Computer Communications 31 (2008) 3153–3162

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Severe congestion handling approaches in NSIS RMD domains with bi-directional reservations

Desislava Dimitrova *, Georgios Karagiannis, Pieter-Tjerk de Boer

University of Twente, DACS Group, P.O. Box 217, 7500 AE Enschede, The Netherlands

ARTICLE INFO

Article history:
Received 14 September 2007
Received in revised form 25 April 2008
Accepted 30 April 2008
Available online 16 May 2008

Keywords:
Quality of service
Diffserv domain
Severe congestion
Bidirectional reservations

ABSTRACT

Real time applications usually impose strict QoS requirements on communication networks. Several QoS frameworks have been developed and standardized to satisfy these QoS requirements. Among them is the NSIS QoS framework that is currently being standardized by the NSIS (Next Steps In Signaling) working group within IETF. Each communication node or a domain on the path that supports the NSIS QoS framework is expected to support a QoS Model (QOSM) appropriate to the characteristics of its underlying technology. One of these QOSMs is the Resource Management in Diffserv QOSM (RMD-QOSM).

RMD-QOSM is based on the reduced-state QoS concept meaning that RMD-QOSM aware interior nodes maintain per Diffserv-class reservation states instead of per flow reservation states. The reduced-state operation has many advantages, among which are scalability and flexibility, but it also results in complex handling of severe congestion situations. A severe congestion may occur when a router or link fails and the traffic is rerouted through another router or link which may become severely overloaded. This paper focuses on the investigation and evaluation of severe congestion handling solutions applied in a RMD-QOSM aware domain which supports bidirectional reservations initiated and maintained by preemption aware services.

1. Introduction

In the past few years, the rapidly increasing use of IP technology in all kinds of networks has greatly boosted the development and deployment of real-time multimedia networking applications such as video conferencing and, most notably, Voice over IP (VoIP). It is predicted that in the foreseeable future, VoIP might as well completely replace the traditional public switched telephone network (PSTN). However, real-time multimedia networking applications impose strict requirements on the underlying communication network, being especially sensitive to packet loss, delay and jitter (variation in delay), see [1]. Furthermore, the underlying communication network should provide support for multiple levels of precedence, or multilevel services. For example, preemption aware services such as emergency calls should be protected at all times, whereas regular, e.g., household, calls could be dropped in favor of emergency calls may the total network capacity be insufficient, e.g., during peak hours, see [2].

In order to extend the best-effort service of IP-based networks, e.g., the Internet, and provide explicit support of multilevel services, various QoS frameworks have been developed. A survey of such QoS frameworks can for example be found in [3]. The IntServ model [4], and its signaling protocol Resource ReserVation Protocol (RSVP) [5], which is specified by the Internet Engineering Task Force (IETF) is able to support end-to-end per flow reservations. RSVP-network intermediary nodes implement Intserv algorithms and manage RSVP reservation requests using per flow states. However such per flow orientation can cause scalability problems, which initiated several attempts to improve the scaling characteristics through flow aggregation.

One attempt is associated with the DPS (Dynamic Packet State) and SCORE (Scalable CORE) service architecture proposed in [6]. In this architecture, per flow management is brought to the edges of an administrative domain and the nodes within the same administrative domain are kept stateless. Furthermore, a dynamic packet state technique is used that uses specific fields of the IP packet to embed the per-flow state. One main disadvantage of this technique is the fact that it is not considered for standardization by the IETF.

A second attempt on improving the scaling characteristics is associated with the specification of RSVP aggregation protocol [7], which is specified by the IETF and which can only be used within one administrative domain. RSVP aggregation aims to avoid per-flow signaling and per-flow state information in the interior/core, by selecting flows with similar QoS requirements and aggregating them.

The Intserv/Diffserv integrated solutions [8], which are specified by the IETF, present yet another attempt. The Differentiated

---

* Corresponding author. Tel.: +31 53489 2013.
E-mail addresses: d.c.dimitrova@ewi.utwente.nl, dim_des@yahoo.com (D. Dimitrova), g.karagiannis@ewi.utwente.nl (G. Karagiannis), p.t.deboer@ewi.utwente.nl (P.-T. de Boer).

0140-3664/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2008.04.031

Services (Diffserv) model [9] can be applied within one administrative domain and it allows the aggregation of flows with similar QoS requirements into Per Hop Behavior (PHB) groups. Nodes at the boundary of the Diffserv aware domains are aggregating the traffic by marking the incoming packets with Diffserv Code Points (DSCPs). In the Intserv/Diffserv integrated solution RSVP signaling messages are carried out across the Diffserv domain but only processed by the edge nodes.

Resource management within Diffserv can be accomplished by using static provisioning and/or dynamic provisioning. Currently, the dynamic provisioning can be accomplished using either the RSVP aggregation protocol or by using Bandwidth Brokers (BBs) [10,11]. BBs are entities that in a centralized way are controlling the information concerning network resources and their usage, domain topology, service policies, and negotiated Service Level Specifications (SLSs). The centralized way of maintaining information has many advantages, but also disadvantages, such as high processing load, congestion and functional dependence on a single entity.

Currently IETF is specifying other means of providing dynamic resource management within Diffserv. One of these means is developed by the Congestion and Pre-Congestion Notification (PCN) working group [12]. PCN develops standards for the marking behavior of the Diffserv interior nodes and for the encoding and transport of the congestion information within the Diffserv domain, see [13]. Another IETF working group, the Next Steps in Signaling (NSIS) working group [14], develops a protocol suite, see [15,16], which provides support for the participation of network entities, such as routers and end nodes, in the signaling procedure and means of communication to the rest of the network.

The NSIS protocol suite is decomposed into a generic, lower layer denoted as NSIS Transport Layer Protocol (NTLP), see [17], and an application specific, upper layer denoted as NSIS Signaling Layer Protocol (NSLP). Currently two application layers are being developed by the NSIS working group – a QoS signaling application (QoS NSLP [18,19]) and a NATFW NSLP for configuring Network Address Translator and Firewalls, see [20]. Other NSLP proposals are as well discussed, for example an NSLP for configuration of metering entities, see [21]. The design of QoS NSLP has conceptual similarities with RSVP, in the sense that QoS NSLP also makes reservations and uses soft-state hop-by-hop refresh messages as a primary state management mechanism. A reservation is made only in a node that supports the QoS NSLP specification. Unlike RSVP, QoS NSLP supports: (1) reservations initiated by both the sender and the receiver; (2) bi-directional reservations; (3) and flexible operation scenarios such as end-to-edge and edge-to-edge. In QoS NSLP a significant decision – no support for multicast – was taken due to the high scale of complexity introduced by the multicast support.

In QoS NSLP, a QoS architecture, such as Intserv and Diffserv, is represented by a QoS Model (QOSM) which specifies directions on how the reservation management should be done. Cooperation between different QoS architectures is supported in NSIS by separating signaling from resource management. QoS NSLP is the functionality responsible for the signaling, while the Resource Management Function (RMF) controls the resource, in accordance to the applied QOSM. The use of QOSMs and the fact that several QOSMs can co-exist in the same node allows a QoS NSLP end system to accomplish generic end-to-end resource reservations across a mixed wireless and wired network and have these reservation implemented accordingly at each point in the network, see [22]. Moreover, the QoS NSLP signaling can include technology-specific requests, whenever an application requires some special treatment by some network technology along the path, e.g., wireless link. One of the QOSMs developed by the IETF NSIS working group is the Resource Management in Diffserv (RMD) QoS Model (RMD-QOSM) as described in [23].

