Available online at www.sciencedirect.com
ScienceDirect
Simulation Modelling Practice and Theory 15 (2007) 426–448
SIMULATION MODELLING PRACTICE AND THEORY
www.elsevier.com/locate/simpat

Providing QoS in OSPF based best effort network using load sensitive routing ☆

Anunay Tiwari, Anirudha Sahoo *

Kanwal Rekhi School of Information Technology, Indian Institute of Technology Bombay, Powai, Mumbai 400076, India

Received 24 February 2006; received in revised form 31 July 2006; accepted 24 November 2006
Available online 16 December 2006

Abstract

In an open shortest path first (OSPF) based best effort network, when a packet experiences congestion, the routing subsystem cannot send it through an alternate path. Thus, it fails to provide desired quality of service (QoS) during congestion. In order to provide QoS we have reported three different load sensitive routing (LSR) protocols in [A. Sahoo, An OSPF based load-sensitive QoS routing algorithm using alternate paths, in: IEEE International Conference on Computer Communication Networks, October 2002; A. Tiwari, A. Sahoo, Providing QoS support in OSPF based best effort network, in: IEEE International Conference on Networks, November 2005; A. Tiwari, A. Sahoo, A local coefficient based load sensitive routing protocol for providing QoS, in: IEEE International Conference on Parallel and Distributed Systems, July 2006]. The LSR protocol forwards packets through alternate paths in case of congestion. The number of alternate paths at any node depends on the value of operating parameter or coefficient used for alternate path calculation. Though the basic protocol in these cases was the same, the methods of choosing operating parameter were different. We referred to these three methods as LSR [A. Sahoo, An OSPF based load-sensitive QoS routing algorithm using alternate paths, in: IEEE International Conference on Computer Communication Networks, October 2002], E-LSR [A. Tiwari, A. Sahoo, Providing QoS support in OSPF based best effort network, in: IEEE International Conference on Networks, November 2005] and L-LSR [A. Tiwari, A. Sahoo, A local coefficient based load sensitive routing protocol for providing QoS, in: IEEE International Conference on Parallel and Distributed Systems, July 2006] coefficient methods. In this paper, we present the LSR protocol along with the three coefficient calculation methods pointing out the reason for going from one method to the next. The main strength of our LSR protocol is that it provides loop free alternate paths in the event of congestion and can interwork with routers running vanilla OSPF protocol. We show through simulation that the LSR protocol based on any of the three different coefficient calculation methods performs much better than OSPF and that out of the three methods proposed by us, L-LSR performs the best.

Keywords: QoS routing; OSPF; Best effort network; Load sensitive routing; Loop free

☆ This research was partially supported by Industrial Research and Consultancy Centre, IIT Bombay under Grant Number 03IR059.
* Corresponding author.
E-mail address: sahoo@it.iitb.ac.in (A. Sahoo).

1569-190X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2006.11.008

1. Introduction

There has been an upsurge in real time applications like Voice over IP, video streaming on the Internet. These applications require quality of service (QoS) to perform satisfactorily. But the current Internet is built on *best effort* infrastructure. Hence there is need for providing QoS on top of best effort network. Usually it is relatively easy to provide QoS to real time application when the application starts. But it is quite difficult to repair the QoS when QoS deteriorates in the middle of running of the application. For example, there are few mechanisms available to provide QoS to VOIP calls when a request for call arrives. Cisco VOIP gateways have Call Admission Control mechanisms in place to admit calls with an accepted level of QoS at the time of call arrival [4]. But when a VOIP call is already connected and the two parties are in conversation, if the QoS of the call deteriorates, then *mid call routing* should be used to reroute the call in a different path to repair the QoS. This should happen transparently without affecting the call. But there is no satisfactory method for providing mid call routing to VOIP or video applications. One effective way would be to provide mid call routing support at the routing layer.

Typically, routing sub-system uses shortest path algorithm [5] like OSPF to route packets. But the routing decision, in this case, is solely based on the destination address of the packets. Hence, packets for a particular destination follow the same path, even though there may be better alternate paths available. Thus, QoS demand of the packets are not considered while routing the packets. If routing protocol can provide support for routing packets along alternate paths, then real time applications like VOIP can perform satisfactorily when the shortest path gets congested. Obviously, this can be exploited for mid call routing.

But routing the packets through better alternate paths is not as straight forward as it may look. One of the challenges is to make the alternate path loop free. If the alternate path protocol is not loop free, then a separate loop detection mechanism has to be put in place. This approach may not be attractive to implementors, since that would mean changing the packet forwarding engine.

In this study, we propose a routing protocol that uses alternate paths to provide QoS along OSPF paths. The network is assumed to be running a link state routing protocol like Open Shortest Path First (OSPF) [6]. Given an OSPF path from a source node to a destination node, the protocol tries to find alternate paths for nodes along the OSPF path. When a node experiences congestion on an outgoing link, it sends congestion notification to all its neighbors except the one connected to it over the congested link. The congestion notification is not flooded, rather it is restricted to only one hop neighbor of the congested node. This node as well as the neighboring nodes then forward packets through alternate paths. The alternate paths are chosen in such a way that the packets do not end up in a loop. Once congestion is over, then the nodes involved in alternate path routing revert back to OSPF routing. Thus, the protocol proposed is very simple, yet is quite effective in providing QoS. But the performance of the protocol depends on being able to find alternate paths for nodes. However, a node cannot arbitrarily choose any neighbor as alternate next hop, rather it has to do so such that the alternate path does not form a loop. The loop free property makes the implementation of the protocol simple, because it does not require a separate loop detection mechanism in the packet forwarding engine. The loop free property of this routing protocol is achieved by adhering to some packet forwarding properties of OSPF protocol, which is loop free. More the alternate paths the better will be the performance of the protocol. The number of alternate paths depends on how the parameter (or coefficient) for finding alternate path is fixed. We present three different methods of finding the parameter. While the basic protocol remains the same the number of alternate paths and the distribution of alternate paths among the nodes change based on the method used. We refer to the protocol as LSR protocol, but choice of alternate path during congestion will change depending on how the operating parameter or coefficient is chosen. Accordingly, we refer to the coefficients as *LSR coefficient* [1], Efficient LSR (*E-LSR coefficient*) [2] and Local LSR (*L-LSR coefficient*) [3]. Note that the operating parameter or coefficient is only calculated one time after OSPF has converged. The same coefficient is used until the topology of the network changes (say due to a link failure), at which time the coefficients are recalculated. Thus, the overhead of calculation of coefficients is quite minimal. In this paper, we discuss how the different coefficients are calculated and present the performance of the protocol when run with different coefficients and compare them with OSPF.