IEEE WCNC, Paris, France, May 2012

Achieving Minimum-Routing-Cost Maximum-Flows in Infrastructure Wireless Mesh Networks

T.H. Szymanski, Dept. ECE, McMaster University, Canada, teds@mcmaster.ca

***Abstract*—A multicommodity *Minimum-Cost Maximum-Flow* algorithm for routing multiple unicast traffic flows in infrastructure wireless mesh networks, represented as commodities to be routed in an undirected or directed graph, is presented. The routing-cost per edge can be any metric, i.e, a delay, a SNR, etc. To minimize resource-usage and transmission power, the routing-cost is formulated as a *Bandwidth-Distance product*, where high-bandwidth backhaul flows are routed over shorter distance paths. The routing algorithm requires the formulation of two linear-programming (LP) problems. The first LP performs constrained multicommodity flow maximization, where the traffic flowing between any source/destination pair is constrained to a sub-graph of the original graph to enforce distance constraints. The second LP performs multicommodity cost minimization, under the constraint that the aggregate flow is maximized. Both LPs can be solved in polynomial time. No other multicommodity unicast routing algorithm can achieve a larger *Maximum-flow*, or the same *Maximum-flow* rate with a lower cost. The algorithm can also be faster than other known *Maximum-Flow* algorithms. Given the physical interference model and an appropriate antenna model, every vector of commodity flow-rates within the *Capacity Region* of an infrastructure network can be scheduled to achieve rigorous throughput and QoS guarantees, using a recently-proposed scheduling algorithm. The algorithm is tested in a hexagonal infrastructure wireless mesh network to maximize backhaul traffic flows.**

***Keywords* - routing, minimum cost, maximum flow, Quality of Service, QoS, scheduling, channel assignment, Capacity Region, wireless mesh network**

I. Introduction

Infrastructure *Multihop Wireless Mesh Networks* (WMNs) [1], consisting of a collection of interconnected wireless mesh routers, represent a promising technology to deliver communication services over large geographic areas [2]. The IEEE 802.11s standard has recently been developed for mesh networks [3]. The WMN will provide communication infrastructure for both *stationary end-users* (i.e., homes, offices) and *mobile end-users* (i.e., smart-phones, tablet computers). The traffic between a wireless router and the end-users within a wireless cell is called *end-user traffic*. The delivery of traffic between the wireless routers in a multihop manner is called backhauling, and this traffic is called *backhaul traffic*.

This paper presents a routing algorithm which can achieve the *Maximum Flow* of multiple unicast backhaul traffic flows through a directed or undirected graph model of a WMN, while simultaneously minimizing the *Routing-Cost*, subject to predetermined cost constraints. The routing cost of an edge can be any metric, i.e., an edge delay, an SNR, an interference measure, a financial cost, etc. In this paper, the routing cost is defined as a *Bandwidth-Distance Product*, where high-bandwidth commodities are routed over shorter distance paths to minimize resource usage (i.e., aggregate edge loads) and aggregate transmission power.

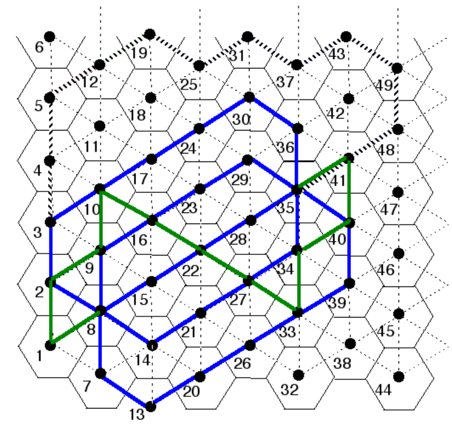

Fig. 1. Routing in a Multihop Infrastructure WMN.

An infrastructure WMN can be represented as a directed or undirected graph $G(V, E)$, where V is the set of fixed wireless routers ($|V| = N$), and E is the set of wireless edges. Each router has multiple wireless transceivers, which can operate over multiple OFDMA channels. Assume that each wireless edge can be provisioned at the physical layer to provide a fixed data-rate λ bits/sec in a scheduling frame (see section 2), as assumed in [4,5,6]. The set of edges E can also be represented by a binary matrix E, where $E(i, j) = 1$ if vertices i and j can communicate over a wireless edge with rate λ bits/sec.

A path $P = (v_1, ..., v_n)$ is a set of vertices representing adjacent wireless edges. Let the distance of a path be the number of wireless edges it traverses. (This definition may include the physical length of the edges.) Traditional routing algorithms may be *single-path*, or *multi-path*. Traditional single / multiple path routing algorithms will route each commodity over single / multiple paths through the network, selected from the set of *all possible end-to-end paths*. It is well known that optimal single-path or multi-path routing in general is NP-Hard, due to the complexity of enumerating paths.