This paper focuses on the development and the evaluation of severe congestion mechanisms for bidirectional reservations in RMD-QOSM aware domain, when preemption aware services are considered. A severe congestion may occur when a router or link fails and the traffic is rerouted through another router or link that may become severely overloaded.

Both bidirectional reservations and preemption aware services set new challenges on how severe congestion is solved. Bidirectional reservations are formed by two associated to each other (or bound) unidirectional reservations. Therefore, it is important to take into account the fact that terminating one bidirectional reservation results in decreased load on both paths, i.e., the forward and reverse. Preemption aware services require prioritization, which may impact the way of how severe congestion mechanism are selecting and terminating flows.

Up to now, several severe congestion handling approaches have been investigated given that connections/reservations use stateful nodes (i.e., keep per flow state) on the control path, see [28–31,5]. However, studies of severe congestion handling associated with stateless nodes (i.e., do not keep per flow state) is a central topic of only few research/standardization activities [26,29]. The authors could not find, up to now, research papers that investigate severe congestion mechanisms, applied to communication paths which: (1) use stateless nodes, (2) support bi-directional reservations; and (3) are established by preemption aware services. Therefore the authors consulted the work on RMD-QOSM, examined the possibilities for improvement and developed own proposals to address the open issues associated with severe congestion handling.

The RMD-QOSM draft, at that time, already included stateless nodes and a mechanism for severe congestion handling with *only* unidirectional reservations. This research study began with evaluating the performance of this mechanism with bi-directional reservations. The results were dissatisfying, since the existing mechanism did not consider the reservation binding in bidirectional reservations. As the next step, the authors discussed several own proposals for optimization out of which two were chosen for implementation. The two severe congestion mechanisms introduce new functionality to the edge nodes which is aware of the reservation binding. The performed tests showed promising results and eventually the RMD-QOSM specification was extended with these optimized mechanisms. Subsequently, these details were included in the IETF standardization draft [23], at the time of the research.

This paper tries to answer the following research questions:

(1) What severe congestion handling mechanisms can be used when bi-directional reservations and preemption services are applied in a RMD-QOSM aware domain?
(2) Which of these severe congestion mechanism satisfies the requirements best?

The rest of this paper is organized as follows. Section 2 describes the RMD-QOSM framework. In Section 3, first the problem of severe congestion is outlined. Subsequently, three severe congestion handling proposals are explained. Section 4 first describes the performance criteria that can be used to evaluate the different severe congestion handling mechanisms followed by the performed simulation experiments and their analysis. Finally, conclusions are drawn and possible future research recommendations are discussed in Section 5.

2. Resource management in Diffserv (RMD) QoS model (RMD-QOSM)

RMD-QOSM is designed to support dynamic resource reservation in accordance to the Diffserv QoS framework, see [24,25]. Multiple reservations are processed in parallel as result of two major

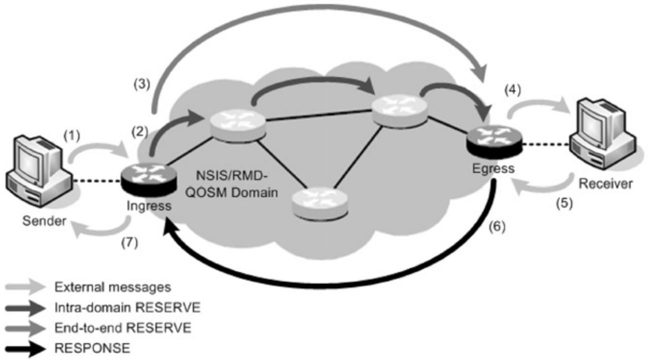

Fig. 1. Resource reservation in the NSIS/ RMD-QOSM domain.

mechanisms, i.e., the admission control and severe congestion management.

Admission control evaluates the resource availability within a node. Nodes inside the RMD domain, i.e., interior nodes, apply a simple verification based on traffic class aggregation, i.e., per PHB. Subsequently, edge nodes evaluate the resource availability for the whole domain. In this process, the edge nodes use the information gathered by interior nodes and an admission decision is taken on a per flow basis. In the admission control process RESERVE and RESPONSE signaling messages are used, see Fig. 1 and [18]. A RESERVE message informs a node about how many resources were requested. A RESPONSE message carries information to edge nodes about the result of the resource inquiry inside the RMD-QOSM aware domain. Once a requesting flow is admitted in the RMD-QOSM domain, then the QoS level associated with this flow is maintained for the duration of the reservation.

If no link or node failures occur then the requested QoS level can be provisioned by the RMD-QOSM aware domain. There are nevertheless situations in which failures do occur, which could cause severe congestion situations, and therefore RMD-QOSM has to be able to react adequately.

Very often link failure triggers re-routing of flows via other still well operating links. Consequently, if these links are highly loaded, they may become overloaded leading to increased delays and probable packet loss. In such circumstances, the QoS experienced by the flows (i.e., both original and re-routed flows) on the overloaded link(s), is likely to be degraded severely. Such undesirable situation is referred to as *severe congestion*, see e.g., [25–27]. To be able to detect severe congestion and to restore the normal link operation, RMD-QOSM was 'armed' with mechanisms for severe congestion detection and handling.

3. Severe congestion management in RMD-QOSM

Traffic re-routing can cause severe congestion on the newly used link. If the occurring overload is not solved fast, all sessions on the particular link – original and re-routed – will suffer unacceptable degradation in the quality of service. To address this problem, severe congestion management is applied. Severe congestion management aims to: first, detect severe congestion situations and second, to decrease the overload down to the link capacity.

Severe congestion can be detected in the interior nodes by performing (excess) traffic measurements. The information obtained from these measurements is encoded by the interior nodes using a packet marking approach. The marked packets are propagated to the egress nodes, which – since they maintain per flow reservation states – are responsible for severe congestion handling. Several packet marking approaches are possible, but in this study the dampened rate proportional approach is used because of its good performance characteristics, see [29,23]. In the dampened rate proportional marking approach the number (or rate) of marked data packets is proportional to the level of overload. We introduce additional, enhance functionality to the edge nodes, which was applied in [23].

Before, we further discuss the severe congestion management procedures, a simple example is presented. The example illustrates the possible types of severe congestion situations that can occur with bidirectional reservations. Gaining insights on the actual processes that are taking place, provides a crucial means to understand the design decisions taken.

In the example, the network topology depicted in Fig. 2a is used. Node 0 has bi-directional communication with node 2, meaning that node 0 sends traffic to node 2 and node 2 sends traffic to node 0. Node 1 has also a bi-directional communication with node 2. The traffic is chosen such that the utilization of each link can go up to 60% of its total capacity.

Three possible situations of severe congestion exist. In the first case, see Fig. 2b, it is assumed that link 0–2 breaks, meaning that the flows (and their traffic) passing from node 0 cannot go directly to node 2, but the flows passing through node 2 towards node 0 can. The 60% load forwarded from node 0 is therefore re-routed via node 1, which causes a severe congestion on link 1–2. This situation is referred as a severe congestion on the forward path.

