In Proc. IPCCC-2004, 23rd IEEE International Performance, Computing, and Communications Conference, Phoenix, AZ (April 2004), pp. 215-222.

Optimization of the Bandwidth Distribution Scheme for Handling Topology Changes

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

[1]Department of Computer Science
Rowan University
201 Mullica Hill Rd.
Glassboro, NJ 08028
hnatyshin@rowan.edu

[2]Department of Computer and
Information Sciences,
University of Delaware,
Newark, DE 19716
sethi@cis.udel.edu

Abstract

The Bandwidth Distribution Scheme (BDS) [Hna03] was designed to combine the advantages of the Integrated and Differentiated Services models and to provide support for scalable per-flow Quality of Service. In recent studies Hnatyshin et al. showed that the variation of the Bandwidth Distribution Scheme called the Exact Requested Bandwidth Range BDS (X-BDS), can support per-flow minimum bandwidth guarantees in a scalable manner [HS03, Hna03]. The X-BDS achieves per-flow QoS by maintaining aggregate flow requirements in the network core and distributing these requirements as needed. Based on the obtained information the edge nodes determine the fair allocation of available bandwidth among the active flows. This paper introduces an optimization which allows the X-BDS approach to operate seamlessly in the event of network topology changes. The primary challenge of addressing this problem is determining how to correctly update the aggregate flow requirements maintained in the network core. This paper examines two instances of topology changes, link failure and link restoration, and presents an algorithm which enables the routers in the X-BDS network to properly update the aggregate flow requirements in each of these cases. This paper examines the performance of the introduced algorithm through simulation [Opn].

1. Introduction

The problem of providing per-flow Quality of Service in a scalable manner still remains one of the most difficult and widely studied problems in the research community. Recently, Hnatyshin et al introduced a new approach called the **B**andwidth **D**istribution **S**cheme (BDS) that provides a framework for building per-flow services in a scalable manner. In particular, [Hna03, HS03, HS02] showed that the variation of the BDS model, called the Exact Requested Bandwidth Range BDS or X-BDS, is capable of supporting minimum bandwidth guarantees on a per-flow basis. This paper examines an optimization of the X-BDS approach that allows support of per-flow minimum bandwidth guarantees in the event of topology changes.

In the Bandwidth Distribution Scheme the edge routers rely on network feedback to discover network characteristics and then use obtained information to adjust bandwidth allocation of individual flows. Generally, the BDS architecture consists of three main components: the admission control unit, the resource management mechanism, and the **R**equested Bandwidth Range **D**istribution and **F**eedback (RDF) protocol. The admission control unit determines if a new flow can be admitted into the network, while the resource management mechanism computes the fair shares of individual flows and allocates bandwidth according to these computations. The RDF protocol is the glue that holds the BDS architecture together. It provides feedback to the edge routers about network changes. Specifically, in the X-BDS approach, the core routers maintain the aggregate flow requirements and the RDF protocol updates and distributes these requirements among the network nodes whenever the flow of traffic through the network changes. The admission control and resource management units of the X-BDS cannot operate without the aggregate flow requirements. That is why correct update of the aggregate flow requirements is vital to the Bandwidth Distribution Scheme.

This paper examines modification to the RDF protocol of the X-BDS approach, which allows the edge routers to provide fair resource distribution in the event of topology changes. Generally, the primary causes of topology changes are either failures of existing link/router or additions of new links/routers. In a mobile environment, node movement usually is the primary cause of topology changes. Although this paper examines modification of the RDF protocol in the events of link failure or link restoration only, we believe that similar logic is applicable to more complex cases such as router mobility. The proposed modification to the RDF protocol is thus the first step toward extending the BDS architecture to a mobile environment.

The rest of the paper is organized as follows. Section 2 briefly introduces the BDS architecture and Section 3 provides an overview of the problem. Section 4 describes the proposed modification of the RDF protocol, while Section 5 evaluates it via a simulation study. Section 6 presents discussion and related work overview and finally, we conclude in Section 7.

