QOS ROUTING IN DIFFSERV MPLS NETWORKS

K.K. Phang, T.C. Ling, T.F. Ang
Faculty of Computer Science and Information Technology,
University of Malaya, Malaysia
Email: {tchaw, kkphang, angtf}@um.edu.my;

ABSTRACT

With the emergence of new network applications and quality of service (QoS) requirements, current Internet technology that supports best-effort service is clearly insufficient. This paper undertakes a detailed study of dynamic quality of service routing in Differentiated Service (DiffServ) Multiprotocol Label Switching (MPLS) networks. A polynomial time, heuristic dynamic routing algorithm, namely TE-QOSPF-Mix, is proposed to provide a better trade-off between hop-count and bandwidth optimization. The algorithm uses well-defined comparative function in path selection with computational complexity equals the original Dijkstra's algorithm. A simulator, UMJaNetSim, is built to simulate and evaluate the proposed algorithm. The simulation results indicate that the proposed QoS routing algorithm exhibits better bandwidth acceptance and lower packet loss ratio.

Keywords: QoS Routing, OSPF, MPLS, DiffServ

1.0 INTRODUCTION

Today, the Internet apart from providing web browsing also supports services such as multimedia applications, critical e-business applications, teleconferencing and video conferencing real time services, and Voice over Internet Protocol (VoIP). These applications or services require high-capacity links, and are sensitive to end-to-end delay, jitter and packet loss. As such, the key design of modern Internet should support quality of service (QoS). Effective methods for supporting network QoS are needed. An effort in this direction is QoS-based routing which takes into account the policies and QoS constraints in route computation. The objectives of QoS-based routing (Crawley et al. 1998) are as follows:

- Dynamic determination of feasible paths.
- Optimization of resource utilization.
- Graceful performance degradation during congestion.

Current QoS routing algorithms suffer from some or all of the following disadvantages:
- Higher computational complexity.
- Difficulty in to fine tuning the tradeoff between metrics.

In this paper, a polynomial time, dynamic routing algorithm, namely TE-QOSPF-Mix, is proposed. TE-QOSPF-Mix uses enhanced Dijkstra's algorithm (Dijkstra, 1959) in its path computation and provides better trade-off between hop-count and bandwidth optimization.

The paper is organized as follows. Section 2 presents literature survey and related works. Section 3 discusses the proposed mechanism, while Section 4 presents the simulation results and the performance analysis. Finally, Section 5 concludes the paper with future work.

2.0 Literature Review

Apostopoulos (1999) investigated the per-flow QoS routing using IntServ. Technical details of extending the OSPF to support per-flow QoS are proposed. Detailed discussion is provided on the choice of QoS metrics and the path selection mechanism. The trade-off between accuracy and computational complexity is also given. This includes the link advertisement mechanisms, and control overhead. The dissemination of network state information for QoS routing can impose a significant burden on the bandwidth and processing resources in the network.

Guerin et al. (1991) investigated the effect of inaccurate network state information on successful identification of feasible path. In the context of the source-directed link state routing protocols, Shaikh et al. (1998) investigated the basic trade-off between routing and signaling overheads. Shaikh (1999) proposed a modified version of Dijkstra's algorithm to compute the shortest path to a destination. Instead of one path, multiple equal-cost shortest paths are computed to each destination. Wang and Crowcroft (1996) proposed a centralized algorithm using available bandwidth and delay as metrics. To find a path in a network, which satisfies these constraints, is NP hard. To achieve polynomial time complexity, the path computation is done in two stages: 1) Eliminates all links that do not meet the bandwidth requirement. 2) Find the minimum delay path using Dijkstra's algorithm. For a network of n nodes, the computation complexity of this algorithm is $O(n^3)$ (Wang and Crowcroft, 1996).

Ma and Steenkiste (1997, 1998) showed that in rate-proportional service such as fair queuing, the end-to-end delay, delay-jitter, and buffer space bounds metrics are mutually dependent. These metrics are determined by the bandwidth reserved for the flow and traffic characteristics. Major polynomial time algorithms proposed are as follow:

1) Widest-shortest.
2) Shortest-widest.

The *Widest-Shortest* path algorithm (a.k.a. QOSPF) applies Dijkstra's algorithm after links with available bandwidth less than the demand bandwidth of LSP have been trimmed. This is a feasible path with the minimum hop count. If there exists more than one path with the minimum hop count, the one with the maximum available bandwidth is selected. Guerin and Orda (1999) investigated QoS routing problem in the presence of inaccurate network states.

Chen and Nahrstedt (1998b) proposed a heuristic algorithm to solve the NP-complete multi-constrained routing problem. A heuristic algorithm is proposed where the first problem is to reduce the NP-complete problem to a simple one. Extended Dijkstra's algorithm is then used to solve the problem in polynomial time. In the multi-constrained routing problem, if all metrics except one take bounded integer values, this problem is solvable in polynomial time. For instance, taking the case of delay-cost-constrained routing, the cost (or delay) of every link from an unbounded real number is mapped into a bounded integer. In this way, the delay-cost-constrained routing problem is reduced to a simpler problem solvable in polynomial time. A feasible path computed from the reduced problem is also a feasible path of the original problem. The shortest-distance path algorithm (Kamei and Kimura, 2001) uses both the hop-count and available bandwidth as metrics. The distance of a link is defined as the reciprocal of the available bandwidth of that link. The link cost of a path is the sum of the reciprocal of the available bandwidth of the link along the path. Extended Dijkstra's algorithm is used in the algorithm.

