Aspects of Quality of Service Routing

Piet Van Mieghem[a] and Hans De Neve[b]
AReNA
Alcatel Corporate Research
Francis Wellesplein 1, B-2018 Antwerp

ABSTRACT

Two pillars of QoS Routing are discussed: the QoS algorithm and the network function to provide each node a consistent view of the topology. Generally, QoS algorithms are believed to be exceedingly complex due to previous announcements that they belong to the class of NP-complete problems. However, a very efficient QoS algorithm, TAMCRA, has been designed which is slightly more complex than the well-known Dijkstra algorithm and far from hard NP-complete.
The topology distribution mechanisms responsible to offer each node in the system a consistent view are complicated due to the coupling of some QoS link metrics (such as available bandwidth) with the state of the network resources. The difficulty lies in the different time scales that impact the process: the slowly flooding of topology information and the more rapid variations of the traffic flowing through the links.

Keywords: Routing, Quality of Service, flooding dynamics, topology updates, TAMCRA

1. INTRODUCTION

Although Quality of Service (QoS) is an inherent ATM concept and being studied for almost a decade, the interests for its problematic nature is today still remarkably vivid. One important reason for this observation, is that the current Internet is moving towards QoS-awareness. Old debates, concept or approximate methods are newly introduced in the IETF's integrated and differentiated services and RSVP working groups. At the same time that the Internet community borrows ATM's QoS concepts, the ATM Forum is pushing ATM towards QoS-aware networking via the PNNI specification, which routing part incorporates aspects of the Internet's experience. Here we will mainly focus on QoS routing which is embedded in PNNI. Also the IETF possesses a QoS-update of its link state routing protocol OSPF and has even started a new working group on QoS Routing. At the time of writing, this IETF QoSR group has produced a framework document (Crawley *et al.,* 1998) broadly covering different facets of QoS routing.

Network routing essentially consists of two identities, the routing algorithm and the routing protocol. The routing algorithm assumes a temporarily static or frozen view of the network topology. It computes some routing instance in the corresponding graph *G(V,E)* with *V* nodes (vertices) and *E* links (edges) where the latter is further characterized by one or more link metrics. Specifically for QoS routing, each link in the graph *G(V,E)* possesses several metrics or it is defined by a link vector with as components values for e.g. delay,

---

[a] Currently at the Technical Unversity of Delft, Faculty of Information Technology and Systems, Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands.
email: P. VanMieghem@its.tudelft.nl
[b] email: hans.de_neve@alcatel.be

cell/packet loss, available bandwidth, cost, etc... . In general, a QoS routing algorithm belongs to the class of multiple parameter routing problems (often) subject to a constraints vector (derived from end-to-end QoS requirements to be obeyed by the path). Since routing problems with at least two additive metrics are known to be NP-complete (Garey and Johnson, 1979), many have regarded QoS routing as not-feasible and, hence, unattractive, though desirable in telecommunications networks. Recently, we succeeded in deducing a QoS algorithm called TAMCRA, Tunable Accuracy Multiple Constraints Algorithm. A number of attractive properties of TAMCRA are discussed in the following section 2 clearly demonstrating that the QoS routing algorithmic aspect does not create an immense a problem as previously was expected.

The routing protocol, on the other hand, provides each node in the network topology with a consistent view of that topology at some moment, say t_1 , from which the graph $G(V,E)$ readily is obtained. In current networks, the routing protocol is dynamic and distributed. The dynamicity means that important topology changes are flooded "automatically" to all nodes in the network while the distributed nature implies that all nodes in the network are equally contributing to the topology information distribution process. Since QoS is associated with resources in the nodes of the network, the QoS link metrics are, in general, coupled to these available resources. The impact of this coupling is discussed section 3.

2. TAMCRA

A network topology supporting QoS consists of link metrics vectors with as components non-negative QoS measures. The QoS measure of a path can either be *additive* in which case it is the sum of the QoS measures along the path or it can be the *minimum(maximum)* of the QoS measures along the path. Min(max) QoS measures are treated by omitting all links (and possibly disconnected nodes) which do not satisfy the requested min(max) QoS measure. We call this topology filtering. Additive QoS measures are expected to cause more difficulties. As mentioned above, the problem of calculating a path which is subject to more than one additive constraint is known to be NP-complete (Garey and Johnson, 1979) and hence, considered as intractable for large networks.