Fig. 1 will illustrate several problems with traditional multipath routing algorithms. Fig. 1 shows a hexagonal WMN with 49 wireless cells, where each cell has a wireless router. Assume each router has a fixed location on a plane. Consider a backhaul traffic flow from nodes 8 to 35. There are a combinatorially-large number of paths between these routers, varying in distance. There is one minimum distance path

P = $(8, 15, 22, 28, 35)$, also denoted $P_{(8,15,22,28,35)}$, which traverses 4 wireless edges. Many paths with distance = 5 are contained within the convex hull shown by the paths $P_{(8,9,16,23,29,35)}$, and $P_{(8,14,21,27,34,35)}$. Many paths with distance = 8 are contained within the convex hull shown by paths $P_{(8,2,3,10,17,24,30,36,35)}$, and $P_{(8,7,13,20,26,33,39,40,35)}$. A path of distance 14 is also shown (dashed lines). To date there are no known algorithms to route multiple unicast commodities which achieve the *Maximum Flow* in a graph, while simultaneously achieving the *Minimum-Cost*. The set of all achievable flow rate vectors defines a polytope in $N\times(N-1)$-dimensional space, and the convex hull of the polytope defines the *Capacity Region* of the network [4,5,6].

A *multicommodity routing algorithm* which minimizes congestion in a graph, defined as the maximum aggregate load on any edge, was presented in [7]. The routing algorithm routes commodities over *all possible edges*, rather than sets of end-to-end paths. The routing algorithm was formulated as a linear-program (LP) which could be solved in polynomial time. More recently, [8] reports similar polynomial-time multi-commodity maximum-flow LPs. These algorithms [7,8] conform to the well known *Minimum-Cut Maximum-Flow* theorem, which states that the maximum achievable flow between any source/destination pair (s,d) is upper bounded by the capacity of the minimum number of edges whose removal results in no flow between (s,d). We call the algorithms in [7,8] the traditional *Multicommodity Maximum-Flow* LPs, which overcome the NP-Hardness of traditional multipath routing algorithms.

However, these traditional LPs share 3 significant problems: (1) They do not minimize cost, i.e., in a *Maximum-Flow* a commodity is equally likely to flow over longer or shorter paths. (2) A maximum-flow routing for commodities may contain cycles, as cost is not minimized. (3) The LPs can be intractable even for relatively small networks. Referring to Fig. 1, there are 49 wireless routers, and $\leq N \times (N-1) \approx 2400$ distinct commodity flows to be routed. There are $\approx$ 158 undirected edges or 296 directed edges (viewing each undirected edge as two directed edges). These prior *Multicommodity Max-Flow* LPs require the specification of a flow-rate variable for every edge and commodity pair, i.e., the network in Fig. 1 may require $\geq$ 700,000 flow-rate-variables to be solved, leading to excessively large and untractable LPs.

We present a routing algorithm to achieve the *Maximum Flow* of multiple unicast commodities while simultaneously achieving the *Minimum-Cost*, subject to constraints on the maximum cost of any commodity. For every commodity, a subgraph containing a set of candidate edges is specified. The removal of undesirable edges results in the specification of a sub-graph $G^c \in G$ for each commodity $c \in C$, which constrains the maximum allowable cost for every commodity. A first LP to maximize the aggregate traffic flow, subject to the constraint that every commodity is routed over its subgraph, is formulated. An efficient algorithm to find useful subgraphs is presented. We also propose an iterative solution algorithm called *Successive Relaxation*, where the subgraphs initially contain minimum-distance paths, and where selective subgraphs are expanded to include longer-distance paths when appropriate. The first LP will find the *Maximum-Flows* subject to routing cost constraints, which are recorded.

Any network can be viewed as having a finite amount of resources, expressed as a *Bandwidth-Distance Product* (or *BD-Product*). A second *Minimum-Cost* LP is formulated, where the maximum-flow rate of each commodity is fixed from the *Maximum-Flow* LP. The second LP will minimize the routing-cost of the *Maximum-Flow*. The cost of an edge can be any metric, a delay, a SNR, a financial cost, etc. However, to minimize resource-utilization and maximize energy-efficiency in our WMN model (see section 2), in this paper the routing-cost is formulated as a *Bandwidth-Distance product*. The second LP achieves the true *Minimum-Cost Maximum-Flow*, and removes any directed cycles. The second LP will minimize the average edge utilization, transmission power, and interference due to unnecessary transmissions. The proposed *Minimum-Cost Maximum-Flow* routing algorithm is tractable and efficient. No other routing algorithm can achieve a larger *Maximum-Flow*, and no other routing algorithm can achieve the same *Maximum-Flow* with a lower (linear) cost.