2. Architecture of the Bandwidth Distribution Scheme

The architecture of the Bandwidth Distribution Scheme consists of three main components: the admission control unit, the resource management mechanism, and the RDF protocol. First we examine the admission control unit and flow requirements. We assume that both the minimum and the maximum transmission rates of a flow are known ahead of time. Thus, 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.

$$RBR^f = [b^f, B^f] \qquad (1)$$

Throughout this paper we often refer to the aggregate flow requirements or the *aggregate RBR*, which is the sum of the RBRs of those flows that travel through a particular link. Thus, the aggregate RBR on link i is the sum of the RBRs of those flows that travel through link i:

$$RBR_i = [b_i, B_i] = \left[\sum_{f \to i} b^f, \sum_{f \to i} B^f \right] \qquad (2)$$

In addition, we define the *edge RBR* on link i, $RBR_{e,i}$, as the sum of the flow RBRs of those flows that enter the network at edge router e and travel through link i. Based on these definitions, the 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. To achieve these guarantees the admission control unit denies network access to those flows whose minimum rate guarantees cannot be met without violating existing flow guarantees, while the resource management unit distributes available bandwidth. In particular, the resource management mechanism allocates minimum requested rate to those flows that are allowed to enter the network and then fairly distributes the excess or leftover bandwidth among individual flows. Please refer to [Hna03] for definitions of fairness used in the BDS architecture.

The RDF protocol supports seamless operation of the admission control and resource management mechanisms by providing network feedback. The RBR Distribution and Feedback protocol consists of three distinct and independent phases or parts: the path probing phase, the RBR update phase, and the notification phase. The path probing phase discovers characteristics of a particular path and is periodically executed by the edge routers. The edge routers maintain the results of the path probing in the Path and Link Tables, where characteristics of individual links on each active path are being stored. The edge routers initiate the RBR update phase to notify the core routers about the changes to the aggregate RBR information due to flow activation or termination. During the RBR update phase, the edge nodes generate control messages on the corresponding paths, and the core routers use information carried in these control messages to update the aggregate flow requirements (e.g. aggregate RBR and edge RBR) maintained in the Interfaces Table. Only in the event of congestion do the core routers initiate the notification phase. In this case, the core routers generate congestion notification messages to the edge routers asking them to adjust allocated rates of their flows. Upon the congestion notification message arrival the edge routers retrieve necessary information from their Path and Link tables, re-compute the fair shares of individual flows, and adjust the per-flow bandwidth allocations accordingly. Additional information about the BDS architecture could be found in [Hna03].

3. Overview of the problem

In the event of link/router failure or restoration, the primary concern of the X-BDS approach is how to correctly update the aggregate flow requirements stored in the network. The main problem arises when due to the topology changes, the edge routers start sending their traffic over new paths. In this case, to update the aggregate flow requirements, the resource reservations of the flows influenced by these topology changes should be removed from their old paths and established on the corresponding new routes.

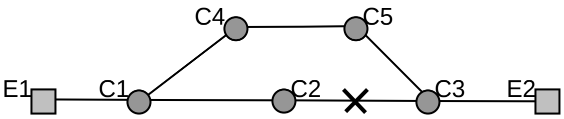


Figure 1. Example of the link failure

To better understand this problem let us consider the example of Figure 1. Let us assume that link C2–C3 fails when traffic from E1 follows path E1–C1–C2–C3–E2 through the network. After discovering link failure, core routers C2 and C3 notify corresponding routers (e.g. C1 and E2) about this event. Subsequently, core routers C1, C2, and C3 update their aggregate RBR values by subtracting the aggregate RBR of the flows that travel through link C2-C3. When edge router E1 discovers the link failure it consults the underlying routing protocol to identify a new route (e.g. E1-C1-C4-C5-C3-E2) for the flows influenced by the failure of link C2-C3. Then, E1 establishes a reservation over the new path by initiating the RBR update phase. Finally, E1 forwards the flows influenced by the link failure over the new path.

Now let us examine what happens when link C2–C3 is restored. We assume that the underlying routing protocol notifies the X-BDS process of edge router E1. Then, E1 removes the resource reservation on path E1–C1–C4–C5–C3–E2, establishes a new reservation over path E1–C1–C2–C3–E2, and sends the flows over the new path. Overall, the primary responsibility of the RDF protocol in the event of topology changes entails removing reservations over inefficient or broken paths, establishing new reservations, and forwarding influenced traffic over the new paths.

4. Modification of the RDF Protocol

This section examines the following two types of topology changes: link failure and restoration of a broken link. In the case of link failure, the RDF protocol should update aggregate RBR values in the routers influenced by the link failure, notify the edge routers about these topology changes, and establish the resource reservations over new paths.

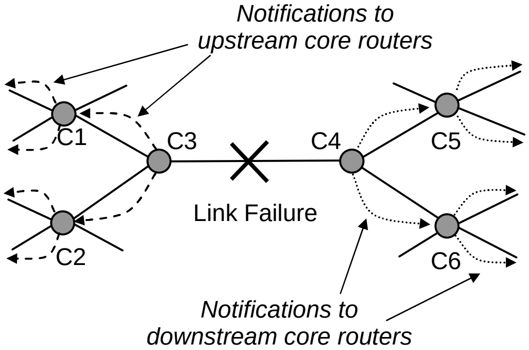

Notifications to upstream core routers
C1
C2
C3
C4
C5
C6
Link Failure
Notifications to downstream core routers

Figure 2. Example of topology changes

If restoration of a broken link causes more efficient paths to appear, then the RDF protocol should remove existing reservations on the old paths and establish new reservations over more efficient routes. We assume that the underlying protocol (e.g., routing) discovers topology changes and propagates corresponding information up to the RDF protocol of the X-BDS process.

