Generalized Multi-Constrained Path (G_MCP) QoS Routing Algorithm for Mobile Ad hoc Networks

Kunagorn Kunavut and Teerapat Sanguankotchakorn
Telecommunications Field of Study
School of Engineering and Technology
Asian Institute of Technology, Thailand
Email: {kunagorn.kunavut, teerapat}@ait.ac.th

***Abstract*— In mobile ad hoc network, efficient routing protocol is required to perform route discovery and maintenance. These protocols can be classified into two main types which are proactive and reactive routing protocols. Most of them usually use the suboptimal path to reach destination without considering QoS parameters. This results in network congestion during high traffic load situation. Hence, many algorithms have been proposed to offer QoS routing to these protocols. However, most of them find the feasible path by using only one or two QoS metrics. This is not enough to support many applications with QoS guaranteed, especially multimedia applications since they have more stringent various QoS requirements. To provide QoS routing in ad hoc network based on such environment, we propose the effective algorithm called Generalized MCP (G_MCP) to find the feasible path based on proposed weighted Connectivity Index (combination of link connectivity and capacity) and non-linear cost (combination of multiple additive QoS metrics using non-linear function). We adopt the fall-back approach. That is, G_MCP will find the path from source to destination by considering weighted Connectivity Index first. If there is a tie, the path with least non-linear cost will be chosen. Based on this approach, G_MCP has the comparable time complexity with the Shortest-Widest Path algorithm. We construct the simulation in a number of scenarios based on proactive protocol called OLSR. According to the simulation results, it is obvious that G_MCP performances are superior than OLSR and Shortest-Widest Path algorithms in terms of throughput, packet delivery ratio, delay and success ratio. Therefore, it can be concluded that G_MCP is able to support various applications and be operated well in highly dynamic mobility environment.**

***Index Terms*— Connectivity Index, non-linear cost, multi-constrained path, QoS routing, mobile ad hoc network**

I. INTRODUCTION

Mobile ad hoc network (MANET) is an infrastructure-less wireless network. Each node in this network behaves like a router to find paths and also forward packets to destinations by using effective routing protocols. These protocols can be approximately classified into two main types which are proactive and reactive protocols. In proactive or table-driven protocols, every nodes maintain their routing tables by periodically exchanging routing information. The examples of proactive protocol are DSDV [1], TBRPF [2] and OLSR [3]. In reactive or on-demand protocols, all nodes do not maintain their topology information. The paths to destinations are necessarily obtained through route discovery process only when they are required. AODV [4] and DSR [5] are the examples of this routing category.

Among these protocols, OLSR is effectively operable in heavy load or congested networks because of its ability to periodically compute paths to destinations [6]. OLSR consists of four main modules: neighbor sensing, message flooding via multipoint relay (MPR) nodes [7], topology information and path computation. Every nodes implementing OLSR collect Hello messages from the others in order to discover the 1-hop and 2-hop neighbors. It also floods the topology control (TC) messages to the others every predefined interval in order to build a partial topology of network. According to this function, only MPR nodes are allowed to forward TC messages to reduce duplicate transmissions of control overhead. However, OLSR generates more control overhead than reactive protocols. After creating partial topological information of the network, each node computes paths to destinations using path computation module which usually finds the shortest hop count paths.

Most of traditional ad hoc routing protocols usually select the suboptimal paths to reach destinations by using either delay or hop count as the routing metric. This causes the network to be easily congested during high load situation. Thus, many QoS routing algorithms [9]–[23] were proposed to allow nodes to find paths by considering various QoS parameters such as bandwidth, delay, reliability and etc. However, most of them apply only one or two QoS metrics which are mainly bandwidth and/or delay in path computation process. Hence, these algorithms are not suitable to support a large number of ubiquitous and different QoS required applications (see Table I).

In general, QoS metrics can be roughly classified into two types which are additive and non-additive QoS metrics. In case of additive QoS metrics, an end-to-end cost of the path is determined by the sum of the individual link cost along the path from source to destination. Whereas the non-additive QoS metrics of a path are given by

Manuscript received November 17, 2011; accepted December 17, 2011

doi:10.4304/jcm.7.3.246-257

TABLE I.
QOS REQUIREMENTS OF COMMON APPLICATIONS [8]

Applications	QoS Requirements*		
	Bandwidth	Delay	Reliability
Web browsing	Medium	Medium	High
Email	Low	Low	High
FTP	Medium	Low	High
Telnet	Low	Medium	High
IP telephony	Low	High	Low
Video conference	High	High	Low

Note: *This is a relative comparison.

the minimum or maximum value of an individual link. Many research works [30]˘[32] have been proposed to solve multi-constrained path (MCP) problem in wireline network by combining multiple additive QoS metrics in a single mixed metric or cost. This combined cost is applied in path computation process as a routing metric to find the feasible paths to destinations. However, only additive QoS parameters are considered in path selection process of these algorithms. The non-additive QoS parameter such as bandwidth is left behind. In [23], the concrete method to apply MCP QoS routing in mobile ad hoc networks was demonstrated. But the considered QoS parameters are only additive type.

Hence, we propose the QoS routing algorithm called Generalized MCP (G_MCP) to find the feasible paths. We introduce weighted Connectivity Index (combination of link connectivity and capacity), and non-linear cost [30] (combination of multiple additives QoS metrics using non-linear function) as non-additive and additive QoS parameters in our proposed algorithm, respectively. Thus, it is capable to support variety of applications since multiple QoS constraints are put into account by combining them into mixed metrics. It is called G_MCP since it is œxible and open to be applied to any networks including both wireline and wireless networks. However, in this work, we consider this algorithm in only mobile ad hoc networks in order to ensure that they are capable to support QoS required applications and be able to operate in highly dynamic topology environment.

II. RELATED WORK

Most of traditional ad hoc routing protocols either proactive or reactive protocols lack capability to support QoS. Thus, many QoS routing algorithms have been researched in decades to offer QoS over ad hoc networks [9]˘[23]. These QoS routings can be classified into two main paradigms which are source and hop-by-hop QoS routings.

A. Source QoS Routings for Ad Hoc Networks

In source QoS routing, source node sends the request packets toward the destination node in order to gather global state information of the network and also monitor if the constrained path satisfies the QoS requirements or not. Consequently, the path will be selected if it has enough resources to support the required application.

In [9], Ticket-based QoS routing mechanism was proposed to find the feasible paths with enough resource to satisfy either delay or bandwidth. Flexible QoS Model for MANETs (FQMM) [10] offers QoS routing by performing additional QoS check function later after the routes to destinations are found to ensure that the generated traffic is not greater than the bandwidth specified in each traffic profile.

The Adaptive Dispersity QoS Routing (ADQR) protocol [11], an extension from the Signal Power Adaptive Fast Rerouting (SPAFAR) protocol [12], finds paths to destinations based on signal strength. However, if there is no single path satisfying bandwidth requirement, the routing algorithm will find multiple disjoint paths with longer-lived connections. The Ad hoc QoS On-demand Routing (AQOR) [13] provides QoS support by selecting the shortest end-to-end delay link satisfying bandwidth requirement.

In QoS-Aware Source-Initiated Ad hoc Routing (QuaSAR) [14], the QoS metrics incorporated in this routing algorithm are namely, battery power, signal strength, bandwidth and delay. The applications have opportunity to independently select the ranking of them since they have different QoS requirements. QoS-Aware Routing Based on Bandwidth Estimation for Mobile Ad hoc Networks (BEQR) [15] considers only bandwidth constraint during route discovery process. This algorithm offers two methods to estimate the available bandwidth which are Listen and Hello methods (see Table II for comparison of these source QoS routings).

B. Hop-by-Hop QoS Routings for Ad Hoc Networks

Algorithms applying hop-by-hop QoS routings locally calculate their own state information such as available bandwidth, signal strength, battery power, loss rate and etc., and distribute these information to the other nodes in the network. These information will be used by the others in path computation process to find the feasible paths to destinations satisfying QoS constraints.

