A Distributed Algorithm for Gateway Load-Balancing in Wireless Mesh Networks

Juan J. Gálvez
Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia, Spain
Email: jjgalvez@um.es

Pedro M. Ruiz
Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia, Spain
Email: pedrom@dif.um.es

Antonio F. G. Skarmeta
Department of Information and Communications Engineering
Computer Science Faculty
University of Murcia, Spain
Email: skarmeta@dif.um.es

***Abstract*—*Wireless Mesh Networks* (WMNs) provide a cost-effective way of deploying a network and providing broadband Internet access. In WMNs a subset of nodes called *gateways* provide connectivity to the wired infrastructure (typically the Internet). Because traffic volume of WMNs is expected to be high, and due to limited wireless link capacity, gateways are likely to become a potential bottleneck. In this paper, we propose a distributed load-balancing protocol where gateways coordinate to reroute flows from congested gateways to under-utilized gateways. Unlike other approaches, our scheme takes into account the effects of interference. This makes it suitable for implementation in practical scenarios, achieving good results, and improving on shortest path routing. Also, it is load-sensitive and can improve network utilization in both balanced and skewed topologies. Simulation results prove the effectiveness of our approach, which outperforms all schemes tested. We have observed throughput gains of up to 80% over the shortest path algorithm.**

I. Introduction and Motivation

A new kind of wireless multi-hop network architecture called *Wireless Mesh Networks* (WMNs) has recently attracted much attention. WMNs are comprised of mesh routers and mesh clients. Mesh routers are generally static (or quasi-static) in nature and are interconnected by wireless links. They serve as the infrastructure and provide connectivity to mesh clients. Typically, a subset of routers have direct connectivity to a fixed infrastructure (e.g. a wired network such as the Internet) and serve as *gateways* to the mesh nodes. WMNs provide a cost-effective way to deploy a network and offer services such as Internet connectivity.

Gateway nodes are a key component of WMNs. In many applications of WMNs most traffic will be directed to/from gateways. Thus, traffic aggregation occurs in the paths leading to a gateway, which can lead to congestion. One important consideration is the strategy employed to associate nodes with a particular gateway, i.e. through which gateway does a node send/receive traffic? We refer to the set of nodes served by a gateway as a *domain*.

Past work has focused on finding high throughput paths in wireless multi-hop networks. To this end, various routing metrics such as ETX [1], ETT [2] and MIC [3] have been proposed. Typical shortest path routing using hop-count or any of the above metrics can lead to load imbalance. Because these metrics are load-agnostic, shortest path routing can lead to situations where a few gateways are overloaded and others are under-utilized. Several factors can lead to this situation. For example, heterogeneous traffic demands, uneven number of nodes served by gateways (motivated for example by unplanned gateway placement), or gateway failures. Load imbalance can lead to inefficient use of network capacity and throughput degradation.

Because the traffic volume in a WMN is expected to be very high, load-balancing becomes important. We address this problem by rerouting traffic from congested to uncongested domains, achieving better capacity utilization and higher network throughput. We have developed a distributed scheme where gateways coordinate and exchange information about nodes and their demands. This enables them to jointly choose a route from the wired network to each active node. When congestion occurs in a domain due to high load, flows are rerouted from the congested domain to another domain. Our scheme is simple and effective. It doesn't introduce overhead inside the WMN and doesn't require computation at mesh routers. Message exchange is done between the gateways via the fixed infrastructure. Unlike some existing protocols, our scheme takes into account the effects of interference, and this enables it to outperform other solutions. Also, it is sensitive to the instantaneous load of the network, so it can achieve improvement in both balanced and skewed topologies.

We analyze performance of our scheme with *ns-2* and compare it with the algorithms presented in [4]. Our scheme outperforms all algorithms as well as the shortest path routing case (where nodes associate with the nearest gateway in terms of the routing metric). We show that most of the algorithms of [4] perform very poorly in real scenarios because they don't take contention into account.

The rest of the paper is organized as follows. In section II we review related work. In section III we describe the network model with a formal problem statement. Section IV explains our gateway load-balancing protocol. Section V shows and analyzes protocol performance based on simulations with *ns-2*. Finally, in Section VI we summarize our conclusions and discuss future work.

