

Intra-Domain Delay-Based Quality of Service using Differentiated Routing

Ioannis Papanagiotou and Michael Howarth

Centre for Communication Systems Research,
Department of Electronic Engineering, University of Surrey,
GU2 7XH Guildford, United Kingdom,
{I.Papanagiotou, M.Howarth}@surrey.ac.uk
<http://www.ee.surrey.ac.uk/CCSR/>

Abstract. Differentiated routing is an approach to providing service differentiation in networks, a field that is currently receiving significant research attention. In this report we present an algorithm, namely Intra-Domain Differentiated Routing (IDDR), which supports qualitative delay differentiation in IP networks. We review existing differentiated routing approaches and then introduce IDDR and present initial results. We demonstrate that using IDDR we can achieve qualitative delay differentiation for two classes of flows.

1. Introduction

The demands placed on computer networks continue to increase, with increased development of multimedia applications and distributed data processing and retrieval systems. These different applications place different requirements on the underlying network. The need for this quality of service (QoS) differentiation has led to the devising of mechanisms which consider flows' requirements before routing them on the network.

The most popular QoS parameters that are relevant for packet-level traffic characteristics are latency, jitter and loss probability, and also bandwidth. Streaming multimedia may require guaranteed bandwidth to ensure that a minimum level of quality is maintained. IP telephony, Voice over IP (VoIP) and video teleconferencing (VTC) require strict limits on jitter and delay. Low delay is essential as it reduces the lag imposed by unforeseen network conditions. Video also requires a low packet drop rate since a single packet loss can give rise to unwanted artefacts on the screen which degrades the video quality. This paper focuses on delay as the QoS metric.

QoS in IP networks has traditionally been provided using differentiated forwarding. Packets from flows that require different QoS are routed along the same paths but are given different forwarding treatment. Packets assigned higher priority (high QoS packets) are forwarded faster than best-effort (BE) packets. Differentiated forwarding has historically been based on one of two frameworks, namely Integrated Services (IntServ) and Differentiated Services (DiffServ). Integrated Services works on a per-flow basis, i.e. it serves each flow differently according to its needs.

Differentiated services classify all flows entering a network into one of a predefined number of QoS classes. Differentiated forwarding has not been widely implemented since it has generally been perceived by both the research community and network operators to be too complex.

Differentiated routing is an alternative that is currently receiving attention and is the focus of the work described in this paper. Using differentiated routing, packets of different QoS classes follow different paths to their destinations. In the literature there are papers where it is claimed that differentiated routing can be used to improve the performance of a network in terms of delay compared to a network running pure Shortest Path First (SPF) routing algorithms [2, 4, 5, 6].

There are two major concerns regarding differentiated routing in an IP environment. Firstly, by routing flows through alternative longer paths we increase the network load, because any individual flow is using more of the network's resources. Secondly, if each router dynamically chooses the next hop independently for each packet to a given destination, undesirable traffic shifts may occur which might in turn lead to service degradation. This for example might result in out-of-order delivery of TCP packets, causing a drop in goodput; or it might result in significant jitter in UDP packet delivery, reducing the effective QoS. We overcome this issue by routing packets of the same flow through the same path.

It is the key hypothesis of the research described in this paper that differentiated routing provides a promising platform for the delivery of QoS. Our objective is to develop mechanisms for differentiated routing for quality of service while minimising or avoiding the two potential drawbacks described above.

In this paper it is shown that differentiated routing can be used to provide delay differentiation between two classes of flows. Our algorithm, Intra-Domain Differentiated Routing (IDDR), is based on the Shortest Path First with Emergency Exits (SPF-EE) algorithm originally developed by Wang and Crowcroft as a mechanism for avoiding congestion [1]. We here extend the algorithm and refocus it for delay-based QoS. In Section 2 we consider previous approaches to QoS using differentiated routing and generally improvements in network performance achieved using differentiated routing. Section 3 describes the IDDR algorithm. Section 4 describes results from a simulation of IDDR, showing the feasibility of delay differentiation using differentiated routing. Finally, Section 5 provides conclusions and describes future work which would improve IDDR's performance.

