RAPPORT DE RECHERCHE

L R I

DISTRIBUTED ALGORITHMS FOR SINGLE AND MULTIPLE-METRIC LINK STATE QoS ROUTING

BADIS H / AL AGHA K

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud-LRI

07/2003

Rapport de Recherche N° 1365

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 650
91405 ORSAY Cedex (France)

Distributed Algorithms for Single and Multiple-Metric Link State QoS Routing

Hakim Badis[1,2] and Khaldoun Al Agha[1,2]
[1]Laboratoire de Recherche en Informatique (LRI)
Bât 490 Université Paris Sud
91405 Orsay Cedex, FRANCE
phone (33)(0)169156591; fax (33)(0)169156586

[2]Institut National de Recherche en Informatique et en Automatique (INRIA)
Domaine du Voluceau - B.P.105
78153 Le Chesnay Cedex, FRANCE

Email: {badis,alagha}@lri.fr

Abstract

Multimedia applications often require guaranteed quality of service (QoS) and resource reservation, which has raised a number of challenging technical issues for routing. Nevertheless, the QoS routing algorithm must be simple because a costly procedure does not scale with the size of the network. In this paper, we present scalable algorithms for single and multiple-metric link state QoS routing problem. Our routing algorithms can treat two, three and four-metrics and provides a polynomial heuristic solution. The algorithms analysis and numerical evaluation are presented. It is shown that our polynomial algorithms are close to the optimal solution computed by an exponential algorithm.

Résumé

Les applications multimédias exigent souvent la qualité de service (QoS) qui présente un défi dans Internet et encore plus dans le contexte d'Internet sans fil. L'algorithme de routage avec la QoS doit être simple parce que le temps d'exécution d'un procédé croît exponentiellement en fonction de la taille du réseau. Dans ce papier, nous présentons des algorithmes distribués pour les protocoles d'état des liens afin de trouver des chemins capables de transporter divers flux (données, audio, vidéo) tout en respectant des contraintes comme le délai de traversée, la bande passante, le coût, le nombre de sauts ou le taux de perte des paquets. Nos algorithmes de routage peuvent traiter deux, trois et quatre métriques et fournissent des solutions approximatives en un temps polynomial. L'analyse des résultats obtenus par des simulations montre que nos algorithmes trouvent les mêmes routes que celles obtenues par des algorithmes qui demandent un temps exponentiel.

1 Introduction

To facilitate the use of multimedia applications in the Internet, new characteristics and services must be added. New classes of service should be offered to provide better guarantees of QoS. An important issue for that purpose is the definition of a routing architecture that considers the QoS requirements of applications. In traditional routing, packets are delivered using a route based on their source and destination addresses while in QoS-based routing, the traffic requirements are also taken into account by the routing algorithm.

Routing protocols usually characterize the network with a single metric such as hop-count or delay, and use shortest path algorithms for the path computation. It is becoming increasingly clear that such routing protocols are inadequate for multimedia applications, such as video conferencing, which often require guaranteed QoS. For a network to support QoS requirements, routing must explicit information on resources available in the network so that applications can make proper resource reservation. To achieve that, it is necessary for routing to have a more complex model of the network, taking into account important network parameters such as bandwidth, delay and loss probability.

The complexity of the algorithm is a function of the number of metrics treated and of their composition rules. Wang and Crowcroft [1] proved that the problem of finding a path subject to constraints on n additive and m multiplicative metrics is *NP-Complete* if $n+m \geq 2$. Bandwidth can be taken into account by pruning links. Therefore, algorithms of acceptable complexity can only handle bandwidth and one additional metric. In the literature, most of the algorithms do not try to solve this complex problem, instead they define simpler problems. In this paper, we present algorithms based on Dijkstra's shortest path algorithm and Lagrange Relaxation (LR) that provide a polynomial solution for multiple-metrics. Lagrange relaxation is a common technique for calculating lower bounds, and finding good solutions. First Held and Karp raised with this technique for the Traveling Salesman Problem in [2, 3].

There are two approaches for packet forwarding: source routing and hop-by-hop routing. In the first one, the packet header has a list of routers that must be traversed from the source to the destination (the source chooses the route). In hop-by-hop routing, each router chooses the next hop that a packet will follow. In link-state routing protocols, packets are forwarded hop-by-hop at each node. Each node maintains a routing table with next hops for all destinations, and this table is usually computed periodically in response to routing updates. When a packet is received, hop-by-hop routing requires only a table lookup to find the next hop and send the packet to it. Link state routing can use fully distributed computation algorithms [4], which has lower memory requirements for the routers. In this paper we present distributed algorithms for single and multiple-metric link state QoS routing.

The remain of this paper is organized as follows. Section 2 presents previous works. In section 3, we present the composition rules of metrics. Section 4 describes the routing protocols with a single metric for path computation. In section 5, we show the single mixed metric approach that combines several QoS parameters (metrics) in one representative value. In section 6, we propose distributed algorithms to find the best path with multiple constraints. In section 7, we show simulation results of our algorithms by comparing them to an exponential algorithm. We propose in 8 to adapt the LR based Hop (LRH) algorithm for Ad hoc networks. We conclude in 9.

2 Previous Work

There are many previous works that investigate the problem of QoS-based routing.

Wang and Crowcroft [1] consider a number of issues in QoS routing. They first examine the basic problem of QoS routing, namely, finding a path that satisfies multiple constraints, and its implications on routing metric selection. They present a centralized algorithm that is suitable for source routing and two distributed algorithms that are suitable for hop-by-hop routing based bandwidth and delay constraints.

Chen [5] proposed a heuristic algorithm for the delay-cost constrained routing problem which is *NP-Complete*. The main idea of this algorithm is to first reduce the *NP-Complete* problem to a simpler one which can be solved in polynomial time, and then solve the new problem by either using an extended

Dijkstra's algorithm or extended Bellman-Ford algorithm.

Al-Fawaz and Woodward [6] propose a routing algorithm to find the shortest path between one source and one destination node while considering the criteria of multiple metric constraints. They have three goals to achieve, 1) Sorting the QoS metrics according to the requested service, 2) Finding the possible routes between the source and the destination and 3) Speed up the path determination by using sliding window and hierarchical clustering technique.

In [7] authors propose and analyze the performance of a distance-vector QoS routing algorithm that takes into account three metrics: propagation delay, available bandwidth and loss probability. Classes of service and metric combination are used to turn the algorithm scalable and as on a two-metric scheme.

Cheng and Nahrstedt [8] give an algorithm to find a path that meets two requirements in polynomial time. The algorithm reduces the original problem to a simpler one by modifying the cost function, based on which the problem can be solved using an extended shortest path algorithm. The shortcoming of this method is that the algorithm has to use high granularity in approximating the metrics, so it can be very costly in both time and space, and it cannot guarantee that the simpler problem has a solution if the original problem has.

The algorithms in [9, 10] are based on calculating a simple metric from the multiple requirements. By doing so, we can use a simple shortest-path algorithm based on a single cost aggregated as a combination of weighted QoS parameters. The main drawback of this solution is that the result is quite sensitive to the selected aggregating weights.