A Core-Extraction Distributed Ad hoc Routing Algorithm (CEDAR) [16] creates core nodes for performing route computation. Once admissible route is set up via core node, shortest-widest path is selected among all available paths. In [17], Widest Path heuristic was proposed in OLSR to find the maximum bandwidth path to destination by modifying both MPR selection and route calculation processes. QoS-Enhanced OLSR Routing in Mobile Ad hoc Networks (QOLSR) [18] was also proposed to allow each node to use Shortest-Widest Path algorithm to find feasible paths to destination by selecting paths with maximum link bandwidth. If there is more than one widest path, a path with shortest delay will be chosen. Many works which adopted the idea of QOLSR [19], [20] were also proposed.

In [21], Widest-Shortest Path algorithm was proposed to offer interference-aware QoS routing by finding a path with minimum hop count. If there is a tie, the widest bandwidth link will be selected. In [22], the new QoS

TABLE II.
COMPARISON OF QOS ROUTING PROTOCOLS

Routing Protocol	Route Discover	QoS Routing Scheme	Routing Metric
Ticket-based [9]	Reactive	Source	Bandwidth, Delay and Cost
FQMM [10]	Reactive	Source	Bandwidth
ADQR [11]	Reactive	Source	Signal Strength and Bandwidth
AQOR [13]	Reactive	Source	Bandwidth and Delay
QuaSAR [14]	Reactive	Source	Battery Power, Signal Strength, Bandwidth and Delay
BEQR [15]	Reactive	Source	Bandwidth
CEDAR [16]	Proactive and Reactive	Hop-by-hop	Bandwidth and Hop Count
Widest Path [17]	Proactive	Hop-by-hop	Bandwidth
QOLSR (Shortest-Widest Path) [18]˘[20]	Proactive	Hop-by-hop	Bandwidth and Delay
Widest-Shortest Path [21]	Proactive	Hop-by-hop	Bandwidth and Hop Count
Shortest-Highest Path [22]	Proactive	Hop-by-hop	Weighted CI and Delay
MCP QoS Routing [23]	Proactive	Hop-by-hop	Multiple Additive QoS Metrics

routing metric was proposed to combine link connectivity defined by Connectivity Index and link capacity into a single metric called weighted Connectivity Index (CI). The Shortest-Highest Path algorithm was also proposed to find the path with highest weighted CI and shortest end-to-end delay. In [23], the Multi-Constrained Path (MCP) QoS routing was proposed to select the feasible path based on only various additive QoS metrics (see Table II for comparison of these hop-by-hop QoS routings).

III. NOTATION AND PROPOSED QOS METRICS

A. Notation

A graph G, denoted by $G = (V, E)$, where V is the set of vertices and E is the set of edges or a relation that associated between two vertices. When we refer to the network, a vertex is a node and an edge is a link between two nodes. A link between two nodes i and j is represented as (i, j) and each link $(i, j) \in E$. The number of links associated with a node x in a graph of network G is called degree of a node x, denoted by $d(x)$. If each link of a graph is associated with some specific values (weights), such graph is said to be weighted.

A subgraph of G originating and covering up to n-hop from node i is defined by $G_i^{n-hop} = (V_i^{n-hop}, E_i^{n-hop})$, where V_i^{n-hop} is the set of all nodes contained within n hops of node i, and E_i^{n-hop} is the set of links associating between two nodes in G_i^{n-hop}.

When a path (or link) from node x to node y exists, where $x, y \in V$, this path is denoted by p_{xy}.

B. Proposed Weighted Connectivity Index

A topological index is a numeric quantity which is mathematical derived from the structural graph of a molecule. In 1975, Randić [24] introduced the Connectivity Index (CI) so called Randić Index which has become the widely used topological index in many applications such as chemical and physical properties [25]˘[27].

The Randić Index is defined in the literature as follows:

$$\chi = \chi(G) = \sum_{(i,j)\in E} \frac{1}{\sqrt{d(i)d(j)}} \quad (1)$$

where the summation is carried out over all links of G.

In this work, we propose to use CI of node, defined as the CI of subgraph originating at each node, to illustrate the link characteristic of every node in the network. It is shown in [28] that the higher value of CI, the better link connectivity of mobile node in ad hoc network is. This implies that nodes with lower link connectivity have higher probability to cause link break in the connecting paths since they may move out of the coverage area of their neighbors. Of course, this leads to increasing dropped packets caused by disconnected links which also affects throughput. Thus, it can be anticipated that the network performances such as throughput and packet delivery ratio can be improved if link connectivity is put into account in path selection process.

In wireless networks, link capacity (available channel bandwidth) indicates transmission capacity of data. This implies somewhat that the higher the link capacity is, the stronger the link connectivity becomes. Therefore, by considering the connectivity index of node, the combined merit of degree of nodes and link capacity can be achieved. In [22], we proposed new QoS routing metric called weighted CI of node which is defined as the Randić Index of subgraph modified to accommodate the link capacity. We verified that weighted CI can be effectively used as the QoS routing metric to improve the network performances.

The weighted CI of any nodes i in graph G can be computed by partitioning the network graph G into subgraph G_i^{n-hop} covering only nodes and links within n-hop from node i. Let u and v represent nodes in a set of V_i^{n-hop} and each link (u, v) is in a set of E_i^{n-hop}. The n-hop weighted CI of node i or $\chi_w(G_i^{n-hop})$ can be defined as

$$\chi_w(G_i^{n-hop}) = \sum_{(u,v)\in E_i^{n-hop}} \frac{q_{(u,v)}}{\sqrt{d(u)d(v)}} \quad (2)$$

where $q_{(u,v)}$ $(0 \leq q_{(u,v)} \leq 1)$ is normalized link capacity and when $q_{(u,v)} = 0$ refers to unavailable linkf. The term $\frac{q_{(u,v)}}{\sqrt{d(u)d(v)}}$ is called the Connectivity Index Contribution Factor (CICF) between node u and v denoted by $\psi(u, v)$ which is defined as follows:

$$\psi(u, v) = \frac{q_{(u,v)}}{\sqrt{d(u)d(v)}} \quad (3)$$

In this work, we define normalized available link bandwidth as $q_{(u,v)}$ which refers to the ratio of free time (idle period) to overall observed time. Free time can be measured by sensing the channel to monitor the traffic status and determine how much link bandwidth is available for transmitting and/or receiving. This bandwidth estimation method is approximately the same as the Listen$f$ scheme which is used in [15], [29].

To calculate weighted CI based on more number of hop counts results in more accurate information about node s connectivity. However, to obtain the node link state information beyond the neighboring nodes (more than 1 hop count), more control overhead and more bandwidth consumption are needed. Thus, there is a tradeoff between information accuracy and bandwidth consumption.

In OLSR which is proactive routing protocol, a node can collect state information up to 2-hop neighbors by using Hello messages. Thus, it is natural for any node i to collect information up to 2-hop neighbors to calculate its own weighted CI. Therefore, weighted CI based on 2-hop information of any node i called 2-hop weighted CI or $\chi_w(G_i^{2-hop})$ can be computed without generating excessive control overhead to the network. For any acyclic graph, 2-hop weighted CI of any node can be calculated simply by using its neighbors 1-hop weighted CI as shown in [22]. However, for any connected graph, 2-hop weighted CI can be generally computed as shown in the Proposition 1:

Proposition 1: For any connected graph, 2-hop based weighted Connectivity Index of any node is equal to the summation of 1-hop based weighted Connectivity Index of its neighbors subtracted by the summation of CICF of all links between its neighbors.

