)\frac{w^f}{\sum_{j \in F_{\mathrm{B}}^k} w^j} \tag{6}$$

It is interesting to note that, although we crafted the fairness definition (6) independently to suit the specific requirements of X-BDS, it is equivalent to the definition of general weighted fair allocation in ATM networks [12]. Based on the fairness definition (6), we introduce the following fairness criteria for bandwidth distribution. The simple and most intuitive way is to distribute bandwidth proportionally to the flow's minimum requested rate (MRR) [9,12]. We call this the proportional fairness criterion. This fairness criterion is achieved by assigning $w^f = b^f$.

$$\mathrm{FS}_f^k = b^f + (C_{\mathrm{B}}^k - b_{\mathrm{B}}^k)\frac{b^f}{b_{\mathrm{B}}^k} = C_B^k \frac{b^f}{b_{\mathrm{B}}^k} \tag{7}$$

This definition of proportional MRR fairness criterion (7) makes it identical to the general weighted fairness proportional to Minimum Cell Rate criterion [12], where each flow is allocated its minimum requested rate b^f plus a share of the leftover bandwidth proportional to b^f.

In the second fairness criterion, the leftover bandwidth is distributed proportionally to the difference between the flow's maximum and minimum requested rates; i.e., the amount of bandwidth a flow needs to be completely utilized. We assume that a flow is completely utilized when it sends traffic at its maximum requested rate, B^f. This fairness criterion is called maximizing utility fairness and is achieved by assigning $w^f = B^f - b^f$.

$$\mathrm{FS}_f^k = b^f + (C_{\mathrm{B}}^k - b_{\mathrm{B}}^k)\frac{B^f - b^f}{B_{\mathrm{B}}^k - b_{\mathrm{B}}^k} \tag{8}$$

The allocated rate of a flow is limited by its RBR and thus may be equal to its maximum requested rate but smaller than its fair share on the bottleneck link.

4.2 Admission Control

The X-BDS network guarantees that each flow would receive at least its minimum requested rate, b^f, while the leftover resources in the network are fairly distributed among participating flows based on the corresponding definition of fairness defined by (7) and (8). To achieve these guarantees, the network allocates to each flow an amount of bandwidth not smaller than the flow's minimum requested rate, and denies network access to those flows whose minimum rate guarantees cannot be met.

The purpose of admission control is to determine whether a new flow can be admitted into the network at its minimum rate without violating existing QoS guarantees of other flows. The problem of admission control has been extensively examined in the literature [13–15]. Traditionally, there are two types of admission control: *parameter-based* and *measurement-based*. In parameter-based admission control, the decision to admit a new flow is derived from the parameters of the flow specification. Usually, this type of admission control relies on worst-case bounds and results in low network utilization, although it does guarantee supported quality of service. Measurement-based admission control relies on measurements of the existing traffic characteristics to make the control decision. Measurement-based admission control supports higher network utilization but it may occasionally cause the quality of service levels to drop below user expectations because of its inability to accurately predict future traffic behavior.

In the X-BDS approach, to determine whether a new flow can be admitted into the network, the edge node examines the current resource allocation on the flow's path. Since the network guarantees that each flow will receive at least its minimum requested rate, the edge router verifies that the sum of the minimum

DOI: 10.1002/nem

requested rates of all the flows that follow a particular path, including a new flow, is smaller than the capacity of the bottleneck link on that path.

Formally, we define the X-BDS admission control as follows. Consider a network that consists of a set of L unidirectional links, where link $k \in L$ has capacity C^k allocated for the X-BDS traffic. This network is shared by the set of X-BDS flows, F, where flow $f \in F$ has the RBR of $[b^f, B^f]$. At any time, the flow enters the network at a rate R^f, called the *allocated rate*, which lies between b^f and B^f. Let $L_f \subseteq L$ denote the set of links traversed by flow f on its way to the destination. Also let $F^k \subseteq F$ denote the set of flows that traverse link k. Then a new flow ϕ with the RBR of $[b^\phi, B^\phi]$ is accepted in the network if and only if

$$b^\phi + \sum_{f \in F^k} b^f \leq C^k \quad \forall k \in L_\phi \tag{9}$$

Thus, new flow ϕ is admitted into the network only if the sum of the minimum requested rates of the active X-BDS flows, including the new flow, is not larger than the X-BDS capacity of each link on ϕ's path to its destination. Equation (9) is often called the *admission control test.* The X-BDS approach employs a variation of the measurement-based admission control. However, unlike traditional measurement-based approaches, the X-BDS admission control does not violate bandwidth requirements of individual flows because the RDF protocol supplies accurate values of link capacities and the aggregate RBRs on the path.

4.3 The X-BDS Resource Allocation Mechanism

To distribute bandwidth according to fairness criteria (7) and (8), the resource management mechanism requires the knowledge of such link characteristics as the aggregate bottleneck RBR and the bottleneck capacity. However, these characteristics are not readily available in the network. Instead the core routers keep track of the capacity, arrival rate, and aggregate RBR for each of their outgoing links. As a result, the edge nodes use the aggregate RBR and link capacity, and not the aggregate bottleneck RBR and the bottleneck capacity, to compute the fair shares of individual flows. To achieve fair resource distribution (6), the X-BDS resource management mechanism consists of two parts: the primary and leftover bandwidth distribution.

When a new flow activates or a congestion notification arrives, the edge node employs the primary bandwidth distribution and computes the fair share of flow f on its bottleneck link k using either the proportional or the maximizing utility fairness criterion as shown below.

$$\mathrm{FS}_f^k = b^f + (C^k - b^k)\frac{b^f}{b^k} = C^k \frac{b^f}{b^k} \tag{10}$$

$$\mathrm{FS}_f^k = b^f + (C^k - b^k)\frac{B^f - b^f}{B^k - b^k} \tag{11}$$

However, the flows that do not have link k as their bottleneck do not adjust their allocated rates. In the section 'Network Architecture', above, we defined link k as a bottleneck link for flow f traveling on path P if k limits transmission rate of f on P, which in context of the X-BDS approach and equations (10) and (11) means that the fair share of flow f on link k is the smallest on path P. As shown in Figure 3, the edge routers maintain complete path information for each flow, which is sufficient to identify the bottleneck link [11].