2. Related Work

This Section reviews a number of algorithms that have been used either for intra-domain service differentiation or for optimised network performance as regards delay and resilience. We initially discuss algorithms that are intended to run on plain IP environments; these are followed by algorithms that make use of Multi-Topology Routing (MTR), and finally those intended for MPLS enabled networks.

2.1. IP-based routing

Wang and Crowcroft in [2] consider the problem of finding paths that satisfy multiple constraints. Both single mixed metric and multiple metric solutions have been evaluated. A single mixed metric is a function of two or more metrics, such as delay and bandwidth, and can only be used for qualitative QoS at best since it acts only as an indicator in path selection. Multiple metric solutions constitute an NP-complete problem when two or more additive or multiplicative constraints are combined. It is shown that it is computationally feasible to utilize bandwidth with any one other additive or multiplicative constraint i.e. this is not an NP complete problem. QoS algorithms that could be integrated in the widely used Open Shortest Path First (OSPF) [3] protocol are presented in [4]. These algorithms differ in complexity with accuracy being the trade-off, and consider only bandwidth and hop-count. In [5-6] Sahoo presents a Load Sensitive Routing (LSR) algorithm using alternate paths. Although the paper's title refers to QoS, the work does not appear to provide service differentiation among groups of flows; instead, it is shown that LSR outperforms OSPF when it comes to delay and jitter.

2.2. Multi-topology routing

Multi-topology routing (MTR) has several planes, with different link weight configurations for the single network topology; this yields as many routing topologies. MTR is considered to be very effective for network resilience by Menth et al. [7]. In their work they point out that currently resilience in IP networks relies mainly on reconvergence after a node failure via the periodic exchange of link state information. MTR offers improved resilience by providing backup paths in case of node failures. Gjessing [8] presents two existing methods for network resilience in IP networks, both of which make use of backup topologies, namely Resilient Routing Layers (RRL) and Multiple Routing Configurations (MRC) [9-10]. The difference between those approaches is that RRL omits certain links in backup topologies while MRC sets a high link costs to them. MTR is being considered as an approach for differentiated routing based on the notion of Network Planes [11]; these may be interconnected across multiple domains to create Parallel Internets that provide differentiated QoS.

2.3. MPLS-based algorithms

MPLS was originally developed to provide faster packet forwarding than traditional IP routing [12]. Xiao et al. [13] describe a path computation algorithm using Constraint Based Routing (CBR) which works both online and offline. A QoS routing scheme using MPLS is presented in [14]. The algorithm utilizes both routing and forwarding differentiation. To account for delay and jitter, Weighted Fair Queuing has been used. Routing differentiation is used for network resilience, but unfortunately the authors do not provide comparison with any well established routing protocol. Wang [15] presents another algorithm which performs service

differentiation according to bandwidth. The assumption is that all packet-level service requirements such as delay, jitter and packet loss rate can be translated into an equivalent bandwidth requirement. The algorithm describes the importance of critical links: the impact of one Label Switched Path (LSP) request on future LSP requests needs to be considered. In [16] Calle et al. present an algorithm that is based both on failure probabilities and failure impact of particular segments in a network. Even though it has proved to be useful it is based on statistical data regarding network failures and therefore cannot handle unpredictable traffic behaviour.

3. Intra-Domain Differentiated Routing (IDDR)

In this Section we present our algorithm, IDDR, which provides service differentiation in terms of delay. The algorithm is based on Shortest Path First with Emergency Exits (SPF-EE) [1]. The latter is intended to improve network resilience, as it modifies plain Shortest Path First (SPF) routing in the event of congestion. SPF-EE is dynamic and highly adaptive to network changes. The algorithm requires routers to keep routing trees of all neighbouring routers, in order to create on-demand alternative paths for flows heading towards congested links. SPF-EE uses these alternative paths - or emergency exits as the authors call them so as to avoid congestion. Although the algorithm was originally intended to address network congestion, we show in this paper that it can be adapted and enhanced to provide delay differentiation to classes of flows. We now explain SPF-EE in brief, since it constitutes the fundamentals of IDDR. We then proceed to describe our modifications to the algorithm which result in IDDR.