The paper is organized as follows. Section II presents our methodology for provisioning traffic in WMNs, and reviews multipath routing in WMNs. Section III presents the *Minimum-Cost Maximum-Flow* routing algorithm. Section IV presents experimental results. Section V concludes the paper.

II. QoS Provisioning in Infrastructure WMNs

Our methodology to provision *Minimum-Cost Maximum-Flows* in a wireless mesh network with *Near-Perfect* throughput and QoS guarantees is summarized in Fig. 2. Assume that every wireless edge can be provisioned to achieve an acceptable *Signal to Interference and Noise Ratio* (SINR), data-rate (DR) and *Packet Error Rate* (PER), over a TDMA scheduling frame, as assumed in [4,5,6]. A TDMA scheduling frame consists of F time-slots, each sufficient to transmit a packet between neighboring nodes. The SINR, DR and PER are periodically recomputed, as required. At the Physical layer, several technologies can be used to achieve the acceptable SINR, DR and PER, including *Time-Division Multiplexing* (TDM), *Orthogonal Frequency Division Multiplexing* (OFDM) and *Space-Division Multiplexing* (SDM) technologies.

Each wireless router has a fixed location (i.e., tower), and may use *Multiple-Input Multiple-Output* (MIMO) antennas to enable high-bandwidth communications with its neighboring routers. A MIMO transmitting antenna can be programmed to minimize interference to unintended receivers [11,12]. A MIMO receiving antenna can be programmed to 'null-out' interference from unintended transmitters [11,12]. As observed by Gupta and Kumar [5], given directional antenna the wireless edges behave more like traditional interference-free edges used in wired networks.

Let $\mathbb{R}$ denote the set of real numbers, and $\mathbb{Z}$ denote the set of integers. A global *Traffic Demand Matrix* $D \in \mathbb{R}^{N \times N}$ is specified for the WMN, where each element $D(i, j)$ specifies

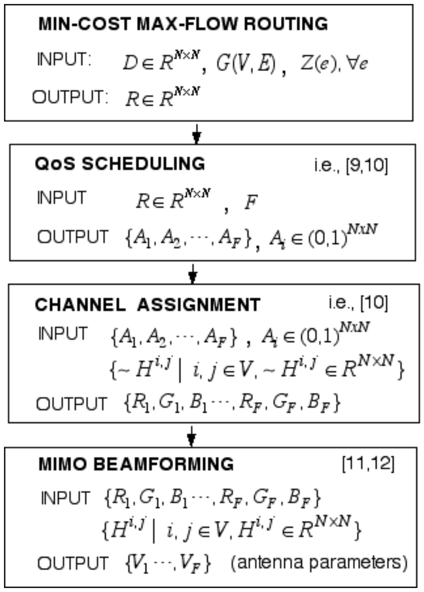

Fig. 2. QoS Provisioning in a Multihop Infrastructure WMN.

the required backhaul traffic rate between a pair of wireless routers (i, j). The matrix $D \in \mathbb{R}^{N \times N}$ is routed using the proposed *Minimum-Cost Maximum-Flow* routing algorithm. After the routing an *Edge Traffic Rate Matrix* $R \in \mathbb{Z}^{N \times N}$ is defined, where $R(i, j)$ specifies the required backhaul traffic rate between a pair of routers (i, j). The matrix R is mathematically decomposed using the recently-proposed scheduling algorithms in [9,10] to yield a TDMA schedule. The TDMA schedule consists of a specification of F sets of *Active Wireless Edges*, each denoted as a binary matrix $A_i \in (0,1)^{N \times N}$ valid for time-slot i in the TDMA scheduling frame. The scheduling algorithm must satisfy $\sum_{i=1}^{F} A_i = R$. To achieve *Near-Perfect* throughput and QoS guarantees, the scheduling algorithm must also satisfy $(\alpha R - K) \leq \sum_{i=1}^{\lceil \alpha F \rceil} A_i \leq (\alpha R + K)$, for any fraction $0 \leq \alpha \leq 1$ and small constant K [9]. Each set of active edges A_i forms a conflict-free *'Matching'* to be realized in one time-slot. The edges in each matching A_i are colored, i.e., assigned to orthogonal OFDMA channels, by a *Channel-Assignment* algorithm CA. Once colored, each matching A_i yields several sets of active edges, one set for each color in each time-slot. In Fig. 2, we assume each A_i for time-slot i is colored into 3 sets denoted by binary matrices $R_i, G_i, B_i \in (0,1)^{N \times N}$. The coloring algorithm must satisfy $(R_i + G_i + B_i) = A_i$ for $1 \leq i \leq F$. The channel-assignment minimizes the joint interference between active edges, and considers estimated channel gain matrices $\sim H^{i,j} \in \mathbb{R}^{N \times N}$ between routers (i,j). The MIMO antenna beamforming parameters and powers are then computed, using the channel gain matrices $H^{i,j}$ between routers. This methodology ensures that given a traffic demand matrix within the *Capacity Region* of the network, every backhaul traffic flow can be scheduled to receive mathematically-provable *Near-Perfect* throughput and QoS guarantees, given that the wireless edges meet the acceptable SINR, DR and PER requirements [9,10].