Recently we have proposed (De Neve and Van Mieghem, 1998, 1998a) a new and promising Tunable Accuracy Multiple Constraints Routing Algorithm (TAMCRA) based on three fundamental concepts: a non-linear measure for the path length, the k-shortest path approach and the principle of non-dominated paths. When m multiple constraints are imposed, a non-linear choice for the definition of the path length proves to be superior to a linear definition (i.e. a weighted sum of the vector components). An important corollary of a non-linear length function is that *the subsections of shortest paths in multiple dimensions are not necessarily shortest paths*. This corollary suggests to consider in the computation more paths than only the shortest one, leading us naturally to the k-shortest path approach. Finally, the multi-dimensional character of QoS routing invites the use of state space reduction which has been implemented via the concept of non-dominated paths. These three fundamental concepts and a thorough performance evaluation of TAMCRA are presented elsewhere (De Neve and Van Mieghem, 1998a).

TAMCRA possesses tuneable accuracy (coupled to the running time) via one integer parameter k which reflects the number of shortest paths taken into account during computation. There always exists a finite value of k for which TAMCRA returns the exact path. Perhaps most important, the worst case complexity of TAMCRA of the order of $O(kV \log(kV) + k^3 m E)$, is pseudo-polynomial which means that the NP-completeness only hides via k behind the granularity of the additive QoS measures and that TAMCRA scales in the number of nodes and links similarly to Dijkstra's algorithm.

The main advantages of the TAMCRA algorithm are the following:

(1) *Exponential decrease of the probability of missing the shortest path as the value of k increases.*

Figure 1 shows the probability that the path which satisfies two constraints is missed in a graph with 100 nodes and 200 links.

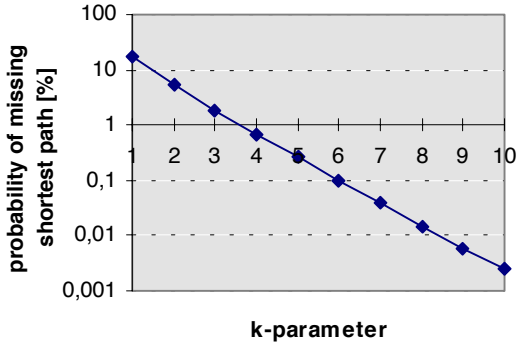

Figure 1: Probability to miss the shortest path versus the tuning parameter k.

Beyond $k = 4$, the probability of missing the shortest path has dropped below 1%.

(2) *The calculation time of TAMCRA increases only linearly with the value of k and saturates beyond a certain value of k.*

This is shown in Figure 2 for the same graph with 100 nodes and 200 links.

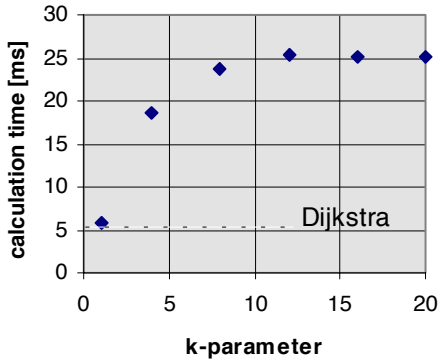

Figure 2: Calculation time of TAMCRA compared to the calculation time of the Dijkstra algorithm.

Beyond $k = 10$, the calculation time saturates. This implies that the value of k can be as large as needed without influencing the calculation time. The level at which the calculation time saturates depends on the size of the graph.

(3) *The value of k needed to solve the multiple constraints problem exactly is a polynomial function of the granularity (i.e. the number of possible values) of the constraints.*

In fact, during the run-time of TAMCRA, it can be verified when possible errors (due to a too small value of k) occur. Therefore, TAMCRA is easily modified into a "self-adapting TAMCRA" which always guarantees an exact result with minimal computation time.