3.1. Shortest Path First with Emergency Exits (SPF-EE)

SPF-EE takes as its starting point conventional shortest path routing using Dijkstra's algorithm. SPF-EE then extends OSPF: each node makes use of the data available in the Link State Database (LSDB) and derives routing trees for itself and each of its neighbours. Consider node A in Fig. 1 running SPF-EE. Using the derived routing trees, the node A calculates for each neighbour the next hop of the alternative path to each and every destination; by browsing each neighbour's routing tree, except the tree of the node that is the SP next hop, node A then checks whether the destination is on a sub-tree rooted under node A itself or the Shortest Path (SP) next hop. If it is neither then the neighbour is considered to be the next hop Alternative Path (AP). Each node will then produce a routing table similar to the one produced using OSPF, but with one more field to record the alternative paths. Fig. 1 illustrates an example of a flow travelling from node A to node F when link A - B is congested. Node A will browse the routing trees of its neighbours, omitting node B's routing tree since B is the next hop of the SP from A to F. The valid candidates for next hop of an AP are nodes C and D. However, in the routing tree of node D, destination node F is on a subtree rooted under source node A; therefore it is invalid because if the packet were forwarded from A to D, D would then forward the packet

back to A, creating a loop. As a result the only valid AP in this case is the one through node C, which can forward a packet on to F.

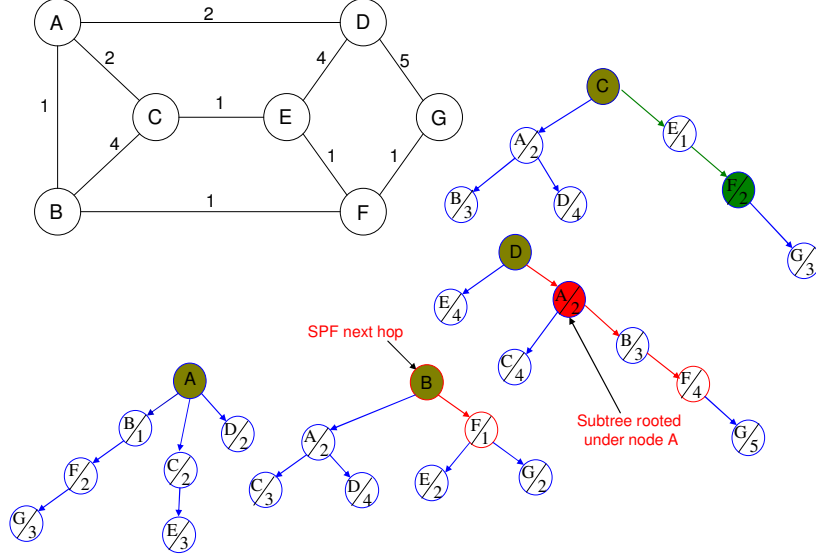


Fig. 1. Example topology and routing trees of node A and its neighbours.

If there is no alternative path for a specific destination a Reverse Alternative Path (RAP) is set up, using the following mechanism: a query message is sent to the current node's neighbours, which in turn look in their neighbours' routing trees for an AP. If an AP is found a reply is sent back to the current node, identifying the neighbour which found the AP and the next hop of the AP from the point of view of the neighbour node. If a RAP cannot be found from the immediate neighbours, the latter send query messages to their neighbours and the procedure goes on until a RAP is established or it is decided that the procedure is too costly to continue. The final reply message that the originating node will receive will include the whole path to the exit. Upon a successful RAP establishment, RAP tables in both the current and exit node are updated.

In SPF-EE, path selection (i.e. the use of either the standard SP or the AP) depends on the length of the outgoing queues of each node. Under normal light loading of the network, packets are forwarding using the standard shortest path. However, if an outgoing queue of a node, e.g. outgoing queue of node A towards node B for destination F (Fig. 1), reaches a certain predefined threshold, then node A redirects the packets to the AP, in this case node C. If the queue to the AP is also above the threshold, then the node looks – in real time – for a new AP and if it does not exist it tries to establish a RAP. If no APs or RAPs can be found for a specific destination then traffic is forwarded to the SP next hop.

