SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 10, No. 1, February 2013, 103-111

UDK: 004.738.2 DOI: 10.2298/SJEE1301103M

Platform for Capacity Reservation in IP Networks

Nataša Maksić[1], Aleksandra Smiljanić[1]

Abstract: The fast development of multimedia devices and services causes the need for increase of the transport capacity of packet networks. OSPF-TE uses both the information about network topology and the link utilization when finding the routing path. Accordingly, it might find path even in the cases when the shortest path routing would cause overloaded link and dropped packets. In this paper we develop the platform for capacity reservation in IP networks. We implement OSPF-TE protocol as an extension of the existing OSPF. In addition, the basic functionalities of the reservation protocol and the user interface are implemented. We present the simulation environment for the verification of our implementation and for the analysis of various routing algorithms based on the information conveyed by OSPF-TE.

Keywords: OSPF-TE, Capacity reservation, OpenVZ.

1 Introduction

OSPF is a predominant intra-domain routing protocol. Path calculation in the OSPF protocol uses Dijkstra's algorithm to find the shortest path in a given graph. Its input data are the network topology presented by the network graph and costs of the network links, i.e., edges of the network graph.

RIP routing protocol was widely used before OSPF. In RIP protocol, each router learns from its neighbouring routers about the costs of their routes to all destinations in the network. The router increases the route cost by the cost of its link to the corresponding neighbour. Each router chooses the shortest path to the destination network, and iteratively conveys this information to the neighbouring routers.

In OSPF, each router sends the information about its links and their costs to all other routers in the network, whenever the network topology changes. OSPF allows each router to have full knowledge about the network topology.

Neither OSPF nor RIP use the information about available network resources when finding routing path. OSPF and RIP find the shortest paths in the network based on the network topology and link costs, and all the traffic is conveyed along these calculated paths. Such routing often leads to uneven link

[1]School of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11120 Belgrade, Serbia; E-mails: maksicn@etf.rs; aleksandra@etf.rs

utilizations in the network, and to the congestion of some links while the others remain underutilized. In order to utilize available link capacities, it is needed to include information about them when the routes to destinations are calculated.

OSPF-TE is an extension of the OSPF protocol which includes the information about the network traffic in the path calculation, and it enables capacity reservations [1]. Capacity reservations are performed for the flows of the particular forwarding equivalence classes. For each new flow, path is calculated based on the network topology, link costs, as well as available link capacities. The calculation finds the best path which satisfies the given constrains. In this way, the network can be evenly utilized and its resources can be used to a greater extent.

OSPF-TE is used together with RSVP-TE protocol. RSVP-TE messages are exchanged between routers on the calculated path of a new flow, and the reservation of the capacity required by the new flow is performed in each router. Upon the reservation, control data with the updated link utilizations is exchanged between routers using OSPF LSA messages, so that each router is informed about current available link capacities.

In IP networks, flow paths can be implemented using source routing. In that case, the role of the RSVP-TE protocol is to perform capacity reservation in the routers along the path calculated by OSPF-TE. Responsibility of the flow end points is to properly generate source routing headers and to send the packets along the reserved path.

MPLS is a widely used protocol for traffic engineering. Besides the elimination of security problems introduced by source routing [2], MPLS routing is more efficient, because it does not require transmission of the source routing headers. In MPLS networks, RSVP-TE protocol encompasses capacity reservation, configuration of the MPLS labels on the links along the paths and configuration of the forwarding tables in the routers along the paths [3]. In MPLS networks, the paths are calculated by OSPF-TE or similar protocols (such as IS-IS-TE).

The OSPF-TE standard specifies the control information which routers need to send to other neighbours [1]. The control information includes the control information defined by the OSPF standard, the available link capacities and possibly other link metrics as well (e.g., delay). Routers exchange the control information using TE LSA messages defined in RFC 3630 [1]. However, standard does not define a particular algorithm for the path calculation based on the control data. The path computation algorithm does not need to be performed simultaneously in all routers in the network, as in OSPF. Possibility of using different algorithms for the calculation of OSPF-TE path enables the optimizations using different criteria.

This paper presents our implementation of the OSPF-TE protocol in C++ programming language. We designed the software that can achieve the high execution speeds and which can be integrated with other parts of the router's control plane. In the process of validation, our design was integrated with the ospfd software, created by John Moy [4]. Implemented software encompasses generation and processing of TE LSA packets, as well as the path calculation algorithm. At the same time, implemented software represents the platform for the analysis of different algorithms for the path calculations as a part of traffic engineering. In this paper, we implemented the algorithm which eliminates links with insufficient available capacities before applying Dijkstra's algorithm for the path calculation. In this way, the algorithm calculates the shortest path with sufficient available link capacities. Also, this paper presents the protocol for the capacity reservations and its implementation, as well as the application which enables the network operator to enter the requests for capacity reservations.