(4) *The probability of missing the shortest path is, above a certain threshold, independent of the number m of constraints.*

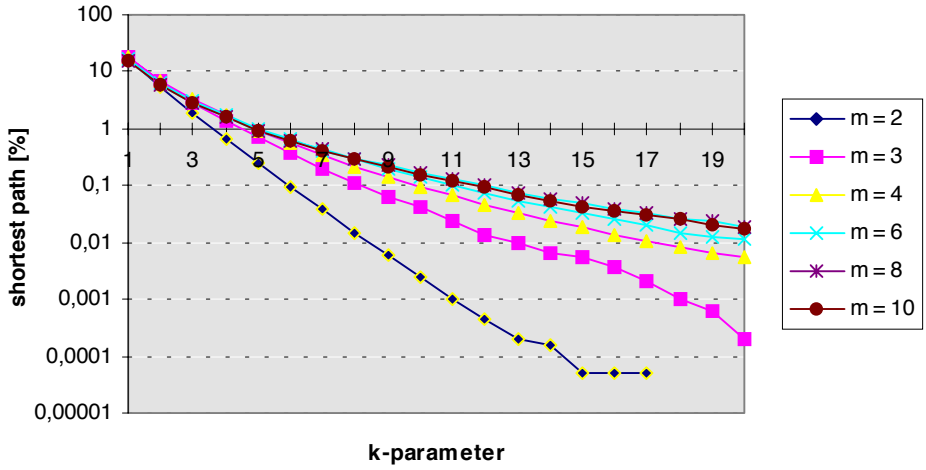

Figure 3. Probability of missing the shortest path as a function of the parameter *k* and the number of constraints *m*.

The insensitivity of *m* - when sufficiently large - on the probability of missing the shortest path is a very desirable property in real networks where most of the links are bi-directional and asymmetric. The latter implies that specifying an asymmetric link needs twice as many vector components as a symmetric link.

(5) *For a constant probability of missing the shortest path, k increases logarithmically in the number of nodes in the graph.*

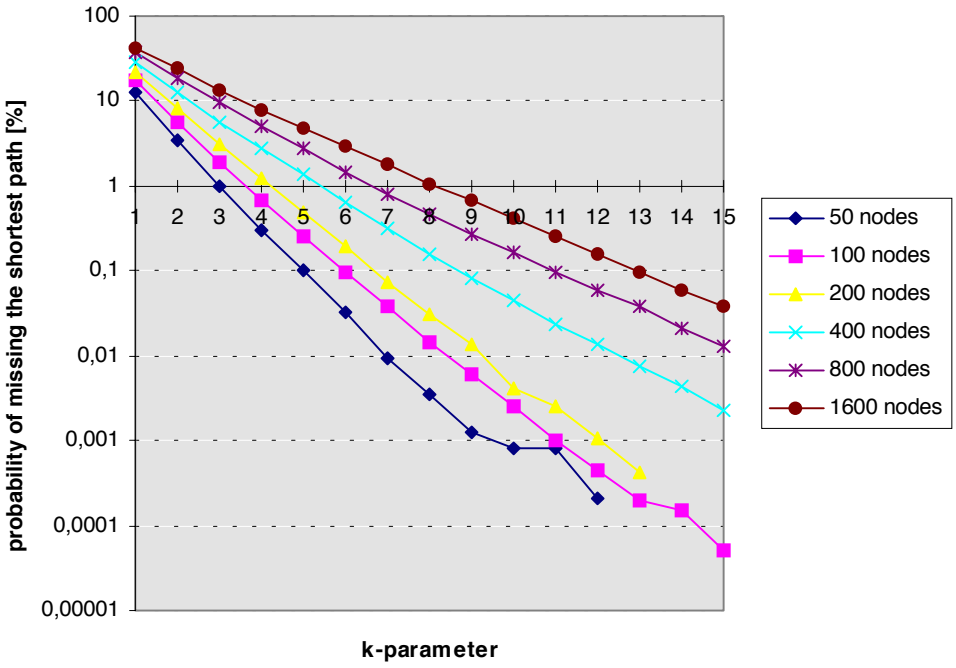

Figure 4. Probability of missing the shortest path as a function of the parameter *k* and the size of the graphs.

The simulated results indicates that $k = O(\log V)$ if we allow for a non-zero probability of missing the shortest path. This means, in practice, that the pseudo-polynomiality or NP-completeness is not that big an obstacle as announced earlier.

