

# Min-Max Congestion in Interference-Prone Wireless Mesh Networks

Sonia Waharte, Arash Farzan and Raouf Boutaba  
D. R. Cheriton School of Computer Science  
University of Waterloo  
Waterloo, Ontario, Canada N2L 3G1  
Email: {spwaharte,afarzan,rboutaba}@cs.uwaterloo.ca

**Abstract**—Users' demand of seamless connectivity has pushed for the development of alternatives to traditional infrastructure networks. Potential solutions should be low-cost, easily deployable and adaptive to the environment. One approach that has gained tremendous attention over the past few years consists in deploying a backbone of access points wirelessly interconnected, offering users access to the wired infrastructure via multi-hop communication. However, the limited transfer capacities and the interference resulting from a shared transmission medium can prevent further deployment if the network performance does not meet users' expectations. In this work, we explore different ways to improve the nominal network capacity while accounting for the phenomena of intra-interference (interference on a single path) and inter-interference (interference among flows on different paths). In particular, under the assumptions of splittable traffic flows, we present an interference-aware linear-programming formulation of the Min-Max congestion problem with and without constraints on the path length. Finally, we evaluate via simulations the potential bandwidth capacity gain resulting from the implementation of these different approaches.

## I. INTRODUCTION

Providing seamless connectivity with bounded guarantees in terms of quality of service represents a challenge that many academic researchers and industrial consortia are currently trying to tackle. To obtain accurate information on the wireless channel quality in a timely manner is rendered difficult by its sensitivity to variations in environmental noise. Therefore, developing efficient algorithms to enhance the network utilization is a challenging problem. When data are carried over the wireless medium via multiple hops, several issues arise: which paths should the packets follow? What is the impact on on-going data transmissions? What is the resulting network performance? These are fundamental questions that need to be addressed for a successful deployment of any wireless multi-hop infrastructure.

The work presented in this paper has been carried through while focusing on its applicability to wireless mesh networks. Contrary to ad hoc networks, wireless mesh networks present a fixed wireless backbone, non energy-constrained, with one (or more) dedicated node acting as a gateway between the wireless domain and the wired infrastructure (typically the Internet). Our study is restricted to the part of the data transfer occurring in the wireless domain. Traffic flows are mainly directed to/from the network gateway, that therefore has a complete view of the traffic transiting through the wireless

backbone. The network gateway can consequently use this information to take informed decisions on which routes the traffic flows should be directed to in order to minimize the overall interference (and consequently the network congestion level), and to optimize the network resource usage. This nonetheless requires that the link capacities remain stable over a sufficiently long period of time. Otherwise, the routing control messages may become stale before to reach their destinations. This condition being respected, we can see the advantages of a centralized administration.

In this paper, we address the problem of traffic flows allocation while minimizing the maximum link utilization. This problem is fundamental for traffic engineering as it relates to minimizing the network congestion level by optimizing the network resource usage. The network congestion problem applied to wireless networks differs from traditional approaches targeted to wired networks by the fact that nodes interfere with each other if they transmit simultaneously on the same channel, within the same geographical area. Consequently, links interrelations have to be integrated in the problem formulation. We also consider an extension to this problem in order to include path length constraints. The intent is to prevent packets to follow long paths that could possibly result in end-to-end delays not acceptable for QoS-sensitive applications (e.g. voice applications). Our contributions are the followings. We present a method that finds the optimal solution to the network congestion problem in polynomial time. To prevent a non-utilized link to become the most congested link, we explore an extension of this problem by adding the constraint that only a link carrying a strictly positive traffic load can become the most congested link. We propose a heuristic that efficiently solves this problem in polynomial time. We validate through simulations the capacity gain of our algorithms with the assumption of splittable traffic flows.

The remainder of the paper is as follows. In Section II, we describe the context of our work. In Section III, we introduce our network model and define some terminologies used in the rest of the paper. We present our formulation of the interference-aware flow assignment problem in Section IV. Performance comparisons are presented in Section V. We conclude the paper in Section VI.