The second section describes the implementation of our platform for capacity reservations in IP networks. The third section presents the simulator used for the validation of the implemented software. Results of the validation are presented in the fourth section. The fifth section discusses the future steps.

2 Implementation of the Platform for Capacity Reservations

Platform for capacity reservation encompasses: the OSPF-TE protocol, the protocol for capacity reservation based on RSVP-TE, and the management application for initiating bandwidth requests. The block scheme of the system is shown in Fig. 1.

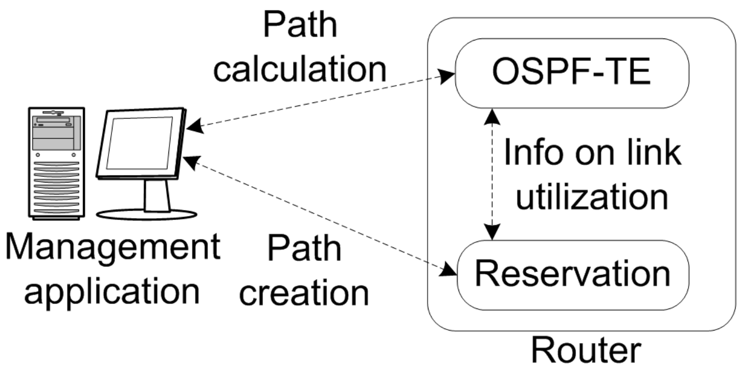

Fig. 1 – *The block scheme of the system for capacity reservations.*

The management application enables operator to establish the paths through the network for the given end-points and flow capacity. The management application can be executed on any general purpose computer.

Other system components, the OSPF-TE protocol and the protocol for capacity reservations are executed on the Internet router.

Our implementation of the OSPF-TE protocol is based on the implementation of the OSPF protocol created by John Moy [5]. In this paper, we added the following functionalities to the existing implementation of the OSPF protocol: creating, sending and receiving OSPF-TE LSA packets, storing and processing the information from these packets, path-computation algorithm for the given constrains as well as the interfaces with the management application and the module for capacity reservations.

OSPF-TE LSA packets, which are generated by routers, convey the information about available link capacities which are obtained from the module for capacity reservations executed on that router. This information is distributed to all OSPF-TE routers in the network by means of LSA mechanism. In this way, each OSPF-TE router is aware of the available capacities in the network and, it is able to calculate paths upon the requests from the management application. The path calculation is performed by using Dijkstra's algorithm which accounts only links with sufficient available capacities to accommodate a new bandwidth request. In other words, the shortest path is found in the graph without links with insufficient capacities.

OSPF-TE and the module for capacity reservations communicate using the UDP protocol. After each successfully finished reservation, module for capacity reservation sends the updated information about available link capacities to the OSPF-TE module which typically run on the same CPU. The OSPF-TE module communicates with the management application using the TCP protocol. The request for path calculation is conveyed from the management application to the OSPF-TE module. In the opposite direction, information if there are sufficient available resources through the network is conveyed from the OSPF-TE module to the management application. In the TCP connection, the OSPF-TE module represents the server. Since all OSPF-TE routers have the information about available capacities of all other routers in the network, the management application can be connected to any OSPF-TE router in the network. We obtained reliable connection through the network by means of the TCP protocol. On the other hand, the OSPF-TE module and the module for capacity reservations are typically located on the same router, which is why we chose the simpler UDP protocol for their communication.

After receiving the information about the calculated path from the OSPF-TE module, management module sends the request for path establishment to the module for capacity reservations on the first router of the path. The path creation itself contains two phases: the first phase of the preliminary reservation and the second phase of the final capacity reservation. In the first phase, request for the capacity reservation is conveyed through all routers on the path, starting from the first router. For each link on the path, the available link capacity is checked, and the requested capacity is preliminarily reserved if available. In the case when links have sufficient capacities and all preliminary reservations are successfully completed, an acknowledgement of successful reservation is sent through all routers on the path, from the last to the first router. A router on the path makes a final reservation when it receives the acknowledgement packet. When the acknowledgement arrives to the first router, the information about the successfully completed reservation is sent to the management application. Exchange of the control information needed for the path establishment is performed by using UDP packets in order to avoid establishment of multiple TCP connections.