In summary, we can see that SPF-EE provides both a shortest path and one or more alternative paths, which in general involves more hops. We hypothesise that this

differentiation of paths can be used to provide delay-based differentiated routing; we therefore adapt SPF-EE as described in the following Section.

3.2. IDDR: Adapting SPF-EE for QoS

We aim to provide two qualitatively differentiated classes of QoS flows: high QoS, i.e. low delay; and best effort, i.e. higher delay. The main difference between IDDR and SPF-EE lies in the path selection. IDDR allows best effort (BE) traffic to be routed *only* through the alternative paths (APs), while high QoS flows are routed through the shortest paths (SPs). If the SPs are over-utilised the additional high QoS flows are also allowed to use APs. The higher cost (usually longer) paths taken by the BE traffic increase the total delay of the best effort flows, increasing service differentiation. To avoid problems such as out-of-order delivery of TCP packets or UDP jitter, packets of the same flow need to follow an identical path.

A feature of IDDR that results in improved network resilience is the computation of all APs along with the SP for each destination during network convergence (i.e. at the time that link state information is propagated through the network); therefore, the lag imposed during path selection is minimised. A real time computation of a secondary AP, as in SPF-EE, would cause further delay and possibly would result in an increase of dropped flows, since during computation the flows would still be routed through the congested link.

IDDR works as follows. After having received link state information about the network, nodes run Dijkstra's algorithm [17] in order to calculate the shortest paths to all destinations. Each node builds its own routing tree and those of its neighbours. Each leaf of the tree consists of the node's ID and the cost of the path from the root of the tree to the node, as shown in Fig. 1. Each node will then find every available AP, following the procedure explained in Section 3.1. The current node retrieves the cost to the neighbour which constitutes an exit and then it adds it to the cost between the neighbour and the destination; therefore each node can deduce the cost of each AP, if more than one, which are then classified and inserted to their routing table in order of ascending cost.

To limit the traffic volume on links used by high QoS flows we introduce a parameter, the IDDR threshold " n ", which bounds percentage link utilisation by the value of " n ". The introduction of this link utilisation gives the network operator some control over the delay of the high QoS flows. The reduced shortest path traffic volume reduces the delay suffered by the high QoS flows. This method using the IDDR threshold, by its nature, also reduces the throughput of the network since it effectively reduces the capacity of its affected links.

The features described above have transformed a congestion avoidance algorithm, SPF-EE to a delay differentiation algorithm, namely IDDR.

4. Simulation Design, Results and Analysis

We now describe the simulation of IDDR. The software, used to simulate the operation of IDDR in a QoS-enabled network, is implemented in C++.

To model the delays encountered by the flows we have initially used a simple model from queuing theory, where the arrival time $1/\lambda$ and service time $1/\mu$ are negative exponentially distributed (Poisson process). For each outbound link in each node we model the link delay as a function of link utilisation and link capacity using the following formula:

$$t_q = \frac{\rho}{1-\rho} \quad (1)$$

where: $\rho = \lambda/\mu$ is the link utilisation and t_q is the link delay.

Whilst this simple model does not take account of the typical heavy tail distribution found in Internet traffic flow [18], it nonetheless provides useful initial insights into the performance of the proposed IDDR algorithm.

4.1. Test topologies

The topologies used to generate our results are: (a) five random topologies created by the BRITE topology generator [19], (b) the test topology by Calle [20] depicted in Fig. 12, and (c) the real topology of the Géant research network [21]. The Géant Topology is of particular interest since it is a real-world network, which implies that its settings incorporates more sophisticated TE, even if that has the objective of optimising the SPF algorithm operation, and not that of IDDR.