Clearly, if the edge routers employ only the primary bandwidth distribution as defined by equations (10) and (11), then link k may often become underutilized because the non-bottleneck flows will not adjust their allocated rates and will traverse link k at rates below their fair shares on link k. A 'water-filling' technique of the max–min fairness [16–18] provides an adequate solution to this problem. The idea of 'water-filling' is to gradually increase allocated rates of individual flows until the leftover bandwidth on the underutilized link k is completely consumed. The X-BDS approach periodically probes the path to discover excess bandwidth, which enables the edge routers to implement a variation of the 'water-filling' technique.

DOI: 10.1002/nem

The periodic path probing delivers information about availability of resources on the path and enables distribution of leftover bandwidth. If the periodic path probe discovers excess bandwidth EB^k on the path (e.g., on link k), then using proportional and maximizing utility definitions of fairness, the flows that travel on that path may increase their fair shares as follows:

$$FS_f^k = {}_{\text{old}}FS_f^k + EB^k \frac{b^f}{b^k} \tag{12}$$

$$FS_f^k = {}_{\text{old}}FS_f^k + EB^k \frac{B^f - b^f}{B^k - b^k} \tag{13}$$

where ${}_{\text{old}}FS_f^k$ is the fair share of flow f on link k prior to discovery of excess bandwidth EB^k. Thus, in the presence of leftover bandwidth, the edge routers increase allocated rates of individual flows until available bandwidth on the path is consumed completely. It has been shown [11] that combination of the primary and leftover bandwidth distribution as defined by equations (10)–(13) approximates fair resource distribution as defined by equations (7) and (8).

5. BDS IMPLEMENTATION

5.1 BDS Implementation at the Edge and Core Routers

To implement the X-BDS approach the flow requirements stored at the edge routers should include the flow RBR values. During the update phase the CNG messages carry and distribute the flow(s) RBR values among the core routers. It should be noted that upon flow termination the flow(s) RBR values carried in the CNG message are negative.

Upon CNG message arrival the core routers update the Core Links table and corresponding aggregate flow requirements. Specifically, the core routers add the RBR values provided by the CNG message to the aggregate RBR (e.g., the aggregate flow requirements) of the corresponding link. In addition, the core routers maintain a 'soft state' for each edge node that sends traffic on the corresponding link. Thus, the arriving CNG message also refreshes the 'soft state' timer associated with the edge router that generated this message. The edge node information is removed from the Core Links table after a 'soft-state' timer expires. Since the aggregate RBR consists of only two values of type double, the 'soft state' of the edge routers is the per-edge node information maintained in each core router.

5.2 Optimizations of the RDF Protocol

During the path probing and notification phases, the aggregate RBR values of the core routers are distributed among the corresponding edge routers. That is why we call the RDF protocol the RBR Distribution and Feedback protocol: the edge nodes distribute the flow(s) RBR among the core routers during the update phase, while during the path probing and notification phases the core routers provide feedback to the edge routers by distributing the aggregate RBR values.

The RDF protocol provides a distributed solution to the problem of maintaining and sharing aggregate flow information among the routers in the network. More specifically, there is no central coordination node responsible for aggregating and distributing flow information. Each core router aggregates information only of those flows that travel through its links; as a result, each core router maintains and shares information about different flow aggregates. Furthermore, each edge router initiates its corresponding phase of RDF protocol independently of other routers in the network based on its specific internal events.

Although all the RDF protocol phases execute independently of each other, it is the case that execution of one phase triggers another. For example, the edge routers always probe the network before the update phase if characteristics of the path are unknown. Likewise, if flow activation causes congestion,

DOI: 10.1002/nem

then the notification phase is preceded by the update phase. Based on this observation, we implemented the following optimization of the RDF protocol. Upon activation of the new flow and before the update phase, the edge router retrieves information about the new flow's path to the destination, which includes capacity and the aggregate RBR of each link on the corresponding path. This information allows the edge router to determine whether admission of a new flow into the network will cause congestion. In the case of upcoming congestion, the edge routers notify corresponding core routers via CNG message of the update phase but delay admission of the new flow into the network. Subsequently, the core routers initiate notification phase and force the edge routers to throttle their flows before the new flow is admitted into the network. As a result, the new flow enters the network at about the same time when existing flows that share the bottleneck links with the new flow adjust their transmission rates. In effect, such optimization eliminates congestion occurrence upon flow activation [11–19].

The X-BDS approach is ideal for long heavyweight flows such as voice and video traffic or large data transfers. In the presence of a large number of short bursty flows, the update phase of the RDF protocol may generate a lot of control traffic because it initiates upon each flow activation or termination. To solve this problem, the edge nodes may combine requests of multiple short flows into a single update phase. Thus, a single CNG message may carry combined information (e.g., aggregate RBR) about multiple activations and terminations of the flows that travel on the same path. This optimization, called *the message aggregation technique* [20], requires the edge routers to provide another level of aggregation for the short and bursty traffic. However, it has no effect on the processing in the network core and operation of the RDF protocol. In the presence of short-lived and bursty traffic the message aggregation technique reduces the control load and processing overhead during the update phase [20].