This proposition can be proved simply by example. Assume that Fig. 1 depicts a subgraph G_X^{2-hop} which is a graph partitioned based on 2-hop criteria of node X. The nodes A, B and C are 1-hop neighbors, while nodes D, E, F, G, H, I, J, K and L are 2-hop neighbors of node X, respectively. Links drawn in dash lines are not included in a subgraph G_X^{2-hop} since they are considered as the 3-hop links from node X. Each 1-hop and 2-hop links are associated with CICF which are denoted by lower case letters (a $\sim$ v). Thus, 2-hop weighted CI based on fresh view of node X or $\chi_w(G_X^{2-hop})$ can be computed as

$$\chi_w(G_X^{2-hop}) = \sum_{(u,v)\in E_X^{2-hop}} \psi(u,v)$$

where E_X^{2-hop} is a set of all links satisfying 2-hop criteria of node X, then

$$\begin{aligned}\chi_w(G_X^{2-hop}) &= a+b+c+d+e+f+g+h+i \\ &\quad +j+k+l+m+n+p+q+r \\ &\quad +s+t+u+v \\ \chi_w(G_X^{2-hop}) &= (a+b+c+d+m+n+p+q) \\ &\quad +(e+f+g+h+n+r+s+t) \\ &\quad +(i+j+k+l+p+s+u+v) \\ &\quad -(n+p+s) \\ &= [\chi_w(G_A^{1-hop})+\chi_w(G_B^{1-hop}) \\ &\quad +\chi_w(G_C^{1-hop})]-[\psi(A,B) \\ &\quad +\psi(A,C)+\psi(B,C)]\end{aligned}$$

Thus, we can conclude that

$$\chi_w(G_X^{2-hop}) = \sum_{j\in V_X^{1-hop};j\neq X} \chi_w(G_j^{1-hop}) - \sum_{(u,v)\in E_X^{1-hop};u,v\neq X} \psi(u,v) \quad (4)$$

where V_X^{1-hop} is a set of all nodes that are contained within 1 hop of node X and E_X^{1-hop} is a set of all links that are contained within 1 hop of node X. In any graph without cycles where there is no link between its neighbors, 2-hop weighted CI computation of node X as expressed in Eq. (4) is reduced to

$$\chi_w(G_X^{2-hop}) = \sum_{j\in V_X^{1-hop},j\neq X} \chi_w(G_j^{1-hop}) \quad (5)$$

which concides with the result shown in [22].

In any connected graph, this proposition provides a simple implementation (in terms of control overhead) for any node $i \in V$ to compute 2-hop weighted CI ($\chi_w(G_i^{2-hop})$), since there is no additional control overhead generated to the networks. In OLSR, every nodes can sense up to 2-hop neighbors, hence, they know the degree of their 1-hop neighbors. Consequently, they are able to compute their own 1-hop weighted CI ($\chi_w(G_i^{1-hop}), i \in V$) which will be piggybacked onto Hello messages and flooded to their neighbors. By collecting these information, each node i knows degree of its neighbors and is able to compute $\sum_{(u,v)\in E_i^{1-hop}} \psi(u,v); u,v \neq i$. Each node i also knows

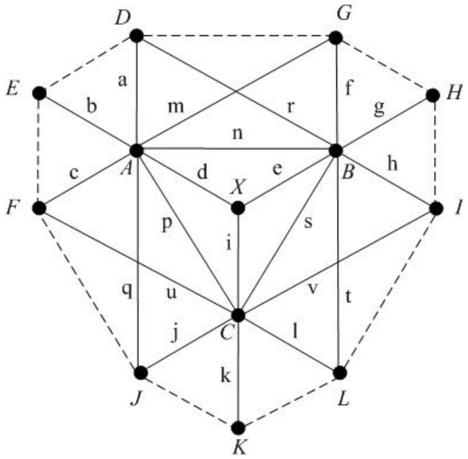

Figure 1. A weighted subgraph G_X^{2-hop} partitioned based on 2-hop information of node X (not include the links drawn in dash lines)

$\chi_w(G_j^{1-hop}); j \in V_i^{1-hop}, j \neq i$ and is able to compute $\chi_w(G_i^{2-hop})$ by using $\sum_{j \in V_i^{1-hop}} \chi_w(G_j^{1-hop}); j \neq i$ subtracted by $\sum_{(u,v)\in E_i^{1-hop}} \psi(u,v); u,v \neq i$ as shown in Proposition 1.

The weighted CI can be classified as one of non-additive QoS parameters since it is the combination of link capacity and link connectivity. Hence, cost of the path can be determined by the value of weighted CI at the bottleneck link or node. Thus, the state information of 2-hop weighted CI of a path p from source node s to destination node d illustrated as $R_w(p_{sd})$ can be defined as

$$R_w(p_{sd}) = min\left\{\chi_w(G_s^{2-hop}), \ldots, \ \chi_w(G_d^{2-hop})\right\} \quad (6)$$

C. Non-linear Cost Function

In this work, the non-linear cost function proposed in [30] is used to non-linearly combine multiple additive QoS metrics e.g., latency, loss rate and etc. For the sake of understanding, we review brieœy here.

Each link $(i,j) \in E$ in graph G is associated with a primary cost parameter $c(i,j)$; K additive QoS parameters $w_k(i,j)$, $k = 1,2,\ldots,K$; K constraints c_k, $k = 1,2,\ldots,K$; all parameters are non-negative. The Multi-Constrained Path (MCP) problem is to find path p from source node s to destination node d that satisfies the following requirement:

- $w_k(p_{sd}) = \sum_{(i,j)\in p_{sd}} w_k(i,j) \leq c_k$ for all k

In MCP problem, primary cost of path $(\sum_{(i,j)\in p_{sd}} c(i,j))$ is not necessary to be minimized as in Multi-Constrained Optimal Path (MCOP) [31]. Hence, the non-linear cost function [30], [31] proposed to solve both MCP and MCOP problems is defined as

$$g_\lambda(p_{sd}) = \sum_{k=1}^{K}\left(\frac{w_k(p_{sd})}{c_k}\right)^\lambda, \lambda \geq 1 \quad (7)$$

Suppose that an algorithm find a path p whose cost function expressed in Eq. (7) is minimized for a given $\lambda \geq 1$, then, the important theorem regarding to the bound on the performance of this heuristic is established as follows [31]:

Theorem 1: Consider the MCP problem. Assume that there is at least one feasible path $p*$ in the network. Let p be a path that minimizes the cost function g_λ for a given $\lambda \geq 1$. Then,

- $w_k(p_{sd}) \leq c_k$ for at least one k, and
- $w_k(p_{sd}) \leq \sqrt[\lambda]{K}c_k$ for all other k

Corollary 1: As λ increases, the likelihood of finding a feasible path also increases.

Proof for Theorem 1 and Corollary 1 can be found in [31]. In Eq. (7), when $\lambda \to \infty$, the largest term of $(\frac{w_k(p_{sd})}{c_k})$ will dominate the other terms. In this case, Eq. (7) can be replaced by

$$g_\infty(p_{sd}) = max\left\{\frac{w_1(p_{sd})}{c_1}, \frac{w_2(p_{sd})}{c_2}, \ldots, \frac{w_K(p_{sd})}{c_K}\right\} \quad (8)$$

IV. Proposed Generalized MCP QoS Routing

As mentioned previously, considering only one or two QoS metrics in path selection process is not enough to support all types of applications, since each application has different QoS requirements (see Table I). However, by including various QoS metrics in path selection algorithm, the feasible path may not exist with all parameters at their optimal values. The problem of finding a feasible path is said to be NP-complete when two or more parameters are used in path computation process [32], [33]. Thus, the routing problem is solvable in polynomial time if we consider only one parameter or define the precedence among multiple QoS metrics (only one metric is accounted at a time) in path computation process.

The proposed Generalized MCP (G_MCP) QoS routing finds the path from source to destination firstly based on 2-hop weighted CI. If there is a tie (finds two or more paths with the same value of weighted CI), the path with least non-linear cost will be chosen. This G_MCP is also called *Least Cost-Highest CI Pathf* algorithm. In addition, this algorithm is solvable in polynomial time by adopting the fall-back approach of generally combined multiple metrics which are weighted CI and non-linear cost.