In the second case, see Fig. 2c, link 2–0 drops and flows (and their traffic) passing from node 0 can reach node 2 but the direct communication from node 2 towards node 0 is not possible The routing protocol will select to route the flows from node 2 via node 1. That would lead to 120% utilization of link 2–1. In this case a severe congestion occurs on the reverse path.

In the last possible scenario, see Fig. 2d, both links 0–2 and 2–0 fail and flows in both directions are re-routed. This is referred to as severe congestion in both directions.

Note that severe congestions in each direction are independent events, i.e., they can occur independently. If flow sizes in both directions are not equal then the occurring overload in the forward and reverse direction is different. Another difference is the time to solve the severe congestion if different forward and reverse paths are used.

3.1. Severe congestion notification by marking data packets

In this section, we give a very brief explanation of the dampened rate proportional severe congestion mechanism. The detailed

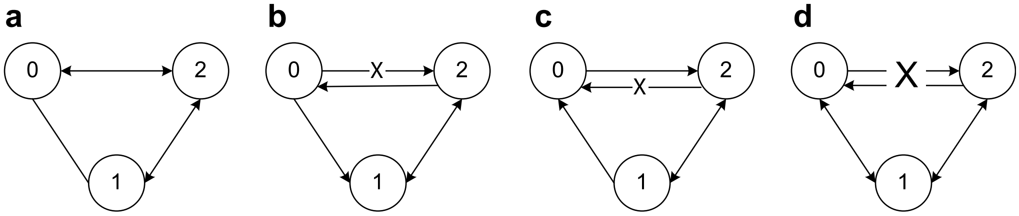

Fig. 2. Delta topology.

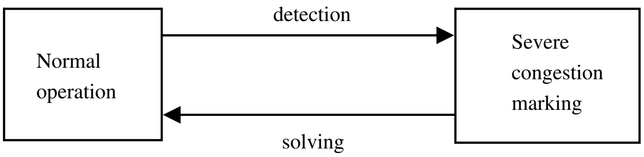

Fig. 3. Interior node operational states.

description of this mechanism can be found in [23]. The RMD-QOSM severe congestion solution is composed of two parts: the severe congestion detection and the severe congestion handling (or solving).

The dampened rate proportional marking approach uses two DSCP values for data packets. If data packets are used for congestion notification, two additional DSCPs are used. The 'affected DSCP' informs the egress nodes that a data packet has passed a congested interior node and the 'encoded DSCP' informs the egress nodes about the level of severe congestion.

Interior nodes are responsible for severe congestion detection and notification by marking data packets. When no overload is occurring then the node operates in a so called 'normal operation' state, see Fig. 3. At the event of overload the node detects severe congestion and its operation state changes to 'severe congestion marking'. In this new state, data packets are specially marked, i.e., 'encoded DSCP', to inform the egress nodes about the level of severe congestion. The egress and ingress nodes take measures to resolve the overload such that eventually the interior node moves back to the 'normal operation' state.

Severe congestion detection is triggered by an increase in the total link rate above the 'severe congestion threshold'. The severe congestion threshold should be higher or equal to the 'admission control threshold' which gives the maximum proportion of the link capacity that can be used for data transfer.

Once it detects a severe congestion, an interior node starts marking data packets. Initially data packets are marked proportionally to the excess traffic rate by using the 'encoded DSCP' encoding. The excess traffic rate is the rate above the severe congestion threshold, which overloads the interior node. The rest of the data packets that pass thorough the severely congested interior node are marked using the 'affected DSCP' encoding.

It is important to note that the rate of the 'encoded DSCP' marked packets is equal to the calculated excess rate divided by a proportionality parameter that we denote in this paper as N. The parameter N is configured to be equal in the whole RMD-QOSM aware domain and is used to be able to notify an overload that is even higher than 100% of the capacity of a link. For example, $N = 1$ allows representing overloads between 0% and 100% overload, while $N = 2$ would allow representing overloads between 0% and 200%.

A severe congestion is considered solved when the total link load (or rate) drops below the 'severe congestion restoration threshold'. The severe congestion solving mechanism itself is performed at the edge nodes of the RMD-QOSM aware domain since they keep per flow state. In particular the egress node calculates the rate of the received 'encoded DSCP' packets. This measured rate, when multiplied by N represents the excess traffic rate that was measured by the severely congested interior node. Since severe congestion is often caused by the fact that too many flows (and their traffic) are forwarded on a link with restricted bandwidth capacity, part of these flows have to be terminated, such that the agreed QoS level of the ongoing flows is maintained. The number of terminated flows should be such that the terminated bandwidth is equal to the excess traffic rate measured by the egress node.

Furthermore, the egress node identifies the flow identity of the received 'encoded DSCP' and/or 'affected DSCP' marked packets. We denote these flows as marked flows. When different priority flows are supported, then the severe congestion mechanism terminates low priority marked flows first. Subsequently, medium priority marked flows are terminated and finally, if necessary, even high priority marked flows are selected for termination.

For each flow that has to be terminated the egress node sends a NOTIFY signaling message, see [18], to the ingress node which terminates the flow.

3.2. Optimized mechanisms for bi-directional reservations

The severe congestion handling mechanism described in Sections 3.1 performs well when only unidirectional reservations are in process in the RMD-QOSM aware domain. The mechanism described in Section 3.1 is denoted in this paper as *without_optimization.* As selection criterion for flow termination this mechanism uses the flow reservation on the severely congested path, starting with the biggest flows. A disadvantage of the *without_optimization* mechanism is that it accounts neither for the reservation size on the path opposite to the severely congested not for the fact that the two reservations, forward and reverse, are bound.

This paper proposes two severe congestion handling optimizations. We begin with accounting for the reservation size on the path opposite to the severely congested. A mechanism, denoted as *with_optimization_1,* selects bidirectional flows for termination using as selection criterion, the smallest reservation on the path that is opposite to the severely congested path. Consider the following example: suppose that the RMD-QOSM aware domain supports bidirectional reservations between nodes A and B and the forward path between node A to B is severely congested. The severe congestion solution denoted as *with_optimization_1* stops flows with the smallest reservation sizes on path B–A. Note that the severe congestion denoted as *without_optimization* will choose first the flows that maintain high (or big) reservation sizes on path A–B, without taking into account the bound reservation size on the reverse path. A comparison of the above mechanisms is presented in Section 4.4.

The *with_optimization_1* mechanism still leaves an open issue – the fact that both reservations are bound. This is especially important when both paths become severely congested. When bidirectional reservations are supported an edge node can operate simultaneously as an ingress and as an egress node. It is always the ingress node however that terminates flows and therefore the time it takes to solve a severe congestion on each path, i.e., forward and reverse, is different. With both paths severely congested if the communicating edge nodes operate independently, as is the case of the *without_optimization* mechanism, it might happen that more reservations are terminated than necessary. Such situation is denoted as a link utilization undershoot.