The random BRITE topologies each consist of 30 nodes with link connectivity using the Waxman model. Each node is randomly placed in space, has at least 2 links attached and the link capacities take values from 10 to 100 units with those values being uniformly distributed. In Section 4.3 we present the results of one of those BRITE topologies since results on all of them are very similar. Calle's topology consists of 15 nodes, 5 of which are interconnected with high capacity links and form the core of the topology. In particular, in Fig. 2, the bold core links have a capacity of 100 units while the remaining links each have a capacity of 50 units. Finally the Géant topology retains all its characteristics unchanged such as the link weights and capacities.

Our link weight settings for both the BRITE and Calle topologies adopt the standard traffic engineering approach of assigning the link weights for each link to be inversely proportional to the capacity of the link. As mentioned above, for the Géant topology we use its real link capacities and link weights.

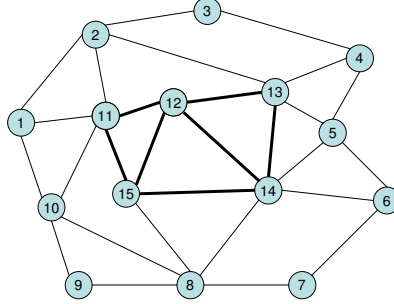


Fig. 2. Calle's Topology, from [20].

4.2. Metrics

Before we proceed to present simulation results, we introduce the following terms:

- *Total traffic* is the sum of all the data rates (bandwidths) of flows injected into the network;
- *QoS ratio* is the percentage of total traffic that requires preferential treatment, i.e. is "high QoS" or low delay traffic.

The total traffic injected to the BRITE random topologies and Calle topology can vary in the simulations. We have varied the number of flows, and the bandwidth of each flow, as well as the QoS ratio. The total traffic is randomly generated from any node to any other node in the case of random BRITE topology. For the Calle topology we have introduced the notions of edge nodes and core nodes. The former are the routers that form the perimeter of the topology (i.e. nodes 1-10 in Fig. 2) while core nodes are all other routers, i.e. those that are not on the perimeter (nodes 11-15 in Fig. 2). Traffic in Calle topology is randomly generated from any edge node to any other edge node in case of the Calle topology. That way we can gain insights as to how IDDR performs on networks that mainly handle transit traffic. For the Géant topology a real traffic matrix from TOTEM has been used [22].

Our analysis of the results is based on the following metrics:

- *Delay*: the sum over all the links along the path used by any individual flow of the queuing delays experienced on each link, as given by equation (1) above. This enables us to depict the extent of service differentiation IDDR can offer.
- *Throughput*: the actual volume of traffic that is successfully passed by the network; because in some tests the utilisation of some links reaches the limiting capacity, some traffic can not be accommodated on the network.
- *Network Utilisation*: defined by the following formula

$$\eta = \frac{\sum b_i}{\sum c_i} \quad (2)$$

where η is the network utilisation, b_i is the traffic volume on link i , and c_i is the capacity of link i . Network utilisation is therefore a measure of the fraction of total network resources that are being used. It illustrates the demand of each algorithm on the network resources

- *Network Utilisation per Flow*: the network utilisation divided by the throughput. This is therefore a measure of the network resources that each flow consumes.

4.3. Results

We have run the simulations for various scenarios. All results presented here have the QoS ratio set to 50%. We have assumed that with the rapidly evolving multimedia applications, which flood networks with real time traffic, the ratio of the total traffic that will need preferential treatment will approach 50%. The total traffic injected on BRITE and Calle's topologies has been set to 2000 and 900 flows respectively. Each flow has a bandwidth of 1 unit. These numbers were chosen so that when SPF runs on those topologies both network utilizations are around 50%. In Géant Topology we have multiplied the bandwidth of each flow by a factor in order to achieve the desirable network utilization. The above settings apply for the results depicted in Figs. 3-5.