6. EVALUATION OF THE X-BDS APPROACH

6.1 Simulation Set-Up

We study the performance of the X-BDS approach using the OPNET Network Simulator [21]. Figure 5 and Table 2 present the network topology and flow configuration used in our study. To simplify the notation, the flow that originates from source *i* is denoted as flow F*i*. For example, the flow that originates from Source 1 is denoted as F1 and the flow of Source 2 as F2. Additionally, links Core 2 to Core 5 and Core 5 to Core 3 are denoted as links C2–C5 and C5–C3, respectively. Each flow in this scenario is carrying multimedia traffic (e.g., video conferencing application) and is configured as shown in Table 2. Each

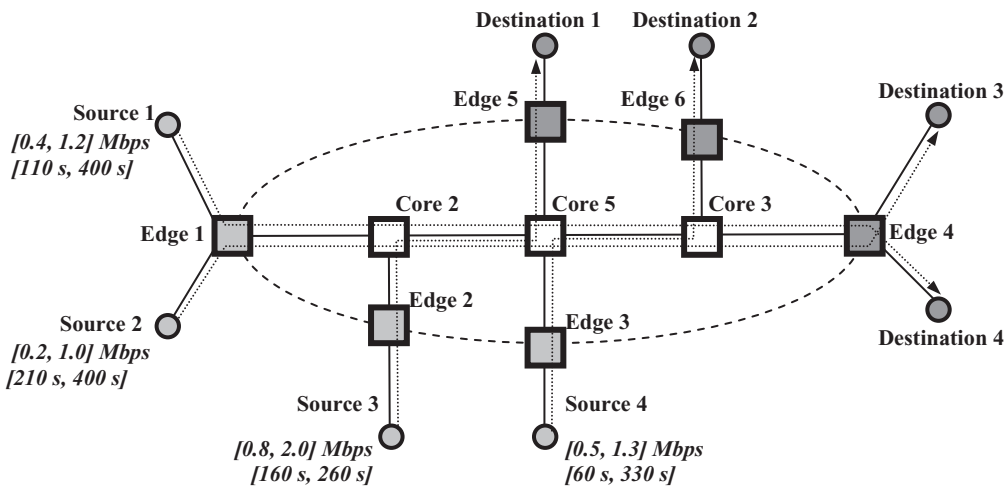


Figure 5. Simulation topology

DOI: 10.1002/nem

Flow Name	Activation Time (seconds)	Termination Time (seconds)	Flow RBR (Kbps)	Source	Destination
F1	110	400	[400, 1200]	1	4
F2	210	400	[200, 1000]	2	3
F3	160	260	[800, 2000]	3	1
F4	60	350	[500, 1300]	4	2
OPNET Application Configuration					
Application Type			Video Conferencing		
Frame Inter-arrival Time Information			30 frames/second		
Frame Size Information			352×240 pixels		
Type of Service			Standard X-BDS traffic		

Table 2. Flow configuration

Flow name	Optimal flow rates for PROPORTIONAL fairness (kbps)					
	[60, 110]	[110, 160]	[160, 210]	[210, 260]	[260, 330]	[330, 400]
F1	0	711.1	533.3	457.1	581.8	1066. 7
F2	0	0	0	228.6	290.9	533. 3
F3	0	0	1066.7	914.3	0	0
F4	1400	888.9	1066.7	914.3	727.3	0
Flow Name	Optimal flow rates for MAXIMIZING UTILITY fairness (kbps)					
	[60, 110]	[110, 160]	[160, 210]	[210, 260]	[260, 330]	[330, 400]
F1	0	750	560	457.1	566.67	900
F2	0	0	0	257.1	366.67	700
F3	0	0	1040	885.7	0	0
F4	1400	850	1040	885.7	666.67	0

Table 3. Optimal per-flow resource allocation for simulation scenario of Figure 5

link in the network is provisioned with 1600 kbps of bandwidth (e.g., link capacity is 2000 kbps; 80% of capacity is allocated for the X-BDS traffic). Duration of the simulation is 400 s.

The network topology used in our study is fairly simple, yet we believe it is sufficiently complex for evaluation of the Bandwidth Distribution Scheme's effectiveness. In this scenario there are four multimedia traffic flows: flows F1 and F2 travel on the main path from edge router 1 to edge router 4, while flows F3 and F4 create cross-traffic by each traversing a single yet different link on the path of flows F1 and F2. The schedule of flow activation and termination forces the bottleneck link of flows F1 and F2 to change from link C5–C3 to link C2–C5 and then back to C5–C3 throughout the simulation, which tests the ability of the X-BDS approach to deal with dynamic changes in the network.

The scenario of Figure 5 is divided into six independent time periods based on the flow activation and termination times. Based on this partitioning Table 3 shows the optimal bandwidth distribution using the proportional and maximizing utility definitions of fairness. Figure 6 illustrates bandwidth distribution among the flows using the maximizing utility definition of fairness. Per-flow resource distribution of Figure 6 was obtained through simulation and it very close to the optimal distribution of Table 3.

To better understand the nature of the X-BDS scheme, let us carefully examine what happens during the time periods [210, 260] and [260, 330] s. At time 210 s, flow F2 activates causing congestion on links C2–C5 and C5–C3. As a result, core router C2 initiates the notification phase requesting flows F1, F2, and F3 to adjust their allocated rates because link C2–C5 is their bottleneck. Meanwhile, core C5 initiates