3 The composition rules for metrics

Let $G = (V, E)$ be the network with $|V| = n$ nodes and $|E| = m$ arcs and met_{ij} a metric for link (i, j). The value of a metric over any directed path $p = (i, j, k, ..., q, r)$ can be one of the following compositions:

- *Additive metrics:* We say metric met is additive if

  $$\text{met}(p) = \text{met}_{ij} + \text{met}_{jk} + ... + \text{met}_{qr}.$$

  It is obvious that delay (del), delay jitter (dej), hop-count (hop) and cost (co) follow the additive composition rule.

- *Multiplicative metrics:* We say metric met is multiplicative if

  $$\text{met}(p) = \text{met}_{ij} \times \text{met}_{jk} \times ... \times \text{met}_{qr}.$$

  The probability of successful transmission (pst) follows the multiplicative composition rule. The composition rule for loss probability (Lp) is more complicated. $\text{met}(p) = 1 - ((1 - \text{met}_{ij}) \times (1 - \text{met}_{jk}) \times ... \times (1 - \text{met}_{qr}))$. It can be transformed to an equivalent metric pst.

- *Concave metrics:* We say metric met is concave if

  $$\text{met}(p) = \min\{\text{met}_{ij}, \text{met}_{jk}, ..., \text{met}_{qr}\}.$$

  It is obvious that Bandwidth (Bw) follows the concave composition rule.

4 Single Metric Approach

In traditional data networks, routing protocols usually characterize the network with a single metric such as hop-count or delay, and use the shortest path algorithms for path computation.

For an additive metric such as delay, hop-count or cost, each arc (i, j) in the path p is assigned a number met_{ij}. When the arc (i, j) is inexistent, $\text{met}_{ij} = \infty$. The routing problem is to find a path

$p*$ between i and r so that met($p*$) is the minimum. In such a case, we use the well-known Dijkstra routing algorithm. Suppose that s is the source node, S the set of vertices whose shortest paths from the source have already been determined, $V - S$ the remaining vertices. Let d array of best estimates of shortest path to each vertex and *previous* an array of predecessors for each vertex. The Dijkstra's shortest path algorithm [11] is as follows:

Step 1: Initially, $S = \varnothing$, $d[s] = 0$, previous$[k] = 0$ and $d[k] = 0$ for all $k \neq s$, $k \in V - S$;

Step 2: Sort the vertex u in $V - S$ that has the maximum $d[u]$;

Step 3: $S = S \cup \{u\}$;

Step 4: For each vertex v which is a neighbour of u:
If $d[v] < d[u] + \text{met}_{uv}$ then
$d[v] = d[u] + \text{met}_{uv}$; previous$[v] = u$;

Step 5: If $V - S = \varnothing$ then the algorithm is complete.
Otherwise, go to step 2.

Step 4 updates the best estimates of all the vertices, $v \in V$, connected to a vertex, u.

For the probability of successful transmission (multiplicative metric), each arc (i, j) in the path p is assigned a number met_{ij}. When the arc (i, j) is inexistent, $\text{met}_{ij} = 0$. The routing problem is to find a path $p*$ between i and r so that met($p*$) is the maximum. In such a case, we propose a variant-Dijkstra routing algorithm.

Step 1: Initially, $S = \varnothing$, $d[s] = 1$, previous$[k] = 0$ and $d[k] = 0$ for all $k \neq s$, $k \in V - S$;

Step 2: Sort the vertex u in $V - S$ that has the maximum $d[u]$;

Step 3: $S = S \cup \{u\}$;

Step 4: For each vertex v which is a neighbour of u:
If $d[v] < d[u] \times \text{met}_{uv}$ then
$d[v] = d[u] \times \text{met}_{uv}$; previous$[v] = u$;

Step 5: If $V - S = \varnothing$ then the algorithm is complete.
Otherwise, go to step 2.

For a concave metric such as bandwidth, each arc (i, j) in the path p is assigned a number met_{ij}. When the arc (i, j) is inexistent, $\text{met}_{ij} = 0$. The routing problem is to find a path $p*$ between i and r so that maximizes met($p*$). In such a case, we propose a variant-Dijkstra routing algorithm.

Step 1: Initially, $S = \varnothing$, $d[s] = max$, previous$[k] = 0$ and $d[k] = 0$ for all $k \neq s$, $k \in V - S$;

Step 2: Sort the vertex u in $V - S$ that has the maximum $d[u]$;

Step 3: $S = S \cup \{u\}$;

Step 4: For each vertex v which is a neighbour of u:
If $d[v] < \min\{d[u], \text{met}_{uv}\}$ then
$d[v] = \min\{d[u], \text{met}_{uv}\}$; previous$[v] = u$;

Step 5: If $V - S = \varnothing$ then the algorithm is complete.
Otherwise, go to step 2.

The worst-case complexity of Dijkstra's algorithm on networks with nonnegative arc length depends on the way of finding the labeled node with the smallest distance label. A naive implementation that examines all labeled nodes to find the minimum runs in $O(n^2)$ time [11]. The implementation using k-ary heaps [12] runs on $O(m\log n)$ time (for a constant k). The implementation using Fibonacci heaps [13] runs in $O(m + n\log n)$ time. The implementation using one-level R-hraps [12] runs in $O(m + (n\log C))$ time and the one using two-level R-heaps together with Fibonacci heaps, in $O(m + n\sqrt{\log C})$ time.

5 Single Mixed Metric Approach

Using a single primitive parameter such as delay is clearly not sufficient for multimedia applications. We need information on other resources particularly bandwidth for supporting resource reservation. One possible approach might be to define a function and generate a single metric from multiple primitive parameters. The idea is to mix various pieces of information into a single measure and use it as the basis for routing decisions. For example, a mixed metric may be produced with bandwidth Bw, delay Del and loss probability Lp using a formula $f(p) = \frac{\text{Bw}(p)}{\text{Del}(p)\times\text{Lp}(p)}$ [14]. A path with a large value is likely to be a better choice in terms of bandwidth, delay and loss probability. Find a path $p*$ between i and r so that met$(p*)$ is the maximum. In such a case, we propose a variant-Dijkstra routing algorithm having the same complexity as that in Dijkstra's shortest path routing algorithm.

Step 1: Initially, $S = \varnothing$, $d[s] = max$, previous$[k] = 0$ and $d[k] = 0$, for all $k \neq s$, $k \in V - S$;

Step 2: Sort the vertex u in $V - S$ that has the maximum $d[u]$;

Step 3: $S = S \cup \{u\}$;

Step 4: For each vertex v which is a neighbour of u:
If $d[v] < \frac{\min\{Bw(u), \text{Bw}_{uv}\}}{(\text{Del}(u)+\text{Del}_{uv})\times(1-(\text{pst}(u)\times\text{pst}_{uv}))}$ then
$d[v] = \frac{\min\{Bw(u), \text{Bw}_{uv}\}}{(\text{Del}(u)+\text{Del}_{uv})\times(1-(\text{pst}(u)\times\text{pst}_{uv}))}$; previous$[v] = u$;

Step 5: If $V - S = \varnothing$ then the algorithm is complete.
Otherwise, go to step 2.

However, a mixed metric can be used as an indicator at best, as it does not contain sufficient information to assess whether user QoS requirements can be met or not. Another problem has to do with mixing parameters of different composition rules. For example, suppose that a path has two segments ab and bc. If metric $f(p)$ is delay, the composition rule is $f(ab+bc) = f(ab)+f(bc)$. If metric $f(p)$ is bandwidth, the rule is $f(ab+bc) = \min[f(ab), f(bc)]$. However, if $f(p) = \frac{\mathrm{Bw}(p)}{\mathrm{Del}(p)\times \mathrm{Lp}(p)}$, neither of the above are valid. In fact, there may not be a simple composition rule at all. Computing a path based on $f(p)$ does not guarantee each QoS parameter individually and the composition rule of this type of measurement is complex to define as the composition rules of the different parameter are different.