Let us consider the example of Figure 2 where link C3-C4 fails and core routers C3 and C4 notify other routers in the network about this failure. Specifically, core routers C3 and C4 distribute the aggregate RBR and the edge RBRs of the flows that traveled through the failed link among their respective upstream and downstream routers. Subsequently, notified core routers update their aggregate flow requirements and forward received information further.

To update the aggregate RBR in the network, the core routers maintain the following information for each of their incoming and outgoing links in the Interfaces Table:

- The aggregate RBR
- Identities (e.g. IP addresses) of the edge nodes that send traffic through the corresponding link
- The edge RBR of the edge nodes that send traffic through the corresponding link

The information maintained in the Interfaces Table is obtained from the control messages forwarded by the edge routers during the RBR update phase.

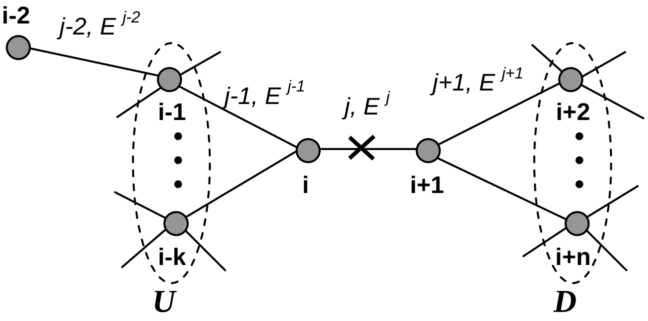


Figure 3. Generalized example of the link failure

Now, let us define the process executed by the edge and core routers in the event of link failure more precisely. First, let us examine the part of the algorithm that updates the aggregate flow requirements of the upstream core routers and notifies the edge routers about the link failure. Let us assume that core router i discovered that its outgoing link j failed. This situation is shown in Figure 3. In this case core router i notifies upstream routers and updates its Interfaces Table. Let set U represent all upstream core routers directly connected to i, set E^{j-1} represent a list of the edge routers whose traffic travels through link $j-1$ between core routers $i-1$ and i, and set E^{j} represent the set of all edge routers whose traffic traveled through failed link j. Then, core router i generates a control message, called the FAIL_UP message, to core router $i-1$ only if set $E^{FAIL\_UP} = E^{j-1} \cap E^{j}$ is not empty. More specifically, core router $i-1$ receives a FAIL_UP message only if there is at least one edge router whose traffic travels through $j-1$ and failed link j.

The FAIL_UP message contains the identity of the failed link and the list of the edge routers $e \in E^{FAIL\_UP}$ with their corresponding edge RBRs on the failed link j. This information is readily available in the Interfaces Table of core router i. After departure of the FAIL_UP messages, core router i updates its Interfaces Table.

Upon the FAIL_UP message arrival, core router $i-1$ performs a similar process. First, router $i-1$ updates its Interfaces Table and then notifies corresponding upstream routers. Similarly, core router $i-1$ generates a control message to core router $i-2$ if set $E^{j-2} \cap E^{FAIL\_UP}$ is not empty. For example, core router $i-2$ is notified about link failure only if there is at least one edge router that sends traffic through links $j-2$, $j-1$, and j (e.g. $e \in E^{j-2} \cap \left(E^{j-1} \cap E^{j}\right)$).

This process continues until the FAIL_UP message arrives at the edge router e. At this point, $E^{FAIL\_UP} = \{e\}$ and the FAIL_UP message contains only the identity of the failed link j, identify of core router e, and the edge RBR of e on j. Subsequently, e discovers an alternative path, initiates the RBR update phase, and re-routes the flows influenced by failure of link j over a new path.

Now let us examine the actions of downstream core router $i+1$ that discovers the failure of its incoming link j. Core router $i+1$ deals with the failure of link j almost the same way as core router i: $i+1$ notifies its downstream routers and updates the Interfaces Table. Let us assume that set D represents all downstream core routers directly connected to $i+1$ and set E^{j+1} represents the list of the edge routers whose traffic travels through link $j+1$ that connects downstream core router $i+2 \in D$ with router $i+1$. Then, core router $i+1$ generates a control message, called the FAIL_DOWN message, to downstream core router $i+2$ only if set $E^{FAIL\_DOWN} = E^{j+1} \cap E^{j}$ is not empty.