A. Traditional Multipath Routing in WMNs

One classical multipath maximum-flow routing problem formulation is a *Integer programming* optimization problem. Let C denote a set of commodities, and $c \in C$ denote one commodity. For unicast networks, each commodity c has a source and destination (s_c and d_c), and a requested traffic rate W^c which can flow over multiple paths. Let P^c be a set of enumerated candidate paths associated with a commodity c, and let $P^c(j) \in P^c$ denote the j-th path in P^c. Let each commodity flow $c \in C$ have a vector of binary decision variables B^c, and let $B^c(j) \in B^c$ denote the j-th decision variable. The decision variable is asserted if any fraction of the commodity c is routed over the corresponding path. A multicommodity maximum-flow optimization problem can be stated as follows:

$$\text{maximize} \sum_{c \in C, p \in P^c} x^c(p) \qquad (1)$$

$$\begin{array}{cccr}
x^c(p) & \geq 0 & \forall c \in C, \forall p \in P^c & (1.1) \\
x^c(e) & \leq Z(e) & \forall c \in C, \forall e \in p, \forall p \in P^c & (1.2) \\
\sum_{c \in C} x^c(e) & \leq Z(e) & \forall e \in p, p \in P^c & (1.3) \\
B^c(p) & \in 0,1 & \forall c \in C, \forall p \in P^c & (1.4) \\
\sum_{p \in P^c} B^c(p) \cdot x^c(p) & \leq W^c & \forall c \in C & (1.5) \\
\sum_{p \in P^c} B^c(p) & \leq K & \forall c \in C, \forall p \in P^c & (1.6)
\end{array}$$

Constraints 1.1-1.3 enforce edge capacity constraints. Constraint 1.5 ensures that the requested flow rate for commodity c, i.e., W^c, is not exceeded. Integer constraint 1.6 asserts that one commodity flows over at most K paths. There are combinatorially-many end-to-end paths to be considered for each flow (in set P^c), and the problem of jointly selecting K optimal paths for each traffic flow is NP-Hard in the general case. Nevertheless, the algorithm can be effective in practice, often yielding solutions within a few percent of the optimal maximum-flow [9].

III. The Minimum-Cost Maximum-Flow Algorithm

This section will present the *Minimum-Cost Maximum-Flow* routing algorithm. To motivate our algorithm, some properties of maximum-flows are first summarized. Referring to Fig. 1, assume all edges have unity distance (1 meter) and the capacity of each edge is Z bits/sec. A minimum-cost routing of the commodity between (8,35) is over the minimum-distance path $P_{(8,15,22,28,35)}$ with a distance of 4. The *BD-Product* of the flow is $4Z$ bit-meters/sec. A *Maximum-Flow* between (8,35) supports 6 paths as shown in bold in Fig. 1, each with capacity Z bits/sec. There are several other maximum-flows with larger costs. A non-unique *Minimum-Cost Maximum-Flow* is shown in bold lines. Notice that there is a significant cost of achieving any *Maximum-Flow*. In Fig. 1, the 2 flows of distance 5 each consume a *BD-Product* of $5Z$ bit-meters/sec, and the 2 flows of distance 8 each consume a *BD-Product* of $8Z$ bit-meters/sec. Therefore, the *Minimum-Cost Maximum-Flow* routing between (8,35) achieves a bandwidth of $6Z$ bits/sec, and consumes $42Z$ bit-meters/sec of resources. Observe that the maximum-flow removes considerable resources from the network that other commodities could use. In practice, it is desirable to constrain the maximum distance that any commodity may flow.