The dynamic-alternative path algorithm (Kamei and Kimura, 2001) is a variant of the *Shortest-Widest* path algorithm where a restriction can be imposed on the difference (a value m) of the hop-count between the shortest path and the alternate path. This algorithm is an extension of the Dijkstra's algorithm.

Assume that the given shortest path has a hop-count of n. The *Widest-Shortest* algorithm is applied to determine paths that can be reached within n+m hops. Lim et al. (2004) proposed the Traffic Engineering QoS OSPF (TE-QOSPF) path computation algorithm. This algorithm uses bandwidth as the primary QoS parameter. The algorithm allows longer paths to be chosen if the path has significantly higher residual bandwidth. The actual meaning of "significantly higher residual bandwidth" is defined as (Lim et al. 2001, 2004):

> "*When a candidate path is one hop longer than another candidate path, its available bandwidth must be twice that of the shorter path. Say, when the hop-count difference is two, the longer path must have three times available bandwidth than that of the shorter path, and so on (This is represented by k and is adjustable). In order to limit the use of very long paths, a hop-count difference threshold, c, is defined so that if two paths have a hop-count difference greater than c, the shorter path is always chosen.*"

An interesting feature of TE-QOSPF is the ability to provide a more balance tradeoff between the hop-count and available bandwidth. A better tradeoff can be achieved by fine-tuning the values of k and c. However, in Lim's proposal the path selection comparative function is not "well order". This may cause inconsistency in path computation. In addition, TE-QOSPF has a higher computational complexity compared to the rest of the algorithms.

(Geleji et al, 2008) proposed an architecture to separate forwarding and routing functionality in small-scale connection-oriented networks, such as MPLS and GMPLS. In this paper, performance analysis of several distributed multi-domain QoS routing algorithms that may be implemented on the suggested platform was carried out. (Alzahrani and Woodward, 2008) presented a localized bandwidth-based QoS routing algorithm, where routing decision is made based on the residual bandwidth that each path can support. In the algorithm, the source node routes packets based on the statistic collected locally.

3.0 Proposed TE-QOSPF-Mix Algorithm

All the algorithms mentioned in section 2 use global network states where paths are computed at the flow arrival time. These algorithms, except TE-QOSPF, compute a path on a per-connection basis, which may have scalability issues. Path caching and other more scalable framework such as MPLS and DiffServ may be employed to reduce processing overhead and improve routing performance. The proposed TE-QOSPF-Mix is designed to overcome these problems.

The proposed TE-QOSPF-Mix algorithm maintains the advantages of TE-QOSPF where fine-tuning between hop-count and available bandwidth can be done. This allows proper balance between the metrics. The algorithm is designed to use comparative function that satisfies the total order relation. In addition, this algorithm has lower computational complexity. The rationales and assumptions behind the design of TE-QOSPF-Mix are as follows:

- If every other parameter remains the same, the shorter path is better.
- If every other parameter remains the same, the wider path is better.
- A longer path with "significantly higher" bandwidth is better.
- The ability for an algorithm to fine-tune the tradeoff between hop-count and available bandwidth is important.
- The comparative function must fulfill the *Total Order* relation requirement.

Let *BetterTEQOSPFMixPath* be a real value function. A better path is indicated by a higher value. The following relations are obtained.

$$BetterTEQOSPFMixPath \propto bandwidth \qquad \text{-----(1)}$$

$$BetterTEQOSPFMixPath \propto \frac{1}{hopCount} \qquad \text{-----(2)}$$

Therefore, by combining formulas (1) and (2), and allows flexible weight adjustment between the metrics to be made, the following single mixed metric is obtained

$$BetterTEQOSPFMixPath = \frac{bandwidth^k}{hopCount^l} \qquad \text{-----(3)}$$

where *k* and *l* are the configurable weights for each metric; *k* and *l* are positive real number greater than 1.

The following shows detailed implementation of TE-QOSPF-Mix using binary heap.

```
TE-QOSPF-Mix
Input: A directed weighted graph G and a vertex U of G
Output: A label D[V], for each vertex V of G
1       Begin
2       Route Tree, R := $\emptyset$
3       For each vertex V $\neq$ u do
4         D[v].parent := null
5         D[v].hop := infinity
6         D[v].bw := infinity
7       For each link from vertex M to N
        bw (M, N) := the advertised available bandwidth
8       Insert all the vertices into the priority queue Q using the
        value of $bw^k/hop^l$ stored in the label D as keys
9       While Q is not empty do
        // remove the minimum vertex (based on the value of $bw^k/hop^l$
           in D) from the Q
10      R := R $\cup$ {U}
11      V := BestNodeIn(Q)
12      for each (directed) edge (V,Z);and Z is in Q do
13        if  D[Z].hop <> infinity // Z is a candidate
14          define temporary vertex W
15          D[w].parent:= V
16          D[w].hop:= D[V].hop+1
17          D[w].bw := min ( D[V].bw, bw(V,Z) ) (Relaxation process)
18          If betterPath (W,Z) = W
19            D[Z].parent:= D[W].parent
20            D[Z].hop:= D[W].hop
21            D[Z].bw := D[W].bw
22          End IF
23        Else
24          D[Z].parent:= V
25          D[Z].hop:= D[V].hop+1
26          D[Z].bw := min ( D[V].bw, bw(V,Z) )
27        update the key of vertex z in Q
28      End (R contains paths to every destination through
             tracing back of parents)
        //function
29        BetterPath(W, V) :
30          avail(W) := Bw(W)k / Hops(W)l
31          avail(V) := Bw(V)k / Hops(V)l
32          If avail(W) > avail(V) then returns W
33          Else returns V
        //function
34        BestNodeIn(Q) :
35          returns the node N with the highest avail(N) as
            calculated in BetterPath() above
```