The FAIL_DOWN message contains the identity of the failed link and the list of the edge routers $e \in E^{FAIL\_DOWN}$ with their edge RBRs on the corresponding outgoing link. For example, the FAIL_DOWN message sent to core router $i+2$ contains the identity of the failed link and the list of the edge routers, whose traffic travels through the failed link, and their edge RBRs on link $j+1$ between core routers $i+1$ and $i+2$. After the FAIL_DOWN messages departure, core router $i+1$ updates its Interfaces Table.

The FAIL_DOWN message contains the edge RBRs of the edge routers on the outgoing link (e.g. $j+1$) of the corresponding router instead of their edge RBR on the failed link (e.g. j), because the traffic that originates from the same edge router and arrives on the same link (e.g., the failed link) may depart from the core router through multiple links. Upon the FAIL_DOWN message arrival, core router $i+2$ identifies a set of downstream core routers that should be notified, sends FAIL_DOWN messages to them, and updates its Interfaces Table. The FAIL_DOWN message terminates its progress when it arrives at an egress router.

Dealing with the event when a broken link comes back up is simpler. We assume that the underlying routing protocol discovers topology changes, finds a new path, and notifies the X-BDS process at the edge router. Subsequently, the edge router identifies the flows that can benefit from a new path, discovers characteristics of that path, and finally, initiates the RBR update phases to update the aggregate flow requirements on the influenced paths.

In summary, in the event of a link failure the RDF protocol executes the following actions:

- The core router downstream of the failed link generates FAIL_DOWN messages to corresponding downstream routers to update the aggregate flow requirements on the downstream portion of the path.
- The core router upstream of the failed link generates FAIL_UP messages to corresponding upstream routers to update the aggregate flow requirements on the upstream portion of the path.
- The edge routers that receive the FAIL_UP message, identify the new paths for the flows influenced by the link failure, initiate the RBR update phase on these paths, and re-route the flows over their corresponding new paths.

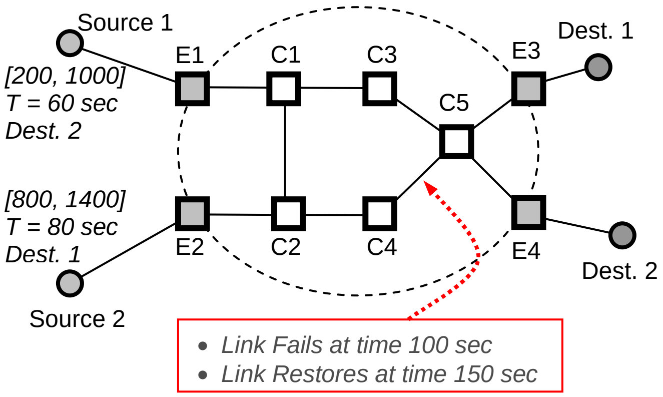
To handle the event of a broken link coming back up or discovery of a more efficient path the RDF protocol performs the following actions.

- The edge router advertises the RBR changes over an old path, removing existing reservations of those flows that can benefit from a more efficient new path.
- The edge router initiates the RBR update phase over a new path (e.g. increase the aggregate flow requirements on a new path) and re-routes corresponding flows over it.

5. Performance Evaluation

We studied the modification of the RDF protocol that deals with the topology changes through simulation. The goal of the simulation was to show that in the event of a topology change the introduced modification to the RDF protocol updates the aggregate flow requirements in the network correctly and allows the X-BDS approach to fairly distribute available bandwidth among active flows. Our experiments were conducted using the OPNET network simulator [Opn]. Figure 4 shows the network topology of the first experiment.

In this experiment two video flows activate at times 60 and 80 seconds and enter the network through edge routers E1 and E2, respectively. We denote a flow that originates from Source 1 as F1 and the flow that originates from Source 2 as F2. Flow F1 travels to Destination 2 over path E1–C1–C3–C5–E4 and has RBR [200, 1000] Kbps, while F2 travels to Destination 1 over path E2–C2–C4–C5–E3 and has RBR [800, 1400] Kbps.