The *with_optimization_2* mechanism[1] avoid link utilization undershoot by maintaining at the ingress node information on the amount of terminated excess rate (bandwidth) on the forward path. At the same time the node also collects excess rate information on the reverse path. Eventually, the ingress can then more efficiently decide, which flows should be terminated to solve the severe congestion. In particular, due to smaller delays on the reverse path the ingress will first terminate flows on the reverse path. Nevertheless, due to reservation binding, also part of the overload on the forward path is solved. The egress node is not aware of that and it chooses enough flows to resolve the overload on the forward path. Thanks to the *with_optimization_2* severe congestion solution, the ingress node does not blindly terminate a flow but it first verifies whether

---

[1] Note that the *with_optimization_2* mechanism incorporates the principle used in the *with_optimization_1* mechanism, i.e., flows with the smallest size on the reverse path are terminated.

link undershoot will occur. If this is the case, then the termination of the requested flow is not processed, otherwise it is. Below, a brief example is explained.

Consider a bidirectional reservation, where each of the generated flows has the same reservation in the forward and reverse direction, i.e., 1 resource unit. A severe congestion on both paths occurs with an overload of 5 resource units on the forward path and 3 resource units on the reverse. The egress sends NOTIFY messages for 5 flows (for the forward path) and the ingress stops 3 flows (for the reverse path). The three stopped flows by the ingress correspond to the release of 3 reservation units on the forward path. This leaves 2 resource units of overload on the forward path. Upon arrival at the ingress node the first three NOTIFY messages are not processed, because 3 resource units of overload were already released on the forward path. The fourth and fifth NOTIFY are processed because the resource units they want to release are not yet released by the ingress node. By processing these two messages the left overload of 2 resource units on the reverse path is solved and the normal link operation is restored.

If the reservation settings are changed such that the forward path has 3 resource units of overload and the reverse path has overload of 5 resource units none of the NOTIFY messages is processed. The reason is that the ingress node has released more reserved bandwidth than the egress needs to release.

4. Performance evaluation

In particular, three experiments are performed. In the first experiment A the impact of the bidirectional reservation sizes on the severe congestion solutions is evaluated, see Section 4.3. In the second experiment B the performance of the *without_optimization* and *with_optimization_1* severe congestion solutions is compared, see Section 4.4. Finally, in the third experiment C the performance of the *without_optimization* and *with_optimization_2* severe congestion solutions is compared, see Section 4.5.

4.1. Simulation settings

To test the desired mechanisms a simulation model was developed. The model was built in the simulation environment of the network simulator[2] ns2. The complete simulation model includes a traffic generator, simulated traffic topology and a simulation part implementing the behavior of the RMD-QOSM protocol. It is important to note that the NTLP layer functionality is not included in the simulation model, and it is assumed to be transparent from the NSLP layer perspective. The functionality of the interior and edge-to-edge nodes was implemented by means of signaling messages and the way they are processed by the network elements, i.e., links and nodes. The model also includes the optimized severe congestion mechanisms introduced in Section 3.2.

The three sets of experiments share some common settings of the simulation model. The topology links use a capacity of 10 Mbps with a propagation delay of 2 ms. Furthermore, each link uses two ns2 physical *dsRED queues*. Physical queue 1 is used for scheduling signaling messages and has the highest priority and size of 44 Kbytes. Physical queue 2 is used for data packets; it receives a lower priority and is split into two virtual queues. The default size of virtual queue 1 (marked data packets) is 65 Kbytes and the size of virtual queue 2 (unmarked data packets) is 58 Kbytes. Note that the sizes of the queues are selected after performing several experiments to achieve a reasonable tradeoff between packet loss and packet delay.

In all experiments CBR (Constant Bit Rate) flows are simulated with varying rate depending on the experiment, see Section 4.2. Additionally each CBR flow is assigned a priority level, i.e., high, medium or low. The flows are generated based on a uniform distribution where the total number of flows are started between the 5th and the 35th second. A link failure is simulated to occur at simulation time equal to 100 s. It is important to note that all flows are generated and stabilized before this event. The holding time of all flows is considered to be higher than the simulation duration (120 s). These choices were made because we want to study the impact of the severe congestion situation on the ongoing flows that are generated before the failure of the link or of the node that causes this severe congestion situation.

An admission threshold of 100% is used, which corresponds to the possible occupation of the total link capacity. The thresholds of severe congestion detection and severe congestion restoration are set to 103% and 100% of the link capacity, correspondingly.

Based on a literature study, see e.g., [28,30,31], we found that the severe congestion mechanisms can be evaluated best using the following performance criteria (or measures):

- *Detection and handling time* is the time it takes to solve the severe congestion. In other words the time from the link failure (e.g., 100 s) until the link utilization drops back to the severe congestion restoration threshold (e.g., 10 Mbps).
- *Link load before and after stabilization* indicates the load on a link before and after the severe congestion has been solved. Optimally, the load after stabilization is as close as possible to (but not above) the restoration threshold, since otherwise unnecessarily much user traffic was terminated.

4.2. Simulation topology

All experiments for bidirectional reservations use the same network topology given in Fig. 4. In Fig. 4, nodes 1 and 2 are interior nodes. Nodes 0 and 3 are ingress edges and emulate the data sources for the forward direction and the data receivers for the reverse direction. The nodes 4 and 5 are the egresses that emulate the data receivers for the forward direction and data sources for the reverse direction. The combination of sources and receivers makes possible to monitor the load coming from each data traffic source during the simulation time.

In the forward direction source 0 sends traffic to node 5 and source 3 sends traffic to node 4. In the reverse direction source 5 sends traffic to node 0 and source 4 sends traffic to node 3, see Fig. 4(a). At link failure time (at 100 s) link 2–3 breaks, see Fig. 4(b) and flows from source 3 are re-routed via path 1–2 and flows from source 4 via path 2–1. As result, severe congestion situation occurs on link 1–2. The type of the severe congestion depends on the used flow reservation sizes.

The effect of the *with_optimization_1* mechanism is better illustrated when a severe congestion occurs only in one direction. Since each flow has a forward and a bound reverse reservation, the flow sizes should be carefully chosen such that only one path becomes overloaded. To achieve this, we used a different combination of flow sizes for each direction of the communication. On the path of severe congestion a combination of 16, 32 and 64 Kbps rates is used while on the opposite path rates of 8, 16 and 32 Kbps are chosen. The chosen rates are presented in Table 1.

In experiments A and B a severe congestion situation on the forward path is simulated. Experiment A examines the performance of the *with_optimization_1* mechanism while experiment B compares this mechanism to the *without_optimization* mechanism.

In experiment C only 16 Kbps flows are used with enough aggregated rate to cause an overload and severe congestion in both directions of the data transfer. The goal of this experiment is to compare the performance of the *without_optimization* and *with_optimization_2* severe congestion mechanisms.

---

[2] http://www.isi.edu/nsnam/ns/.

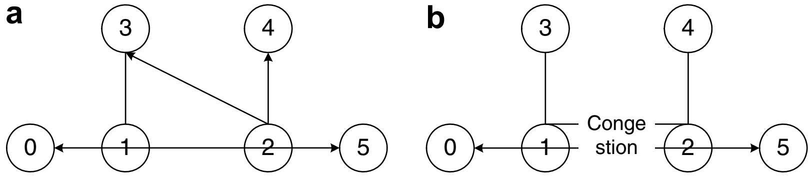

Fig. 4. Network topology.