DOI: 10.1002/nem

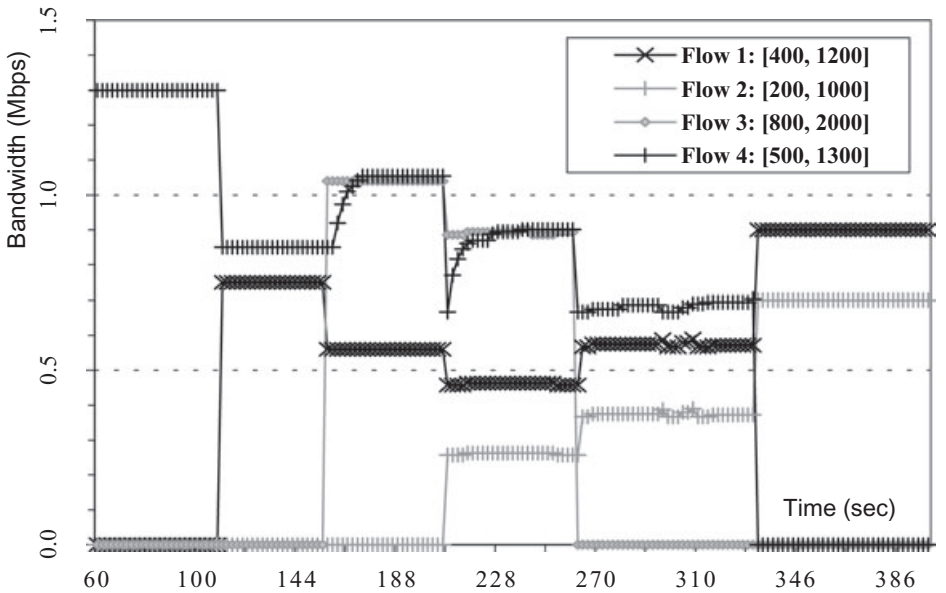


Figure 6. Example of resource distribution using the X-BDS approach

notification phase and notifies Edge 1, Edge 2, and Edge 3 to throttle their flows. However, only Edge 3 adjusts transmission rate of its flow, because link C5–C3 is a bottleneck only for flow F4. After the rate reduction, Edge 3 conducts the path probing, discovers that link C5–C3 is underutilized, and gradually increases transmission rate of F4 until all excess bandwidth is consumed.

At time 260 s, flow F3 terminates and link C5–C3 becomes a new bottleneck for flows F1, F2, and F4. Based on the path probing results, Edge 1 discovers that flows F1 and F2 send data at rates below their fair shares. Edge 1 increases allocated rates of F1 and F2, causing congestion on link C5–C3. Subsequently, F4 receives CN from C5 and reduces its sending rate.

6.2 Evaluation of the Resource Management Mechanism

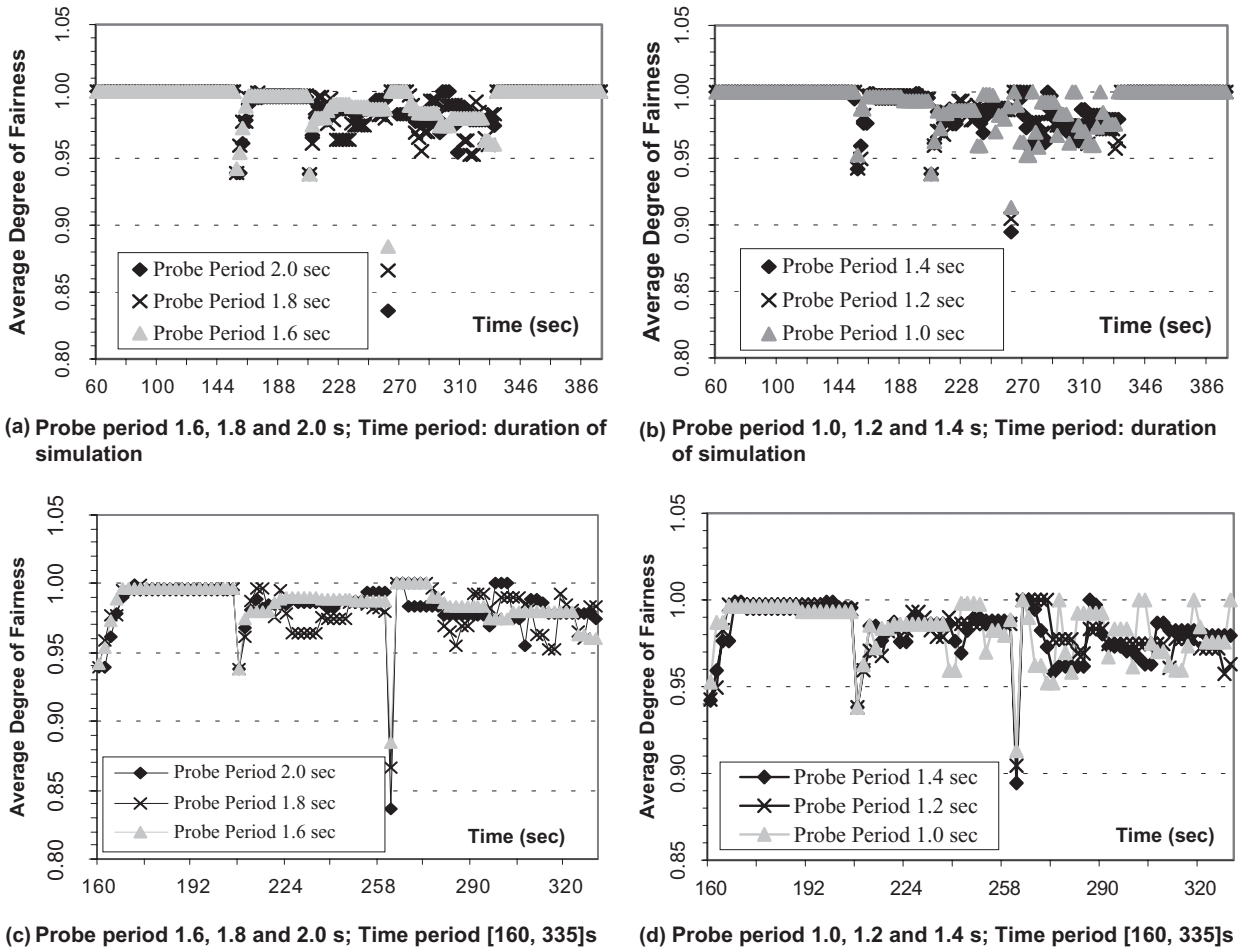
To evaluate the effectiveness of the resource management mechanism in achieving a fair distribution of the bandwidth, we define a metric called *degree of fairness*. The degree of fairness at time t for flow f is a measure of the ratio between the allocated rate $R^f(t)$ and the optimal allocated rate $R^f_{\mathrm{OPT}}(t)$:

$$\mathrm{DF}^f(t) = \begin{cases} 1 - \left| 1 - \dfrac{R^f(t)}{R^f_{\mathrm{OPT}}(t)} \right|, & \text{if} \quad \dfrac{R^f(t)}{R^f_{\mathrm{OPT}}(t)} < 2 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

The degree of fairness is a statistic whose value lies between 0 and 1 and which shows how fairly the resources were distributed at a particular point in time. High degree of fairness values (e.g., 0.95–1.00) correspond to a situation when resources are distributed fairly and the allocated rates of the flows are close to the corresponding optimal rates. Small degree of fairness values correspond to a situation when resources are not distributed fairly and the allocated rates of the flows diverge from the corresponding optimal rates. We often average the degrees of fairness of all the active flows during a particular time period and refer to the obtained value as the *average degree of fairness.*

We compared the average degree of fairness achieved during the simulation for different probe period values. We varied the probe period between 1.0 and 2.0 s and averaged the degree of fairness values over 2 s intervals. Figure 7 presents two sets of graphs: Figure 7(a) and 7(b) show achieved fairness through the whole simulation: Figure 7(c) and 7(d) zoom in on the time period [160, 335]s. The results presented in Figure 7 were collected using the maximizing definition of fairness. The proportional definition of fairness yields a similar outcome.