Fig. 3 illustrates the percentage delay difference between high QoS flows and BE flows, given by:

$$\text{Delay difference} = 1 - \frac{\delta_{QoS}}{\delta_{BE}}$$

where δ_{QoS} is the mean delay of all QoS flows and δ_{BE} is the mean delay of all BE flows. This shows the percentage reduction in delay encountered by the high QoS flows compared to the delay encountered by the BE flows. It is evident that in all cases even for an IDDR threshold close to 1 (i.e. when links carrying QoS traffic are allowed a high utilisation) a reasonable delay differentiation is found. For the BRITE network the delay differentiation decreases significantly as the IDDR threshold increases. In the Calle topology the delay differentiation does not change substantially with IDDR threshold. The Géant topology delay differentiation drops for higher IDDR threshold values, however from threshold value of 0.4 onwards it outperforms BRITE topology. It is encouraging to note that the Géant topology, a real topology, retains high delay differences even for high values of IDDR threshold.

We next compare the throughput of the networks when running IDDR compared to their throughput with normal SPF routing. In Fig. 4 we see that in case of BRITE and Géant topologies SPF routing outperforms IDDR in throughput, i.e. the SPF throughput is higher than that of IDDR. This is to be expected as the IDDR threshold restricts the utilisation of all links which are used by high QoS flows. In Calle's topology, however, the flexibility offered by APs seems to balance out the restrictions set by the IDDR threshold. The IDDR threshold value of great interest though is around 0.9 (i.e. on any link which carries high QoS flows the link utilisation cannot exceed 0.9). At this value in the Calle topology the IDDR throughput is slightly higher than that of SPF.

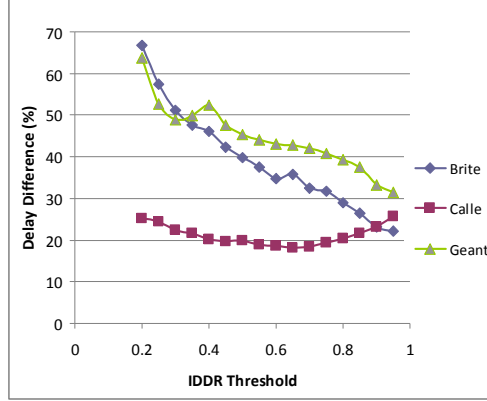


Fig. 3. Percentage Delay Difference between High QoS- and BE-Flow

In Fig. 5 we consider the network utilisation per flow, i.e. the resources consumed by each flow. It is evident that IDDR consumes more resources than SPF because some flows take paths that have a higher cost than the least cost. IDDR makes use of APs for BE flows and, in case of lack of congested SPs, for high QoS flows as well. This means that all the BE flows will follow paths which have a higher cost than the SPs, and therefore typically tend to comprise more links. Therefore it is not surprising that IDDR has a higher utilisation, since this is the cost-tradeoff for the delay differentiation. In summary, in Fig. 5 we see that the reduced throughput in IDDR in conjunction with the increased network utilization means that IDDR is slightly more expensive resource-wise than plain SPF routing.

Finally, in Fig. 6 we compare IDDR with OSPF for their delay as a function of throughput. In the case of the Calle topology IDDR keeps the delays of high QoS flows significantly lower than SPF flows for a wide range of throughput and from 450 flows onwards delays of BE flows are significantly lower as well. However, in the case of the BRITE topology delay of BE flows is constantly higher than that of the flows in SPF, while high QoS flows are only slightly better than SPF flows in

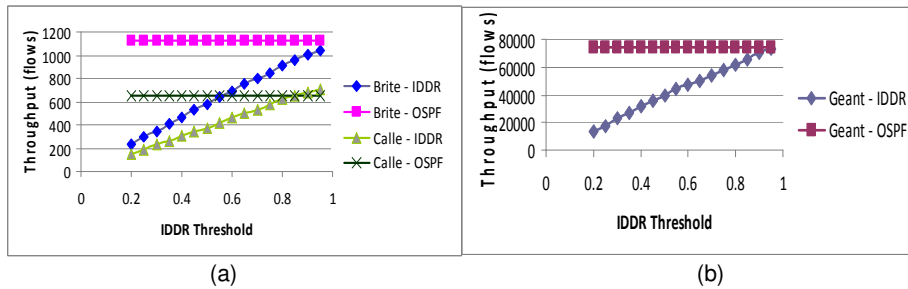


Fig. 4. Throughput vs. IDDR Threshold (a) for BRITE and Calle topologies and (b) for Géant topology. (The Géant topology results have been plotted on a different graph since its throughput is much higher than that of the other two topologies)

terms of delay. Nevertheless assuming that the best-effort traffic is mainly non delay-sensitive traffic, it is evident that IDDR is a promising solution for service differentiation in a network.