6 multiple metrics approach

Multiple metrics can certainly model a network more accurately. However, the problem of finding a path with n additive and m multiplicative metrics in *NP-Complete* if $n + m \geq 2$ [1]. Including a single metric, the best path can be easily defined. Otherwise, including multiple metrics, the best path with all parameters at their optimal values may not exist. For example, a path with both maximum bandwidth and minimum delay may not necessarily exist.

6.1 one concave and one additive as metrics

Let us come back to delay, delay jitter, hop-count, cost, loss probability and bandwidth. It is clear that any two or more of delay, delay jitter, hop-count, cost and loss probability in any combination as metrics are *NP-Complete*. The only feasible combinations are bandwidth and one of the five additive metrics or bandwidth and one multiplicative metric to provide a polynomial solution. In this section, we choose the bandwidth and delay as the routing metrics. However, the algorithm presented is generic and apply to other routing metrics with similar composition rules, for example, bandwidth and delay jitter or bandwidth and cost.

As we have mentioned before, a path with both maximum bandwidth and minimum delay may not necessarily exist. Thus, we must decide the precedence among the metrics in order to define the best path. The delay has two basic components: queuing delay and propagation delay. The queuing delay is more dynamic and traffic-sensitive, thus bandwidth is often more critical for most multimedia applications. If there is no sufficient bandwidth, queuing delay and probably the loss rate will be very high. So, we define the precedence as bandwidth and then the propagation delay. Our strategy is to find a path with maximum bandwidth (a widest path), and when there is more than one widest path, we choose the one with shortest delay. We refer to such a path as the shortest-widest path. The widest path problem is to find a path $p*$ between i and j that maximize $\mathrm{Bw}(p*)$. For a given topology, there are usually many widest paths with equal width, and loops can be formed as a result. However, shortest-widest path is always free of loops. Intuitively, the delay metric eliminates the loops.

Theorem 1: *shortest-widest paths are loop-free in a distributed computaion.*

Proof: By contradiction. Suppose that node A and node B are involved in a loop for destination C (Figure 1). Path p_1p_2 is the shortest-widest path from node A to node C and path $p_1^*p_2^*$ is the shortest-widest path from node B to node C.

By the definition of the shortest-widest paths, we have

$$\mathrm{width}(p_2^*) \leq \mathrm{width}(p_1p_2) \qquad (1)$$

$$\mathrm{width}(p_2) \leq \mathrm{width}(p_1^*p_2^*) \qquad (2)$$

Note that

$$\mathrm{width}(p_1^*p_2^*) = \min[\mathrm{width}(p_1^*), \mathrm{width}(p_2^*)]$$

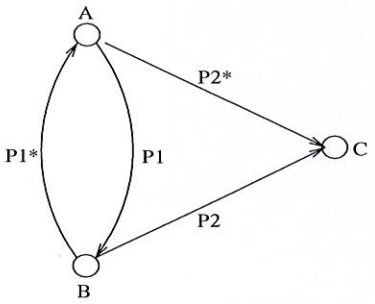

Figure 1: A loop involving node A and node B

$$\leq \text{width}(p_2^*) \qquad (3)$$

Similarly,

$$\begin{aligned} \text{width}(p_1 p_2) &= \min[\text{width}(p_1), \text{width}(p_2)] \\ &\leq \text{width}(p_2) \qquad (4) \end{aligned}$$

From (1), (3) and (4), we have

$$\text{width}(p_1^* p_2^*) \leq \text{width}(p_2) \qquad (5)$$

Comparing (5) with (2), we have

$$\text{width}(p_1^* p_2^*) = \text{width}(p_2) \qquad (6)$$

Similarly, we have

$$\text{width}(p_1 p_2) = \text{width}(p_2^*) \qquad (7)$$

Equation (6) shows that path $p_1^* p_2^*$ and path p_2 are equal widest paths, since path $p_1^* p_2^*$ is the shortest-widest path, we have

$$\text{length}(p_2) \geq \text{length}(p_1^* p_2^*) > \text{length}(p_2^*) \qquad (8)$$

Similarly, equation (7) shows that path $p_1 p_2$ and path p_2^* are equal widest paths, since path $p_1 p_2$ is the shortest-widest path, we have

$$\text{length}(p_2^*) \geq \text{length}(p_1 p_2) > \text{length}(p_2) \qquad (9)$$

Equation (8) and (9) contradict each other. This completes the proof.

Each arc (i, j) in the path is assigned the following values: del_{ij}, which is the propagation delay, Bw_{ij}, which is the available bandwidth. When the arc (i, j) is inexistent, $\text{del}_{ij} = \infty$ and $\text{Bw}_{ij} = 0$. Let $\text{del}(p) = \text{del}_{ij} + \text{del}_{jk} + ... + \text{del}_{qr}$ and $\text{Bw}(p) = \min\{\text{Bw}_{ij}, \text{Bw}_{jk}, ..., \text{Bw}_{qr}\}$. Let $d1$ array of best estimates of widest paths and $d2$ the length of those paths to each vertex respectively. The shortest-widest path algorithm based on dijkstra algorithm is as follows:

Step 1: Initially, $S = \varnothing$, $d1[s] = max$, $d2[s] = 0$, previous$[v] = 0$, $d1[v] = 0$ and $d2[v] = \infty$ for all $v \neq s$ in $V - S$;

Step 2: Sort the vertex u in $V - S$ that has the maximum $d1[u]$;

Step 3: If u has more than one element, find $k \in u$ so that length$(s, ..., j, k) = \min_{j \in S}[d2[j] + \text{del}_{jk}]$, $S = S \cup \{k\}$;

Step 4: For each vertex v which is a neighbor of $u = k$:
If $d1[v] < \min\{d1[u], \text{Bw}_{uv}\}$ then
$d1[v] = \min\{d1[u], \text{Bw}_{uv}\}$; $d2[v] = d2[u] + \text{del}_{uv}$; previous$[v] = u$

Step 5: If $V - S = \varnothing$, the algorithm is complete.
Otherwise, go to step 2.

Step 2 finds a node with maximum width from s, if there are more than one widest path found, step 3 chooses the one with minimum length. Step 4 updates the width and length for neighbors of u. The time complexity of the shortest-widest path algorithm is equal to that Dijkstra's shortest path algorithm.

6.2 Two additive metrics

It has been proven in [1] that a routing problem is *NP-Complete*, if the number of additive QoS parameters that should be minimized are more than or equal to two. We cannot hope an algorithm that can find the theoretical optimum and runs in polynomial time. In the literature, most of the algorithms does not try to solve this complex problem, instead they define simpler problems. In this section, we present an algorithm to find a path that is minimal for one additive metric, and the other additive metric of it remains under a given bound. For example, find a path that is minimal for a hop-count or cost, and the delay of it remains under a given bound (Delay Constrained Least Hop-count path problem (DCLH), Delay Constrained Least Cost path problem (DCLC)).