DOI: 10.1002/nem



(a) Probe period 1.6, 1.8 and 2.0 s; Time period: duration of simulation

(b) Probe period 1.0, 1.2 and 1.4 s; Time period: duration of simulation

(c) Probe period 1.6, 1.8 and 2.0 s; Time period [160, 335]s

(d) Probe period 1.0, 1.2 and 1.4 s; Time period [160, 335]s

Figure 7. Achieved average degree of fairness

As Figure 7 shows, throughout the duration of the experiment the X-BDS scheme achieves fair resource distribution: the values of the average degree of fairness vary between 0.95 and 1.00. Only in the events of flow activation or termination does the degree of fairness briefly drop below 0.95. More specifically, upon flow activation or termination, the average degree of fairness reaches lower values when the probe period value is large. For example, when flow F3 terminates at time 260 s the degree of fairness falls to 0.83 with the probe period of 2.0 s. However, when the probe period is 1.0 s the degree of fairness drops only to 0.91.

This happens because small values of the probe period cause the edge routers to probe the network more frequently and thus increase probability of discovering the path changes quickly. Meanwhile, the larger probe period values cause the edge routers not to discover the path changes for a longer period of time and thus cause the degree of fairness average to reach lower values. However, large values of the probe period cause the network feedback to provide a more accurate estimate of the excess bandwidth available on the path and thus results in less fluctuation of the flow rates. Our study suggests that the value of probe period between 1.2 and 1.4 s causes the resource distribution mechanism to perform reasonably well.

6.3 Link Utilization

To compare the link utilization achieved by the maximizing utility and proportional fairness criteria, the simulation scenario of Figure 5 was slightly modified. The RBR of flow F4 was changed from [500, 1300]

DOI: 10.1002/nem

to [400, 450] kbps. Figure 8 illustrates the achieved utilization on links C2–C5 and C5–C3 using both fairness criteria.

As Figure 8 shows, both fairness criteria are able to achieve close to 100% link utilization. However, upon activation of flow F1 at time 110 s and termination of flow F2 at time 260 s, proportional fairness is unable to distribute all available bandwidth among active flows at once and relies on periodic path probing to distribute leftover bandwidth. It can easily be shown that proportional fairness fails to utilize all available bandwidth when the fair share of the flow is larger than its maximum requested rate [10,11]:

$$C^k \frac{b^f}{b^k} > B^f \tag{15}$$

For example, at time 260 s the bandwidth of bottleneck link C5–C3 is distributed among flows F1, F2, and F4. Since the fair share of flow F4 is $1600 * 400/(400 + 200 + 400) = 640$ kbps, while its maximum requested rate is 450 kbps, 190 kbps of the bandwidth dedicated for F4 are left unused. Thus, using the proportional fairness criterion, the edge nodes often rely on periodic path probing to discover and utilize available bandwidth. On the other hand, the maximizing utility fairness does not suffer from this deficiency because by definition the fair share of the flow can be larger than the maximum requested rate only if maximum requested rate of all flows on the link is smaller than the link's capacity.

$$b^f + (C^k - b^k)\frac{B^f - b^f}{B^k - b^k} > B^f \Rightarrow C^k > B^k \tag{16}$$

Figure 9 zooms in on the time period [258, 268] s to better illustrate this phenomenon. As Figure 9 shows, the maximizing utility fairness allows the edge routers to allocate all of the available bandwidth by time 262 s. On the other hand, the resource distribution using proportional fairness allocates only 90% of avail-

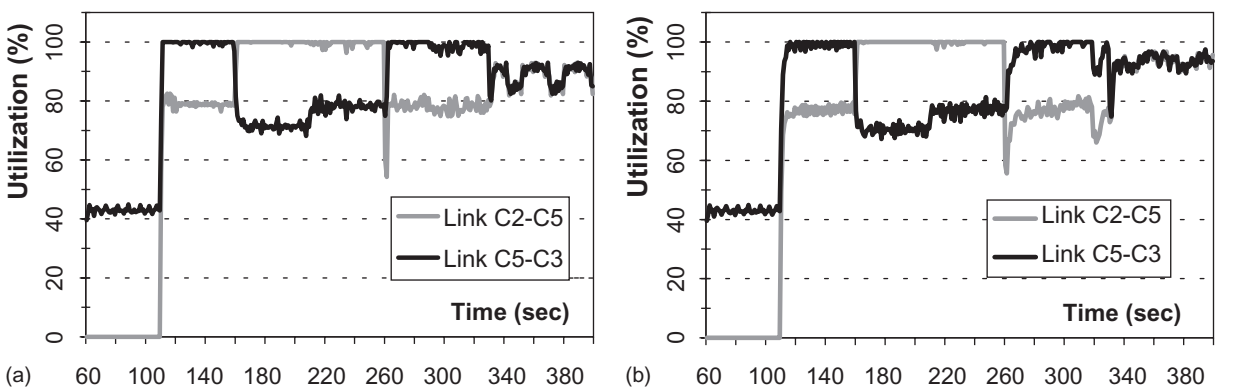


Figure 8. Utilization of links C2–C5 and C5–C3: (a) maximizing utility fairness; (b) proportional fairness