Define the *BD-Expansion* of a commodity flow between a vertex pair (s,d) in a graph G(V,E) as the ratio of the achieved *BD-Product* per unit rate given a routing, over the *Minimum BD-Product* per unit rate for the commodity (when all other commodities are unrouted). The *BD-Expansion* illustrates the effectiveness of a given topology G(V,E) and routing algorithm to realize a particular commodity flow(s). An expansion close to unity indicates the network and routing algorithm are well-suited the handle the commodity flow. A larger *BD-Expansion* indicates the network and routing algorithm consume excessive resources to achieve the maximum flow. The *Max-Flow* in Fig. 1 has a *BD-Expansion* of 1.75.

A. Determining Feasible Edge Sets

Assume that each commodity is constrained to flow over a set of feasible edges. An efficient algorithm to determine a feasible edge set for each commodity is specified. Assume a planar graph G(V,E), as shown in Fig. 1. The algorithm is shown in Fig. 2. Given an traffic demand matrix $D \in \mathbb{R}^{N \times N}$, there are up to $N(N-1)$ commodities to be routed. The objective is to compute a vertex-induced subgraph $G^c(V^c, E^c) \in G(V,E)$ for each commodity, with distance constraints.

For every node in G, the algorithm initially computes a minimum distance path to every other node using Dijkstra's algorithm. The complexity of Dijkstra's algorithm is $O(|E| + |V| log|V|)$. For every commodity c to be routed between a source/destination pair (s,d), the algorithm initializes a set of candidate nodes and and a set of candidate edges to be NULL. The algorithm then visits all intermediate nodes in the set V. Let $M(s,v)$ denote the length of a minimum-distance path between (s,d). If $M(s,v)$ + $M(v,d) \leq M(s,d)$ plus a distance-threshold DT, then the node v is included in a set of feasible vertices V^c for the commodity. The set of feasible edges E^c consists of the edges in the vertex-induced subgraph V^c, i.e., the set of edges in E whose endpoints are in V^c, subject to distance constraints. The computation of the subgraph for each commodity has complexity $O(|V|)$, assuming matrix M is precomputed. Referring to figure 1, for the commodity flow between (8,35), all edges contained within the convex hull of distance 5 are found using this algorithm with threshold DT = 1. Once a subgraph G^c is computed for every commodity $c \in C$, the *Minimum-Routing-Cost Maximum-Flow* LP can be formulated.

B. The Maximum-Flow LP

The constrained *Maximum-Flow* of a single commodity c over a source/destination pair (s_c, d_c) is found by Eq. 2:

$$\text{Maximize: } r^* \qquad (2)$$

```
for s ∈ V
      M(s,:) = minimum distance vector
               (using Dijkstra's algorithm)
end;
for c ∈ C
  Vc= NULL;
  for v ∈ G
        if (M(sc,v) + M(v,dc)) ≤ (M(sc, dc) + T)
           V'= V' ∪ v
        end;
  end;
end;
Ec = NULL;
for u ∈ Vc, v ∈ Vc
   if ((u,v) ∈ E) & (M(sc,u)+M(v,dc)) ≤ (M(sc, dc)+T)
        Ec = Ec ∪ (u,v)
   end;
end;
```

Fig. 3. Algorithm 1 : Finding feasible subgraphs for commodities.

Subject to:

$$
\begin{array}{cclc}
0 & \leq r^c(e) & \forall e \in E^c & (2.1) \\
r^c(e) & \leq Z(e) & \forall e \in E^c & (2.2) \\
r^c_{in}(v) & = r^c_{out}(v) & \forall v \in V^c - (s_c, d_c) & (2.3) \\
r^c_{in}(s_c) & = 0 & & (2.4) \\
r^c_{out}(d_c) & = 0 & & (2.5) \\
r^c_{out}(s_c) & \leq W^c & & (2.6)
\end{array}
$$

$$r^* = r^c_{in}(d_c)$$

Let $r^c(e)$ denote the flow rate of commodity c over edge e. Let $r^c_{in}(v)$ and $r^c_{out}(v)$ denote the sum of flows into and out-of vertex v due to commodity c, respectively. This LP constrains the flow of every commodity c to edges in the subgraph $G^c(V^c, E^c)$. The *Constrained Multicommodity Maximum-Flow LP* (LP #1) is given by Eq. 3:

$$\text{Maximize: } r^* \qquad (3)$$

Subject to:

$$
\begin{array}{cclc}
0 & \leq r^c(e) & \forall c \in C, \forall e \in E^c & (3.1) \\
r^c(e) & \leq Z(e) & \forall c \in C, \forall e \in E^c & (3.2) \\
\sum_{c \in C} r^c(e) & \leq Z(e) & \forall c \in C, \forall e \in E & (3.3) \\
r^c_{in}(v) & = r^c_{out}(v) & \forall c \in C, \forall v \in V^c - (s_c, d_c) & (3.4) \\
r^c_{in}(s_c) & = 0 & \forall c \in C & (3.5) \\
r^c_{out}(d_c) & = 0 & \forall c \in C & (3.6) \\
r^c_{out}(s_c) & \leq W^c & \forall c \in C & (3.7)
\end{array}
$$

$$r^* = \sum_{c \in C} r^c_{in}(d_c)$$

Constraint 3.3 enforces edge capacity constraints. Flow-balance constraints 3.4-3.7 are restricted to the subgraph G^c for each commodity c. The LP is solved and the maximum-flows are determined. (Due to space limitations, only the LPs for the directed graphs are presented.)

In the proposed *Successive Relaxation Algorithm* if the desired flow rate of a commodity c cannot be achieved, then the sub-graph $G^c(V^c, E^c)$ must be expanded. The subgraph G^c can also be processed to determine which other commodities, called interfering commodities, flow over saturated edges in G^c. The subgraphs for those interfering commodities can also be expanded, thereby relieving the congestion in G^c.

C. The Minimum-Cost LP

To obtain the minimum-achievable routing-cost, a second LP (LP #2) is formulated in Eq. 4. Let Γ^c denote the flow-rate achieved by commodity c in the *Maximum-Flow* LP.

$$\text{Minimize:} \quad y^* \tag{4}$$

Subject to:

$$\begin{array}{llll} r^c_{out}(s_c) & = \Gamma^c & \forall c \in C & (4.1) \\ \sum_{c\in C} r^c(e)) & \leq Z(e) & \forall c \in C, \forall e \in E & (4.2) \\ r^c_{in}(v) & = r^c_{out}(v) & \forall c \in C, \forall v \in V' & (4.3) \\ r^c_{in}(s_c) & = 0 & \forall c \in C & (4.4) \\ r^c_{out}(d_c) & = 0 & \forall c \in C & (4.5) \end{array}$$

$$y^* = \sum_{c\in C} \sum_{e\in E} x^c_e \times M(e)$$

where $V' = V - (s_c, d_c)$. Constraint 4.1 requires that every commodity flow-rate is fixed as determined by LP #1. The remaining constraints are similar to those in the preceding LP. The objective function is to minimize is the sum of all fractional commodity flow-rates over each edge e times the distance of the edge $M(e)$. However, any linear cost function can be used.

D. Properties

Property 1: By setting the distance-threshold DT for every unicast commodity equal to the maximum distance Δ in G, LP #1 finds the maximum achievable aggregate flow of all commodities, subject to capacity constraints. LP# 1 finds a routing which adheres to the well-know *Minimum-Cut Maximum-Flow* theorem for each commodity, and no other routing algorithm can achieve a larger *Maximum-Flow*.

Proof: By setting the threshold to the maximum distance in G, then the subgraph associated with every commodity is the full graph G(V,E), and there are no additional constraints associated with any commodity. The LP #1 reduces to the *Maximum-Flow* LP in [7,8], which also adheres to the well-know *Minimum-Cut Maximum-Flow* theorem for each commodity. ■

Property 2: By solving LP #2, given the solution vector $\{\Gamma^c\}$ of LP #1 for which the distance-threshold DT for every commodity equals to the maximum distance in G, the routing-cost of the *Maximum-Flow* is minimized. No other routing algorithm can achieve a *Maximum-Flow* for multiple unicast commodities with a lower linear cost.

Proof: By contradiction. If it is not true, then either the solution to LP #1 did not yield a maximum-flow, or the solution to LP #2 did not yield a minimum-cost. ■

Theorem 1: Given a traffic demand matrix D, a minimum-distance matrix M and a topology G where the edge capacities are Z bits/sec and edges have normalized (unit) distance, a necessary condition for the D to lie within the *Capacity Region* of G is that the bandwidth-distance product of D must not exceed the bandwidth-distance product of G, i.e.,

$$\sum_{s=1}^{N} \sum_{d=1}^{N} D(s,d)M(s,d) \leq |E|Z \tag{5}$$

where $|E|$ is the number of edges.

Proof: By contradiction. Suppose a matrix D which violates Eq. 5 can be routed. Consider routing commodities iteratively along minimum-distance paths, decrementing the capacities of all traversed edges appropriately after each commodity is routed. At some point, the remaining Bandwidth-Distance capacity on the RHS will be exhausted before all the commodities have been routed. The routing of any remaining commodity along a minimum-distance path will result in an edge capacity violation. ■

Theorem 1 yields a necessary condition for any requested traffic demand matrix to lie within the *Capacity Region* of a network. It yields a simple test to determine if a requested traffic demand matrix can be realized by any *Multicommodity Maximum-Flow* routing algorithm.