Each arc (i, j) in the path is assigned two additive values: met1 and met2. The problem is to minimize the met1 of the path, while keep the met2 under a given constraint Δ_{met2}. In a formal description, we are looking for

$$\min\{\text{met1}(p) : p \in P(s,t) \text{ and met2}(p) \leqslant \Delta_{met2}\} \qquad (10)$$

Where $P(s,t)$ is the set of paths from the source node s to the destination node t. Our heuristic is based on the Lagrange relaxation. Lagrange relaxation is the common technique for calculating lower bounds, and finding good solutions for this problem. First Held and Karp raised with this technique for the Traveling Salesman Problem in [2, 3]. We neglect the constraining conditions (this is the relaxation), and build them into the objective function. The solutions feasible to the original problem can certainly suit the relaxation conditions as well, so we can get a lower bound of the original problem. If the path found is not feasible to the constraining conditions, we increase the dominance of it in the modified metric1 function, enforcing the solution to approach to the optimal solution. Moreover, decrease the difference between the obtained lower bound and the optimum of the original problem as well. This is the base of the Lagrange relaxation [15]. First we study Lagrange relaxation for the optimization problem with inequality constraints described by

$$\min\{f(x) : g(x) \leqslant 0 \text{ and } x \in X \subseteq \Re^n\} \qquad \text{(a)}$$

That is, for $\lambda \in \Re^m$ we construct the problem

$$\min\{f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) : x \in X \subseteq \Re^n\} \qquad \text{(b)}$$

The λ_i are called Lagrangians or Lagrange multipliers. Not that lagrangians are not variables but parameters. For every λ we have an optimization problem (b) in the variable x. The function $f_\lambda(x) := f(x) + \lambda^t g(x) = f(x) + \sum \lambda_i g_i(x)$ is called the Lagrange function to (a).

Proposition 1: If $\lambda \geqslant 0$ then (b) is a relaxation of (a).

Proof: See [15].

Proposition 2: Assume that $\overline{x}_\lambda$ is the optimal to (b). Then $\overline{x}_\lambda$ is optima

1. $g(\overline{x}_\lambda) \leqslant 0$
2. $\lambda_i g_i(\overline{x}_\lambda) = 0, i = 1, ..., m$
3. $\lambda \geqslant 0$.

Proof: See [15].

The presented **Lagrange Relaxation based Met1** (LRM) algorithm consists of the heuristic that minimizes $\text{met1}_\lambda := \text{met1} + \lambda \times \text{met2}$ modified objective function. For a given (fixed) λ, we can easily calculate the minimal path (p_λ). If $\lambda = 0$ and $\text{met2}(p_\lambda) \leq \Delta_{met2}$ we found an optimal solution for the original problem as well. If $\text{met2}(p_\lambda) > \Delta_{met2}$ we must increase λ, to increase the dominance of the met2 in the modified objective function. So, we increase λ while the optimal solution of met1_λ suits the delay requirements. With the help of Lagrange relaxation we have an algorithm which can find the optimal λ for a given source destination pair. Moreover it gives an estimation for the optimal solution, although we have no guarantee of finding the optimal solution, we always get a bound for the solution, which tells that at most how far is it from the optimal solution. To find the value of λ that gives the best result, we use the following Claim.

Claim 1: Let $L(\lambda) := \min\{\text{met1}_\lambda(p) : p \in P(s,t)\} - \lambda \times \Delta_{met2}$. Then $L(\lambda)$ is a lower bound to problem (10) for any $\lambda \geq 0$. (11)

Proof: let $p*$ denote the optimal solution of (10). Then

$$\begin{aligned} L(\lambda) &:= \min\{\text{met1}_\lambda(p) : p \in P(s,t)\} - \lambda \times \Delta_{met2} \\ &\leq \text{met1}_\lambda(p*) - \lambda \times \Delta_{met2} \\ &= \text{met1}(p*) + \lambda(\text{met2}(p*) - \Delta_{met2}) \\ &\leq \text{met1}(p*). \end{aligned}$$

Proves the claim.

To obtain the best lower bound we need to maximize the function $L(\lambda)$, that is we are looking for the value

$L^* := max_{\lambda \geq 0} L(\lambda)$, (12)

and the maximizing λ^*. Now, some properties of the function $L(\lambda)$ are given. The simple proofs are left to the reader.

Claim 2: L is a concave piecewise linear function, namely the minimum of the linear functions $\text{met1}(p) + \lambda \times (\text{met2}(p) - \Delta_{met2})$ for all $p \in P(s,t)$.

Claim 3: For any $\lambda \geq 0$ and met1_λ-minimal path p_λ, $\text{met2}(p_\lambda)$ is a supgradient of L in the point λ.

Claim 4: Whenever $\lambda < \lambda^*$, then $\text{met2}(p_\lambda) \geq \Delta_{met2}$ and if $\lambda > \lambda^*$, then $\text{met2}(p_\lambda) \leq \Delta_{met2}$ for each met1_λ-minimal path p_λ.

Claim 5: A value that maximizes the function $L(\lambda)$ if and only if there are paths p_{met1} and p_{met2} which are met1_{λ^*}-minimal and for which $\text{met2}(p_{met1}) \geq \Delta_{met2}$ and $\text{met2}(p_{\text{met2}}) \leq \Delta_{\text{met2}}$. ($p_{\text{met1}}$ and p_{met2} can be the same, in this case $\text{met2}(p_{\text{met2}}) = \text{met2}(p_{\text{met1}}) = \Delta_{\text{met2}}$).

Claim 6: let $0 \leq \lambda_1 < \lambda_2$, and $p_{\lambda_1}, p_{\lambda_2} \in P(s,t)$ λ_1-minimal and λ_2-minimal paths. Then $\text{met1}(p_{\lambda_1}) \leq \text{met1}(p_{\lambda_2})$ and $\text{met2}(p_{\lambda_1}) \geq \text{met2}(p_{\lambda_2})$.

These two latter Claims together give that the λ^* maximizing the function $L(\lambda)$ gives the best modified cost function, that is λ^* is the smallest value for which there exists a met1_{λ^*}-minimal path p_{met2} which satisfies the metric2 constraint. The LRM algorithm is as follows:

Step 1: $p_{\text{met1}} := \text{Dijkstra}(s, t, \text{met1})$;

Step 2: If $\text{met2}(p_{\text{met1}}) \leq \Delta_{\text{met2}}$ then return p_{metr1}, the algorithm is complete.

Step 3: $p_{\text{met2}} := \text{Dijkstra}(s, t, \text{met2})$;

Step 4: If $\text{met2}(p_{\text{met2}}) > \Delta_{\text{met2}}$ then return "there is no solution", the algorithm is complete.

Step 5: $\lambda := \frac{\text{met1}(p_{\text{met1}}) - \text{met1}(p_{\text{met2}})}{\text{met2}(p_{\text{met2}}) - \text{met2}(p_{\text{met1}})}$;

Setp 6: $p_{\text{Update}} := \text{Dijkstra}(s, t, \text{met1}_\lambda)$

Step 7: If $\text{met1}_\lambda(p_{\text{Update}}) = \text{met1}_\lambda(p_{\text{met1}})$ then return p_{met1}.
Otherwise,
If $\text{met2}(p_{\text{Update}}) \leq \Delta_{\text{met2}}$ then
$p_{\text{met2}} := p_{\text{Update}}$; Otherwise,
$p_{\text{met1}} := p_{\text{Update}}$;

Step 8: Go to step 5.