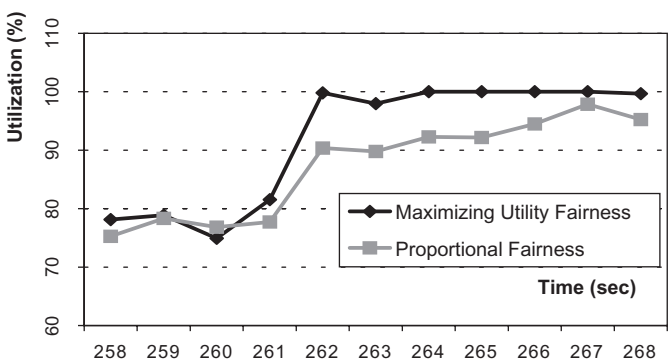


Figure 9. Achieved utilization of link C5–C3 during the time period [258, 268] s

DOI: 10.1002/nem

able bandwidth. The rest of the bandwidth is allotted within the span of the next 5–8 s based on feedback from the periodic path probes. In summary, the simulation results showed that the proportional fairness criterion in certain cases fails to distribute available bandwidth completely. However, in such instances the overall performance of the X-BDS does not suffer much because the periodic path probing notifies the edge nodes about excess bandwidth available. As a result, the edge routers fairly distribute remaining resources within a short period of time.

6.4 Control Load Overhead

To examine the overhead caused by the X-BDS approach we added 12 additional FTP sources that activate and terminate randomly throughout the simulation. Each FTP source is configured to generate upload requests of random size between 30 and 1500 kbytes. Even though each FTP source has only a single application active at any point of time, the applications are randomly restarted throughout the duration of the simulation. On average the number of active FTP applications varies between 4 and 10. To reduce the total number of X-BDS control messages generated in this scenario, we apply the X-BDS message aggregation optimization [20]. We run the simulation for 400 s.

We define the control load overhead caused by the RDF protocol as the ratio between the amount of the control data and the total amount of data generated in the system. The overhead of the protocol was computed in terms of the number of packets and the number of bits generated in the system. Figure 10 illustrates how the overhead of the RDF protocol varies with the change of the path probing period.

As expected, the overall overhead of the X-BDS diminishes as the probe period decreases. For example, control traffic incurs only 0.03% of the bits overhead, when the edge routers probe the network every 2.0 s. In the worst case studied, when the edge routers generate probe messages every 1.0, the RDF protocol incurs only 0.06% of the bits overhead. However, the overhead in terms of packets is an order of magnitude larger than the overhead in terms of bits. The X-BDS packet overhead ranges from 1.7% to 3.5%. Such behavior is expected, since the size of a control packet is significantly smaller than the average size of a data packet.

7. DISCUSSION AND RELATED WORK

Most of the current architectures that support QoS in the Internet have emerged from various proposals by the Internet Engineering Task Force (IETF). In 1994, the IETF introduced Integrated Services [22] architecture, followed by the Differentiated Services [2] model in 1998. Although both of these approaches address the same problem of supporting quality of service in the Internet, they are different in terms of implementation and provided services. Integrated Services provides end-to-end guarantees on a per-flow basis, while DiffServ attempts to provide end-to-end guarantees based on per-hop assurances for a small

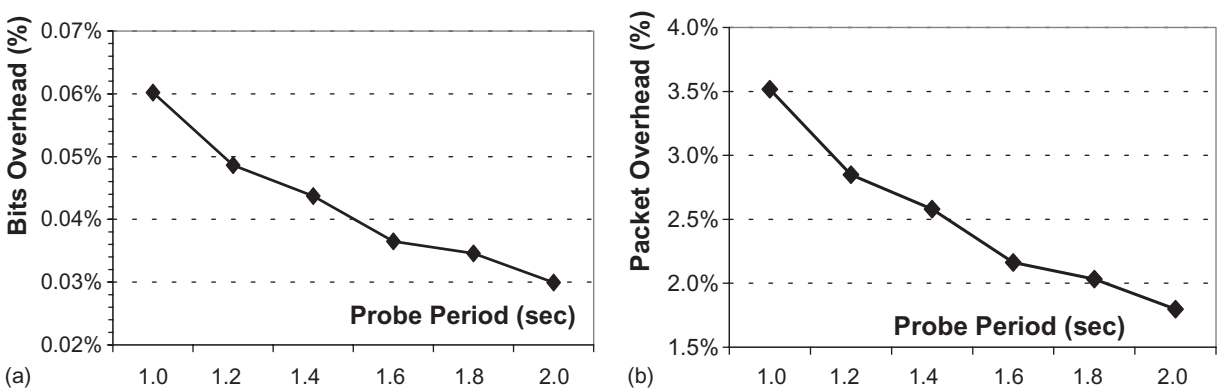


Figure 10. Control load overhead: (a) overhead in bits; (b) overhead in packets

DOI: 10.1002/nem

set of predefined traffic classes. At the implementation level, Integrated Services requires per-flow management in the network core, while the Differentiated Services model employs a network architecture that pushes per-flow management to the network edges.

This paper describes a novel approach called the Exact Bandwidth Distribution Scheme (X-BDS) for providing minimum per-flow bandwidth guarantees and for dynamic distribution of available bandwidth in a fair manner. Similarly to the Differentiated Services Architecture, the edge nodes in the X-BDS network manage per-flow information, while the core routers deal only with the traffic aggregates. The X-BDS scheme aggregates flow requirements in the network core and distributes these aggregated requirements in the network, so that they may be used for fair sharing of available bandwidth among individual flows. The X-BDS employs a combination of periodic probing and explicit network feedback for distributing the aggregate flow requirements in the network.