Table 1
Flow sizes: experiments A, B, and C

Source–Destination	Total load (Mbps)	High priority flows (Mbps)	Medium priority flows (Mbps)	Low priority flows (Mbps)	Used rates (Kbps)	Used rate combination
Experiment A and B, severe congestion in forward direction						
0–5, 3–4	10	1	3	6	16, 32, 64	big–big, big–small
5–0, 4–3	5	0.5	1.5	3	8, 16, 32	
Experiment C, severe congestion in both directions						
0–5, 3–4 5–0, 4–3	10	1	3	6	16	None

Different rates also mean that different combinations between the forward and reverse reservation can be set. When the flows with the highest forward reservation have also the highest bound reverse reservation they are referred to as *big–big flows* and when the highest forward reservation have the smallest bound reverse reservation size, they are referred to as *big–small flows*.

4.3. Experiment A – One direction severe congestion

The *goal* of this experiment is to observe the impact of the bi-directional reservation sizes on severe congestion solutions. In this type of experiments, all reservations are bi-directional and only one severe congestion point occurs on either the forward or the reverse path. Furthermore, three types of flow priorities are used: high, medium and low. The used bi-directional severe congestion mechanism is described in Section 3.2 and denoted as *with_optimization_1*.

This experiment is performed twice, the first time using *big–big flows* and the second time using *big–small flows,* with rates as defined in Table 1.

Fig. 5 shows the link load versus simulation time obtained on the severely congested link 1–2, when the *big–small flows* setting is used. Fig. 6 shows the link load versus simulation time on the opposite direction link, i.e., link 2–1, for the *big–big flows* setting (left part of the figure) and for the *big–small flows* setting (right part of the figure). Fig. 7 shows the message signaling load versus simulation time for the *big–big flows* setting (left part of the figure) and for the *big–small flows* setting (right part of the figure).

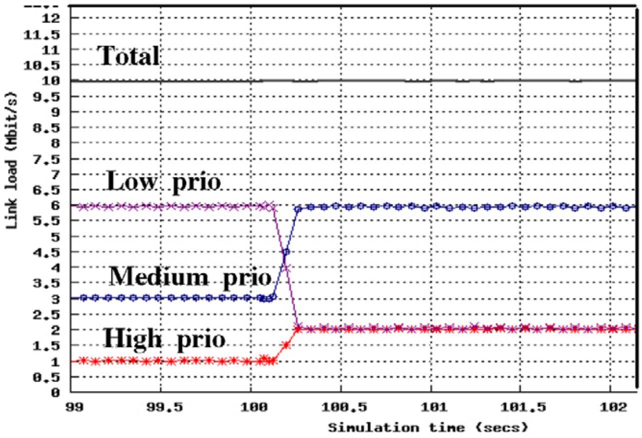

Fig. 5. Experiment A, big–small flows: Link 1–2 – one path congested.

The link load on the forward path of the overloaded link 1–2 in both experiments (*big–big flows* and *big–small flows*) is the same and the load for *big–small flows* case is presented in Fig. 5. After the severe congestion is solved the link utilization is 100% but the proportion of different priority groups is rearranged in order to keep all high priority flows and as much as possible medium priority flows. The *detection and handling time* is 0.25 s and the priority of the flows is maintained.

However, the link load on the reverse path after stabilization for both simulation experiments differs. In both graphs after the link failure (at 100 s) the link load initially rises to 10 Mbps due to the re-routed flows from the reverse path source 4, see Fig. 6. For the *big–big flows* experiment, see Fig. 6, termination of the half of the bandwidth on the forward path results in a termination of also the half of the bandwidth on the reverse path. The reason is that the forward – reverse reservations ratio is 2:1. To solve the severe congestion, total reservations summing up to 10 Mbps have to be maintained on the forward path, which are associated/bound with a reservation of 5 Mbps on the reverse path.

In the *big–small flows* experiment the link load on the reverse path does not drop to 50% but it stays above it, see Fig. 6. The proportion of forward – reverse bandwidth is different for each of the flow size combinations. The flow termination starts with the smallest reverse bandwidth, which is in this case the biggest forward bandwidth. As a result, the congestion is solved by stopping fewer flows than in the *big–big flows* experiment and the flows with the biggest reverse reservation are still maintained in the network. The reader's attention might be drawn to one peculiar drop in the total link load that is more visible in the *big–small flows* case, see Fig. 6. If the graphs of the message signaling loads are consulted (Fig. 7), the explanation is obvious. RMD-QOSM uses in-band signaling and the signaling packets have the highest priority. When the NOTIFY messages are sent they use part of the link capacity and only the left over capacity is used for data transfer, which causes the drops.

Additionally, the signaling load for the *big–big flows* case is higher (Fig. 7), which shows that more flows have to be stopped than in comparison with the case of *big–small flows*.

4.4. Experiment B – Comparison of without_optimization and with_optimization_1 mechanisms

The *goal* of this experiment is to compare the performance of the *without_optimization* and *with_optimization_1* mechanisms. In

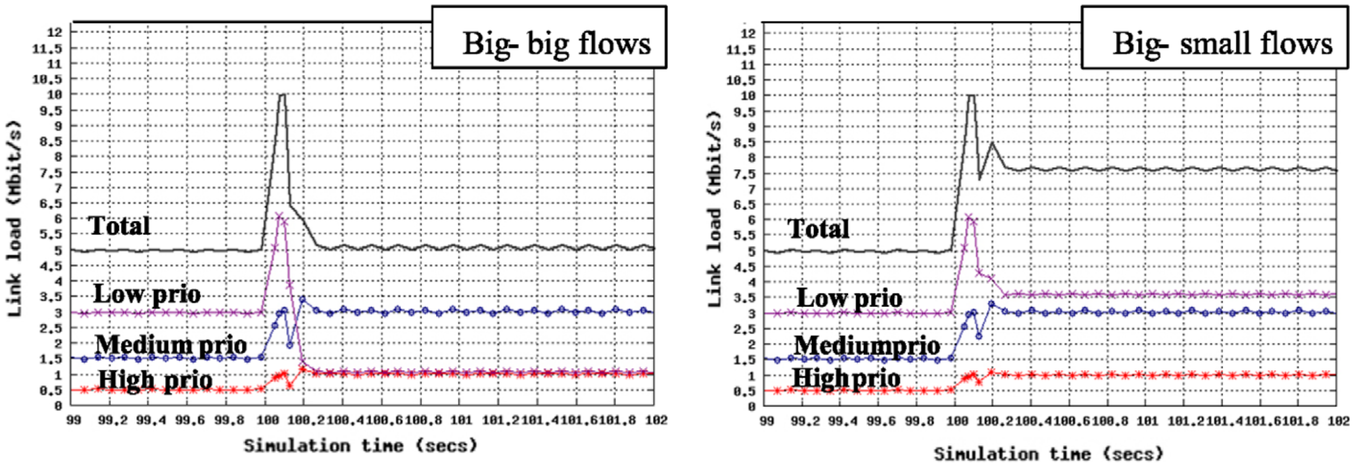

Fig. 6. Experiment A: Link 2–1 – one path congested.