Even though G_MCP considers various QoS routing parameters in path selection process, the computational complexity of this algorithm is comparable to Shortest-Widest Path algorithms [18]˘[20]. This is because the ranking of multiple QoS metrics is defined. In this work, the high precedence is given to weighted CI. Since if the requirement on channel bandwidth or link capacity cannot be met, or link connectivity is lost due to absent nodes in the path, it will affect the other additive QoS parameters which are combined into non-linear cost.

Fig. 2 illustrates the pseudo-code of routing procedure in G_MCP. For a graph $G = (V,E)$, where V is the

G_MCP$(G = (V,E);\ N \subset V)$
Step 1) Initially, $N := \{s\}$
  For all $i \neq s$
    $\zeta_i := R_w(p_{si})$ and $\gamma_i := g_\lambda(p_{si})$
Step 2) $M := \{\}$
  Find $m \notin N$ so that $\zeta_m = max_{i\notin N}\ \zeta_i$
    $M := M \cup \{m\}$
Step 3) **If** there is more than one element in M **then**
    Find $m \in M$ so that $\gamma_m = min_{i\notin N}\ \gamma_i$
  $N := N \cup \{m\}$
  If N contains all nodes in V **then**
    Algorithm is completed.
Step 4) **For** all $i \notin N$,
    Tmp $:= \zeta_i$
    $\zeta_i := max\{\zeta_i, min(\zeta_m, R_w(p_{mi}))\}$
    If $\zeta_i \neq$ Tmp **then**
      $\gamma_i := \sum_{k=1}^{K}\left(\frac{w_k(p_{sm}+p_{mi})}{c_k}\right)^\lambda, \lambda \geq 1$
Step 5) Go to step 2)

Figure 2. The Generalized MCP (G_MCP) algorithm

set of nodes and E is the set of links, M is temporary set and N is the subset of set V. Assume node s is the source or computing node. Let $\zeta_i = R_w(p_{si})$ and $\gamma_i = g_\lambda(p_{si})$ denote the weighted Connectivity Index and non-linear cost of the paths from node s to any node i, respectively ($\zeta_i = 0$ and $\gamma_i = \infty$, if no link exists from node s to node i). By convention, $\zeta_s = \infty$ and $\gamma_s = 0$. In step 1) each node calculates the state information: 2-hop weighted CI and non-linear cost of the known paths from node s to node i. In Step 2), algorithm finds node(s) which provides the highest value of 2-hop weighted CI to tentatively selected node i. If there are more than one node with identical maximum value of 2-hop weighted CI, step 3) will select the node m which provides the least non-linear cost link to node i. Step 4) updates the new state information of the tentatively selected nodes around the newly selected node m. The algorithm iterates to step 2) until all nodes in set V are added as the destination nodes.

In this work, we implement G_MCP QoS routing on OLSR. However, only implementing QoS routing on OLSR may not be effective enough since some better feasible paths in terms of least cost-highest CI paths may be hidden by using the native MPR [7] due to the fact that the native MPR aims at optimizing control overheads by minimizing number of MPR nodes. Since only MPR nodes are allowed to forward topology information, only partial graph of network are known to each node. Thus, we also implement our proposed MPR computation process [34] to optimize the feasible paths found in each node

Heuristic MPR($G = (V, E)$; $N1, N2, MPR \subset V$)
Step 1) Initially, $MPR := \{\}$
Step 2) **For** all nodes $x \in N1$, **Compute** $d(x)$
Step 3) **Find** $n \in N1$ which provide the only path to reach some nodes in $N2$
$MPR := MPR \cup \{n\}$
Step 4) **While** there exist nodes in $N2$ which are not covered by at least one node in the MPR
Step 4a) **For** all $x \in N1, x \notin MPR$
Compute numbers of nodes in $N2$ which are connected to node x and not yet covered by at least one node in MPR
Step 4b) $T := \{\}$
Find $n \in N1$ that provides the hightest weighted CI and least non-linear cost link
$T := T \cup \{n\}$
Step 4c) **If** there is more than one element in T **then**
Find $n \in T$ that has the maximum number of nodes in $N2$ which are not yet covered
$MPR := MPR \cup \{n\}$

Figure 3. Heuristic for MPR selection process [34]

by allowing it to select its MPR nodes based on weighted CI and non-linear cost as illustrated by the pseudo-code in Fig. 3.

In Fig. 3, let $N1$, $N2$, MPR, T, $d(x)$ denote the sets of 1-hop neighbors, 2-hop neighbors, multipoint relay (MPR) nodes, temporary nodes and degree of node x in a graph of network G, respectively. Nodes in $N1$ and $N2$ are already assigned and updated whenever a MPR selector (MPR computing node) receives Hello messages. In steps 1) and 2), each MPR selector resets its own MPR set to an empty set and computes the degree of its 1-hop neighbors. It will select the node(s) from 1-hop neighbors set to be MPR nodes if it is the node(s) which provides the only path to reach some 2-hop neighbors as shown in step 3). However, in step 4), if there are some 2-hop neighbors which are not yet discovered, it will find node(s) which provides the highest weighted CI link(s) from itself to its 1-hop neighbors. And if there is a tie (two or more nodes with the same value of weighted CI), node(s) providing the least non-linear cost link(s) to its 1-hop neighbors will be assigned to a set T. However, if there is another tie, a node in temporary set T providing the maximum number of 2-hop neighbors which are not yet discovered will be selected as MPR node as shown in step 4c).

V. Performance Evaluation and Simulation Results

In this section, we investigate the proposed G_MCP algorithm by constructing simulation using NS-2 simulator [35]. We measure the efficiency and effectiveness of the proposed G_MCP algorithm in wireless mobile ad hoc networks and compare it with traditional *OLSR* [3] and *Shortest-Widest Path* algorithm with the optimal path selection [20]. To indicate the reliability of the results obtained in the proposed G_MCP algorithm, the performances measurement with 95 % confidence interval are computed.

In this simulation, λ in non-linear cost function in our proposed G_MCP, expressed in Eq. (7) is set to its largest possible value ($\lambda = \infty$) to achieve the highest probability to find the feasible paths [23], [31]. G_MCP can support any number of additive QoS constraints by combining them into a non-linear cost. However, only delay and packet loss rate are considered as additive QoS constraints in all simulations here. Since both parameters can indicate timeliness and precision which used to measure the output performances of a routing process.

According to the subjective tests, the International Telecommunication Union (ITU) G.114 specifications recommend that more than 400 ms one-way end-to-end delay for real-time traffic is unacceptable [36]. Thus, the delay constraint used in these simulation scenarios is set to 400 ms. For the packet loss rate constraint, we set to 10 % as used in [23], since ad hoc networks are highly mobility networks with limited resources.

The performance evaluation metrics are described below:

- Throughput: the amount of data that are delivered in the network over the time
- Packet delivery ratio (PDR): the ratio of the total number of packets received by destinations to the total number of packets sent by sources
- Average end-to-end delay: the average amount of time it takes all packets to reach the destinations
- Success Ratio (SR): the ratio of the total number of connection requests whose feasible paths are found by routing algorithm to the total number of connections requested by sources

To demonstrate that the proposed G_MCP algorithm works well and is open enough to support various applications, the simulations are constructed in two main scenarios: CBR and MPEG-4 (which is encoded in VBR) services.

A. Constant Bit Rate (CBR) Services

In this scenario, we measure the performances of G_MCP using CBR traffic. The parameters used in this simulation scenario are listed in Table III. Two conditions are setup: load-varying and speed-varying, in order to demonstrate how well the G_MCP algorithm can operate and handle the CBR services when the offered load and movement speed are changed.

TABLE III.
PARAMETERS USED IN CBR AND MPEG-4 SERVICES SCENARIO

Parameters	CBR		MPEG-4	
	Load-Varying	Speed-Varying	Load-Varying	Speed-Varying
Area	1000 m by 1000 m			
MAC Protocol	IEEE 802.11		IEEE 802.11b	
Channel Capacity	2 Mbps		11 Mbps	
No. of Nodes	50 nodes			
Speed	2 m/s	1 to 30 m/s	2 m/s	1 to 30 m/s
Pause Time	0 s			
No. of S-D Pairs	10 connections		5 connections	
Offered Load	25 to 150 kbps per flow	100 kbps per flow	Rate Factor* 1 to 6	Rate Factor* 4
Delay Constraint	400 ms			
Loss Rate Constraint	10 %			
Simulation Time	300 seconds			
No. of Simulations per Scenario	10 times			

Note: *Rate Factor is a parameter used in MPEG-4 video traffic generator to scale up or scale down video input.