## II. INTERFERENCE PHENOMENON IN WIRELESS MULTI-HOP NETWORKS

In wireless communications, severe performance degradation can result from interference of concurrent data transmissions. The shared transmission medium constrains all nodes within interference range of a sender or receiver to inactivity until completion of the ongoing communication. In their seminal work [4], Gupta and Kumar showed that in a wireless network with  $n$  identical nodes, the achievable per node throughput is  $\Theta(1/\sqrt{n \log n})$  with random node placement and communication pattern. Under the assumption of an optimal node placement and communication pattern, this throughput becomes  $\Theta(1/\sqrt{n})$ . Interference can occur within a single flow (Figure 1) or between multiple flows (Figure 2). This phenomenon is known as *route coupling* effect.

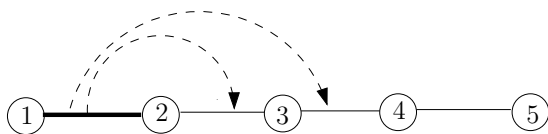


Fig. 1. Intra-flow Interference: Communication on Link 1-2 blocks Link 2-3 and Link 3-4

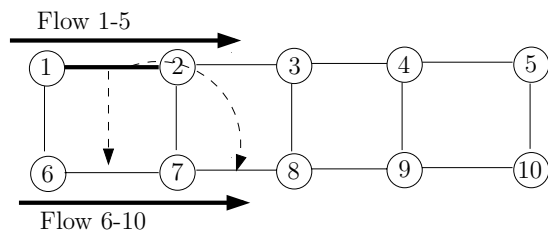


Fig. 2. Inter-flow Interference: 2 flows transit between Node 1 and Node 5, and between Node 6 and Node 10. Communication on link 1-2 blocks Link 6-7 and Link 7-8

To evaluate the impact of such topologies on the maximal achievable throughput, let us assume that a source node and a destination node are 5 hops apart, and that all links have the same nominal network capacity  $B$  (throughput that can be achieved at the MAC layer in one hop [6]). In the worst case scenario, the maximum achievable throughput for the source-destination pair depicted in Figure 3 is limited to  $B/5$  because of the bottleneck link (represented in bold) which blocks four other links.

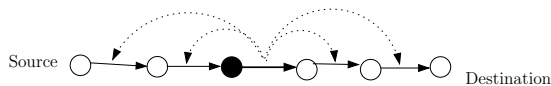


Fig. 3. Throughput is bounded by  $B/5$  if the path length exceeds 4 hops

If two traffic flows are routed on paths within interference range, the maximal throughput is reduced to  $B/7$  (Figure 4).

If these two flows share a common node along their respective path, the maximal achievable throughput goes down to  $B/9$  (Figure 5).

These simple examples illustrate the importance to account for the phenomenon of interference as it significantly impacts

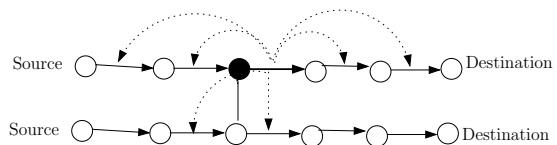


Fig. 4. Throughput is bounded by  $B/7$  when 2 paths are interconnected by 1 link

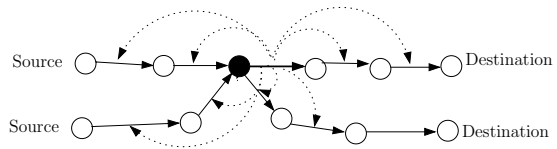


Fig. 5. Throughput is bounded by  $B/9$  when 2 paths share a common node

the network performance and consequently the way the routing policies should be designed.

## III. NETWORK MODEL

### A. Assumptions

Wireless mesh networks exhibit unique salient characteristics that need to be considered in the design of efficient solutions to the problems we are tackling. Their differences with existing multi-hop wireless networks mainly concern the following two aspects:

- The wireless backbone is composed of non energy-constrained static nodes;
- Traffic flows converge towards the node acting as the gateway between the wireless backbone network and the wired infrastructure network (e.g. the Internet).

Given these characteristics, we adopt a centralized approach to address the network congestion problems. The network gateway plays a central role as the traffic flows converge towards it. However, for the traffic information gathered by the network gateway to be useful, it is necessary to assume that the link status and traffic characteristics remain stable over a period of time sufficiently long to allow the routing information to propagate and therefore to avoid oscillation situations.