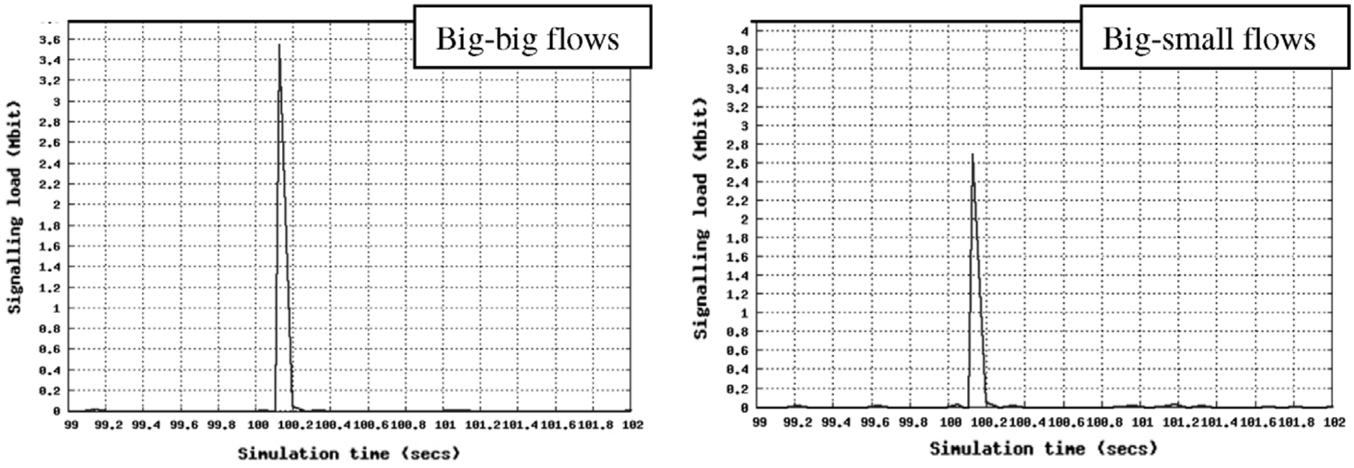

Fig. 7. Experiment A, signaling load: Link 2–1 – one path congested.

this type of experiments, the reservations are bidirectional and only one point of severe congestion occurs on the forward path. Furthermore three flow priorities are used: high, medium and low.

In the previous section, the results were generated using the *with_optimization_1* severe congestion mechanism. This set of experiments uses the same setting that was used for experiment A, but now the *without_optimization* solution is used, instead of the *with_optimization_1* solution. Again combinations of *big–big flows* and *big–small flows* is used with the rates specified in Table 1.

Fig. 8, similar to Fig. 6, shows the link load versus simulation time on the link 2–1, for the *big–big flows* setting (left part of the figure) and for the *big–small flows* setting (right part of the figure).

Beginning with the *big–big flows* experiment the *with_optimization_1* severe congestion mechanism starts terminating flows with the smallest bound reverse reservation size, which corresponds to the smallest forward reservation size. To solve the severe congestion level of 100% the mechanism has to terminate, say *X* flows.

The *without_optimization* mechanism starts selecting the bidirectional flows with biggest forward reservation size, which in this case have also the biggest reverse reservation size. To solve the severe congestion level of 100%, the mechanism stops, say *Y* flows, where *Y* is smaller than *X*. This phenomenon is observed on the reverse link 2–1. In particular, when the *with_optimization_1* severe congestion mechanism is used, the drop in the total load of data packets (see Fig. 6, big–big flows, drop to 6.5 Mbit/s) is bigger than the situation when the *without_optimization* severe congestion mechanism is used (see Fig. 8, big–big flows, drop to 8 Mbit/s). The utilization on the forward link, i.e., link 1–2 is not shown in this paper but it has been measured and found to be similar to the one shown in Fig. 5.

The second scenario uses the *big–small flows* combination. No differences in link utilization and signaling load are expected. The reason is very simple. The *with_optimization_1* severe congestion mechanism stops first the flows with smallest reverse reservation size. These are the flows with biggest forward reservation size. The *without_optimization* severe congestion mechanism, on the other hand, begins with the highest forward reservation size flows, which corresponds to the smallest reverse size. It can be concluded that in the case of big–small flows both mechanisms terminate the same number of flows. Therefore the same number of NOTIFY messages are generated and the signaling load is the same. Note that this phenomenon has been observed during various performance experiments, but due to the fact that the derived conclusion is obvious, the output of these experiments is not included in this paper.

4.4.1. Conclusions

- Both mechanisms, i.e., *with_optimization_1* and *without_optimization*, solve a severe congestion situation by terminating the same amount of bandwidth, i.e., 10 Mbps on the forward path and 5 Mbps on the reverse path.

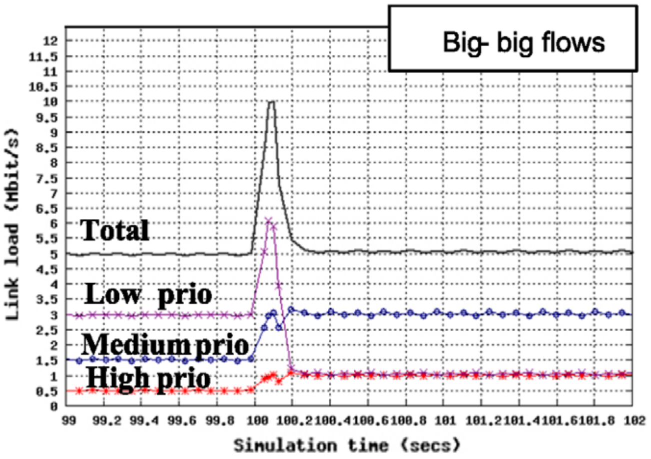

Fig. 8. Link 2–1: *Without_optimization* – one path congested.

- The number of flows terminated by each severe congestion mechanism (i.e., *with_optimization_1* and *without_optimization*) and consequently the generated signaling load depend on the combination of the forward and reverse reservation sizes, i.e., whether *big–big* or *big–small* flows are used.
- The *detection* and *handling* time for both severe congestion mechanisms is the same.
- The priority of the flows is not affected regardless whether the optimization is used or not.

4.5. Experiment C – Comparison of without_optimization and with_optimization_2 mechanisms

The *goal* of this experiment is to compare the performance of the *without_optimization* and *with_optimization_2* mechanisms, see Section 3.2. In this type of experiments, the reservations are bidirectional and one point of severe congestion occurs in the forward direction and another one, almost simultaneously, occurs on the reverse direction. Furthermore, three flow priorities are used: high, medium and low.

When bidirectional reservations are used and when both paths are severely overloaded then both the ingress and the egress nodes choose flows to terminate. Each flow keeps a forward reservation and a bound reverse reservation. As result, when the severe congestion situation is solved, more flows might be terminated than is necessary to solve the severe congestion. To observe such behavior a simulation experiment was performed using the *without_optimization* severe congestion mechanism. The exact same experiment was then repeated with the *with_optimization_2* mechanism in order to evaluate whether there is some improvement. The network topology is depicted in Fig. 4 and the traffic parameters are given in Table 1, i.e., 16 Kbit/s CBR flows of three different priorities. The link utilization in each direction is chosen to be 100%.