The complexity of Dijkstra's algorithm is estimated as follows: In line 8, the algorithm takes *2n* time to insert the vertices into the initial priority queue of *n* vertices. It is observed that if a better-cost path is obtained through V, then the path cost for V is decreased and the vertex priority changed in the queue. This is done at the relaxation process listed from line 13 to 22. Each of the subsequent priority queue extraction operations takes time (log *a*), where *a* is the current size of the queue. Thus the priority Q operations take O (log *n*) time (Cormen, 2001). Within the while loop (from line 9 to 28), there is only one extraction and deg (*V*) relaxation.

Therefore, the entire extraction and relaxation will take time log n + deg (V) log n. In total, the running time of the while loop is given by:

$$\sum_{v \in G} (1 + \deg(v)) \log n \qquad \text{-----(4)}$$

Since G is a graph with m edges (Goodrich and Tamassia, 2002), then

$$\sum_{v \in G} \deg(v) = 2m \qquad \text{-----(5)}$$

Therefore the complexity of TE-QOSPF-Mix is O(2n) + O((n + m) log n) which is equivalent to O(n + mlog (n)). It can also be shown that the relation BetterTEQOSPF-MixPath (≥) is a total order relation (Goodrich and Tamassia, 2002) satisfying the reflexive, anti-symmetric, transitive and comparability properties.

As TE-QOSPF-Mix is implemented in DiffServ MPLS networks, the available bandwidth metric needs to be included in the link-state advertisements. Opaque LSA (Coltun, 1998) or TE-OSPF (Srisuresh et al., 2002) can be used to carry the additional link metric information to other routers. Paths for EF class (path with reserved bandwidth) are calculated on-demand (during the path-setup request). Paths for other classes (without bandwidth guarantee) are calculated when new route advertisements are received (a new route table is constructed each time a new advertisement is received).

4.0 SIMULATION ENVIRONMENT AND RESULTS

This section presents the simulation results, and analysis of the performance of the proposed TE-QOSPF-Mix. Figure 1 depicts the MCI topology. MCI backbone represents a well-known existing ISP topology. This topology has been used in many network performance studies (Shaikh 1999).

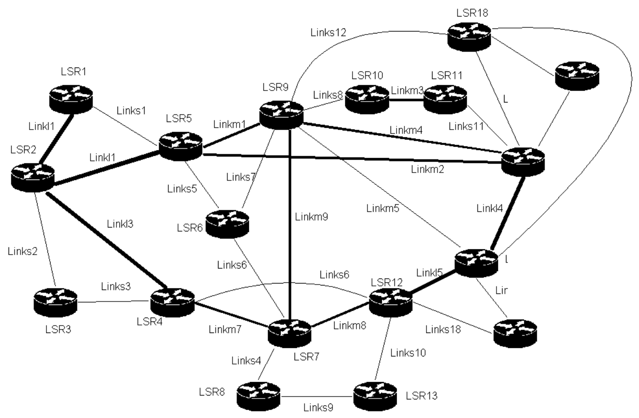

Fig. 1: The MCI Topology

The MCI topology consists of 18 nodes, 47 links, max degree of 7, a diameter of 6 and the number of traffic source is 360.

The performance evaluation metrics used are as follows:

- Long-term link utilization
- Average Link Utilization
- Peak Link Utilization
- End-to-End Delay
- Packet Loss Ratio

An important aspect of QoS routing and traffic conditioning is to improve or optimize network resource utilization. The long-term, average and peak link utilization are measured to reflect the network resource utilization and throughput. End-to-End delay and packet loss ratio are usually negotiated before a real time connection is established. In the case of non-real time application, usually only the packet loss ratio is negotiated. The bandwidth-blocking ratio is used to indicate the percentage of call connection requirements that are rejected by the network. This metric takes into consideration the rate requirement of a call when calculating the blocking ratio.

Due to the poor performance of the static algorithms, subsequent sections evaluate only the dynamic algorithms.

1. Shortest widest
2. Widest Shortest
3. Dynamic Alternative+1 (Dyn Alt; C=1)
4. Dynamic Alternative+2 (Dyn Alt; C=2)
5. Shortest distance path (Sum of 1/(avail. bw))
6. TE-QOSPF with k := |hops1-hops2|+1; c=1 (TE-F3)
7. TE-QOSPF-Mix with k := 1; l :=2 (Mix-F2)
8. Shortest path(OSPF-Static)
9. Inverse capacity (OSPF-InvCap)

The CBR traffic parameters are as follows:

- Bit Rate: 0.2 Mbps
- Amount to be sent: 2 Mbit
- Delay between call: 3 s

The CBR traffic is sent continually with a delay between call of 3 s. In every call, each source will send exactly 2 Mbit of data to a random destination.