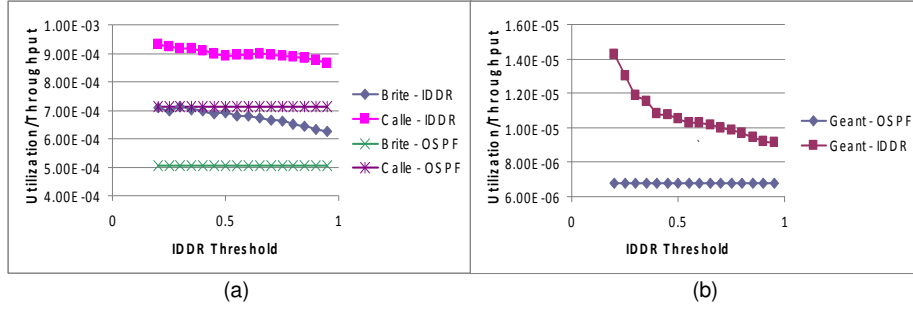


Fig. 5. Network Utilisation per flow vs. IDDR threshold (a) for BRITE and Calle topologies and (b) for Géant topology.

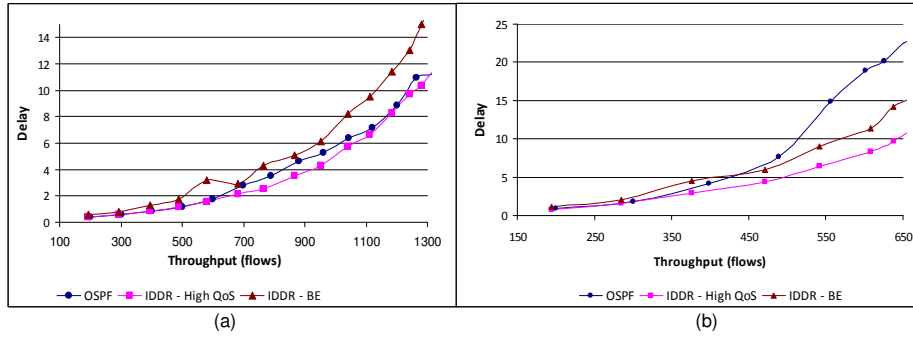


Fig. 6. Delay vs. throughput graph for (a) BRITE and (b) Calle topologies for IDDR threshold set to value 0.9 and total traffic in the range of 200-2100 flows

5. Conclusion

In this paper we have presented an algorithm for delay differentiation and tested it through simulation using a simple delay model from queuing theory. We have found significant qualitative delay differentiation by using IDDR. The network throughput is reduced only slightly compared to standard SPF routing. We believe that both throughput and delay differentiation could be further improved by performing traffic engineering on the network optimised for IDDR. In addition a future more refined simulation of the algorithm, using a package such as Network Simulator 2 (ns-2), that accounts for more complex queuing models and source packet modelling, would give us more advanced insights into the operation of IDDR.

Acknowledgement

Ioannis Papanagiotou is supported by a Doctoral Training Grant from the UK Engineering and Physical Sciences Research Council (EPSRC).