At link failure time (at 100 s) links 2–3 and 3–2 break, see Fig. 4 (b), and flows from source 3 are re-routed via path 1–2 and flows from source 4 via path 2–1. As a result 100% severe congestion occurs on link 1–2 and link 2–1.

The performance results show that the use of the *without_optimization* mechanism leads to a link utilization undershoot on both paths, see Fig. 9. The reason of this is the lack of communication between the egress and ingress node, causing each of them to independently terminate bandwidth proportional to the excess rate. However, when speaking about bidirectional reservations, both severe congestions are related and when a flow is stopped actually resources in both directions are released.

Note that it might be expected that the drop in link load will be 50% when the *without_optimization* mechanism is used. This does not happen because a flow that receives marked data packets on the forward path can also receive marked data packets on the reverse path. As result a double amount of packet rate can be

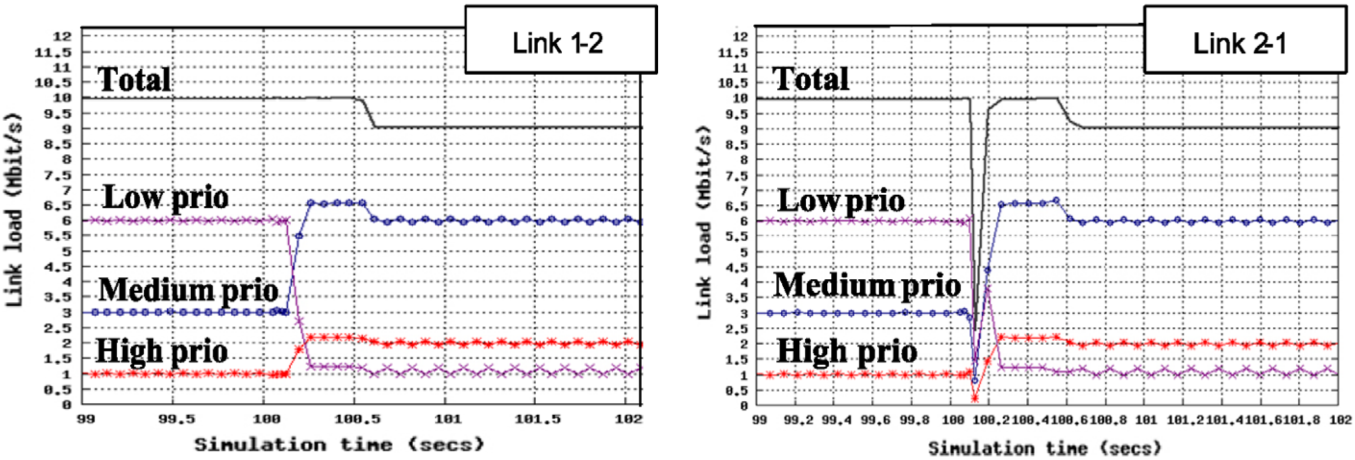

Fig. 9. *Without_optimization* – both paths congested.

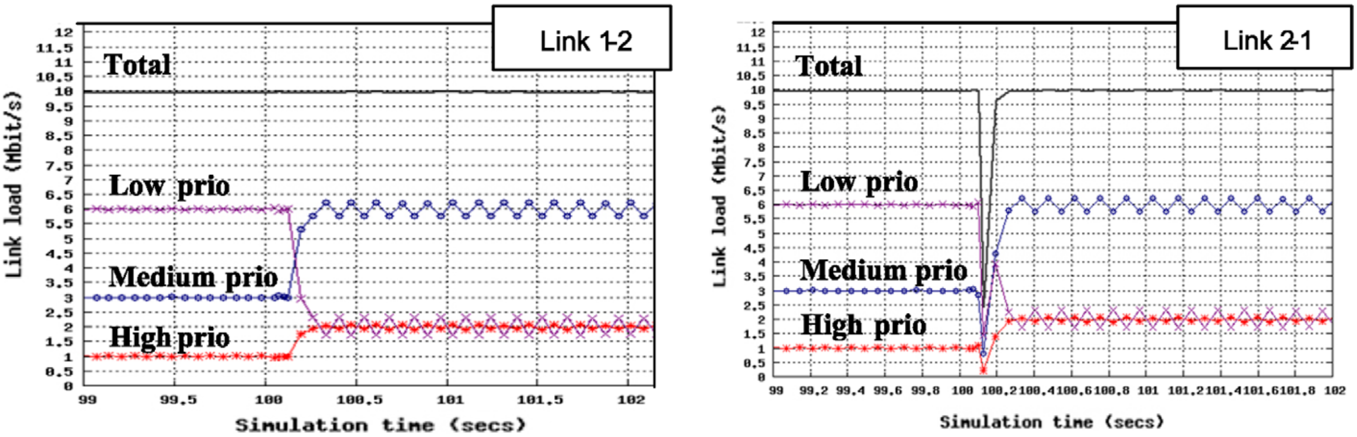

Fig. 10. *With_optimization_2* – both paths congested.

marked, but the same flow can also be chosen for termination by both the ingress and the egress. Therefore, the link utilization undershoot is not as big as expected.

The *detection and handling time* is about 0.55 s and the priority of the flows is not affected by other factors besides the link utilization undershoot.

When the *with_optimization_2* severe congestion mechanism is used, it is expected that the link undershoot issue is solved. Since the ingress node keeps information on the flows to be terminated in both directions and it can therefore, compensate for the undesired flow termination drop. The performance evaluation results obtained when the*with_optimization_2* severe congestion mechanism is used confirm these expectations. From the graphs that show the overloaded links 1–2 and 2–1 (Fig. 10), we can conclude that the *with_optimization_2* mechanism successfully solves the link undershoot problem. The priority of the flows is maintained and the *detection and handling time* is actually decreased to the value of 0.25 s on both paths, i.e., forward and reverse. Again the temporary drop in the data flow, due to NOTIFY messages, is observed. The size of the drop is big because when only 16 Kbps flow sizes are used, a large number of flows have to be terminated to solve the severe congestion.

4.5.1. Conclusions

- If no measures are taken to prevent double termination of flows in the edge nodes then a link undershoot occurs on the reverse and on the forward direction.
- The use of optimizations, i.e., the use of the *with_optimization_2* mechanism, solves the link undershoot on the forward and on the reverse path. The *detection and handling time* is faster when the optimization is used.
- The flows priority is not affected regardless whether optimization is used or not.

5. Conclusions and future work

With the presented research we aim to cover only some of the aspects of the RMD-QOSM protocol behavior described in [23]. This paper tries to answer two main research questions, see Section 1. Section 3 relates to our first research question and discusses three severe congestion mechanisms. The first severe congestion mechanism, denoted as *without_optimization*, was included originally in a previous version of the RMD-QOSM draft. When this mechanism is applied, then flows forwarded on a severely congested path are chosen for termination starting with flows with the biggest reservation size maintained on the same path. However, this mechanism was developed to work with unidirectional reservations and does not perform very well with bi-directional reservations. The second severe congestion mechanism, denoted as *with_optimization_1,* uses a policy, where first flows are terminated that maintain smallest reservations on the path opposite to the severely congested path. In the third severe congestion mechanism, denoted as *with_optimization_2,* an edge node maintains information on the amount of excess rate already terminated by its communicating edge node. By using this information an edge node can more efficiently decide which flows to terminate on the forward path and which flows on the reverse path.