The VBR traffic used for this simulation consists of the following parameters.

- Bit Rate: 1 Mbps
- Mean Burst Length: 5000 usec
- Mean interval between burst: 15000 usec
- Number of bits to be sent: 2 Mbit
- Delay between call: 3 s
- ON-OFF model with Poisson distribution

The VBR traffic is sent continually with a delay between call of 3s. In every call, each source will send exactly 2 Mbit of data to a random destination. Simulation starts with a normalized load of 1. After each run, the load of the CBR and VBR traffic is increased at a rate of 10% by increasing the "number of bits to be sent" of each source by 10%.

Bandwidth Acceptance

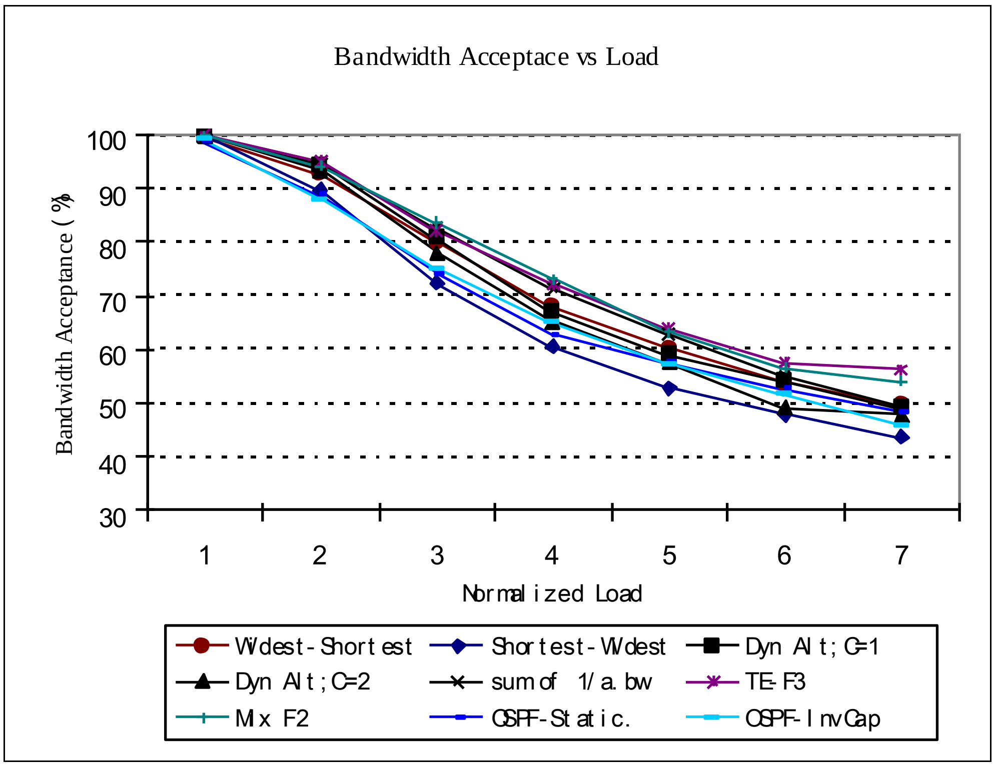

Fig. 2: Bandwidth Acceptance

For ease of analysis, the normalized traffic loads from Figure 2 are divided into three regions: light (1,2), moderate (3,4) and heavy (5,6,7). Figure 2 shows the bandwidth acceptance rate that the routing algorithms can accommodate. At stage 1, when the normalized load is light, *TE-F3* has the best performance. The results at this stage indicate that non-shortest path is always preferred. The additional resources consumed by the alternative path algorithm are insignificant compared to the abundance of resources. At this stage, the introduction of longer paths did not introduce any adverse effect to the network.

At stage 2, when the normalized load is moderately high, *Mix-F2* has the best performance. The results in this stage indicate that *Mix-F2* is able to balance the trade-off between the use of paths with wider bandwidth and longer paths. Algorithms *Sum of 1/a.bw*, *TE-F3*, *Dyn Alt;C=1* still perform considerably well. However, algorithms such as *Dyn Alt;C=2*, *OSPF-InvCap*, *OSPF-Static* and *Shortest-Widest* begin to show signs of the adverse effect of over utilization of network resources. This is especially true for the *Shortest-Widest* since at this stage even the static *OSPF-Static* outperforms the dynamic *Shortest-Widest*.

At stage 3, when the normalized load is heavy, both *TE-F3* and Mix-F2 outperform the rest. The ranking of *sum of 1/a.bw* drops from the second position to third. One possible explanation of the drop of performance of this algorithm is that, as the network becomes more congested, the value of the available bandwidth of more links becomes smaller. This implies that the sum of the reciprocal of these values become very large. The computed sum could have been dominated by the reciprocal value of links with the smaller available bandwidth. The sum of the reciprocal of very small number may introduce more errors or more uncertainty in the path selection. On the other hand, *TE-F3* and *Mix-F2* continue to outperform the rest because longer alternate paths are selected based on more stringent condition and valuable network resource is not over utilized to make the short term gain.