Step 1 calculates shortest path on met1_λ modified metric1 function with Dijkstra algorithm by setting λ at 0. $\text{Dijkstra}(s, t, \text{met1})$ return a met1-minimal path between the nodes s and t. In step 2, if the path found meets the metric2 requirement Δ_{met2}, this is the optimal path, and the algorithm stops. Otherwise, step 3 calculates the shortest path on the metric 2. If the obtained path suits the metric 2 requirements (Δ_{met2}), a proper solution exists. Otherwise, there is no suitable path from s to t that can fulfill the metric 2 requirement, so the algorithm stops. In step 5, by using the claim 5, we can calculate λ because p_{met1} and p_{met2} are met1_λ-minimal. In this case, the only possible λ is $\lambda := \frac{\text{met1}(p_{\text{met1}}) - \text{met1}(p_{\text{met2}})}{\text{met2}(p_{\text{met2}}) - \text{met2}(p_{\text{met1}})}$. After in step 6, we find a met1_λ-minimal path p_{Update}. If $\text{met1}_\lambda(p_{\text{Update}}) = \text{met1}_\lambda(p_{\text{met1}})$ then p_{met1} and p_{met2} are met1_λ-minimal, so the optimal path is p_{met2}. Otherwise, we replace either p_{met1} or p_{met2} with p_{Update} according the metric 2 constaints.

Theorem 3: *The final path constructed by LRM for a given source s and destination t does not contain any loops.*

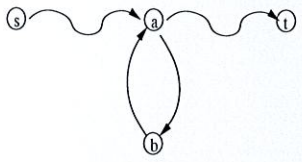

Figure 2: A loop involving node a and node b

Proof: By contradiction. Let *s-a-b-t* the optimal path found by LRM the optimal λ (Figure 2). *s-a-b-t* is the shortest path in term of met_λ i.e.,

$\forall p \neq (s\text{-}a\text{-}b\text{-}t) \in P(s,t)\ \text{met1}_\lambda(p) > \text{met1}_\lambda(s\text{-}a\text{-}b\text{-}t)$ (13)

We have $\text{met1}(s\text{-}a\text{-}t) \leq \text{met1}(s\text{-}a\text{-}b\text{-}t)$ and $\text{met2}(s\text{-}a\text{-}t) \leq \text{met2}(s\text{-}a\text{-}b\text{-}t)$. So,

$$\text{met1}(s\text{-}a\text{-}t)+\lambda\times(\text{met2}(s\text{-}a\text{-}t)-\Delta_{\text{met2}}) \leq \text{met1}(s\text{-}a\text{-}b\text{-}t)+\lambda\times(\text{met2}(s\text{-}a\text{-}b\text{-}t)-\Delta_{\text{met2}}) \quad (14)$$

Equation (13) and (14) contradict each other. This completes the proof.

Theorem 4: *The LRM algorithm terminates after $O(m\log^3 m)$ iteration, so the running time of the algorithm is $O(m^2\log^4 m)$* (Dijksta algorithm runs on $O(m\log n)$ time and $n << m$).

Because there are only finite number of different path, the algorithm finds the optimal λ in a finite number of steps. The LRM algorithm can not find the optimal solution if there are several path the same met1_λ at the optimal λ (Figure 4), therefore the LRM cannot decide which path is the optimal on the base of met1_λ.

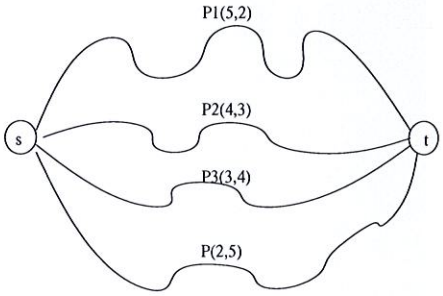

Figure 3: A network of four paths from s to t

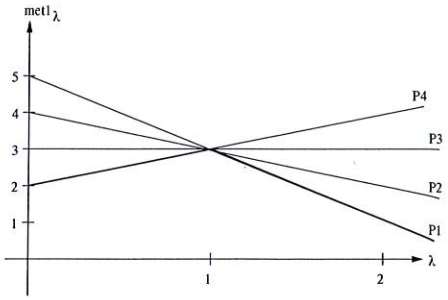

Figure 4: Sevral paths with the same met1_λ

If we run the LRM algorithm in the network of Figure 3 ($\Delta_{met2} = 4$), we first calculate shortest

path on met1_λ modified metric1 function with Dijkstra algorithm by setting λ at 0. We obtain the path $p4$. We have $\text{met2}(p_4) = 5 > \Delta_{\text{met2}} = 4$. We calculate the shortest path on the metric 2. We obtain p_1. We have $\text{met2}(p_1) = 2 < \Delta_{\text{met2}} = 4$. So, there is a solution. $\lambda = 1$ is given by the step 5 of the LRH algorithm. Now, we calculate P_{Update} with Dijkstra algorithm by setting λ at 1. We obtain four paths with the same $\text{met1}_\lambda = 3$ (p_1: $5+1\times(2-4) = 3$, $p2$: $4+1\times(3-4) = 3$, p_3: $3+1\times(4-4) = 3$, p_4: $2+1\times(5-4) = 3$). One solution possible of this problem is to take the path P, so that $\text{met1}(p)$ is the minimum and $\text{met2}(p) \leq \Delta_{met2}$.

6.3 Two additive and one concave metrics

It is clear that two additive and one concave metrics problem is *NP-Complete*. In this case, we take hop-count and delay as additive metrics and bandwidth as concave metric. We consider the Delay and Bandwidth Constrained Least Hop path problem (DBCLH).

Given two constants, the minimum bandwidth $\Delta_{bandwidth}$ and the maximum delay Δ_{delay}. The Delay and Bandwidth Constrained Least Hop path problem (DBCLH) is to find a path p from i to r minimal for a hop-count, satisfying $del(p) \leqslant \Delta_{delay}$ and $\text{Bw}(p) \geqslant \Delta_{bandwidth}$. The formal description is: $\min\{\text{hop}(p) : p \in P(s,t)$ and $\text{del}(p) \leqslant \Delta_{delay}$ and $\text{Bw}(p) \geqslant \Delta_{bandwidth}\}$, where $P(s,t)$ is the set of paths from the source node s to the destination node t, and $\text{hop}(p)$ is the hop-count. Our heuristic is based on the Lagrange Relaxation. As we have mentioned before, we define the precedence as bandwidth and then the propagation delay.

Theorem 5: *A path has a width no less of $\Delta_{bandwidth}$, if and only if each link in the path has a bandwidth no less than $\Delta_{bandwidth}$.*

Proof: if each link in p has a bandwidth no less than $\Delta_{bandwidth}$, it is obvious that $\text{width}(p) \geq \Delta_{bandwidth}$. Suppose that $\text{width}(p) \geq \Delta_{bandwidth}$ but there is a link (i,j) with Bw_{ij} less than $\Delta_{bandwidth}$. We then have $\text{width}(p) = \text{Bw}_{ij} < \Delta_{bandwidth}$, which contradicts the assumption that $\text{width}(p) \geq \Delta_{bandwidth}$.