The idea of using explicit network feedback for dynamic rate and congestion control is not new. In particular, the Explicit Congestion Notification (ECN) extension to IP [23] uses binary feedback to notify ECN-capable transports about congestion occurrences. Unlike the ECN extension, the network feedback in the X-BDS model not only notifies the edge routers about congestion but also carries additional information such as the arrival rate and aggregate RBR on the congested link. A similar idea is used in ATM networks for Available Bit Rate (ABR) congestion control [24] where the feedback carried by the resource management cells also includes rate information. However, the ABR congestion control relies on per-flow information stored in the network core and tries to achieve utilization goals first and only then seeks fairness. In contrast, the X-BDS model does not store per-flow information in the network core. Instead, the core routers maintain a soft state for the edge routers that send traffic though the outgoing links of the core router and an aggregate RBB for those links. Furthermore, the X-BDS approach tries to achieve utilization and fairness goals simultaneously: the edge nodes compute the fair shares of individual nodes so as to consume all bandwidth allocated for X-BDS traffic, and in the presence of excess bandwidth individual flows increase their transmission rates so as to preserve fairness. The Explicit Control Protocol (XCP) [25] generalized the ECN proposal by sending additional information about congestion. XCP also does not require per-flow information in the network core. However, unlike X-BDS, XCP is not a rate-based but a window-based protocol that separates utility control from the fairness control.

The X-BDS approach is designed primarily for support of per-flow bandwidth guarantees. A similar feedback-based idea of providing dynamic per-flow bandwidth allocation for elastic traffic sources called simple rate control algorithm was introduced in reference 26. A traffic source is called elastic if it does not require a fixed rate and can adjust its transmission rate as needed. Unlike X-BDS, the boundary nodes in the simple rate control algorithm employ knowledge of the level of network congestion and the user utility functions to determine a fair resource distribution among elastic sources. End-users obtain the level of congestion through the explicit acknowledgements (ACK) that carry the number of congested links on a particular path.

The Stateless-Core approach [6] provides an interesting solution for supporting per-flow QoS without keeping per-flow information in the network core. The main idea of this scheme relies on the Dynamic Packet State (DPS), where control information is carried in the IP header of the data packets [27]. The routers use the DPS information to provide per-flow guarantees without maintaining per-flow state in the network core. The main differences between the X-BDS and Stateless-Core models are information storage in the network core, distribution of flow requirements, and resource allocation for individual flows. The Stateless-Core approach does not keep any additional information in the network core, while X-BDS requires each core router to maintain a 'soft-state' for the edge routers that sends traffic though its links. However, the DPS mechanism used by the Stateless-Core approach to carry flow requirements in the packet's header requires additional processing of each data packet in the network core. On the contrary, X-BDS employs a lightweight distributed message exchange protocol that carries control information separately from data and thus does not require additional per-packet processing. The RDF protocol used by X-BDS shares aggregate flow requirements among the nodes in the network which requires negligible amount of processing in the network core and, as shown in Control load overhead (Section 6.4), introduces a small amount of traffic due to exchange of control information.

DOI: 10.1002/nem

In the Stateless-Core approach, the core routers provide per-flow rate allocation via a FIFO queue with probabilistic drop, where the probability of dropping a packet is a function of the estimated rate carried in the packet's header and the fair share at that router which is estimated based on measurements of the aggregate traffic [6]. In this scheme, if a flow transmits at a rate higher than its fair share rate, then non-conforming packets of the flow will be dropped in the network core. By traveling through the network, instead of being dropped right away at the boundaries, these non-conforming packets will waste valuable network resources such as buffer space, bandwidth, etc. The X-BDS model adjusts transmission rates of the flows before they enter the network, which avoids this problem. However, adjustment of per-flow rates in the core has the advantage of not being subject to the propagation delay of the notification messages required in the X-BDS model. Another disadvantage of the Stateless-Core model is its inability to distribute excess bandwidth due to use of the upper bound of the aggregate reservation for admission control, which could possibly lead to network underutilization [6]. On the contrary, the X-BDS approach fairly distributes excess bandwidth and maximizes network throughput.

Similarly to the simple rate control algorithm [26], the X-BDS approach can be used to support bandwidth guarantees and is most suitable for elastic sources that can tolerate and benefit from frequent changes of the allocated rates. FTP and video flows are examples of such elastic traffic. The main advantage of the X-BDS approach is its ability to fairly distribute available resources among individual flows, while supporting minimum bandwidth guarantees of each flow and maintaining high link utilization. In addition, the RDF protocol of the X-BDS approach is fairly simple and does not incur significant load overhead.

8. CONCLUSIONS

This paper introduced a novel idea, called the Exact Bandwidth Distribution Scheme, for support of per-flow bandwidth guarantees. In the X-BDS approach, each flow that enters the network is guaranteed its minimum requested rate regardless of the network conditions. The leftover bandwidth is fairly distributed among the flows that can utilize it. The flows whose requests cannot be satisfied without violating guarantees of other active flows are not allowed to enter the network. Furthermore, in the X-BDS approach the active flows often degrade their resource consumption to accommodate the arrival of a new flow.

The X-BDS approach relies on a network architecture where the edge routers maintain per-flow requirements while the network core deals with the traffic aggregates only, which follows the Internet philosophy of keeping the network core simple and moving all processing complexity to the network edges. The Requested Bandwidth Range Distribution and Feedback protocol distributes aggregate flow requirements among the nodes in the network. The aggregate flow requirements provided by the RDF protocol support the admission control and resource management units of the edge routers. Admission control determines whether a new flow can be admitted into the network, while the resource management unit computes the bandwidth fair shares for individual flows.

Evaluation of the X-BDS approach shows that the X-BDS is capable of supporting fair per-flow distribution of available bandwidth, which could be used for building services with per-flow bandwidth guarantees. Furthermore, the X-BDS supports high link utilization, allows the new flows to enter the network without causing congestion, eliminates congestion fast, and causes negligible overhead in the network. Even though the X-BDS is primarily suited for large elastic flows, in reference 20 we examined an optimization that allows the X-BDS to perform well in the presence of large number of small flows. Currently, we are investigating a number of additional mechanisms for improving the X-BDS approach. We also are examining the possibility of extending the X-BDS approach to a multi-domain environment, applying the X-BDS idea to traffic aggregates in DiffServ [2] and MPLS [7] networks, defining requests with different traffic priority and utility functions, examining stability issues of the X-BDS approach, and the possibility of introducing the X-BDS approach into a mobile environment.