Table 1: Simulation set A--Overall Ranking

Ranking	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MCI (light)	TE-F3	Dyn Alt with C=1	sum of 1/a.bw	Mix F2	Dyn Alt with C=2	Shortest -Widest	Widest-Shortest
MCI (moderate)	Mix-F2	sum of 1/a.bw	TE-F3	Dyn Alt;C=1	Widest-Shortest	Dyn Alt;C=2	Shortest -Widest
MCI (heavy)	TE-F3	Mix F2	sum of 1/a.bw	Widest-Shortest	Dyn Alt;C=1	Dyn Alt;C=2	Shortest -Widest

Based on the overall simulation results listed in Table 1, *TE-F3* and *Mix-F2* appear to be the overall winner in terms of bandwidth acceptance. Another interesting observation of the *Dyn Alt* algorithms can be derived from their performance. *Dyn Alt with C=1* (Except in only one instance) always outperforms the *Dyn Alt with C=2* counter part. This indicates that longer paths should not be used without constraint.

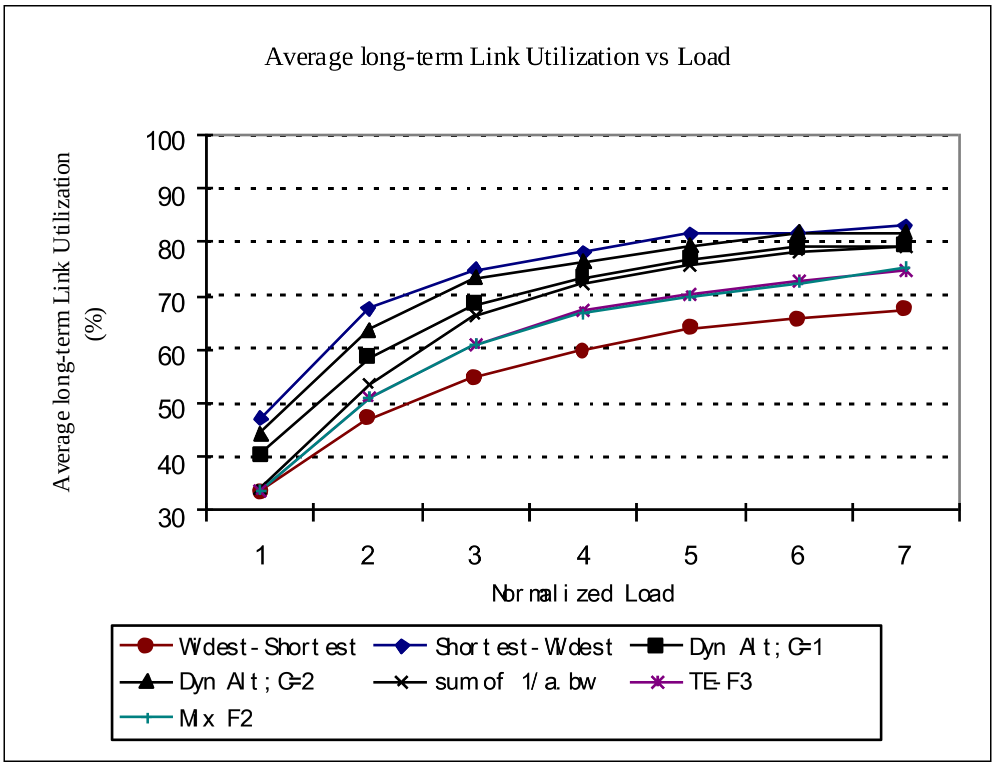

Fig. 3: Average Long-Term Link Utilization

Average Long-Term Link Utilization

As in simulation set A, again for ease of analysis, the normalized traffic loads in Figure 3 are divided into three regions: light (1,2), moderate (3,4) and heavy (5,6,7).

Table 2: Overall Ranking of ALTLU

Ranking	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Load (Light)	Shortest-Widest	Dyn Alt;C=2	Dyn Alt;C=1	sum of 1/a.bw	TE-F3	Mix F2	Widest-Shortest
Load (Moderate)	Shortest-Widest	Dyn Alt;C=2	Dyn Alt;C=1	sum of 1/a.bw	Mix F2	TE-F3	Widest-Shortest
Load (Heavy)	Shortest-Widest	Dyn Alt;C=2	Dyn Alt;C=1	sum of 1/a.bw	TE-F3	Mix F2	Widest-Shortest

For all stages of traffic load, that is from light to heavy, algorithm *Shortest-Widest*, *Dyn Alt;C=2*, *Dyn Alt;C=1*, *sum of 1/a.bw* and *Widest-Shortest* always rank consistently at the position 1, 2, 3, 4 and 7 respectively in terms of ALTLU. On the other hand, the ranking of *TE-F3* and *Mix-F2* fall either at position 5 or 6. In terms of network resources utilization, *Shortest-Widest* uses the most resources. Table 2 indicates that *Shortest-Widest* has over utilized the resource while *Widest-Shortest* underutilized the resources. The former will perform well when the network is non-congested while the latter when the network is congested. Both algorithms are inefficient as both went to the opposite extreme. A better algorithm should be able to balance the two extremes. *Dyn Alt;C=1*, *sum of 1/a.bw*, *TE-F3* and *Mix-F2* are able obtain a better balance between longer path and available bandwidth. The performance of these algorithms is more consistent.