These results demonstrate that TAMCRA can solve multiple constraints routing problems with a high accuracy within a polynomial time-frame and scales very well as a function of the number of constraints and the size of the graphs. In conclusion, TAMCRA is perfectly suited to compute in connection oriented networks the desired path, such e.g. the designated transfer list (i.e. the hierarchical path subject to the user's QoS requirements) in PNNI.

In the Internet, a basically connection-less network, most routing protocols are based on hop-by-hop forwarding and pre-computed routing tables (Sales and Van Mieghem, 1998). Recently, we have shown (De Neve and Van Mieghem, 1998b) that 'hop-by-hop destination based only' quality of service routing based on TAMCRA is promising because in most cases, the correct (even best) QoS-path is found.

In summary, the algorithmic part of the QoS routing problem is feasible provided TAMCRA disposes of the precise knowledge of the network topology. This brings us to the 'updating or protocol' part of the QoS routing problem, discussed in the next section 3.

3. FLOODING DYNAMICS

3.1 TWO TIME SCALES

In a network topology as illustrated in Figure 5, we distinguish between (1) changes that occur infrequently and between (2) those that rapidly change in time. The first kind reflects topology changes due to failures and the joining/leaving of nodes. In the current Internet, only this kind of topology changes is considered. Its dynamic is relatively well understood. The key point is that the time between two 'first kind' topology changes is long compared to the time needed to flood this information over the whole network. Thus, the topology databases on which routing relies, converge rapidly with respect to the frequency of updates to the new situation and the transient period where the databases are not synchronized (which may cause routing loops), is generally small.

The second type of rapidly varying changes are typically those related to the consumption of resources or to the traffic flowing through the network. The metrics coupling to state information seriously complicates the dynamics of flooding because the flooding convergence time T can be longer than the change rate Δ of some metric (such as available bandwidth). Figure 5 illustrates how the bandwidth BW on a link may change as a function of time. In contrast to the first kind changes where $T << \Delta$, in the second kind changes, T can be of the same order as Δ. Apart from this, the second type changes necessitates the definition of a significant change that will trigger the process of flooding. In the first kind, every change was significant enough to start the flooding. The second kind significant change may be influenced by the flooding convergence time T and is, generally, strongly related to the traffic load in (a part of) the network. As will shown below, an optimal update strategy for the second type changes is highly desirable.

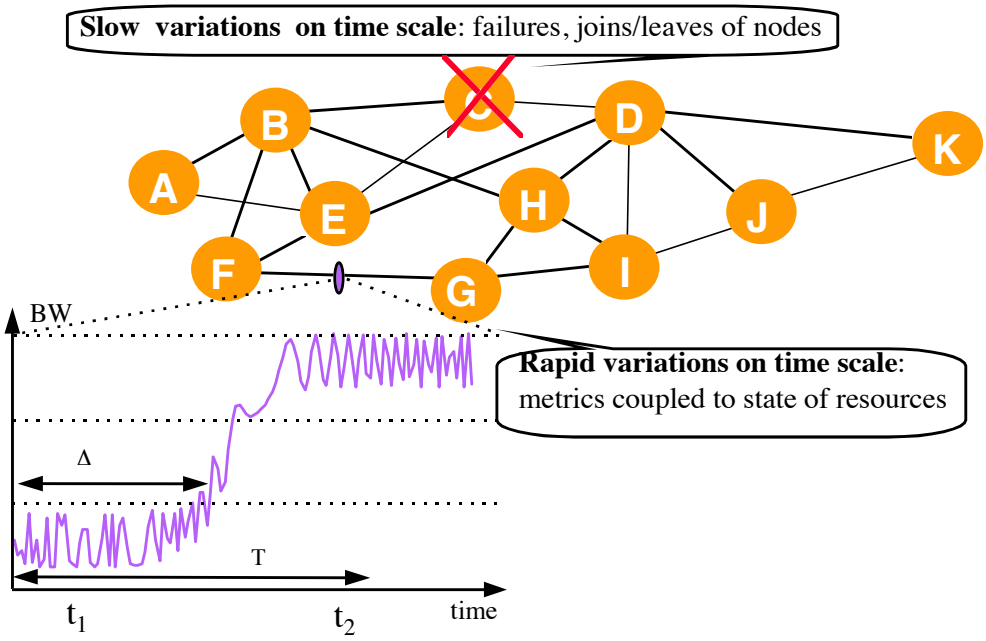

Figure 5. Network topology changes on different time scales

As far as we know, no detailed topology update strategy for the second type changes has been published, although some partial results and preliminary studies have already appeared (Apostolopoulos et al., 1998; Sivabalan and Mouftah, 1998).