DOI: 10.1002/nem

REFERENCES

1. Hnatyshin V, Sethi A. Estimation based load distribution in the Internet. *Computer Networks* 2005; **48**(4): 525–554.
2. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. *IETF RFC 2475*, December 1998.
3. Clark D, Fang W. Explicit allocation of best effort packet delivery service. *IEEE/ACM Transactions on Networking* 1998; **6**(4): 362–373.
4. Feng W, Kandlur D, Saha D, Shin K. Understanding and improving TCP performance over networks with minimum rate guarantees. *IEEE/ACM Transactions on Networking* 1999; **7**(2): 173–187.
5. Kumar V, Lakshman T, Stiliadis D. Beyond best effort: router architecture for the differentiated services of tomorrow's Internet. *IEEE Communications Magazine* 1998; May: 152–164.
6. Stoica I. Stateless core: a scalable approach for quality of service in the Internet. PhD thesis, Carnegie Mellon University, 2000.
7. Rosen E, Viswanathan A, Callon R. Multiprotocol label switching architecture. *IETF RFC 3031*, January 2001.
8. Dharwadkar P, Siegel H, Chong E. A heuristic for dynamic bandwidth allocation with preemption and degradation for prioritized requests. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, April 2001; 547.
9. Banchs A. User fair queuing: fair allocation of bandwidth for users. In *Proceedings of IEEE INFOCOM'02*, June 2002.
10. Hnatyshin V, Sethi A. Avoiding congestion through dynamic load control. In *Proceedings of ITCom-2001, SPIE's International Symposium on the Convergence of Information Technologies and Communications*, Denver, CO, August 2001; 309–323.
11. Hnatyshin V. Dynamic bandwidth distribution techniques for scalable per-flow QoS. PhD thesis, University of Delaware, 2003.
12. Vandalore B, Fahmy S, Jain R, Goyal R, Goyal M. General weighted fairness and its support in explicit rate switch algorithms. *Computer Communications* 2000; **23**(2): 149–161.
13. Breslau L, Jamin S, Shenker S. Comments on the performance of measurement-based admission control algorithms. In *Proceedings of IEEE INFOCOM'00*, March 2000.
14. Gibbens R, Kelly E. Measurement-based connection admission control. In *Proceedings of the 15th International Teletraffic Congress*, Amsterdam, Netherlands, June 1997; 781–790.
15. Kelly F, Key P, Zachary S. Distributed admission control. *IEEE Journal on Selected Areas in Communications* 2000; **18**: 2617–2628.
16. Marbach P. Priority service and max–min fairness. In *Proceedings of IEEE INFOCOM'02*, June 2002.
17. Radunovic B, Le Boudec J. Unified framework for max–min and min–max fairness with applications. *Technical Report IC-200248*, EPFL, July 2002.
18. Tzeng H, Sui K. On max-min fair congestion control for multicast ABR service in ATM. *IEEE Journal on Selected Areas in Communications* 1997; **15**(3): 545–556.
19. Hnatyshin V, Sethi A. Fair and scalable load distribution in the Internet. In *Proceedings of the International Conference on Internet Computing*, June 2002: 201–209.
20. Hnatyshin V, Sethi A. Reducing load distribution overhead with message aggregation. In *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference*, April 2003; 227–234.
21. *OPNET Modeler*. OPNET Technologies Inc. http://www.opnet.com
22. Braden R, Clark D, Shenker S. Integrated services in the Internet architecture: an overview. *IETF RFC 1633*, June 1994.
23. Ramakrishnan K, Floyd S, Black D. The addition of explicit congestion notification (ECN) to IP. *IETF RFC 3168*, September 2001.
24. Kalyanaraman S, Jain R, Fahmy S, Goyal R, Vandalore B. The ERICA switch algorithm for ABR traffic management in ATM networks. *IEEE/ACM Transactions on Networking* 2000; **8**(1): 87–98.
25. Katabi D, Handley M, Rohrs C. Internet congestion control for future high bandwidth-delay product environments. In *Proceedings of ACM SIGCOMM'02*, Pittsburgh, PA, August 2002.
26. Kar K, Sarkar S, Tassiulas L. A simple rate control algorithm for maximizing total user utility. In *Proceedings of INFOCOM'01*, April 2001; 133–141.
27. Stoica I, Zhang H. Providing guaranteed services without per-flow management. In *Proceedings of ACM SIGCOMM*, September 1999.

DOI: 10.1002/nem

AUTHOR'S BIOGRAPHIES

Vasil Y. Hnatyshin is an Assistant Professor in the Computer Science Department at Rowan University, Glassboro, New Jersey, USA. He started his education at L'viv State University, L'viv, Ukraine. He received his B.S. (summa cum laude) from Widener University, Pennsylvania, and M.S. and Ph.D. in Computer Science from University of Delaware, Delaware. Dr. Hnatyshin's research interests include service differentiation, resource management, and quality-of-service in IP networks, as well as network management, BGP and inter-domain routing, mobile ad hoc networks, and wireless technologies.

Adarshpal S. Sethi is a Professor in the Department of Computer & Information Sciences at the University of Delaware, Newark, Delaware, USA. He has an MS in Electrical Engineering and a PhD in Computer Science, both from the Indian Institute of Technology, Kanpur, India. He has served on the faculty at IIT Kanpur, was a visiting faculty at Washington State University, Pullman, WA, and Visiting Scientist at IBM Research Laboratories, Zurich, Switzerland, and at the US Army Research Laboratory, Aberdeen, MD. Dr. Sethi is on the Editorial Advisory Board for the *Journal of Network and Systems Management*, and on the editorial boards of *eTNSM* (*IEEE electronic Transactions on Network and Service Management*), *International Journal of Network Management*, and *Electronic Commerce Research Journal*. He was co-Chair of the Program Committee for ISINM '95, and was General and Program Chair for DSOM '98; he is also active on the program committees of numerous conferences. Dr. Sethi's research interests include architectures and protocols for network management, fault management, quality-of-service and resource management, and management of wireless networks.

DOI: 10.1002/nem