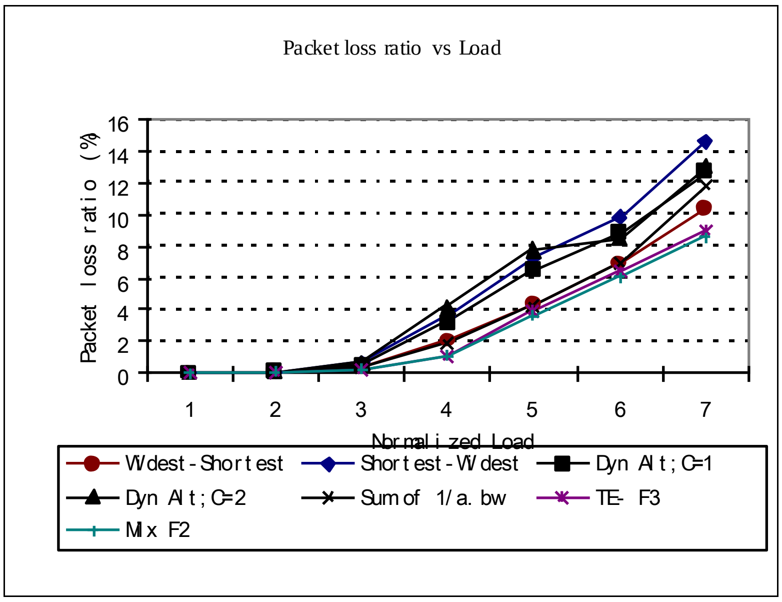

Fig. 4: Packet Loss Ratio

Packet Loss Ratio

In Figure 4, both algorithms *Mix-F2* and *TE-F3* emerge as the overall winners in terms of having a lower packet loss ratio. Algorithms *Shortest-Widest*, *Dyn Alt;C=1* and. *Dyn Alt;C=2*, outperform the *Widest-Shortest* only when the traffic load is low. On the other hand, algorithm *Sum of 1/a.bw* is able to outperform *Widest-Shortest* when the traffic load is low and moderate. However, as the traffic load becomes higher, this algorithm experiences a higher packet loss ratio.

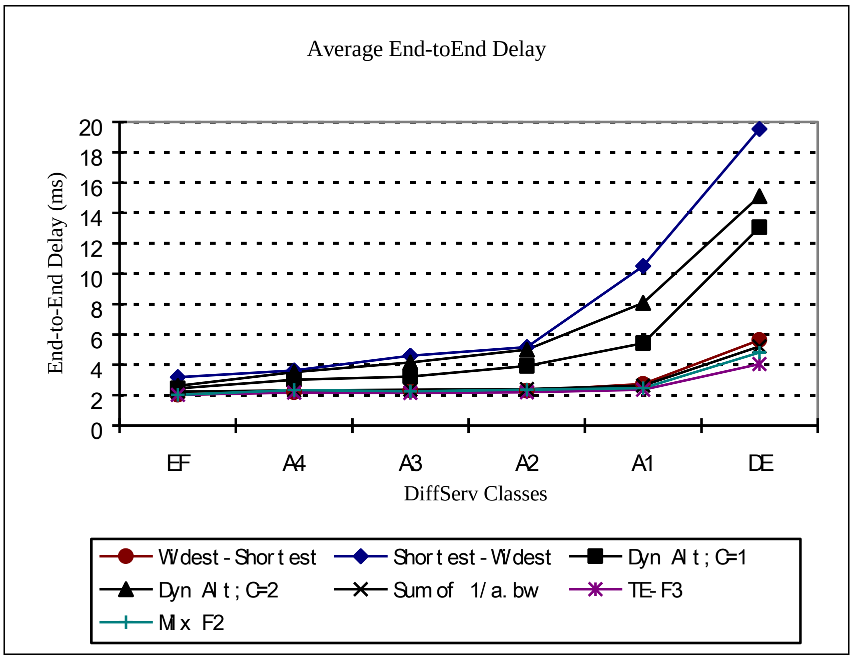

Fig. 5: Average End-to-End Delay (Light Load)