- Link Fails at time 100 sec
- Link Restores at time 150 sec

Figure 4. First Experiment: simulation topology

Each link in the network is provisioned with 1800 Kbps, e.g., link capacity is 3000 Kbps and 60% of this capacity is allocated for the X-BDS traffic. At time 100 seconds, link C4–C5 fails and flow F2 starts traveling on new path E2–C2–C1–C3–C5–E3 causing congestion in the network. At time 150 seconds, link C4–C5 comes back up and flow F2 is re-routed back over path E2–C2–C4–C5–E3. The simulation was executed for 200 seconds and used the Routing Information Protocol (RIP) implemented within the OPNET network simulator for the routing table updates. Figure 5 shows resource distribution among the flows and Figure 6 illustrates utilization of the resources allocated for the X-BDS traffic on links C1–C3 and C2–C4. Links C1–C3 and C2–C4 were examined because the network conditions on these links reflect the overall situation on the paths of flows F1 and F2, respectively.

As Figure 5 shows, flows F1 and F2 activate at times 60 and 80 seconds respectively, and since they are the only active flows on their paths, they transmit traffic at their maximum rates of 1000 Kbps and 1400 Kbps, respectively. Flows F1 and F2 continue transmitting at their maximum rates until time 100 seconds, when link C4-C5 fails. As shown in Figure 6, during the time period between the flow activations and failure of link C4–C5, links C1–C3 and C2–C4 were utilized at 56% and 78% respectively.

After link C4–C5 fails, edge router E2 discovers a new path and re-routes flow F2. At time 100 seconds, F2 starts traveling over new path E2–C2–C1–C3–C5–E3 causing congestion in the network. As a result, core router C3 generates congestion notification messages to edge routers E1 and E2. Subsequently, E1 and E2 reduce allocated rates of their respective flows F1 and F2 eliminating congestion in the network.

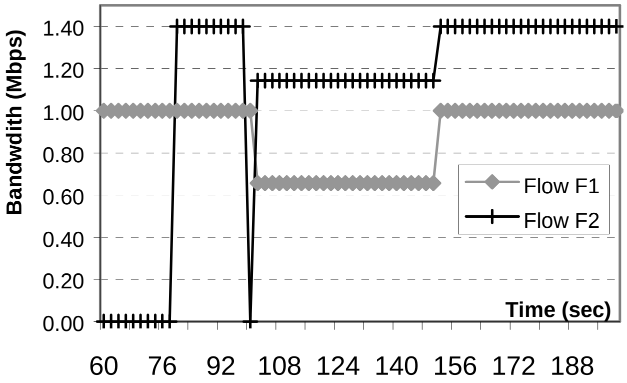


Figure 5. First Experiment: resource distribution

As Figure 6 shows, after link failure, flow F2 travels on a new path through link C1–C3 and not via link C2–C4, which causes utilization of links C1–C3 and C2–C4 to change to 100% and 0% respectively. As Figure 5 shows, during the time period [100, 150] seconds, when flows F1 and F2 share available resources on link C1–C3, F1 and F2 are allocated 657 and 1147 Kbps of bandwidth, respectively, which is a fair distribution of the link bandwidth based on the flow requirements.

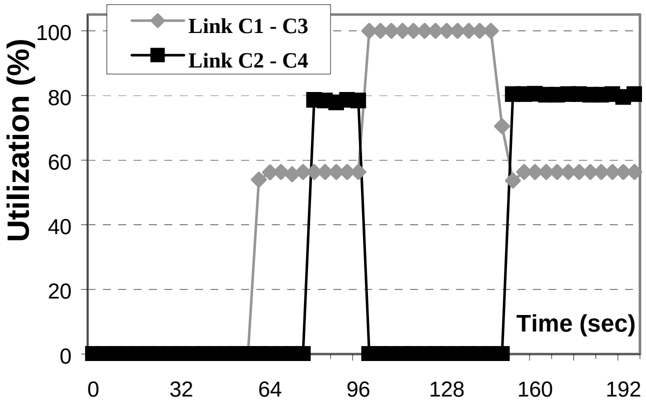


Figure 6. First Experiment: utilization of resources allocated for the X-BDS traffic

Upon link failure at time 100 seconds, core routers C4 and C5 update the aggregate RBR in the network. In particular, C4 generates a FAIL_UP message that travels to edge router E2 and removes resource reservations of the traffic that traveled through link C4–C5 (e.g., flow F2). At the same time, core router C5 generates a FAIL_DOWN message to egress router E3 removing resource reservation of flow F2. After edge router E2 receives the FAIL_UP message, it probes a newly discovered path and then initiates the RBR update phase. The flows that traveled through the failed link are suspended until characteristics of a new route are discovered. This event is illustrated in Figure 5, which shows that at time 100 seconds the allocated rate of flow F2 is 0 Kbps.