### B. Definitions

As previously mentioned, wireless communications differ from their wired counterparts as interference can occur within a single flow (*intra-flow interference*) as well as between paths located within interference range (*inter-flow interference*). To account for this phenomenon, we define the following terminology that we will use in the rest of the paper.

*Definition 1:* Given a link  $(i, j)$ , the *interfering set of  $(i, j)$*  called  $Inf_{ij}$  is defined as the group of links for which at least one endpoint is at interference range of  $i$  and/or  $j$  (including link  $(i, j)$ ).

*Definition 2:* Given a directed graph  $G(V, E)$  in which each link  $(i, j) \in E$  carries a traffic  $f(i, j)$ , the *normalized link*

utilization  $I_{ij}$  of  $(i, j)$  is defined as the sum of the traffics of the links in the interfering set of  $(i, j)$  divided by their respective capacities.

$$I_{ij} = \sum_{(k,l) \in \text{Inf}_{ij}} \frac{f(k,l)}{C(k,l)}$$

A link utilization of 1 means that the link is fully utilized and that no further traffic can be supported by this link.

#### IV. MINIMIZING THE MAXIMUM CONGESTION

In this section, we are interesting in solving the following problem.

*Given a network topology and a traffic demand matrix, determine the load repartition that minimizes the maximum network congestion if a feasible assignment exists.*

Minimizing the network congestion is equivalent to minimizing the maximum link utilization (as defined in the previous section). Traffic congestion control is a classic traffic engineering problem. With appropriate flows allocations, more users can be handled and more traffic can be supported by the networks.

Two different scenarios can be distinguished depending if the most congested link (with the highest utilization) is carrying a positive traffic load (and therefore can be considered as active) or not. To illustrate the importance of making such a distinction, let us consider the following example depicted in Figure 6.

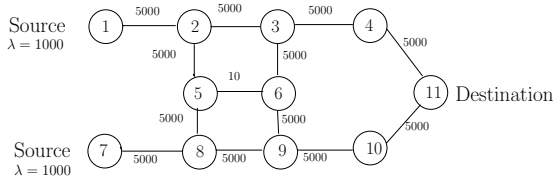


Fig. 6. 2-source network

Node 1 and Node 7 are sending 1000 units of traffic to Node 11. In this particular example, Link 5-6 is congested if the traffic flows follow the paths as highlighted in Figure 7. If each link utilization is accounted for, Link 5-6 would constitute the bottleneck link and theoretically, to remain under the congestion level (*i.e.* to remain below a link utilization of 1), only 5/6 of the initial traffic load can be supported by the network. However, if we restrict the problem to considering only the active links, all the traffic demands can be accommodated as none of the active links is over utilized.

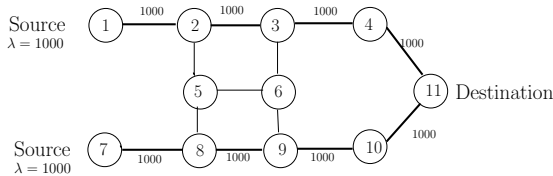


Fig. 7. 2-source network: Link 5-6 is congested

#### A. Problem Formulation

The problem we are tackling can be formulated as a linear programming problem and therefore can be solved in polynomial time. The basic idea of the algorithm consists in determining the optimal capacity ratio  $\alpha$  to apply to the link capacities that minimizes the maximum link utilization with respect to the flow conservation and interference constraints. The rationale behind this approach relies on the observation that the utilization  $I_{ij}$  of a link  $(i, j)$  is bounded by 1. Therefore, in order to minimize the maximum link utilization, it is sufficient to find the minimum value of  $\alpha$  such that  $I_{ij} \leq \alpha$ .

The network can be abstracted by a directed graph  $G = (V, E)$  in which each edge  $(u, v) \in E$  has a capacity  $C(u, v) \geq 0$ . Given a traffic demand matrix in which source node  $s_i$  has a traffic demand  $d_i$  strictly positive,  $i = 1, \dots, k$ , the resulting linear problem is similar to the maximum-flow problem with additional constraints on the interference and a new objective function minimizing the cost:

$$\left\{ \begin{array}{l} \text{Minimize } \underline{c^T X} \text{ subject to:} \\ \sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u) \quad \forall u \in V \setminus \{s_1, \dots, s_k, t\} \\ \sum_{v \in V} f(s_i, v) = \sum_{v \in V} f(v, s_i) + d_i \\ \sum_{u \in V} f(u, t) = \sum_{u \in V} f(t, u) - \sum_{i=1..k} d_i \\ I_{uv} \leq \alpha \quad \forall (u, v) \in E \end{array} \right.$$

where  $X = [f(1, 2), \dots, \alpha]$ , and  $c^T = \underbrace{[0; 0; \dots; 0; 1]}_{2m}$ .