1) CBR Services in Load-Varying Condition: G_MCP is designed to let each node in network to be able to balance the load, since it considers the link capacity in path computation process. However, if it finds two or more path with the same weighted CI, non-linear cost will be considered. Thus, it is expected to perform well in either light or heavy load situation. Fig. 4 shows the simulation results of all performance metrics versus offered load.

Fig. 4(a) illustrates that throughputs of all routing algorithms are approximately the same in light load situation (around 25 kbps). However, when the offered load increases, OLSR has the worst performance comparing to the other QoS routing algorithms (G_MCP and Shortest-Widest Path), since it always selects the paths with shortest hop count to reach destinations without considering any QoS parameters. Whereas the Shortest-Widest Path and G_MCP consider bandwidth in their path finding processes. By comparing G_MCP with Shortest-Widest Path, even though both algorithms consider bandwidth in their path finding processes, however, G_MCP put into account Connectivity Index (in addition to bandwidth) which intuitively illustrates the probability of link break (the higher the CI is, the lower probability of link break [28]) in path finding process as well.

In Fig. 4(b), PDRs of all protocols continuously decrease when more CBR packets are generated into the network, since the congestion occurs and some packets are discarded when there is no more space for buffering the incoming packets. By comparing all protocols, it is obvious that both G_MCP and Shortest-Widest Path algorithms outperform OLSR. The reason is similar as that of Fig. 4(a), that is, both algorithms consider bandwidth in path finding process. Therefore, the bandwidth is guaranteed which results in higher packet deliver ratio comparing to OLSR. G_MCP has slightly better PDR than Shortest-Widest Path (3.11 % when offered load is at 75 kbps per connection) since, similarly, packet loss rate is also considered in G_MCP.

Fig. 4(c) highlights the improvement of G_MCP in terms of end-to-end delay over OLSR and Shortest-Widest Path. The delays of all protocols are not much different when offered load is low. However, the delay increases when offered load is high which causes the congestion and long end-to-end delay. Both G_MCP and Shortest-Widest Path algorithms can improve the end-to-end delay by considering delay as one of QoS routing metric. However, G_MCP has slightly lower delay than Shortest-Widest Path (up to 8.56 % at heavy load situation) since packet

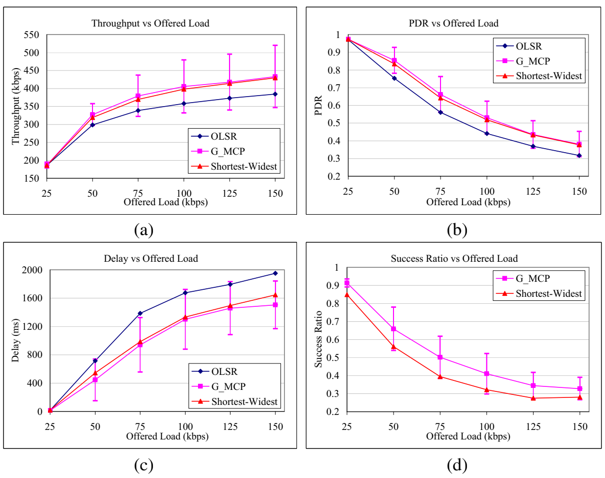

Figure 4. Effect of offered load on CBR traffic measured by (a) throughput, (b) packet delivery ratio, (c) end-to-end delay and (d) success ratio

loss rate, another additive QoS metric, is also considered in path finding process (via non-linear cost function). Thus, G_MCP has the ability to avoid the congesting path and experiences the lowest end-to-end delay.

One of the performance metrics which is especially used for measuring performance of QoS routings is the success ratio of finding the feasible path. Therefore, Fig. 4(d) depicts the success ratios of only QoS routing algorithms. It is obvious that success ratio decreases when offered load increases. Since some feasible paths may not be found by sources due to the fact that less channel bandwidth is available, more packets are discarded and time to take packets to reach destinations is longer.

Between both QoS routing algorithms, G_MCP has better success ratio (more feasible paths are found) than Shortest-Widest Path algorithm (up to 27.33 %), due to the fact that both algorithm consider the same non-additive QoS parameter (bandwidth) as primary QoS metric. However, G_MCP considers multi-constrained QoS parameters simultaneously as the secondary additive QoS metric whereas Shortest-Widest Path algorithm considers only delay in their path finding process. Therefore, the paths satisfying only delay constraint found by Shortest-Widest Path algorithm is not necessary to satisfy multi-constraints QoS in G_MCP which results in lower success ratio in Shortest-Widest Path algorithm.

2) CBR Services in Speed-Varying Condition: G_MCP is expected to improve the performances of ad hoc networks to handle services in highly dynamic topology network and is robust to link failures. Therefore, the simulation results of all performance metrics versus speed are carried out and illustrated in Fig. 5.

In Fig. 5(a) and (b), it is obvious that the throughputs and PDRs of all algorithms decrease when the speed of nodes increases. Considering the fact that OLSR is the proactive protocol which has to send the Hello and TC messages every 2 and 5 seconds (default values) in order to update the network link states. Therefore, when the speed increases, nodes may either traverse a long distance which causes the topology change or not be able to update the link state due to the link break, thus, nodes may use the out-of-date link states to forward the packets. This, of course, results in higher packet loss and decreasing throughput. In case of G_MCP and Shortest-Widest Path algorithms, they are also proactive protocols which have to send Hello and TC messages to update the link states (to calculate weighted CI in case of G_MCP), therefore, they follow the same characteristics as typical OLSR.

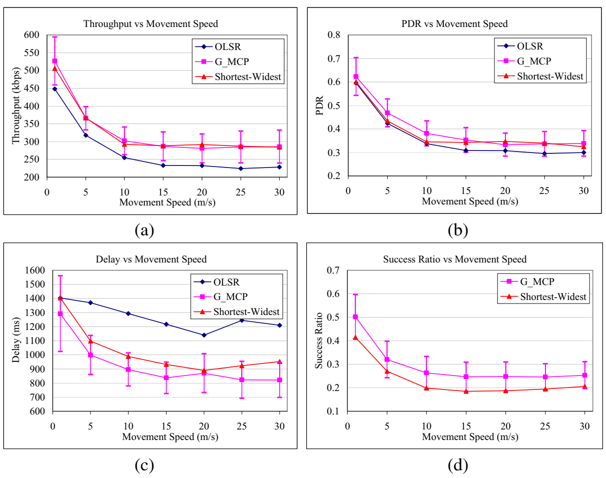

Figure 5. Effect of mobility on CBR traffic measured by (a) throughput, (b) packet delivery ratio, (c) end-to-end delay and (d) success ratio

When we compare G_MCP and Shortest-Widest Path algorithm with OLSR, it is obvious that both algorithms outperform OLSR (up to 27.94 % in terms of throughput), since both of them consider the bandwidth in path calculation. Therefore, even though the out-of-date link states are used to forward the packets, as long as the links do not break, the forwarded packets still receive a certain level of bandwidth guarantee. When the speed is not more than 15 m/s, G_MCP achieves the highest PDR (up to 10.43 % comparing to Shortest-Widest Path and 14.61 % comparing to OLSR). At high speed, throughput and PDR of G_MCP is approximately the same as Shortest-Widest Path algorithm, since in G_MCP, the calculation of weighted CI is needed. However, when speed increases, the node almost cannot exchange the Hello and TC messages, therefore the weighted CI calculation could not be done properly, so there is almost no effectiveness of weighted CI.