Theorem 2: Consider a random uniform traffic demand matrix D, and a planar mesh topology G(V,E) , where (i) the $|V| = N$ routers are uniformly distributed over the square of dimension $\sqrt{N} \times \sqrt{N}$, (ii) the node degree is bounded, (iii) the edge capacities are Z bits/sec, and (iv) the edges have unit normalized length. Then the expected throughput per node is upper bounded by:

$$O\left(Z|E|/(N\sqrt{N}\ )\right) \tag{6}$$

Proof: By contradiction. The BD-Product of G is given by the numerator of Eq. 6, where the number of edges $|E|$ is $O(N)$. The expected aggregate distance demanded by N nodes is given by $O(N\sqrt{N})$, i.e., each of the N nodes demands a distance of $O(\sqrt{N})$. If Eq. 6 is not true, then a routing for the random uniform traffic matrix D exists where the bandwidth-distance product used exceeds the bandwidth-distance capacity available, which contradicts theorem 1. ■

Result 1: By Theorem 2, the expected throughput per node in a hexagonal WMN as shown in Fig. 1 is upper bounded by $O(Z/\sqrt{N})$. Result 1 is consistent with the asymptotic upper bound established by Gupta and Kumar in [5], and illustrates the usefulness of the *Bandwidth-Distance-Product* metric.

IV. EXPERIMENTAL RESULTS

This section summarizes the results of the LPs for a 36-node hexagonal mesh, using the topology shown in Fig. 1. Table 1 illustrates the results of the traditional *Multicommodity Max-Flow LP* [8], and of our *Constrained Maximum-Flow* LP #1 with distance constraints relaxed, on a 36-node hexagonal mesh. Let the capacity of every edge = 4 Mbits/sec, and the length be 1 meter. The *Bandwidth-Distance product* of the

TABLE I
MAXIMUM-FLOW LP, DISTANCE THRESHOLD = ∞

Traffic	BD(M)	BD(R)	$\lvert x \rvert$	α	β	ExT	Flow
M1	237	652.8	12,240	2,592	314	6.40	71
M2	223	657.1	12,240	2,592	314	5.72	69
M3	209	634.9	12,240	2,592	314	5.38	69
M4	217	642.1	12,240	2,592	314	5.67	71
100 Ms	231.3	617.6	12,240	2,592	314	5.57	69.9

TABLE II
MAXIMUM-FLOW LP, DISTANCE THRESHOLD = 0.

Traffic	BD(M)	BD(R)	$\lvert x \rvert$	α	β	ExT	Flow
M1	237	237	507	444	302	0.18	71
M2	223	223	463	415	302	0.14	69
M3	209	209	409	381	299	0.18	69
M4	217	217	421	391	293	0.10	71
100 Ms	231.3	231.3	545.5	460.4	314	0.130	69.9

network is 170 edges (1 meter) * 4 Mb/sec= 680 Megabit-meters/second (Mbm/s). From theorem 2, a necessary condition for a traffic pattern to be achievable is that its *BD-product* must not exceed 680 Mbm/s.

Each traffic pattern consist of 2 random permutations, where every node transmits to 2 nodes each with rate = 1 Mb/sec. Rows 1-4 each represent the results of routing one traffic pattern. The last row represents the results of routing 100 randomly selected traffic patterns. Let $|x|$, α, and β denote the size of the vector solution x, the number of equality constraints, and the number of inequality constraints (using summations) in the LP, respectively. Let *ExT* and *Flow* denotes the execution time and aggregate flow rate, respectively. According to Table 1, all traffic patterns are achievable, i.e., they lie within the *Capacity Region* of the WMN. The average *Minimum BD-Product* of a traffic pattern is 231.3 Mbm/s, which is below the network capacity of 680 Mbm/s. The network consumes most of its resources to route each permutation. The average *BD-product* is 617.6 Mbm/s, representing a *BD-Expansion* of 2.67. The average edge utilization is 90.8%, a very high load.