Theorem 5 implies that any links with a bandwidth less than $\Delta_{bandwidth}$ are not parts of the path we want. Hence to find paths satisfying $\text{Bw}(p) \geqslant \Delta_{bandwidth}$, we eliminate any links with a bandwidth less than $\Delta_{bandwidth}$ so that any paths in the resulting graph satisfy $\text{width}(p) \geq \Delta_{bandwidth}$. The LR algorithm for the DBCLH problem is as follows:

Step 1: Set $\text{del}_{ij} = \infty$ and $\text{hop}_{ij} = \infty$, if $\text{Bw}_{ij} < \Delta_{bandwidth}$;

Step 2: $p_{\text{del}} := \text{Dijkstra}(s,t,\text{del})$;

Step 3: If $\text{del}(p_{\text{del}}) = \infty$ then return "there is no solution"; the algorithm is complete.

Step 4: If $\text{del}(p_{\text{hop}}) \leq \Delta_{\text{del}}$ then return p_{hop}; the algorithm is complete.

Step 5: $p_{\text{del}} := \text{Dijkstra}(s,t,\text{del})$;

Step 6: If $\text{hop}(p_{\text{del}}) > \Delta_{\text{del}}$ then return "there is no solution", the algorithm is complete.

Step 7: $\lambda := \frac{\text{hop}(p_{\text{hop}}) - \text{del}(p_{\text{del}})}{\text{del}(p_{\text{del}}) - \text{del}(p_{\text{hop}})}$;

Setp 8: $p_{\text{Update}} := \text{Dijkstra}(s,t,\text{hop}_\lambda)$;

Step 9: If $\text{hop}_\lambda(p_{\text{Update}}) = \text{hop}_\lambda(p_{\text{hop}})$ then return p_{hop}.
Otherwise,
If $\text{del}(p_{\text{Update}}) \leq \Delta_{\text{del}}$ then $p_{\text{del}} := p_{\text{Update}}$;
Otherwise, $p_{\text{hop}} := p_{\text{Update}}$;

Step 10: Go to step 7.

Theorem 6: *The final path constructed by LR algorithm for the DBCLH problem does not contain any loops.*

proof: see the proof of the theorem 3.

The LR algorithm for the DBCLH problem has the same complexity as the LRM algorithm described before. The step 2 runs in $O(m)$ time. The other steps in $O(m^2 \log^4 m)$ time.

6.4 Four-Metrics Heuristic

There is no routing algorithm in literature that treat four-metrics. We consider the Delay, Bandwidth and Loss probability Constrained Least Hop path problem (DBLCLH). Here, we use the single mixed concept. This single mixed metric combines delay and loss probability, but uses the absolute value of the logarithmic transmission-success probability function (*slog*) instead of the loss probability to avoid complex composition rules. $slog(p) = |log(1 - loss(p))|$. Furthermore, we assume that routes cannot have more than 90% of loss probability, that is, $0 < slog(p) < 1$. As ms unit is appropriate [16] to represent delay, we assume that delay is integer. As a consequence, we have a simple single-metric representation of delay and loss, where the integer part is delay and the decimal part represents loss. Each arc (i, j) in the path is assigned the value $\text{sm}_{ij} = \text{slog}_{ij} + \text{del}_{ij}$. When the arc (i, j) is inexistent, then $\text{sm}_{ij} = \infty$. Let $\text{sm}(p) = \text{sm}_{ij} + \text{sm}_{jk} + ... + \text{sm}_{qr}$. Given three constants, The minimum bandwidth $\Delta_{bandwidth}$, The maximum delay Δ_{delay} and the maximum logarithmic transmission-success probability $\Delta_{success}$, where the maximum single mixed metric is $\Delta_{ms} = \Delta_{delay} + \Delta_{succes}$. The Delay, Bandwidth and Loss probability Constrained Least Hop path problem (DBLCLH) is to find a path p from i to r that is minimal for a hop-count, satisfying $\text{sm}(p) \leqslant \Delta_{sm}$ and $\text{Bw}(p) \geqslant \Delta_{bandwidth}$. The DBLCLH's formal description is: $\min\{\text{hop}(p) : p \in P(s,t)$ and $\text{sm}(p) \leqslant \Delta_{sm}$ and $\text{Bw}(p) \geqslant \Delta_{bandwidth}\}$. The DBLCLH problem is *NP-complete*. The LR algorithm for the DBLCLH problem is as follows:

Step 1: Set $\text{sm}_{ij} = \infty$ and $\text{hop}_{ij} = \infty$, if $\text{Bw}_{ij} < \Delta_{bandwidth}$;

Step 2: $p_{\text{sm}} := \text{Dijkstra}(s, t, \text{sm})$;

Step 3: If $\text{sm}(p_{\text{sm}}) = \infty$ then return "there is no solution"; the algorithm is complete.

Step 4: If $\text{sm}(p_{\text{hop}}) \leq \Delta_{\text{sm}}$ then return p_{hop}; the algorithm is complete.

Step 5: $p_{\text{sm}} := \text{Dijkstra}(s, t, \text{sm})$;

Step 6: If $\text{hop}(p_{\text{sm}}) > \Delta_{\text{sm}}$ then return "there is no solution"; the algorithm is complete.

Step 7: $\lambda := \frac{\text{hop}(p_{\text{hop}}) - \text{sm}(p_{\text{sm}})}{\text{sm}(p_{\text{sm}}) - \text{sm}(p_{\text{hop}})}$;

Setp 8: $p_{\text{Update}} := \text{Dijkstra}(s, t, \text{hop}_\lambda)$;

Step 9: If $\text{hop}_\lambda(p_{\text{Update}}) = \text{hop}_\lambda(p_{\text{hop}})$ then return p_{hop}.
Otherwise,
If $\text{sm}(p_{\text{Update}}) \leq \Delta_{\text{sm}}$ then $p_{\text{sm}} := p_{\text{Update}}$;
Otherwise, $p_{\text{hop}} := p_{\text{Update}}$;

Step 10: Go to step 7.

proof: see the proof of the theorem 3.

The LR algorithm for the DBLCLH problem has the same complexity as the LRM algorithm described before. There are two cases, when the algorithm cannot find the optimal solution. In the first case, if there is no suitable path p^* from s to t that can fulfill the bandwidth requirement. Here, we use the best effort or distributing traffic among multiple routes. In the second case, the algorithm finds several paths with the same hop_λ at the optimal λ, therefore the algorithm can not decide which path is the optimal on the base of hop_λ. Here we propose an algorithm that can be used to make decisions in situations involving multiple and prioritized constraints.

6.5 Four-Metrics routing decision

When multiple constraints are important to a routing algorithm, it is too difficult to choose between alternatives. We propose an algorithm that can be used to make decisions in situation involving multiple and prioritized constraints. Similar problems can be found in the operations research area. For example, Thomas saaty's analytic hierarchy process (AHP) is a well-known technique in such cases [17]. In our case, for example the priority order is bandwidth (1), delay (2), loss probability (3) and hop-count (4). The algorithm is as follow:

Step 1: Find the nominated paths that satisfy the all QoS constraints;

Step 2: Generate priority normalized weights, pw_i, $i \in \{1, 2, 3\}$;

Step 3: Calculate the normalized path scores nsp_{ij} for each path j on each metric i;

Step 4: Calculate the total normalized score of path $\text{tnsp}_j = \sum_{i=1}^{3} \text{pw}_i \times \text{nsp}_{ij}$, for each path j;

Step 5: Choose the path with the highest total score.