Let  $m$  be the number of links and  $n$  the number of nodes. The interfering sets for each link can be easily computed in  $O(m \log m)$ . The most complex subroutine is to solve the linear program as defined above. Linear programs have been shown to be solvable on the worst case in  $O((m+n)^3 L)$  arithmetic operations [3] [8], where  $L$  is a parameter measuring the precision needed to perform the arithmetic operations exactly [7]. Therefore, determining a load-balanced flow allocation with Min-Max interference can be realized in  $O(m \log m + (m+n)^3 L) = O((m+n)^3 L)$ .

#### B. Problem Extension with Active Links Only

This problem extends the previous one as we are interested in minimizing the maximum congestion while ignoring the links that are not carrying any traffic load. Indeed, if no traffic is exchanged between two given nodes, to consider the link utilization of the link between these two nodes is not meaningful. Finding a solution to this problem is harder as it necessitates an exhaustive search consisting in computing for all subsets of links the minimum achievable utilization while ignoring the interference on the remainder of the links as they are considered as idle. As the solution can only be found in exponential time, we propose the following heuristic that runs

in polynomial time. Algorithm 1 iteratively determines the link with the highest utilization. This link can be in two states:

- 1) **Case 1: Idle.** This link is ignored in the subsequent iterations, its flow is set to zero and its capacity is set to infinity.
- 2) **Case 2: In use.** Either there exists another feasible flow assignment in which the flow on this link is null but this link still has the highest interference level. In that case Case 1 applies. If there is no such alternative, the algorithm terminates.

Compared to the previous algorithm, this algorithm only requires  $O(m)$  additional iterations to terminate.

---

**Algorithm 1** Minimize the Maximum Utilization with Active Links Only

---

- 1: Calculate the interfering set for each link
  - 2: Calculate the optimal load assignment, which yields a capacity ratio *CapacityRatio*
  - 3: *CurrentCapacityRatio*=*CapacityRatio*
  - 4: **while** true **do**
  - 5:     Set the flow on the bottleneck link to 0
  - 6:     Calculate the optimal load assignment
  - 7:     **if** *CapacityRatio*  $\leq$  *CurrentCapacityRatio* **then**
  - 8:         *CurrentCapacityRatio* = *CapacityRatio*
  - 9:     **else**
  - 10:         Return traffic flow matrix
  - 11:     **end if**
  - 12: **end while**
- 

### C. Problem Extension with Paths Length Constraints

We extend the network congestion problem by adding a constraint on the path length. The feasible set of paths between a source and a destination should only include disjoint paths with bounded length. The path disjointness constraint ensures some degree of network reliability, whereas the maximum path length constraint enforces that the end-to-end delay remains within certain bounds (if the path length exceeds a certain number of hops, even without congestion, the end-to-end delay can exceed the requirements of delay-sensitive applications).

---

**Algorithm 2** k-shortest simple paths algorithm with minimum path length

---

- 1:  $k=1$
  - 2: Compute minimum path length
  - 3: **while**  $k \leq \min(\text{edge} - \text{connectivity}(s), \text{edge} - \text{connectivity}(t))$  **do**
  - 4:     Compute paths
  - 5:     **if** Maximum path length  $>$  Minimum path length **then**
  - 6:         break
  - 7:     **end if**
  - 8: **end while**
- 

The k-shortest paths problem has been studied for different scenarios and different networks environments [2], [5]. In this work, we use a simple implementation described in Algorithm

2 that performs well for a small number of paths. More complex algorithms can be implemented if the number of paths considered becomes more important. To determine the running time of this algorithm, let us consider a non-directed graph  $G = (V, E)$  with  $n$  vertices and  $m$  edges, two vertices  $s$  and  $t$ , and  $k$  a positive integer. The k-shortest paths between  $s$  and  $t$  can be found in  $O(k(m+n \log n))$  [9] [10].  $k$  can not exceed the degree of  $s$  and  $t$ . Therefore in at most  $m$  iterations, the k-shortest paths with minimum path length can be found.