References

1. Wang, Z.; Crowcroft, J. "Shortest path first with emergency exits", SIGCOMM '90, ACM, Vol. 20, Issue 4, August 1990
2. Wang, Z. and Crowcroft, J. "Quality-of-service routing for supporting multimedia applications", IEEE Journal on Selected Areas in Communication, IEEE Journal, Vol. 14, issue 7, pp. 1228-1234 September 1996
3. Moy, J.; Corp., Cascade Communication. "OSPF Version 2", IETF, RFC 2178, July 1997.
4. Guerin, R.A.; Orda, A.; Williams, D. "QoS routing mechanisms and OSPF extensions", GLOBECOM '97, IEEE, Vol. 3, pp.1903-1908, November 1997
5. Sahoo, A. "An OSPF based load sensitive QoS routing algorithm using alternate paths", Computer Communications and Networks Proceedings, IEEE, pp. 236-241, October 2002.
6. Sahoo, A. "A load-sensitive QoS routing algorithm in best-effort environment", MILCOM Proceedings, IEEE, Vol. 2, pp. 1206-1210, October 2002.
7. Menth, M.; Martin, R. "Network resilience through multi-topology routing", 5th International Workshop on Design of Reliable Communication Networks Proceedings, 2005 (DRCN 2005), IEEE, pp 271-277, October 2005
8. Gjessing, S. "Implementation of two Resilience Mechanisms using Multi Topology Routing and Stub Routers", International Conference on Internet and Web Applications (ICIW '06), IEEE, pp. 29-29, February 2006 (1)
9. Hansen, A.F.; Kvalbein, A.; Cicic, T.; Gjessing, S.; Lysne, O. "Resilient routing layers for recovery in packet networks", International Conference on Dependable Systems and Networks Proceedings, 2005 (DSN 2005), IEEE, pp. 238-247, June 2005
10. Kvalbein, A.; Hansen, A. F.; Cicic, T.; Gjessing, S.; Lysne, O. "Fast IP Network Recovery Using Multiple Routing Configurations", INFOCOM 2006, IEEE, pp. 1-11, April 2006
11. Wang, N; Ho, K-H.; Pavlou, G. "Adaptive Multi-topology IGP Based Traffic Engineering with Near-optimal Network Performance", Proc. of the IFIP Networking Conference (Networking'2008), Singapore, May 2008.
12. "MPLS stack software, a mechanism for packet forwarding in networks". [Online] Data Connection , 2007. [Cited: 24 October 2007.] <http://www.dataconnection.com/mpls/whatis.htm>.
13. Xipeng Xiao; Hannan, A.; Bailey, B.; Ni, L.M. "Traffic engineering with MPLS in the Internet", IEEE Network, Vol. 14, Issue 2, pp. 28-33, March 2000
14. Li, Z.; Zhang, Z.; Wang, L. "A novel QoS routing scheme for MPLS traffic engineering", International Conference on Communication Technology Proceedings, 2003 (ICCT 2003), IEEE, Vol. 1, pp. 474-477, April 2003
15. Wang, B.; Su, X.; Chen, C.L.P. "A new bandwidth guaranteed routing algorithm for MPLS traffic engineering", International Conference on Communications, 2002 (ICC 2002), IEEE, Vol. 2, pp. 1001-1005, April 2002
16. Calle, E.; Marzo, J.L.; Urra, A.; Vila, P.; "Enhancing MPLS QoS routing algorithms by using the network protection degree paradigm" , GLOBECOM '03, IEEE, Vol. 6, pp 3053-3057, December 2003
17. Tanenbaum, A. "Computer Networks", Ed. Prentice Hall, 2003

18. **Crovella, M.E.; Bestavros, A.** "Self-similarity in World Wide Web traffic: evidence and possible causes", *Networking, IEEE/ACM*, Vol. 5, Issue 6, December 2007
19. "*Boston university Representative Internet Topology generator*". [Online] [Cited: 10 March 2008]. <http://www.cs.bu.edu/brite>.
20. **Calle, E.; Marzo, J.; Urra, A.** "Protection performance components in MPLS networks", *Computer Communications, Science Direct*, Vol. 27, Issue 12, pp.1220-1228, July 2004
21. "*Topology : The Network : GÉANT*". [Online] ,October 2004, [Cited: 20 March 2008]. <http://www.geant.net/server/show/nav.159>
22. "*TOTEM (TOolbox for Traffic Engineering Methods) Project*". [Online], 4 March 2008, [Cited: 20 March 2008]. <http://totem.run.montefiore.ulg.ac.be/datatools.html>.