Finally, when at time 150 seconds link C4–C5 is restored, edge router E2 re-routes its flow F2 over new path E2–C2–C4–C5–E3. First, E2 probes the new path to discover its characteristics, then E2 initiates two RBR update phases one of which removes resource reservation of flow F2 on old path E2–C2–C1–C3–C5–E3, while the other adds resource reservation of flow F2 on new path E2–C2–C4–C5–E3. As Figures 5 and 6 show, after link C4–C5 comes back up, the network starts to operate the same way as before the link failure. Once again, flows F1 and F2 transmit data at their maximum sending rates of 1000 Kbps and 1400 Kbps, respectively, while links C1–C3 and C2–C4 are utilized 56% and 78%.

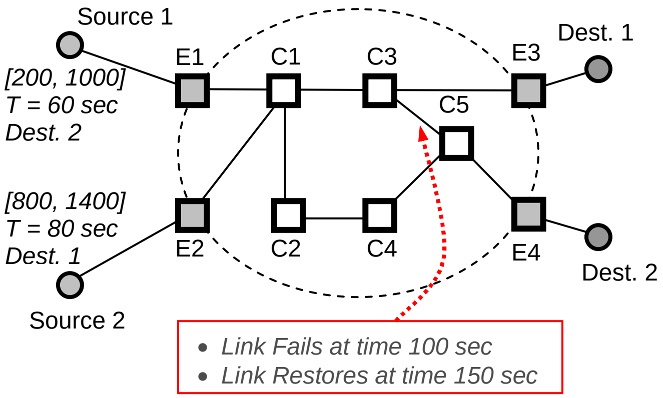


Figure 7. Second Experiment: simulation topology

We conducted the second experiment for the same network set-up but using a different network topology. Figure 7 shows network topology and simulation set-up for the second experiment. In this experiment link C3–C5 fails and gets back up at times 100 and 150 seconds, respectively. We present results of the second experiment in Figures 8 and 9. Figure 8 shows achieved resource distribution, while Figure 9 shows utilization of the X-BDS resources on links C1–C3 and C2–C4.

As before, in this simulation the routers rely on the RIP routing protocol, which uses "the-shortest-path-first" approach for routing the traffic. That is why at the beginning of the simulation flow F2 follows a shorter path (E2–C1–C3–C5–E4) even though traveling over a longer route (E2–C1–C2–C4–C5–E4) would have resulted in more resources allocated for F2.

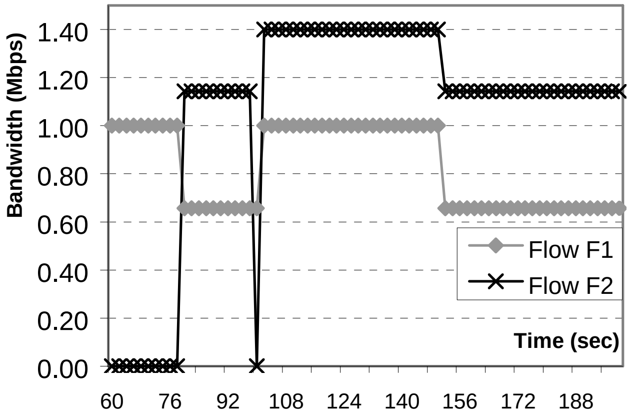


Figure 8. Second experiment: resource distribution

Before the failure of link C3–C5, both flows F1 and F2 travel through link C1–C3 and thus share available resources among them. Thus, as Figures 8 and 9 show, during this time period flows F1 and F2 transmit data at rates 567 Kbps and 1143 Kbps, respectively, while links C1–C3 and C2–C4 are utilized 100% and 0 %, respectively.

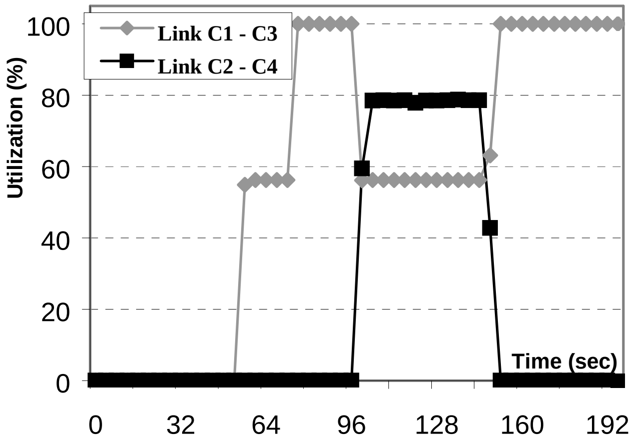


Figure 9. Second Experiment: utilization of the resources allocated for the X-BDS traffic

When link C3–C5 fails, core router C3 generates the FAIL_UP message to remove resource reservations of the flows that traveled through the failed link. When edge router E2 receives the FAIL_UP message, it identifies F@ as the flow that traveled through the failed link. Then, E2 consults the routing table and retrieves an alternative path to Destination 2. Subsequently, E2 probes new path E2–C1–C2-C4–C5–E4 and re-routes flow F2 over it. As before, F2 is suspended until the characteristics of a new path are discovered. At the same time, core router C5 generates the FAIL_DOWN message to egress E4 updating the aggregate flow requirements on the downstream portion of the path.