Due to the path constraint, the problem needs to be reformulated as a multicommodity-flow problem [1]. Let  $W$  be the set of all OD (Origin-Destination) pairs. To a given OD pair  $w$  is associated a traffic demand  $d_w$ .  $P_w$  represents the set of all shortest paths for this OD pair with each path  $p \in P_w$  carrying a traffic flow  $x_p$ . Let  $f_i$  be the flow originated by the OD pair  $i$  and  $f$  the aggregate traffic flow. For each OD pair  $w$ , we therefore have the following constraints:

$$\begin{cases} \sum_{p \in P_w} x_p = d_w, & \forall w \in W \\ x_p \geq 0, & \forall p \in P_w, w \in W \\ f(u, v) = \sum_{1, \dots, k} f_i(u, v) \end{cases}$$

The optimal routing problem as previously described can therefore be formally expressed as:

$$\begin{cases} \text{Minimize} & c^T X \text{ subject to:} \\ \sum_{v \in V} f_i(u, v) = \sum_{v \in V} f_i(v, u) & \forall u \in V \setminus \{s_i, t\} \\ \sum_{v \in V} f_i(s_i, v) = \sum_{v \in V} f_i(v, s_i) + d_i & \\ \sum_{u \in V} f(u, t) = \sum_{u \in V} f(t, u) - \sum_{i=1, \dots, k} d_i & \\ I_{uv} \leq \alpha & \forall (u, v) \in E \end{cases}$$

where  $X = [f_1(1, 2), \dots, \alpha]$ , and  $c^T = \underbrace{[0; 0; \dots; 0; 1]}_{2km}$ .

### D. Problem Extension with Path Lengths Constraints and Active Links Only

This problem is a combination of the two problems described in the previous subsections. For each source, the feasible paths are restricted to those respecting the path length constraints. In the optimization problem, only the links carrying a strictly positive traffic flow can be eventually selected as most congested links. The implementation details are given in Algorithm 3.

## V. EVALUATIONS

### A. Simulation Setups

In order to evaluate the performance of our algorithms, we ran a large number of simulations for different network configurations. The parameters that we modified are:

**Algorithm 3** Minimize the Maximum Interference for Maximum Network Utilization with Bounded Number of Hops

```

1: Calculate the interfering set for each link
2: Calculate the k-shortest simple paths with minimum length
   for each source to the destination
3: Calculate the optimal load assignment that yields a capacity
   ratio CapacityRatio
4: CurrentCapacityRatio=CapacityRatio
5: while true do
6:   Find the most interfering links
7:   for each of these interfering links do
8:     Remove the link from the connectivity graph
9:     Calculate the optimal load assignment
10:    if CapacityRatio ≤ CurrentCapacityRatio then
11:      CurrentCapacityRatio = CapacityRatio
12:    end if
13:  end for
14:  if CurrentCapacityRatio ≥ CapacityRatio then
15:    Return traffic flow matrix
16:  end if
17: end while

```

- the total number of nodes
- the traffic demand matrix

The network topologies are randomly generated in an area of 1000x1000. All nodes have equal transmission range set to 20 and the interference range is assumed to be of the same order as the transmission range. Although that to assume a uniform signal propagation is not a realistic assumption, it does not affect our results as it only concerns the way adjacency matrices and interference matrices are computed. The test network topologies we used are generated without any additional constraint than to ensure network connectivity (and that therefore there exists a path between any pair of nodes).

Link capacities are randomly chosen between 1 and a maximal capacity  $C_{max}$  such as the maximum traffic demand in  $k$  times smaller than the maximal capacity (in order to favor the appearance of congestion points when the network grows).  $k$  has arbitrarily been set to 50.

Simulations have been performed 50 times for each configuration set.

In these simulations, we primarily tried to establish:

- 1) The influence of the network size on the interference level with a constant percentage of sources generating some traffic load.
- 2) The influence of an increasing traffic demand on the interference level for a fixed number of nodes.