Fig. 5(c) illustrates the great improvement of G_MCP over OLSR (up to 33.92 %) and Shortest-Widest Path (up to 13.86 %) in terms of end-to-end delay. Since the path finding in OLSR is done without putting into account any QoS parameters whereas both G_MCP and Shortest-Widest Path always select the path satisfying delay requirement (delay requirement is considered in both algorithms). Comparing G_MCP to Shortest-Widest Path algorithm, the end-to-end delay of G_MCP is lower than that of Shortest-Widest Path in all speeds. This is because of the effect of weighted CI (the links are more stable) and non-linear cost function (where delay requirement is included). The weighted CI, by its definition as shown in Eq. (2), implies how strong the link connections of the node are. Therefore, the path with the highest weighted CI obtains the highest stability, thus, results in lower end-to-end delay.

Success ratios of both QoS routing algorithms are measured and illustrated in Fig. 5(d). It is obvious that G_MCP improves success ratio over Shortest-Widest Path (up to 33.51 %). Because G_MCP considers both non-additive and multiple additive QoS metrics using weighted CI and non-linear cost, respectively. Thus, it has higher probability to select the path satisfying multiple requirements defined by applications than Shortest-Widest Path algorithm which considers only bandwidth and delay constraints. Without considering another QoS constraint i.e. loss rate in Shortest-Widest Path algorithm, the lower success ratio will be obtained since the selected path may not satisfy this omitted QoS constraint.

B. MPEG-4 Services

In this scenario, MPEG-4 traffics are generated based on the Transform Expand Sample (TES) methodology [37], [38] which is an approach for modeling any set of given observations in a time series. We use MPEG-4 traffic generator [39] to generate the sequences of I (Intra-coded), P (Predictive coded) and B (Bidirectional coded) frames every 1/30 second and import them to NS-2 [35]. We also set two scenarios which are load-varying and speed-varying conditions to verify the performances of each routing protocol. The parameters used in this scenario are shown in Table III.

In MPEG-4 video traffic generator, there is a parameter called Rate Factor which is used to control the MPEG-4 bitstreams. Rate Factor is a parameter defined to scale up or scale down video input while preserving the same sample path and autocorrelation function for the frame size distribution. In load-varying condition, Rate Factor is varied to control MPEG-4 traffic, whereas in speed-varying condition, Rate Factor is kept constant for fair comparison.

1) MPEG-4 Services in Load-Varying Condition: Fig. 6 illustrates throughput, PDR, end-to-end delay and success ratio of all considered routing algorithms (OLSR, Shortest-Widest Path, G_MCP) when they deliver MPEG-4 bitstreams in load-varying condition.

As demonstrated in Fig. 6(a), throughputs of all protocols increase when Rate Factor increases (more MPEG-4 traffics are offered to the network). It shows that throughput of OLSR is the lowest comparing to the other algorithms, since OLSR itself doesn t provide any QoS mechanisms. Whereas G_MCP shows significant improvement over OLSR in terms of throughput (up to 24.48 %) and it has also slightly better throughput than Shortest-Widest path (up to 3.6 %). Both G_MCP and Shortest-Widest Path algorithms consider bandwidth in their path finding processes. However, G_MCP also considers link connectivity in form of weighted CI before selecting a path. Moreover, this weighted CI also indicates the reliability of a selected path since it indicates connectivity level of each intermediate node along this path. This results in lower loss rate which directly affects the throughput.

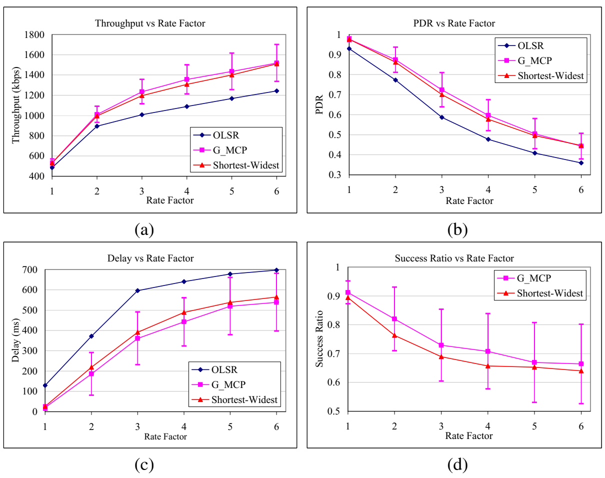

Figure 6. Effect of offered load on MPEG-4 traffic measured by (a) throughput, (b) packet delivery ratio, (c) end-to-end delay and (d) success ratio

Fig. 6(b) shows that all protocols can reliably transmit MPEG-4 frames in light load situation (Rate Factor = 1). However, when offered loads increase, G_MCP can transmit MPEG-4 frames with the highest PDR (up to 25.16 % and 3.47 % comparing to OLSR and Shortest-Widest Path algorithm, respectively) due to the same reason explained in Fig. 6(a). That is, path returned by G_MCP is a path with the highest reliability. In addition, loss rate, one of additive QoS metrics included in non-linear cost, is also considered as the second QoS metric in G_MCP while this metric is omitted in Shortest-Widest Path algorithm.

As mentioned previously in ITU G.114 specification,it is recommended that one-way end-to-end delay for real-time traffic should be less than 400 ms [36]. In Fig. 6(c) which depicts delays of all algorithms, end-to-end delay of OLSR is acceptable when Rate Factor is not more than 2, while delays of both G_MCP and Shortest-Widest Path algorithms are acceptable when they transmit MPEG-4 bitstreams with Rate Factor that is not more than 3, since they consider delay constraint when selecting a path. Between G_MCP and Shortest-Widest Path, G_MCP has the lower end-to-end delay because it has ability to avoid the congested path by considering loss rate due to congestion.

It is obvious that G_MCP outperforms Shortest-Widest Path algorithm in either light or heavy load situation when measuring success ratio (up to 7.76 %) as illustrated in Fig. 6(d). Because multiple constraints i.e. weighted CI (bandwidth and link connectivity), non-linear cost (delay and loss rate) are considered as mentioned previously in Section IV, whereas only bandwidth and delay are considered in Shortest-Widest Path algorithm. Therefore, the ratio that all predefined QoS requirements are satisfied when using G_MCP algorithm is higher.

2) MPEG-4 Services in Speed-Varying Condition: Performances of all protocols when they deliver MPEG-4 bitstreams in speed-varying condition are shown in Fig. 7.

In Fig. 7(a), throughputs decrease when node speed increases from 1 m/s to 10 m/s and remains roughly constant when speed is beyond 10 m/s. In this figure, both G_MCP and Shortest-Widest Path algorithms achieve lots of improvement over OLSR (up to 26.16 %) since both of them consider bandwidth in their path finding process. PDRs of all protocols follow the same trend as throughput, as illustrated in Fig. 7(b). Both G_MCP and Shortest-Widest Path algorithms have better performances in terms of PDR than OLSR (up to 27.10 %). Between them, G_MCP can transmit MPEG-4 bitstreams with slightly better PDR than Shortest-Widest Path (up to 3.01 %) when speed is less than 30 m/s, since link connectivity and packet loss rate are also considered in G_MCP when selecting a path. Thus, a path returned by G_MCP is a little bit more stable than another one which results in better throughput and PDR. However, at high speed, performances of G_MCP are not much improved because CI calculation in each node is not accurate due to highly dynamic topology change, as explained in Section V-A.2.

End-to-end delay of G_MCP is the lowest (highest performance) among all routing protocols, as shown in Fig. 7(c). Its delay is always less than 400 ms, which is an acceptable level, as recommended in the ITU G.114 specification regardless of the node speed. Because a path returned by G_MCP is the best feasible path in terms of highest weighted CI and least non-linear cost. Thus, this path is the highest stability path as mentioned in Section V-A.2 which results in lowest end-to-end delay. Whereas only bandwidth and delay are considered in Shortest-Widest Path, loss rate which is another important QoS metric is left behind. Moreover, paths returned by this algorithm may not be stable. Thus, delay of Shortest-Widest Path oscillates a little bit around 400 ms and is longer than that of G_MCP. OLSR itself doesn t provide QoS routing which results in the lowest performance in terms of end-to-end delay.