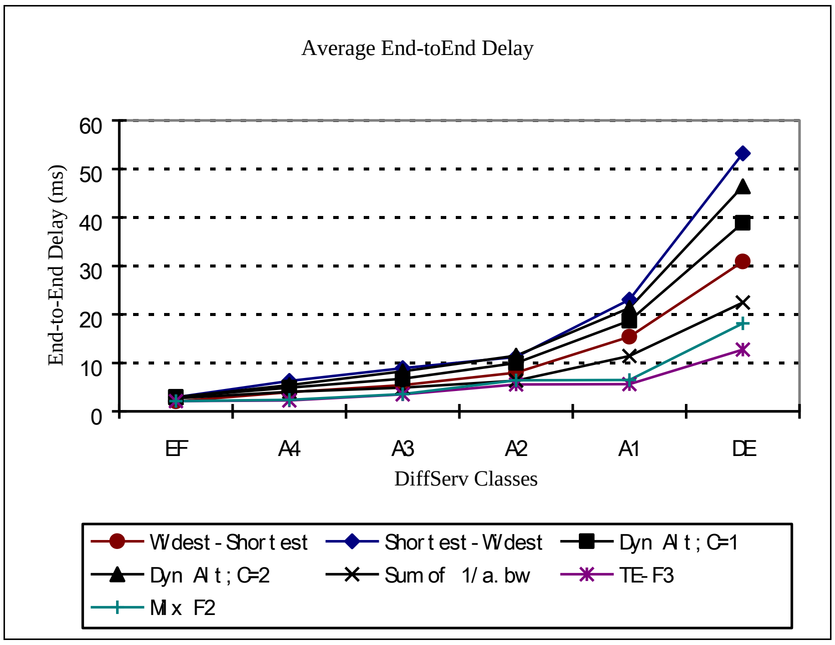

Fig. 6: Average End-to-End Delay (Moderate Load)

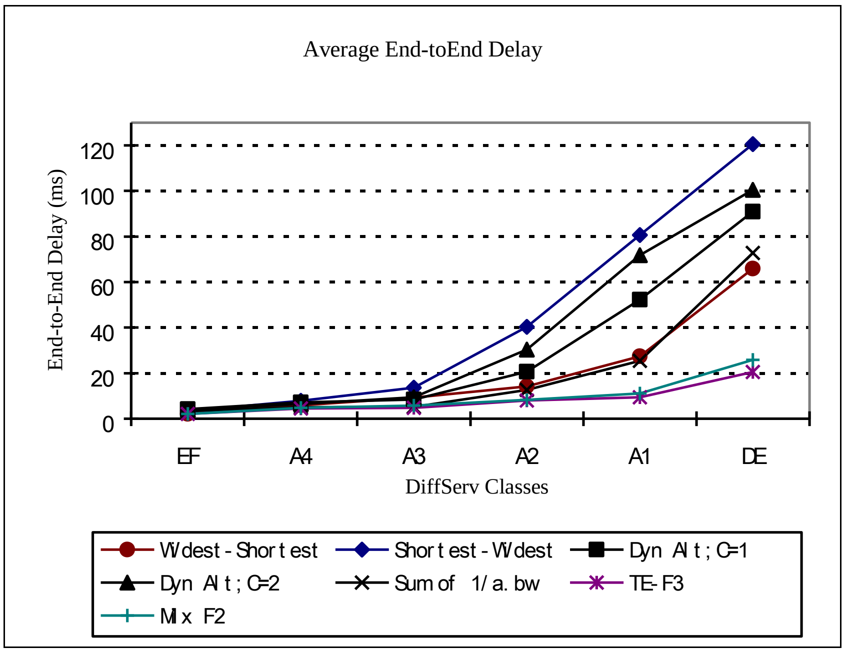

Fig. 7: Average End-to-End Delay (Heavy Load)

Average End-to-End Delay

Under the three types of traffic loads of light, moderate and heavy, as depicted in figure 5, 6 and 7, the average end-to-end delay performance of various DiffServ using each algorithm is consistent. From the simulation results, DiffServ classes arrange in ascending order of average end-to-end delay are as follows: EF, AF3, AF2, AF1 and DE. This order is consistent with the definition of the DiffServ subclasses.

Among all algorithms, algorithm *TE-F3* and *Mix-F2* outperform the rest in terms of having the lowest average end-to-end delay.

The overall simulation results show that *TE-F3* and *Mix-F2* are the overall winner in terms of bandwidth acceptance. In terms of long-term link utilization, *TE-F3* and *Mix-F2* are able to maintain a better balance between longer path and available bandwidth. The packet loss ratio and average end-to-end delay of *Mix-F2* and *TE-F3* are the lowest among the evaluated algorithms. The end-to-end delay observed is consistent with the definition of the DiffServ subclasses (EF, AF3, AF2, AF1 and DE). Generally, TE-QOSPF-Mix algorithm has the best overall performance in terms of bandwidth acceptance, long-term link utilization, packet loss ratio and average end-to-end delay. TE-QOSPF-Mix is able to maintain better balance between the hop-count and available bandwidth metrics.

5 SUMMARY AND OPEN ISSUES

The paper begins by investigating QoS routing. Next, the state-of-the-art development in IP-based QoS routing, DiffServ MPLS are explored and discussed. These form the fundamental to the enhancement of traffic engineering and conditioning in IP network. We propose to enhance QoS-based routing by extending Dijkstra's algorithm with an attempt to balance the hop count and available bandwidth. A QoS routing mechanisms, namely TE-QOSPF-Mix, is subsequently proposed. In general, simulation results show that these approaches to the design of

QoS routing algorithms are feasible, versatile and have many advantages. TE-QOSPF-Mix is a "simplified" solution to the NP-hard QoS routing problem, whereupon both the metrics (hop-count and bandwidth) are optimized. The advantages of the proposed TE-QOSPF-Mix algorithm are as follows:

- Allow network administrators to fine-tune the tradeoff between hop-count and available bandwidth.
- The computational complexity is compatible with the original Dijkstra's algorithm.

In general, the simulation results show that the performance gain in terms of bandwidth acceptance, long-term link utilization, packet loss ratio and average end-to-end delay is significant.