We consider a network with seven paths from a source (s) to a destination (t) illustrated in Table 1.

path p_2 did not meet the bandwidth constraint ($10 < 20$), p_4 did not meet the hop-count constrain ($10 > 8$), p_5 did not meet the delay constraint ($30 > 25$), p_6 did not the loss probability constraint ($30e - 6 > 20e - 6$). Suppose that for this example, the generated priority normalized weights are represented in Table 2.

The normalized scores of each path on each metric are given in Table 3.

Computing each path's total score, we obtain

- p_1 total score = 0.260999
- p_3 total score = 0.407021
- p_7 total score = 0.331958

We choose the path with the highest total score. So, the path p_3 is selected.

Path	Bw	del	Lp	hop
1	40	10	2e-6	5
2	10	20	10e-6	6
3	60	15	5e-6	7
4	40	22	8e-6	10
5	25	30	15e-6	5
6	30	14	30e-6	4
7	30	18	10e-6	2
REQ	20	25	20e-6	8

Table 1: Paths with QoS constraints

Priority	Bw	del	Lp	hop
pw	0.465819	0.27714	0.16107	0.0959699

Table 2: Priority normalized weight

Path	Bw	del	Lp	hop
1	0.30769	0.23255	0.11764	0.35714
3	0.46153	0.34883	0.29411	0.50000
7	0.23076	0.41860	0.58823	0.14285

Table 3: Normalized path scores on each metric

7 Simulation Results

We used simulation for our evaluation of the performance of the LR method in case of two, three and for metrics. We built 3 random networks with 100 nodes (Figure 5).

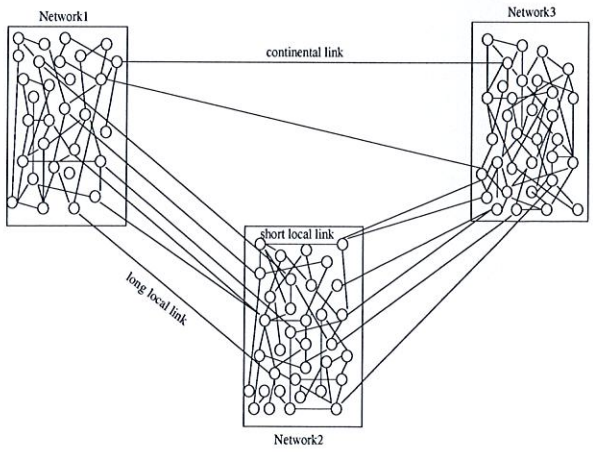

Figure 5: Topology model

A random generator was used to create links interconnecting the nodes [18]. The output of this random generator is always a connected network in which each node's degree is at least 2. We adjust the parameters of the random generator carefully to obtain realistic network topologies with an average node degree of 4, which is close to the average node degree of current internetworks. For the LRM

method, we consider that met1 as the cost and met2 as the propagation delay. The cost value on links varies from 1 to 30 based on uniform distribution. The propagation delay on links is selected from three ranges to resemble delay characteristics of nationwide network e.g. in the US. The first range (1-8 ms) represents shortest local links; the second range (8-12 ms) represents longer local links, while the third range (20-30 ms) represents continental links. On each network, 20% of nodes have longer local links and 2% have continental links.

7.1 Average Cost

Figure 6 shows the average cost of the paths found by the LRM algorithm compared with the Constrained Bellman-Ford (CBF) [19] algorithm that gives the optimal solution between a source and a set of destination nodes, but the running time is exponential.

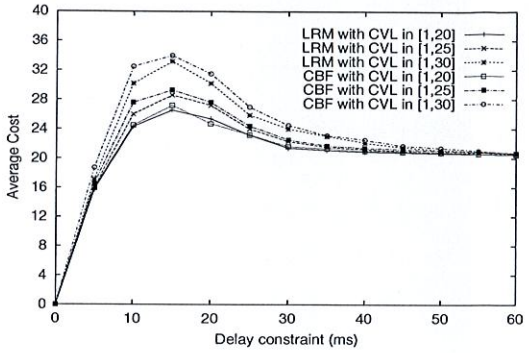

Figure 6: Average cost

Based on uniform distribution, the Cost Value on Links (CVL) varies from 1 to 20, 1 to 25, and 1 to 30. The LRM algorithm found almost the same paths as CBF. For $\Delta_{delay} < 15$, it is impossible to find short paths for all source destination pairs with a small Δ_{delay}, thus the cost of them are also small. As the delay bound increases the algorithm find more and more (longer) paths, therefore the average of cost of the paths increases. At $\Delta_{delay} = 15$, the algorithm can find path for all source destination pairs. For $\Delta_{delay} > 15$, the LRM algorithm is able to find the paths with lower costs and higher delay. So, the average cost of the paths will decrease with the delay bound.

7.2 Average Number of setps

We use the average number steps to measure the running time of the algorithm. The number of steps represents the events when the algorithm changes the contents of the heap. The curve of the Figure 7 is obtained by varying the CVL from 1 to 20.

For $\Delta_{delay} < 15$, as the LRM algorithm finds more and more paths, the number of Dijkstra executions increases. After $\Delta_{delay} > 15$, the LRM algorithm can find the optimal solution after the execution of the first Dijkstra because the minimum cost paths computed can more and more satisfy the delay bounds.

8 Future work: DBCLH problem in QOLSR protocol

Most routing protocols for Mobile Ad hoc NETworks (MANETs) [20], such as OLSR [21], are designed without explicitly considering QoS of the routes they generate. The number of hops is the most common criterion adopted by such proposed routing protocols. It is necessary to take into account important network parameters such as bandwidth and delay. To achieve that, we have developed the QOLSR protocol (OLSR based QoS) [22] that propagates MPR's delay and bandwidth information to each

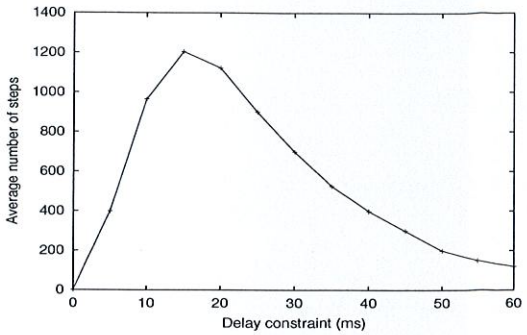

Figure 7: Average number of step

node in the network. Now, we propose to adapt the LR based Hop (LRH) algorithm for link-state protocols.

OLSR protocol is an optimization over the classical link-state protocol, tailored for Mobile Ad hoc NETworks (MANETs). The key concept used in the protocol is that of multipoint relays (MPRs). The idea of MPR is to minimize the overhead of flooding messages in the network by reducing duplicate retransmission in the same region. In route calculation, The MPRs are used to form the route from a given node to any destination in the network.

Let $G = (V, E)$ be the network with $|V|$ nodes and $|E|$ arcs and $p = (i, j, k, ..., q, r)$ a directed path. When the arc (i, j) is inexistent or j is not a MPR of i (due to the OLSR routing mechanism), then $\text{del}_{ij} = \infty$ ($\text{sm}_{ij} = \infty$), $\text{hop}_{ij} = \infty$ and $\text{Bw}_{ij} = 0$ for the DBCLH problem (DBLCLH problem). The application of LRH algorithm will be only over the MPR nodes.