As Figure 8 shows, after link failure, flow F2 is suspended for a short time and does not transmit any traffic. However, once the characteristics of the new path are discovered, E2 allows flow F2 to travel at its maximum allocated rate because F2 is the only active flow on the new path. Similarly, once E1 discovers that the aggregate RBR on the path to Destination 1 has been reduced, E1 increases allocated rate of flow F1 to its maximum rate. As Figure 9 shows, at this point, link C1–C3 becomes underutilized (e.g., link utilization drops to 56%), while utilization of link C2–C4 increases to 78%.

Finally, at time 150 seconds link C3–C5 comes back up and the routing protocol notifies edge router E2 about this event. As a result, E2 probes a new path, removes the resource reservation of flow F2 from old path E2–C1–C2–C4–C5–E4, and establishes a new reservation for flow F2 over new path E2–C1–C3–C5–E4. Subsequent re-routing of flow F2 causes congestion on link C1–C3, and as a result, core router C3 initiates the notification phase sending CN messages to edge routers E1 and E2. After the edge routers receive congestion notifications and adjust allocated rates of their corresponding flows the network situation is restored to that before the link failure. As before, link C1–C3 is completely utilized, while link C2–C4 is idle. At the same time, flows F1 and F2 share resources on link C1- C3 and transmit data at rates 567 Kbps and 1143 Kbps, respectively.

6. Related Work Overview and Discussion

This paper presents a mechanism that allows the Bandwidth Distribution Scheme to fairly allocate available resource among individual flows in the event of topology changes. In particular, a set of optimizations presented in this paper, show how the aggregate information stored in the network is updated upon link failure or link restoration. A similar mechanism was developed for the Resource Reservation protocol (RSVP) of the Integrated Services model [BCS94, Whi97]; RSVP includes provisions for updating reservation states in the network in the event of an error. In particular, when an error is detected the RSVP router generates *PathErr* message which informs the sender about the problem. In addition, RSVP uses a "soft-state" mechanism to remove reservation states that were not refreshed (e.g. no *Path* message arrived within a specific time period). Thus, in event of a link failure the routers rely on the "soft-state" mechanism to remove all the reservation states influenced by this event, while subsequent *Path* messages install resource reservation states for influenced flows on new routing paths [Whi97].

A load-balancing and fault tolerance mechanism for MPLS networks was discussed in [LG01]. The idea of the proposed approach is to distribute the flows or the packets of the flows influence by the failure over multiple disjoint paths. Such technique provides a more even load distribution in the network and reduces the effect of the link failure on the flows in the network. [BF03] examined a problem of fault tolerance in the networks with advance reservations where resource allocation is allowed before the actual transmission occurs. [BF03] proposed and evaluated a set of re-routing post-failure strategies for dealing with link failures in advance reservation environment. [AK01] presented an extension to QoS architecture which in addition to QoS specifications also maintains resilience requirements. The idea is that the applications provide their resilience requirements to the edge nodes. Subsequently, the network uses provided information to determine proper resource management and traffic handling. This approach was designed to work in the IP-based networks with MPLS, where resilience requirements are mapped into corresponding MPLS recovery options [AK01]. An overview of the issues related to the fault tolerance and resilience in IP networks was discussed in [AK00].

In this paper we presented a set of optimizations that allow the X-BDS approach to seamlessly operate in the event of the topology changes. Our solution relies on the underlying routing protocol to discover alternative paths and does not consider the problem of load balancing. Instead, the proposed solution addresses the problem of updating the aggregate flow requirements in the event of link failure or link restoration. Simulation results suggest that introduced extension to the RDF protocol efficiently handles topology changes. Our solution correctly updates the aggregate flow requirements in the network core and supports normal operation of the X-BDS approach in the event of topology changes.

Overall, the total time required by the BDS approach to handle topology changes is influenced by the following parameters:

- The total time to discover an alternative route and to notify the edge routers about it
- The time to remove the old resource reservation and to establish a new one

The first parameter depends only on the efficiency of the routing protocol, while the second parameter depends on the efficiency of the path probing and the RBR update phases of the RDF protocol. As it was reported in [Hna03, HS02], the total time to execute the path probing and the RBR update phases is limited by two round trip times from the ingress to egress routers, which is considered to be adequate.