3.2 GUARANTEED QOS ROUTING: A TWO PHAZES PROCESS

In this section we confine ourselves to *guaranteed* QoS provision which necessitates a connection oriented approach as demonstrated in Sales and Van Mieghem (1998). The establishment of a connection between two nodes in the network, say node A and node K as in Figure 6, takes place in two phazes. Based on the network topology reflecting a snap shot at time t_1 and flooded to the last node at $T+t_1$, the routing algorithm (e.g. TAMCRA) computes the path from A to K subject to some QoS requirements. Subsequently, in the second phaze, the needed resources along that path are installed in all nodes constituting that path. This phaze is known as the 'connection prerequisite' and the network function or protocol that executes this 'installation' (connection set-up) is called signaling. The signaling operates in a hop by hop mode: it starts with the first node and proceeds further to the next node if the 'installation' is successful. Due to the rapidly changing nature of the traffic in the network (especially in broadband or multimedia), at time $t_2 > t_1$ and at a certain node I (as exemplified in Figure 6), the connection set-up process may fail because the topology situation at time t_1 may significantly differ from that at time t_2 (Figure 5). Rather than immediately blocking the path request from A to K, current protocols (such as PNNI) invoke in this case an emergency process, called crankback. The idea is similar to back tracking. The failure in node I returns the previous node D with the responsibility to compute immediately an alternative path from itself towards K, in the hope that along that new path the set-up will succeed. The crankback process consumes both much CPU-time in the nodes as control data and yet, does not guarantee a successful connection set-up. When the crankback process returns back to the source node A and this node also fails to find a path to K, the connection request is blocked or rejected and much computational effort of cranking back was in vain.

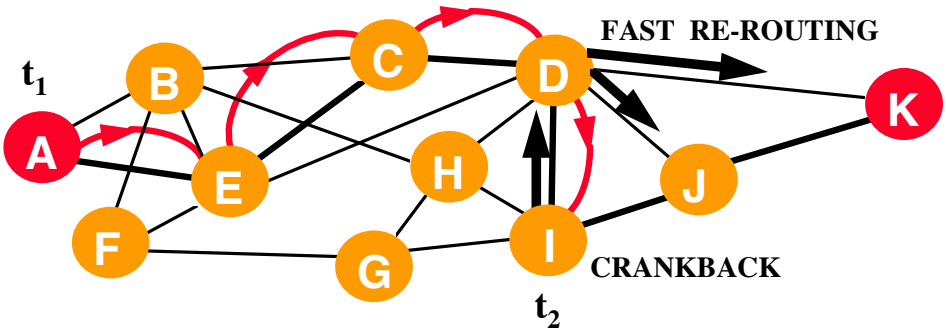

Figure 6. Crankback and fast-rerouting

Although the crankback process seems an interesting emergence 'exit' to prevent blocking, the efficiency certainly needs further study. But, just in those emergence cases due to heavy traffic, the crankback processes generates additional control traffic possibly causing a triggering of topology flooding, and hence even more control data is created, eventually initiating a positive feedback loop with severe consequences. Therefore, we suggest to prevent invoking crankback as much as possible by developing a good topology update strategy.

3.3 TWO FLOODING TECHNIQUES

In order to supply each node with a consistent view of the network topology, we discuss two different 'distribution' techniques, topology flooding and multicasting via a spanning tree. In topology flooding - in short, flooding - each node forwards the packet(s) containing the topology update information on all interfaces, except for the incoming interface. To avoid explosive generation of this packet, the flooding process is damped by some control function, often via lifetime or sequence number checking. Flooding is the simplest and most used technique. In addition, flooding is also the most robust and fastest method because the packet is forwarded along all possible paths from the originating node to each other node in the network. However, as flooding relies on using all possible paths and, consequently, as duplicates at nodes may arrive, it may consume considerable processing resources and bandwidth in the network.