From Fig. 7(d), it is obvious that G_MCP has better success ratio than Shortest-Widest Path (up to 15.45 % at speed = 10 m/s) regardless of node speed. It means that G_MCP has higher probability to find the feasible paths when it delivers MPEG-4 bitstreams. It can achieve higher success ratio because multiple constraints are put into account in G_MCP when selecting the feasible paths. Thus, the returned path in G_MCP has higher probability to satisfy bandwidth, delay and loss rate constraints defined by application in our simulation setting. Whereas Shortest-Widest Path does not consider an important QoS metric which is loss rate, therefore, the returned path found by Shortest-Widest Path algorithm is not necessary to satisfy loss rate constraint. This results in lower number of feasible returned paths.

VI. Efficiency of G_MCP

Sections V demonstrates the performance and effectiveness of G_MCP in terms of throughput, packet delivery ratio, end-to-end delay and success ratio. However, to study the efficiency of G MCP QoS routing, the comparison of time complexity and communication complexity (namely, control overhead) of all algorithms is needed. Therefore, this simulation scenario is set up to illustrate the time and communication complexities of each routing algorithm. We use the same simulation parameters as listed in Table III in CBR traffic for load-varying condition except that the number of nodes are varied and simulation time is reduced to 60 seconds. Table IV details the platform of the machine used to perform this simulation.

The time complexities of all algorithms can be compared by simply measuring execution time (of ns-2 for each scenario) of each protocol in the same simulation environment and parameters to select paths to any reachable nodes in the network. Fig. 8(a) depicts that the execution time for running the simulation of each routing protocol increases when there are more number of nodes in the network, since the complexities of these routing algorithms are directly proportional to number of nodes (up to about 25 minutes for 100 nodes). The execution time here is the time taken by ns-2 simulator for each scenario in exactly the same environment and parameters. Therefore, the amount of time difference occurs due to only routing algorithm itself. By comparing the execution time of these routing algorithms, their complexities can be measured indirectly and compared relatively.

It is obvious that the execution times of all routing algorithms are not much different when number of nodes in the networks is not more than 50 nodes. However, when number of nodes increases. The execution times of both G_MCP and Shortest-Widest Path algorithms are approximately the same but a bit larger than OLSR, since they require more computation of multiple QoS con-

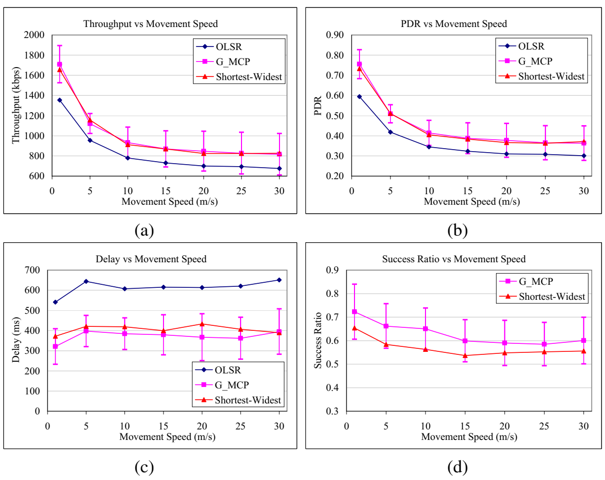

Figure 7. Effect of mobility on MPEG-4 traffic measured by (a) throughput, (b) packet delivery ratio, (c) end-to-end delay and (d) success ratio

TABLE IV.
Specifications of the Platform Used in This Simulation

Processor	Intel Quad Core Q9650 processor 3.0 GHz
Memory	2.0 GB
Operating System	Fedora Core 7
Simulator	NS version 2.29

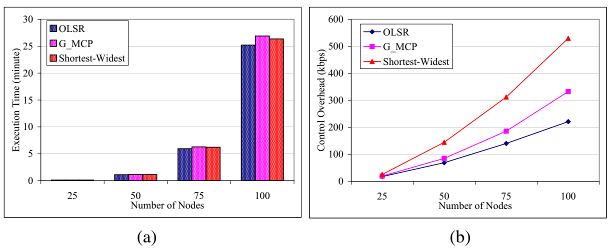

Figure 8. Efficiency of each algorithm on (a) execution time to find paths to any reachable nodes and (b) control overhead ooded to the networks

straints (weighted CI and non-linear cost in G_MCP, and bandwidth and delay in Shortest-Widest Path algorithm) by defining the precedence among them.

Fig. 8(b) shows the communication complexity which is illustrated by the total control overheads of each routing algorithm. In this figure, it is obvious that control overheads of all algorithms tend to increase when there are more nodes in the networks.

In Shortest-Widest Path algorithm, it modifies MPR computation process to select MPR nodes based on bandwidth and delay to improve the performances in terms of throughput, error rate and packet loss rate [20]. However, this algorithm does not consider topological information which results in the largest number of control overhead flooded to the network due to the increasing number of MPR nodes.

Control overhead of G_MCP is moderate among all algorithms. It is more than OLSR about 23.10 % but less than Shortest-Widest Path algorithm about 41.24 % at 50 nodes. In G_MCP, MPR computation process is also modified to select the optimal path based on weighted CI and non-linear cost. Therefore, each node selects its MPR nodes based on topological information or link connectivity to the others. This leads to more number of MPR nodes than OLSR which results in more control overheads. It should be noted that the increasing control overhead in G_MCP is due to only the increasing number of MPR nodes. There is no additional control packet sent to compute weighted CI since this is carried out by piggybacking onto Hello messages.

VII. Conclusions

In this paper, G_MCP is proposed to incorporate QoS routing into mobile ad hoc network by considering both additive and non-additive QoS parameters in path computation process. The weighted Connectivity Index (combined parameter of link connectivity and capacity) and non-linear cost (combined multiple additive QoS metrics) are proposed as non-additive and additive QoS metrics to be used in this type of network, respectively.

In simulations, we compared performances of the proposed G_MCP with some routing algorithms namely, OLSR and Shortest-Widest Path algorithms using both CBR and MPEG-4 (encoded into VBR) traffics. We can conclude that G_MCP outperforms OLSR in terms of throughput, PDR and end-to-end delay regardless of the offered load and node speed. It also gains lots of advantages over Shortest-Widest Path QoS routing algorithm in terms of success ratio of finding the feasible paths.

However, G_MCP is a little bit more complex than OLSR but it has approximately the same level of complexity as Shortest-Widest Path algorithm since they requires the additional computation in selecting the paths based on multiple QoS metrics. Control overhead of G_MCP is larger than OLSR but less than Shortest-Widest Path algorithm. Thus, Shortest-Widest Path algorithm consumes more bandwidth than the others in order to exchange the control overhead when there are more number of nodes in the networks.

In G_MCP, multiple QoS parameters are effectively combined and considered in path finding process. It can find the feasible paths satisfying multiple QoS constraints which are differently required by various applications. Since link connectivity is one of QoS parameters that is taken into account in selecting paths, therefore, ad hoc networks implementing G_MCP are more robust to link failures and are capable to operate in highly mobility network.

References

[1] C.E. Perkins and P. Bhagwat, Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, Proceedings of ACM SIGCOMM, pp. 234˘244, August 1994.

[2] R. Ogier, F. Templin and M. Lewis, Topology Dissemination Based on Reverse-Path Forwarding (TBRPF), RFC 3684, February 2004.

[3] T. Clausen and P. Jacquet, Optimized Link State Routing Protocol (OLSR), RFC 3626, October 2003.

[4] C. E. Perkins, E. Belding-Royer and S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, July 2003.

[5] D. Johnson, Y. Hu and D. Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728, February 2007.

[6] C. Mbarushimana and A. Shahrabi, Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshop (AINAW), pp. 679˘684, May 2007.

[7] A. Qayyum, L. Viennot and A. Laouiti, Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks, INRIA Research Report RR-3898, 2000.