There are many issues related to this work that require further research. TE-QOSPF-Mix uses two metrics, that is, hop-count and available bandwidth. Future work on TE-QOSPF-Mix may consider the incorporation of additional constraints within the path finding metrics. TE-QOSPF-Mix is an intra-domain routing algorithm and can be extended to support inter-domain routing. The proposed QoS routing mechanism can be incorporated into the current multicast and anycast routing algorithms to provide better QoS.

References

1 Alzahrani, A.S. and Woodward, M.E.; Residual bandwidth as localized QoS routing metric16th International Conference on Software, Telecommunications and Computer Networks, 2008, pp. 125 - 129

2 Apostolopoulos, G. et al. (1999a). Improving QoS Routing Performance Under Inaccurate Link State Information, Proceeding of ITC '16, Edinburgh, Scotland, June 1999.

3 Apostolopoulos, G. Guerin, R., Kamat, S. and Orda, A. (1999b). QoS Routing Mechanisms and OSPF Extensions, RFC2676, IETF, August 1999.

4 Chen, S. and Nahrstedt, K. (1998b). An Overview of Quality-of-Service Routing for the Next Generation High Speed Networks: Problems and Solutions.
IEEE Network, 12(6), Nov./Dec. 1998, pp. 64-79.

5 Coltun, R. (1998). The OSPF Opaque LSA Option, RFC 2370, July 1998.

6 Cormen, T.H. et al (2001). Introduction to Algorithms. Cambridge, Mass.: MIT Press, 2001.

7 Crawley, E. et al. (1998). A Framework for QoS-based Routing in the Internet, RFC 2386, August 1998.

8 Dijkstra, E. (1959). A Note on Two Problems in Connexion with Graphs, Numerische Mathematik, 1959, Vol. 1, pp. 269-71.

9 Geleji, G. et al. A Performance Analysis of Inter-Domain QoS Routing Schemes Based on Path Computation Elements, International Symposium on High Capacity Optical Networks and Enabling Technologies, 2008, pp. 146 – 152

10 Goodrich, M. T. and Tamassia, R. (2002). Algorithm Design. John Wiley & Sons, Inc., 2002.

11 Guerin, R. A. and Orda, A. (1999). QoS Routing in Networks with Inaccurate Information: Theory and Algorithms, IEEE/ACM TRANSACTIONS ON NETWORKING,
VOL. 7, NO. 3, JUNE 1999, pp. 350 –364.

12 Guérin, R. et al. (1991). Equivalent Capacity and Its Application to Bandwidth

Allocation in High-Speed Networks, IEEE Journal on Selected Areas in Communication,
September 1991, Vol. 9, No. 7, pp. 968-981.

13 Kamei, S and Kimura, T. (2001). Evaluation of Routing Algorithms and Network Topologies for MPLS Traffic Engineering, Global Telecommunications Conference, 2001.
GLOBECOM '01. IEEE , Volume: 1 Nov. 2001, pp. 25-29.

14 Lim, S.H. (2001). Traffic Engineering Enhancement to OSPF for IP QoS with Diffserv and MPLS. Paper (Master). University of Malaya. April, 2001

15 Lim, S. H., Mashkuri Yaacob, Phang, K. K and Ling, T.C. (2004). Traffic engineering enhancement to QoS-OSPF diffserv and MPLS networks, IEE Proceedings Communications,
Volume 151, Issue 1, Feb 2004 Page(s):101 – 106

16 Ma, Q. and Steenkiste, P. (1997). Quality-of-Service Routing with Performance Guarantees, Proc. 4th Int'l. IFIP Wksp. QoS, May 1997.

17 Ma, Q. and Steenkiste, P. (1998). Routing Traffic with Quality of Service Guarantees in Integrated Services Networks, in Proceedings of NOSSDAV '98, July 1998.

18 Shaikh, A. et al. (1998). Evaluating the overheads of source-directed quality-of-service routing. Proc. IEEE International Conference on Network Protocols (ICNP '98), October 1998, Austin, TX, pp. 42-51.

19 Shaikh, A. (1999). Efficient Dynamic Routing in Wide-Area Networks, Ph.D. Paper, University of Michigan, May 1999.

20 Srisuresh, P. et al. (2002). TE LSAs to extend OSPF for Traffic Engineering, Internet Draft, IETF, September 2002.

21 Wang, Z. and Crowcroft, J. (1996). Quality-of-Service Routing for Supporting Multimedia Applications, IEEE Journal on Selected Areas in Communications, September 1996, Vol. 14, No. 7, pp. 1228-1234.

BIOGRAPHY

K. K. Phang is an Associate Professor at the Faculty of Computer Science & Information Technology, University of Malaya, Malaysia. He obtained his PhD in 2004 from University of Malaya. His current research interests include high speed computer network, network QoS, grid computing, traffic conditioning and traffic engineering, and fuzzy logic.

T.C. Ling obtained his PhD in 2005 from University of Malaya, Malaysia. He is an Associate Professor at Faculty of Computer Science & Information Technology, University of Malaya. His research areas include core network research, inter-domain Quality of Service (QoS), Voice over IP (VoIP), grid computing, and network security.

T.F. Ang obtained his M. Comp. Sc. in 2001 from University of Malaya. He is a lecturer at the Faculty of Computer Science & Information Technology, University of Malaya, Malaysia. His research areas include web services, Voice over IP (VoIP), grid computing and nework QoS.