9 Conclusions

QoS routing provides better QoS guarantees to applications and improves the network resource utilization. Nevertheless, routing algorithms have to be carefully designed or the complexity will compromise their implementation. There are no routing algorithms in literature that treat four-metrics in polynomial time. Our routing algorithms can treat two-metrics, three-metrics and four-metrics and provides a polynomial heuristic solution. For a single metric approach, we use Dijkstra shortest path algorithm and its variants. For a multiple metric approach, we have proposed algorithms that provide a polynomial loop-free solution. We have compared the performance of the LRM algorithm with the CBF algorithm. The LRM algorithm is close to the optimal solution in a polynomial time. We have proposed to adapt our algorithms for link-state protocols in Ad hoc networks to satisfy the end-to-end QoS requirement.

An ad

References

[1] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications*, vol. Vol.14, no. 7, pp. 1228-1234, September 1996.

[2] M. Held and R. Karp, "Quality of service routing for supporting multimedia applications," *Operation Research 18*, vol. pp. 1138-1162, 1970.

[3] R. Karp and M. Held , "The traveling salesman problem and minimum spanning trees, Part II," *Mathematical Programming 6*, vol. pp. 62-88, 1971.

[4] J. Garcia-Luna-Aceves, "A Unified Approach to Loop-Free Routing Using Distance Vectors or Link States," *In Proc. Sigcomm'89, Texas, USA*, September 1989.

[5] S. Chen, "Routing Support for Providing Guaranteed End-to-End Quality of Service," *PhD, in Engineering Collage. Urbana*, 1999.

[6] M. Al-Fawaz and M. E. Woodward, "Fast Quality of Service Routing Algorithms with Multiple Constraints," *8th IFIP Workshop on ATM&IP, Ilkely, UK*, 2000.

[7] L. H. Costa, S. Fdida and M. B. Duarte, "Distance-vector QoS-based Routing with Three Metrics," *NETWORKING*, pp. 847–858, 2000.

[8] S. Cheng and K. Nahrstedt, "On finding multi-constrained paths," *ICC'98, Atlanta, Georgia*, 1998.

[9] H. NEVE, P. Mieghem , "A multiple quality of service routing algorithm for PNNI ," *IEEE ATM'98 Workshop, Fairfax, Virginia*, pp. 306–314, 1998.

[10] L. Guo and I. Matta, "Search Space Reduction in QoS Routing," *Technical Report NU-CCS-98-09*, October 1998.

[11] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numer. Math., 1:269-271*, 1959.

[12] T. H. Cormen, C. E. Leiserson and R. L. Rivest, "Introduction to algorithms," *MIT Press, Cambridge, MA*, 1990.

[13] M. L. Fredman and R. E. Tarjan, "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms," *J. Assoc. Comput, 34:596-615*, Mach 1987.

[14] Z. Wang and J. Crowcroft , "Bandwidth-Delay Based Routing Algorithms," *IEEE GlobeCom'95, Singapore*, November 1995.

[15] M. Guignard, "Lagrange Relaxation," *Belgian Journal of Operation Research, Special Issue Francoro, Vol. 35 (3-4)*, 1995.

[16] A. Fei, G. Pei, R. Liu and L. Zhang, "Measurements on delay and hop-count of the Internet," *IEEE GlobeCom'98- Internet Mini-Conference*, 1998.

[17] W. L. Winston, "Operations research: Applications and Algorithms," *3rd edition: International Thomson Publishing*, 1994.

[18] H. Salama, "Multicast Routing for Real-time Communication on High-Speed Networks," *PhD thesis, North Carolina State University, Department of Electrical and Computer Engineering*, 1996.

[19] R. Widyono, "The Design and Evaluation of Routing Algorithms for Real-time Channels ," *Technical Report TR-94-024, University of California at Berkeley*, June 1994.

[20] "http://www.ietf.org/html.charters/manet-charter.html."

[21] T. Clausen P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol," *In IETF Internet Draft, draft-ietf-manet-olsr-09.txt*, April 2003.

[22] H. Badis, A. Munaretto, K. Al Agha and Guy pujolle, "QoS for Ad hoc Networking Based on Multiple-Metric: Bandwidth and Delay," *MWCN'03, Singapore*, October 2003.

RAPPORTS INTERNES AU LRI - ANNEE 2003

N°	Nom	Titre	Nbre de pages	Date parution
1345	FLANDRIN E LI H WEI B	A SUFFICIENT CONDITION FOR PANCYCLABILITY OF GRAPHS	16 PAGES	01/2003
1346	BARTH D BERTHOME P LAFOREST C VIAL S	SOME EULERIAN PARAMETERS ABOUT PERFORMANCES OF A CONVERGENCE ROUTING IN A 2D-MESH NETWORK	30 PAGES	01/2003
1347	FLANDRIN E LI H MARCZYK A WOZNIAK M	A CHVATAL-ERDOS TYPE CONDITION FOR PANCYCLABILITY	12 PAGES	01/2003
1348	AMAR D FLANDRIN E GANCARZEWICZ G WOJDA A P	BIPARTITE GRAPHS WITH EVERY MATCHING IN A CYCLE	26 PAGES	01/2003
1349	FRAIGNIAUD P GAURON P	THE CONTENT-ADDRESSABLE NETWORK D2B	26 PAGES	01/2003
1350	FAIK T SACLE J F	SOME b-CONTINUOUS CLASSES OF GRAPH	14 PAGES	01/2003
1351	FAVARON O HENNING M A	TOTAL DOMINATION IN CLAW-FREE GRAPHS WITH MINIMUM DEGREE TWO	14 PAGES	01/2003
1352	HU Z LI H	WEAK CYCLE PARTITION INVOLVING DEGREE SUM CONDITIONS	14 PAGES	02/2003
1353	JOHNEN C TIXEUIL S	ROUTE PRESERVING STABILIZATION	28 PAGES	03/2003
1354	PETITJEAN E	DESIGNING TIMED TEST CASES FROM REGION GRAPHS	14 PAGES	03/2003
1355	BERTHOME P DIALLO M FERREIRA A	GENERALIZED PARAMETRIC MULTI-TERMINAL FLOW PROBLEM	18 PAGES	03/2003
1356	FAVARON O HENNING M A	PAIRED DOMINATION IN CLAW-FREE CUBIC GRAPHS	16 PAGES	03/2003
1357	JOHNEN C PETIT F TIXEUIL S	AUTO-STABILISATION ET PROTOCOLES RESEAU	26 PAGES	03/2003
1358	FRANOVA M	LA "FOLIE" DE BRUNELLESCHI ET LA CONCEPTION DES SYSTEMES COMPLEXES	26 PAGES	04/2003
1359	HERAULT T LASSAIGNE R MAGNIETTE F PEYRONNET S	APPROXIMATE PROBABILISTIC MODEL CHECKING	18 PAGES	01/2003
1360	HU Z LI H	A NOTE ON ORE CONDITION AND CYCLE STRUCTURE	10 PAGES	04/2003
1361	DELAET S DUCOURTHIAL B TIXEUIL S	SELF-STABILIZATION WITH r-OPERATORS IN UNRELIABLE DIRECTED NETWORKS	24 PAGES	04/2003
1362	YAO J Y	RAPPORT SCIENTIFIQUE PRESENTE POUR L'OBTENTION D'UNE HABILITATION A DIRIGER DES RECHERCHES	72 PAGES	07/2003
1363	ROUSSEL N EVANS H HANSEN H	MIRRORSPACE : USING PROXIMITY AS AN INTERFACE TO VIDEO-MEDIATED COMMUNICATION	10 PAGES	07/2003