[8] A. S. Tanenbaum, Computer Networks, Fourth edition, New Jersey, Prentice Hall, 2003.

[9] S. Chen and K. Nahrstedt, Distributed Quality-of-Service Routing in Ad hoc Networks, IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1488˘1505, August 1999.

[10] H. Xiao, W. K.G. Seah, A. Lo and K. C. Chua, A Flexible Quality of Service Model for Mobile Ad hoc Networks, Proceedings of IEEE Vehicular Technology Conference, Vol. 1, pp. 445˘449, September 2000.

[11] Y. Hwang and P. Varshney, An Adaptive QoS Routing Protocol with Dispersity for Ad Hoc Networks, Proceedings of 36th Hawaii International Conference on System Sciences (HICSS), January 2003.

[12] Y. Hwang and P. Varshney, An Adaptive Routing Protocol for Ad-hoc Networks using Multiple Disjoint Path, Proceedings of IEEE Vehicular Technology Conference, vol. 3, pp. 2249˘2253, May 2001.

[13] Q. Xue, and A. Ganz, Ad Hoc On-demand Routing (AQOR) in Mobile Ad Hoc Networks, Journal of Parallel and Distributed Computing, vol. 63, no. 2, pp. 154˘165, 2003.

[14] S. Medidi and K. Vik, Quality of Service-aware Source-Initiated Ad Hoc Routing, Proceeding of IEEE Sensor and Ad Hoc Communications and Networks (SECON), pp. 108˘117, October 2004.

[15] L. Chen and W. Heinzelman, QoS-aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks, IEEE Journal on Selected Areas in Communications, vol. 23, no. 3, pp. 561˘572, March 2005.
[16] P. Sinha, R. Sivakumar and V. Bharghavan, CEDAR: A Core-Extraction Distributed Ad hoc Routing Algorithm, Proceedings of IEEE INFOCOM, March 1999.
[17] Y. Ge, T. Kunz and L. Lamont, Quality of Service Routing in Ad Hoc Networks Using OLSR, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS 03), January 2003.
[18] H. Badis, A. Mauaretto, K. A. Agha and G. Pujolle, QoS for Ad Hoc Networking Based on Multiple Metrics: Bandwidth and Delay, Proceedings of the 5th IEEE International Conference on Mobile and Wireless Communications Networks, pp. 15˘18, October 2003.
[19] H. Badis and K. A. Agha, QOLSR Multi-path Routing for Mobile Ad Hoc Networks Based on Multiple Metrics: Bandwidth and Delay, Proceedings of IEEE Vehicular Technology Conference, vol. 4, pp. 2181˘2184, May 2004.
[20] H. Badis, A. Mauaretto, K. A. Agha and G. Pujolle, Optimal Path Selection in a Link State QoS Routing Protocol, Proceedings of IEEE Vehicular Technology Conference, vol. 4, pp. 2570˘2574, May 2004.
[21] D. Nguyen and P. Minet, QoS Support and OLSR Routing in a Mobile Ad hoc Network, Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, April 2006.
[22] K. Kunavut and T. Sanguankotchakorn, QoS-aware Routing for Mobile Ad hoc Networks Based on Multiple Metrics: Connectivity Index (CI) and Delay, Proceedings of IEEE Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2010), pp. 46˘50, May 2010.
[23] K. Kunavut and T. Sanguankotchakorn, Multi-Constrained Path (MCP) QoS Routing in OLSR based on Multiple Additive QoS Metrics, Proceedings of IEEE Symposium on Communications and Information Technologies 2010 (ISCIT 2010), pp. 226˘231, October 2010.
[24] M. Randić, On Characterization of Molecular Branching, Journal of The American Chemical Society, vol. 97, no. 23, pp. 6609˘6615, November 1975.
[25] I. Gutman, O. Araujo and D.A. Morales, Estimating the connectivity index of a saturated hydrocarbon, Indian Journal of Chemistry, vol. 39, pp. 381˘385, 2000.
[26] I. Gutman, O. Miljković, G. Caporossi and P. Hansen, Alkanes with small and large Randić connectivity indices, Chemical Physics Letters, vol. 306, no. 5, pp. 366˘372, June 1999.
[27] G. Caporossi, I. Gutman, P. Hansen and L. Pavlović, Graphs with maximum connectivity index, Computational Biology and Chemistry, vol. 27, pp. 85˘90, 2003.
[28] M. A. Rajan, M. G. Chandra, L. C. Reddy and P. Hiremath, A Study of Connectivity Index of Graph Relevant to Ad Hoc Networks, International Journal of Computer Science and Network Security, vol. 7, no. 11, pp. 198˘203, November 2007.
[29] L. Shi, A. Fapojuwo, N. Viberg, W. Hoople and N. Chan, Methods for Calculating Bandwidth, Delay, and Packet Loss Metrics in Multi-hop IEEE 802.11 Ad Hoc Networks, Proceedings of IEEE Vehicular Technology Conference, pp. 103˘107, May 2008.
[30] H. D. Neve and P. V. Mieghem, A Multiple Quality of Service Routing Algorithm for PNNI, Proceedings of IEEE ATM Workshop, pp. 324˘328, May 1998.
[31] T. Korkmaz and M. Krunz, Multi-constrained Optimal Path Selection, Proceedings of the IEEE INFOCOM, vol. 2, pp. 834˘843, April 2001.
[32] J. M. Jaffe, Algorithms for finding paths with multiple constraints, Networks, vol. 14, pp. 95˘116, 1984.
[33] Z. Wang and J. Crowcroft, Quality of Service Routing for Supporting Multimedia Applications, IEEE Journal on Selected Areas in Communications, vol. 14, no. 7, pp. 1228˘1234, September 1996.
[34] K. Kunavut and T. Sanguankotchakorn, Optimized Path Selection Process in OLSR Based on Weighted Connectivity Index and Delay, Proceedings of IEEE Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2011), pp. 348˘351, May 2011.
[35] Network Simulator 2, http://www.isi.edu/nsnam/ns
[36] Chen, T. Farley and N. Ye, QoS Requirements of Network Applications on the Internet, International Journal on Information-Knowledge-Systems Management, vol. 4, pp. 55-76, January 2004.
[37] B. Melamed, TES: A Class of Methods for Generating Autocorrelated Uniform Variates, ORSA Journal on Computing, vol.3, pp. 317˘329, 1991.
[38] B. Melamed, D. Raychaudhuri, B. Sengupta and J. Zdepski, TES-Based Modeling for Performance Evaluation of Integrated Networks, Proceedings of IEEE INFOCOM, 1992.
[39] Video Traffic Generator, http://www.sce.carleton.ca/∼amatrawy/mpeg4

Kunagorn Kunavut was born in 1979. He received the B.S. and M.S. in Telecommunications Science from Assumption University, Thailand, in 2000 and 2004, respectively. He is currently pursuing the doctoral degree in Telecommunications Field of Study at Asian Institute of Technology, Thailand under Royal Thai Government (RTG) Fellowship. He has been served as a faculty member at Assumption University since 2001. His current research interests include QoS, medium access control, wireless multimedia and wireless networks protocol.

Teerapat Sanguankotchakorn was born in Bangkok, Thailand on 08th December 1965. He received the B.Eng in Electrical Engineering from Chulalongkorn University, Thailand in 1987, M.Eng and D.Eng in Information Processing from Tokyo Institute of Technology, Japan in 1990 and 1993, respectively under Japanese Government Scholarship. In 1993, he joined Telecommunication and Information Systems Research Laboratory at Sony Corporation, Japan, where he holds two patents on Signal Compression. He joined Asian Institute of Technology in October 1998 as an Assistant Professor. Currently, he is an Associate Professor at Telecommunications Field of Study, School of Engineering and Technology, Asian Institute of Technology. His research interests include Digital Signal Processing, High Speed Network Protocol, IP Network and Broadband Access Network, QoS in IP and Mobile Ad Hoc Network, and Cross-layer Design in Wireless Network. He is the senior member of IEEE and member of IEICE, Japan.