Multicasting the topology information along a minimum spanning tree just optimizes the use of bandwidth and resources. But, multicasting requires the computation of a minimum spanning tree of the network. Even tough very efficient algorithms such as Prim's or Kruskal's minimum spanning tree algorithm (Cormen et al, 1995) exist, an additional computation and a storage of the multicast forward table in each node must be taken into account. In addition, multicasting is less robust than flooding because it is sensitive to changes in the network connectivity and because the forwarding only uses one path from source node to each other node.

The brief description of the properties of both flooding techniques suggests that the first kind of topology changes (joining/leaving of nodes) are best advertised via flooding while the second kind (rapid variations of metrics) may benefit from multicasting. The latter statement is conditional because it strongly depends on the topology graph $G(V,E)$ and no rigorous analysis has been published yet. However, we give some intuitive arguments. For second kind of rapid topology changes, the network connectivity remains stable - only the metrics vary - and the computation of the minimum spanning tree needs to be performed only when first kind of topology changes occur because only these alter the connectivity of the graph $G(V,E)$. Further, we expect that in heavy traffic more updates are needed. Hence, in this regime the impact of additional control traffic is more critical. Just in this regime, multicasting is advantageous over flooding, although the convergence of the

flooding process is faster on average. The slower convergence of multicast may be compensated by a more frequently triggering of the updates. Previous reasoning implicitly assumed that only min(max) measures (actually only available bandwidth) are considered as fast varying metrics. Indeed, we recommend that worst case values (corresponding to maximum allowed loading of the nodal queues) for additive measures (such as delay, log(1-clr), price,...) are advertised only when first kind topology changes occur. Thus, the minimum spanning tree algorithm may choose an additive metrics (e.g. delay) as optimization criterion.

In summary, we have discussed both the algorithmic as protocol part of the QoS routing problem. We feel confident that the first part is sufficiently solved by TAMCRA. The dynamic part containing an intelligent update-strategy is, however, still uncultivated ground waiting for further research exploitation.

4. REFERENCES

Apostolopoulos, G., R. Guerin, S. Kamat and S. K. Tripathi, 1998a, " Quality of Service Based Routing: A Performance Perspective", to appear in ACM Sigcomm'98,

ATMF-PNNI, 1996, Private Network Network Interface (PNNI), Specification Version 1.0, March 1996.

Calvert K., M. Doar, E. Zegura, 1997, "Modelling Internet Topology", IEEE Comm. Magazine, pp.160-163

Cormen, T. H., C. E. Leierson and R. L. Rivest, 1995, *Introduction to Algorithms*, The MIT Press, Cambridge.

Crawley, E., R. Nair, B. Rajagopalan and H. Sandick, 1998, "A framework for QoS-based Routing in the Internet", draft-ietf-qosr-framework-06.txt

De Neve, H. and P. Van Mieghem, 1998, "A Multiple Quality of Service Routing Algorithm for PNNI", IEEE ATM Workshop, (Fairfax, May 26-29), pp. 324-328.

De Neve, H. and P. Van Mieghem, 1998a, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm", submitted to IEEE Transactions on Networking.

De Neve, H. and P. Van Mieghem, 1998b, " Hop-by-hop Quality of Service Routing", submitted to IEEE INFOCOM'99.

Garey M. R. and D. S. Johnson, 1979, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freeman, San Francisco

Nichols, K, V. Jacobson and L. Zhang, 1998, "A Two-bit Differentiated Services Architecture for the Internet", draft-nichols-diff-svc-arch-00.pdf

Sales, B. and P. Van Mieghem, 1998, "Dual-mode routing: A Generic Framework for IP over ATM Integrated Routing", IEEE Symposium on Computer & Communications, ISCC'98, (Athens, June 30-July 2), pp. 326-328.

Sivabalan, M. and H. T. Mouftah, 1998, "Design Considerations for Link-state Routing Protocols", IEEE Symposium on Computer & Communications, ISCC'98, (Athens, June 30-July 2), pp. 53-57.