According to Table 1, the size of the LP problem remains fixed as expected. The problem was solved in a 4-core 2.8 GHz workstation with 16 Gigabytes of main memory. The average execution time per problem is 5.57 seconds. *The LP was intractable for a hexagonal mesh with 64 nodes.*

Table 2 illustrates the results of the *Constrained Maximum-Flow* algorithm, when the distance threshold = 0, i.e., every commodity is constrained to follow a minimum distance path. The average number of flow-variables to be solved is reduced to 545.5 (versus 12,240 for the LPs in [7,8]). The average number of equality constraints is reduced to 460.6 (vs. 2,592). The average number of inequality constraints is 305.6 (vs. 314). The average execution time to find a *Minimum-Cost Maximum-Flow* using unconstrained LP #1 followed by LP # 2 is $\approx$ 12.1 seconds. The average execution time using the *Constrained Maximum-Flow* LP is 0.13 seconds, representing a speedup of a factor of $\approx$ 90. Perhaps the most surprising result is the quality of the solution. The *BD-Product* of each traffic pattern was 231.3 Mbm/s, compared to 617.6 Mbm/s, i.e., all traffic patterns were routed along minimum-distance paths. The average edge load was 34%, much lower than 90.8% for the algorithm in [8]. The proposed algorithm has been tested on several network topologies and traffic patterns, and the results are consistent. The proposed *Constrained Maximum-Flow* LPs result in considerably better resource utilizations.

V. CONCLUSIONS

A polynomial-time constrained unicast multicommodity *Minimum-Cost Maximum-Flow* routing algorithm for general directed or undirected graphs has been presented. No other unicast multicommodity routing algorithm can achieve a larger *Maximum-Flow*, or the same *Maximum-Flow* with a lower (linear) cost. By constraining the problem size, the routing algorithm can be considerably faster than other known *Maximum-Flow* algorithms, and can achieve considerably better resource utilizations. The algorithm has been tested on several wireless mesh topologies, and often achieves speedups between factors of 50...100. By relaxing distance constraints, the algorithm can find all *Maximum-Flow* routings within the *Capacity Region* of a network, addressing a problem identified in [4,5,6]. The true power of the algorithm is illustrated when computing *Maximum-Flow* routings for larger mesh networks, which can be intractable with the conventional *Multicommodity Maximum-Flow* LPs in [7,8]. The proposed routing algorithm removes undesirable edges from consideration, resulting in significantly smaller LPs to solve, with significantly faster solutions.

REFERENCES

[1] I.F. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks", *IEEE Radio Communications*, pp. S23-S30, 2005.

[2] Y. Zhang, N. Ansari, and H. Tsunoda,"Wireless Telemedicine Services over Integrated IEEE 802.11/WLAN and IEEE 802.16/WIMAX Networks", *IEEE Wireless Comm.*, Feb. 2010.

[3] M.J. Lee, J. Zheng, Y-B Ko and D.M. Shrestha,"Emerging Standards for Wireless Mesh Networks", *IEEE Wireless Commun.*, April 2006.

[4] M. Kodialam and T. Nandagopal, "Characterizing Achievable Rates in Multi-Hop Wireless Mesh Networks With Orthogonal Channels", *IEEE/ACM Trans. Networking*, Vol. 13, No. 4, Aug. 2005

[5] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", *IEEE Trans. Inform. Theory*, Vol. 46, No. 2, March 2000

[6] S. Toumpis and A.J. Goldsmith, "Capacity Regions for Ad-Hoc Wireless Networks", *IEEE Trans. Wireless Comm.*, Vol. 2, No. 4, July 2003

[7] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke, "Optimal Oblivious Routing in Polynomial Time," *Proc. 35th ACM Symp. Theory of Computing (STOC)*, 2003.

[8] Z. Li, B. Li, and LC. Lau, "On Achieving Maximum Multicast Throughput in Undirected Networks", *IEEE Trans. Inform. Theory*, Vol. 52, No. 6, June 2006

[9] T.H. Szymanski and D. Gilbert, "Provisioning Mission-Critical Telerobotic Control Systems over Internet Backbone Networks with Essentially-Perfect QoS", *IEEE JSAC,* June 2010.

[10] T.H. Szymanski, "Provisioning Backhaul Traffic Flows in Infrastructure Wireless Mesh Networks with Near-Perfect QoS", *IEEE Sarnoff Symp.*, Princeton NJ, April 2010.

[11] R.A. IItis, SJ Kim, and D.A. Hoang, "Noncooperative Iterative MMSE Beamforming Algorithms for Ad Hoc Networks", *IEEE Trans. Communications*, Vol. 54, No. 8, pp. 748-759, 2006.

[12] M. Cao, X. Wang, S-J Kim, and M. Madihian, "Multi-Hop Wireless Backhaul Networks: A Cross-Layer Design Paradigm", *IEEE JSAC*, Vol. 25, No. 4, pp. 738-748, May 2007.