The process of capacity reservations is shown in Fig. 2. The content of the control packets which are exchanged between routers during the path establishment are shown next to the links. The path goes through routers' ports with IP addresses which are marked by symbols A, B, C and D. Field *Hop count* represents the number of router's ports on the path which is calculated by OSPF-TE. The goal of the field *Type* is to distinguish between the packet which is used for preliminary reservation (*Type* = 0) and the returning packet. Field *Status* has useful information in the case of the returning packet: the reservation can be successful (*Status* = 1) or not (*Status* = 0). Field *Capacity* carries the requested path capacity. Inside the field *Path*, we store an array of router's addresses through which the capacity is reserved, and which are calculated according to OSPF-TE. Field *Next hop* contains the index of the router's address inside the field *Path*, taking into account that the first address has index 0. Index contained in the field *Next hop* points to the address to which the next router should forward the request. The value of index beyond the bounds of the array *Path* represents the end of the path. In Fig. 2, packets which have the field *Next hop* equal to 5 and –2 indicate to the last and the first router respectively that they are the end points of the calculated path. The field *Path* contains the addresses with indices from 0 to 3.

If any of the links along the calculated path has insufficient capacity during the capacity reservation, process of the capacity reservation is stopped. Then, the message, which conveys the indication of error (*Status* = 0), is sent from the current toward the first router. In the routers along the path, the preliminary reservations are being cancelled as they receive the control packets with *Status* = 0. When this message comes to the first router on the path, the information about unsuccessful path establishment is sent to the manager. The unsuccessful capacity reservations are possible when multiple managers generate requests simultaneously. Failure of the reservation process can happen only when the available link capacity is lower than the sum of the capacity requests for the concurrent reservations. If the request for the capacity reservation is not realized, an operator can send the capacity reservation again. Then, an alternative path might be found.

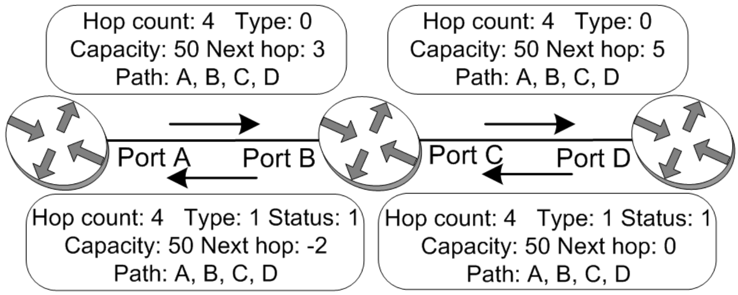

Fig. 2 – *Capacity reservation.*

3 Simulation Network

In this section, we will describe the simulation network which has been used for the validation of the platform for capacity reservations. The simulation network is based on the virtualization technique in which virtual machines represent the routers. Since several virtual machines can be executed on one computer, this technique enables cost-effective simulation environment. Furthermore, software inside the virtual machine is executed in the same way as on the physical computer. As a consequence, it is not necessary to modify software for the simulations as it is the case with the network simulator ns-2 [6].

We use the virtualization software OpenVZ, which is executed in the Linux operating system [7]. In OpenVZ, the virtual machines share common operating system with physical computer. Such virtualization requires less resources and enables execution of more virtual machines on the same physical computer compared to other virtualization methods.

The topology of the simulation network used for the validation of our platform for capacity reservations is shown in Fig. 3. Each router shown in Fig. 3 executes on one virtual computer. Software modules implementing the OSPF-TE protocol and the protocol for capacity reservations execute inside each router. During the simulation, all virtual routers were executed on one physical computer. Connections between virtual machines were created using Linux command *brctl*, which creates bridges in the kernel of the operating system. Network ports of the virtual machines are connected to the ports of bridges. We wrote bash script for automated creation of bridges in the Linux operating system, initiating virtual machines, configuring the network ports of the virtual machines as well as the automated initiation of the OSPF-TE module and the module for capacity reservations [8]. Furthermore, the script which stops the virtual machines was written.