For simplicity, we refer to the schemes we implemented by:

- 1) *MLU*: Algorithm minimizing the maximum link utilization
- 2) *MLU-A*: Algorithm minimizing the maximum utilization with active links only

- 3) *MLU-PC*: Algorithm minimizing the maximum utilization with paths length constraints
- 4) *MLU-APC*: Algorithm minimizing the maximum utilization with active links only and path length constraints
- 5) *SP*: Shortest path routing strategy

*B. Influence of Network Size*

Figure 8 depicts the variation of the link utilization when the number of nodes varies from 10 to 45. The percentage of sources is set to 33% of the total number of nodes. It is worth noting that we do not constrain the link utilization level to be less than 1 in order to differentiate the degrees of congestion experienced by the network in the different scenarios considered.

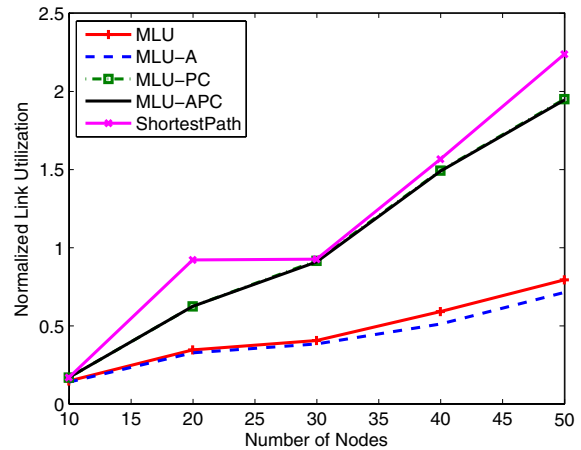


Fig. 8. Normalized Link Utilization vs. Network Density

We can observe in Figure 8 that the five algorithms perform as expected. *MLU* and *MLU-A* outperform *MLU-PC* and *MLU-APC* due to the path constraints that prevent the use of longer paths less prone to congestion. The shortest path routing algorithm, without the use of load balancing strategy, leads to higher link utilizations. Compared to *MLU-PC* and *MLU-APC*, the shortest path routing strategy does not perform as poorly as one could expect as the topologies considered initially are very small (10 nodes). Therefore the number of possible shortest paths for each source-destination pair is limited. But we can observe that as the network grows, the difference in performance becomes more significant. The advantage of the optimization introduced with *MLU-A* appears more distinct compared to *MLU* as the network size increases.

Table I reports the standard deviation for the simulations sets performed. As previously mentioned, we ran 50 simulations for each scenario. *MLU-PC*, *MLU-APC* and *SP* are very sensitive to the topology, and if it appears that a link of low capacity is on the shortest path to the destination, the level of congestion increases drastically.

Nb Nodes	MLU	MLU-PC	MLU-A	MLU-APC	SP
10	0.1947	0.1947	0.3697	0.3697	0.3690
15	0.3548	0.3395	0.7615	0.7615	0.7601
20	0.2205	0.2205	0.4373	0.4374	0.4295
25	0.2806	0.2791	0.5327	0.5346	0.5753
30	0.7833	0.7846	0.9984	0.9984	0.9759
35	0.3220	0.3222	0.8239	0.8249	0.9312
40	0.4435	0.4408	0.9623	0.9627	1.0336
45	0.5190	0.5178	1.1493	1.1439	1.1768

TABLE I  
STANDARD DEVIATION: INFLUENCE OF THE NETWORK SIZE

Src (%)	MLU	MLU-PC	MLU-A	MLU-APC	SP
2	0.0367	0.0367	0.3067	0.3067	0.3059
6	0.0856	0.0857	0.2236	0.2236	0.4442
10	0.1328	0.1298	0.9291	0.9334	1.2065
14	0.2883	0.293	0.7444	0.7444	0.8084
18	0.1188	0.1166	0.5813	0.5813	0.5423
22	0.9794	0.9855	1.3435	1.344	1.2392
26	0.5916	0.5953	1.1136	1.1136	1.2973
30	0.8928	0.8788	1.3679	1.3679	1.3266

TABLE II  
STANDARD DEVIATION: INFLUENCE OF THE TRAFFIC DEMAND

### C. Influence of Increasing Traffic Demand

We then study the impact of an increasing traffic demand on the performance of the algorithms. We consider a network topology of 30 nodes, and progressively increase the percentage of sources to up to 50% of the total number of nodes.