The second research question is discussed in Section 4, where the three different severe congestion mechanisms are evaluated and compared to each other using two performance criteria: *detection and handling time* and the *link load before and after the stabilization*. In particular, this is done when bidirectional reservations, originating from preemption aware services, are in process. A simulation model of the protocol, implemented in ns2, was used.

In the first experiment A, the *without_optimization* severe congestion mechanism was evaluated when reservations with different sizes on the forward and reverse paths were applied. It was observed that the size of the reservations in both paths (forward and reverse) can impact the link utilization severely.

In the second experiment B, the *without_optimization* and the *with_optimization_1* severe congestion mechanisms were analyzed and compared to each other. The experiment results show that the *with_optimization_1* severe congestion mechanism performs better than the *without_optimization* severe congestion mechanism in terms of the selected performance criteria.

The third experiment C aims at evaluating and comparing the operation of the *without_optimization* and the *with_optimization_2* mechanisms. A central issue in this experiment is the termination of bidirectional flows, which leads to the release of bandwidth on both communication paths, i.e., forward and reverse. When the *without_optimization* mechanism is used, an edge node does not have information about how much excess rate (i.e., bandwidth) has been already terminated by its communicating edge node. Consequently, the node might terminate too many flows and eventually cause link utilization undershoot. The *with_optimization_2* severe congestion mechanism uses the available information on the terminated flows (and their reserved bandwidth) and it does not terminate flows unnecessary. In this way the link undershoot issue is avoided and the link utilization is improved.

Regarding future activities, we would like to recommend further research work to be done in the area of the bi-directional reservation. In particular, the research should be extended to include the interoperation between end-to-end signaling and edge to edge signaling and its impact on the severe congestion solutions. Furthermore, it should be evaluated if security attacks could severely impact the operation and performance of the severe congestion mechanisms.

References

[1] Cisco Systems, Quality of Service for VoIP, Available from: <http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/qossol/qosvoip.pdf>, September 2002.
[2] H. Schulzrinne, J. Polk, Communications resource priority for the session initiation protocol (SIP), draft-ietf-sip-resource-priority-10.txt, July 2005.
[3] S.R. Lima, P. Carvalho, V. Freitas, Admission control in multiservice IP networks: architectural issues and trends, IEEE Communications Magazine 45 (4) (2007) 114–121.
[4] R. Braden, D. Clark, S. Shenker, Integrated services in the internet architecture: an overview, RFC 1633, IETF Informational RFC, 1994.
[5] R. Braden, L. Zhang, S. Berson, A. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification, IETF RFC 2205, 1997.
[6] I. Stoica, H. Zhang, Providing guaranteed services without per flow management, ACM SIGCOMM'99, October, 1999.
[7] F. Baker, et al., Aggregation of RSVP for IPv4 and IPv6 Reservations, IETF RFC 3175, September, 2001.
[8] Y. Bernet, R. Yavatkar, P. Ford, F. baker, L. Zhang, M. Speer, R. Braden, B. Davie, E. Felstaine, Framework for integrated services operation over Diffserv networks, IETF RFC 2998, 2000.
[9] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, IETF RFC 2475, December, 1998.
[10] K. Nichols, V. Jacobson, L. Zhang, A two-bit differentiated services architecture for the internet, RFC 2638, July, 1999.
[11] Z. Zhang, et al., Decoupling QoS control from core routers: a novel bandwidth broker architecture for scalable support of guaranteed services, ACM SIGCOMM 2000, 2000.
[12] Charter of the IETF PCN working group, Available from: <http://www.ietf.org/html.charters/pcn-charter.html>.
[13] P. Eardley, Pre-congestion notification architecture, draft-ietf-pcn-architecture-03 (work in progress), February 2008.
[14] Charter of the IETF NSIS working group, Available from: <http://ietf.org/html.charters/nsis-charter.html>.
[15] R. Hancock, et al., Next Steps in Signaling (NSIS): Framework, Request For Comments 4080, 2005.
[16] X. Fu, H. Schulzrinne, A. Bader, D. Hogrefe, C. Kappler, G. Karagiannis, H. Tschofenig, Van den S. Bosch, NSIS: a new extensible IP signaling protocol suite, IEEE Communications Magazine 43 (10) (2005) 133–141.
[17] H. Schulzrinne, R. Hancock, GIST: general internet messaging protocol for signaling, draft-ietf-nsis-ntlp-15, Internet draft, work in progress, February 2008.
[18] J. Manner, et al., NSLP for Quality-of-service signaling, draft-ietf-nsis-qos-nslp-16.txt (work in progress), February 2008.
[19] J. Ash, A. Bader, C. Kappler, D. Oran, QoS-NSLP QSPEC Template, draft-ietf-nsis-qspec-20.txt, Internet draft (work in progress), 2008.
[20] M. Stiemerling, H. Tschofenig, M. Martin, C. Aoun, NAT/Firewall NSIS signaling layer protocol (NSLP), draft-ietf-nsis-nslp-natfw-18.txt, Internet draft (work in progress), February 2008.
[21] F. Dressler, G. Carle, C. Fan, C. Kappler, H. Tschofenig, NSLP for metering configuration signaling, Internet draft (work in progress), October, 2004.
[22] A. Báder, G. Karagiannis, L. Westberg, C. Kappler, T. Phelan, H. Tschofenig, G. Heijenk, QoS signaling across heterogeneous wired/wireless networks: resource management in Diffserv using the NSIS protocol suite, in: The Second International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine), August 2005.
[23] A. Bader, et al., RMD-QOSM, The resource management in Diffserv QOS model, draft-ietf-nsis-rmd-12.txt (work in progress), November 2007.
[24] L. Westberg, et al., Resource management in Diffserv (RMD): a functionality and performance behavior overview, Lecture Notes in Computer Science 2334, Springer-Verlag, 2002.
[25] A. Császár, et al., Comparative performance analysis of RSVP and RMD, Lecture Notes in Computer Science 2811, Springer-Verlag, 2003.
[26] A. Császár, et al., Severe congestion handling with resource management in Diffserv on demand, Lecture Notes in Computer Science 2345, Springer-Verlag, 2002.
[27] L. Westberg, et al., Resource Management in Diffserv Framework, draft-westberg-rmd-framework-04.txt, (work in progress), September 2003.
[28] S. Rai, B. Mukherjee, D.O. Deshpande, IP resilience within autonomous system: current approaches, challenges, and future directions, IEEE Communications Magazine (2005) 142–149.
[29] A. Császár et al., Resilient reduced-state resource reservation, KICS/IEEE Journal of Communications and Networks 7 (4) (2005).
[30] T. Cinkler, P. Demeester, A. Jajszczyk, Resilience in communication networks, IEEE Communications Magazine 40 (1) (2002) 30–32.
[31] J.T. Park, Resilience in GMPLS path management: model and mechanism, IEEE Communications Magazine 42 (7) (2004) 128–135.