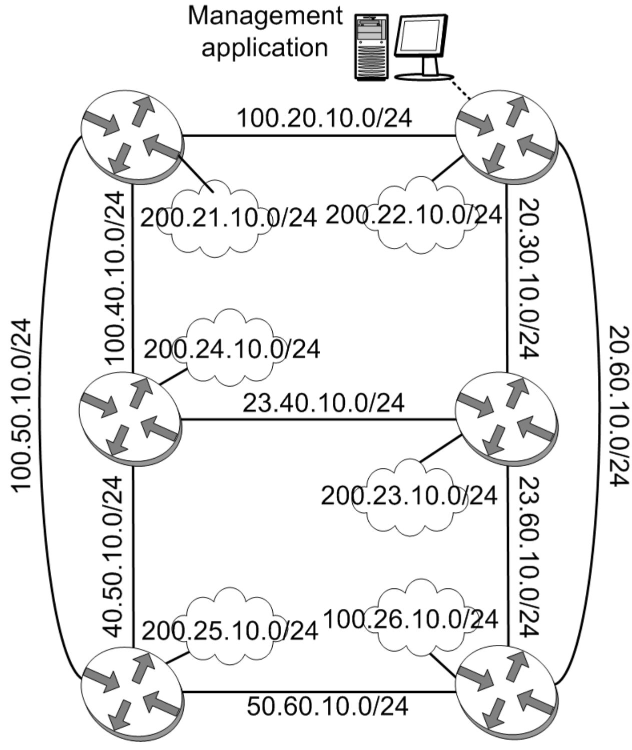

Fig. 3 – *Simulation network.*

In the simulation network, the management application is executed on the separate computer. The process of the path establishment, in order to validate the platform, is automated by bash script. Bash script generates path end-points randomly. These end-points are represented by the addresses from the stub local networks connected to the routers. Then, implemented script issues the request for path establishment to the management application. Based on that request, the management application sends the request for path creation to the OSPF-TE router. Choice of the router is configurable. In our simulation, the request is sent to the router with a direct connection to the manager without loss of generality. If the OSPF-TE router finds the path, the management application forwards the request for path establishment to the first router on the calculated path. Then, the path establishment continues as it is described in the previous section. If OSPF-TE cannot find the path due to the lack of available capacity, the management application will forward that information to the validation script. Information on successfully established paths is also forwarded to validation script. Validation script saves the received information for further analysis.

4 Validation Results

Our platform for bandwidth reservations was validated using the simulation network presented in Fig. 3.

Capacities of the links in the simulation network were set to 1000Mbit/s, and requests were generated to demand capacities of 50Mbit/s. The simulation consisted of multiple runs, and each assumed the network without any reservations. During each run, the validation script generated requests for finding and creation of the path iteratively. Generation of the requests was stopped when the first request was dropped due to the lack of available capacity in the network. **Table 1** presents the number of created paths in each run of the simulation. We measured the remaining available capacity of the links in the network after the termination of the simulation run. Mean value of the available capacity of network links at the end of the run varied between 16% and 26%.

Table 1
The number of created connections before the first dropped request.

Simulation run	1	2	3	4	5	6
Number of created connections	187	186	205	181	209	188

Goal of the presented simulation was to validate functionalities of the software components included in the platform for capacity reservations. Improvement of the path-calculation algorithm might improve the obtained results.

5 Following Steps

Presented platform for capacity reservation is developed with two goals in mind. Main goal of the platform is the deployment within the high-capacity router which has been developed in our lab. Since the router will feature MPLS functionality, the capacity reservations will be performed by the full implementation of RSVP-TE protocol which is being developed as the part of the same project. Our platform might be used for the design of optimized path-calculation algorithms. We plan to integrate our solution into some of the existing open-source routing platforms, such as Xorp.

6 Conclusion

This paper presents the implementation of the platform for capacity reservations in IP networks, which incorporates implementation of the OSPF-TE protocol, the protocol for capacity reservation, and the interactive management application developed for the network operator. Platform can be used in routers of different types, ranging from software Linux routers to the high-performance hardware routers.

Described test environment confirmed correct functioning of our implementation. Presented test environment can be used for the verification and evaluation of different algorithms for bandwidth reservations.

6 Acknowledgment

This work was supported by the Serbian Ministry of Science and Education, and companies Telekom Srbija and Informatika.

7 References

[1] D. Katz, K. Kompella, D. Yeung: RFC 3630 – Traffic Engineering (TE) Extensions to OSPF Version 2, Sept. 2003. Available on: http://www.ietf.org/rfc/rfc3630.

[2] S.M. Bellovin: Security Problems in the TCP/IP Protocol Suite Variable, Computer Communication Review, Vol. 19, No. 2, April 1989, pp. 32 – 48.

[3] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow: RFC 3209 – RSVP-TE: Extensions to RSVP for LSP Tunnels, Dec. 2001. Available on: http://www.ietf.org/rfc/rfc3209.txt.

[4] J.T. Moy: OSPF Complete Implementation, Addison-Wesley Professional, Reading, MA, USA, 1998.

[5] OSPFD Routing Software Resources. Available on: www.ospf.org.

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

[7] http://wiki.openvz.org.

[8] http://www.gnu.org/software/bash/.