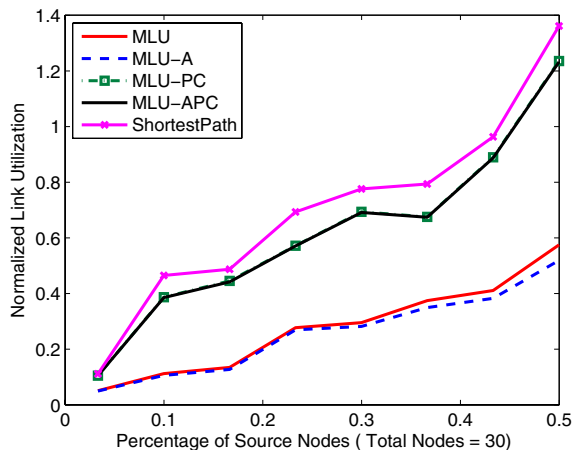


Fig. 9. Normalized Link Utilization vs. Number of Source Nodes

Figure 9 depicts the normalized link utilization averaged over 50 simulations runs with an increasing number of source nodes. We can observe that although the relative performance of the algorithm remains the same, *MLU* (and therefore *MLU-A*) exhibits a more stable behaviour and less sensitivity to an increasing traffic load as a result of a better load repartition. *MLU-PC* and *MLU-APC* rapidly suffer from congestion and we can observe that their performance progressively degrades as the number of source nodes increases.

The standard deviation reported in Table II supports the con-

clusion drawn previously and illustrates the lack of adaptability of pure shortest paths schemes to congestion situations.

## VI. CONCLUSIONS AND FUTURE WORK

Multi-hop wireless networks are gaining a growing importance in the technological landscape. This trend will undoubtedly become more and more significant as transmission speeds increase and mechanisms enhancing network reliability are developed. However, this vision might be compromised if proper routing schemes are not implemented. The performance degradation resulting from non-optimal traffic allocation strategies can waive the benefits of any technological advance.

In this work, we focused on a specific problem: how to enhance the nominal capacity (and therefore reduce the overall interference) in wireless multi-hop networks with multiple sources and a single destination. We demonstrated that with a proper flow assignment, significantly more traffic can be injected into the network than with a multipath or shortest-path routing strategy. Although many variants of these routing algorithms have been proposed and can be compared with, our aim was to provide some upper and lower bounds to the link utilization that can be reached under various scenarios. This highlights the benefits of exploiting the salient characteristics of the network (presence of a centralized administration) to distribute more efficiently the traffic load for a more efficient network utilization.

In future works, we intend to extend our study to the multiple frequencies scenario, and to provide solutions taking into consideration other performance criteria such as end-to-end delay or overhead cost. We also aim at implementing the flow allocation strategies presented in this paper in a real testbed for a more complete evaluation of the advantages/drawbacks of each routing strategy.

## REFERENCES

- [1] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. Mc Graw Hill, 2001.
- [2] D. Eppstein. Finding the k-shortest paths. *SIAM Journal on Computing*, 28(2), 1998.
- [3] C. C. Gonzaga. An algorithm for solving linear programming problems in  $O(n^3)$  operations. *Progress in Mathematical Programming*, pages 1–28, 1988.
- [4] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.
- [5] J. Hersherberger, M. Maxel, and S. Suri. Finding the k shortest simple paths: A new algorithm and its implementation. In *5th Workshop on Algorithm Engineering and Experiments (ALENEX03)*, 2003.
- [6] Jangeun Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5):8–14, Oct. 2003.
- [7] D. Spielman and S.-H. Teng. Smoothed analysis of termination of linear programming algorithms. In *In Proceedings of the 18th International Symposium on Mathematical Programming*, volume 97 of *B*, July 2003.
- [8] P. M. Vaidya. An algorithm for linear programming which requires  $O((m+n)^2 + (m+n)^{1.5}n)$  arithmetic operations. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, 1990.
- [9] J. Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, (17):712–716, 1971.
- [10] J. Y. Yen. Another algorithm for finding the k shortest loopless network paths. In *In Proceedings of 41st Mtg. Operations Research Society of America*, volume 20, 1972.