7. Conclusions

This paper presented a set of optimizations that allow the X-BDS approach to perform well in the event of such topology changes as link failure and restoration of the failed link. We studied the introduced optimization through simulations using OPNET network simulator [Opn]. Evaluation of this approach suggests that the introduced modification to the RDF protocol handles topology changes in an efficient manner: it properly updates aggregate flow requirements stored in the network core and re-routes the flows influenced by the topology changes. This work provides a first step towards extending the X-BDS framework to a mobile environment. However, to better understand the properties of this algorithm, the RDF protocol should be studied in a more complex network topology where link failures influence multiple paths. In particular, such study should examine the amount of overhead caused by the RDF protocol in an attempt to re-route traffic over multiple new paths.

REFERENCES:

[AK01] A. Autenrieth, A. Kirstädter, "Resilience-Differentiated QoS – Extensions to RSVP and DiffServ to Signal End-to-End IP Resilience Requirements," In Proceeding of the 3rd International Workshop on the Design of Reliable Communication Networks (DRCN2001), Budapest, Hungary, October 7 - 10, 2001.

[AK00] A. Autenrieth, A. Kirstädter, "Fault-Tolerance and Resilience Issues in IP-Based Networks," In Proceedings of the Second International Workshop on the Design of Reliable Communication Networks (DRCN2000), Munich, Germany, April 9 - 12, 2000.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. "An Architecture for Differentiated Services," December 1998, IETF RFC 2475.

[BCS94] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," June 1994, IETF RFC 1633.

[BZB+97] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, “Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification,” September 1997, IETF RFC 2205.

[BF03] Lars-Olof Burchard, Marc Droste-Franke: "Fault Tolerance in Networks with an Advance Reservation Service," In Proceedings of Eleventh International Workshop on Quality-of-Service (IWQoS), June 2003.

[CF98] D. Clark, W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, Vol. 6, No. 4, pp. 362-373, August 1998

[CL99] H. Chow, A. Leon-Garcia, "A Feedback Control Extension to Differentiated Services," March 1999. Internet Draft: draft-chow-diffserv-fbctrl.txt (work in progress).

[GK98] R. Gibbens and E. Kelly, "Measurement-based connection admission control," In Proceedings 15th International Teletraffic Congress, Amsterdam, Netherlands, June 1997.

[Hna03] V. Hnatyshin, "Dynamic Bandwidth Distribution Techniques For Scalable Per-Flow QoS," Ph.D. Thesis, University of Delaware, 2003.

[HS03] V. Hnatyshin and A.S. Sethi, "Reducing load distribution overhead with message aggregation," In Proceedings 22nd IEEE International Performance, Computing, and Communications Conference, April 2003.

[HS02] V. Hnatyshin and A.S. Sethi, "Fair and Scalable Load Distribution in the Internet," In Proceedings International Conference on Internet Computing, June 2002.

[HS01] V. Hnatyshin and A.S. Sethi, "Achieving Fair and Predictable Service Differentiation Through Traffic Degradation Policies," In Proceedings Conference on Quality of Service over Next-Generation Data Networks, August 2001.

[KKZ00] F. Kelly, P.B. Key, and S. Zachary, "Distributed Admission Control," *IEEE Journal on Selected Areas in Communications*, Vol. 18 No. 12, pp.2617-2628, December 2000.

[KLS98] V. Kumar, T. Lakshman, D. Stiliadis, "Beyond Best Effort: Router Architecture for the Differentiated Services of Tomorrow's Internet," *IEEE Communications Magazine*, Vol.36, No. 5, pp. 152-164, May 1998.

[KST01] K. Kar, S. Sarkar, L. Tassiulas, "A Simple Rate Control Algorithm for Maximizing Total User Utility," In Proceedings of INFOCOM '01, April 2001.

[LG01] S. Lee and M. Gerla, "Fault-Tolerance and Load Balancing in QoS Provisioning with Multiple MPLS Paths," In Proceedings of IFIP Ninth International Workshop on Quality of Service (IWQoS), 2001.

[Opn] OPNET Modeler. OPNET Technologies Inc. http://www.mil3.com

[Sto00] I. Stoica, "Stateless Core: A Scalable Approach for Quality of Service in the Internet," Ph.D. Thesis, Carnegie Mellon University, 2000.

[Whi97] P. White, "RSVP and Integrated Services in the Internet: A Tutorial," *IEEE Communications Magazine*, Vol. 35, No. 5, pp. 100-106, May 1997.