II. Related Work

There are a number of works concerning the problem of load-balancing in WMNs. Various solutions are proposed in [4], which focus on evening the load of all gateways. Most of these don't take into account the effects of contention and interference of flows and perform poorly in practice, as we will show in this paper. In WCETT-LB [5] and AODV-ST [6], extensions are made to existing solutions to allow nodes to switch to alternate gateways when their default gateway is congested. In [7] and [8], authors propose load-balancing schemes with similar objectives. Most of these can suffer from route flapping or poor performance for not taking contention into account. Bejerano et al. study the problem of load-balancing in mesh networks in [9]. The load-balancing algorithm is executed in a centralized point outside the mesh network, using a interference-free graph-theoretic model. Several studies use balanced trees rooted at the gateways and route traffic along the tree paths. Hsiao et al. [10] propose a distributed algorithm to find a fully top load-balanced tree, using a interference-free model. Raniwala et al. propose the *Hyacinth* architecture for multi-radio multi-channel WMNs in [11]. Routing and channel assignment is done distributively and dynamically. The performance results are mainly concerned with the increase in throughput due to the use of multiple channels and the different channel assignment strategies. Authors in [12] seek a delay-optimal routing forest, where a tree is rooted at each gateway. The cost function is not load-dependent, therefore their scheme won't achieve appreciable gains over shortest path routing in balanced topologies.

III. Network Model

We consider WMNs which comprise of wireless static or quasi-static mesh *routers*, also called *nodes*. These nodes form a wireless multi-hop network. Mesh *clients*, also called *users*, connect to the mesh routers. A subset of nodes, referred to as *gateways*, are directly connected to a fixed infrastructure, which we will assume is the Internet for the rest of the paper. Each router has one radio interface (e.g. 802.11b or g) for communication with other routers, using a single common channel. Communication between nodes and users is done via a separate interface (wired or wireless).

Although intra-WMN communication is possible, we assume that most of the traffic will be directed to the Internet. A user can access the Internet through one or more links leading from its router to a gateway.

A. Problem statement

Let G be the set of gateways in the WMN. A *sink* is a router that receives Internet traffic. Traffic directed to a sink must be served by a gateway node. The choice of serving gateway for each particular sink must be made by the load-balancing protocol. A *domain* d_i is the set of sinks served by gateway GW_i. Fig. 1 shows an example WMN.

A set D of domains constitutes a solution to the problem. Let us call *nearest gateway* (NGW) the solution adopted by

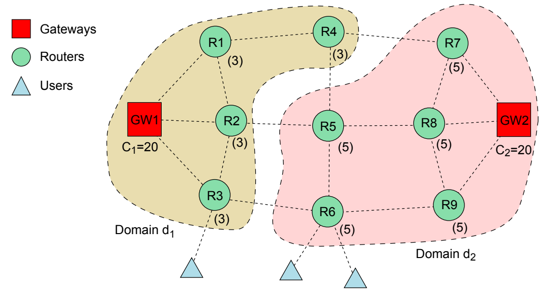

Fig. 1. Example WMN. There are two gateways GW_1 and GW_2, with their corresponding domains d_1 and d_2. All routers are acting as sinks, with their demand shown in parentheses. The capacity of each domain is 20.

shortest path routing using a particular metric (e.g. hop-count). With NGW all sinks are associated with their nearest gateway in terms of the routing metric. Fig. 1 shows a possible NGW solution using hop-count. NGW can result in domains which are congested while others are not. In the example, if we assume that the capacity of both domains is 20, d_2 will be overloaded (its load is the sum of demands of its sinks, i.e. 25). While the load of d_1 is 12. If, for example, the traffic of $R6$ is rerouted to d_1 both domains would be uncongested: the loads of d_1 and d_2 would be 17 and 20, respectively.

The load of a domain is the sum of demands of its sinks:

$$load(d) = \sum_{s \in d} demand(s) \tag{1}$$

The overload of a domain d is given by:

$$overload(d) = \begin{cases} 0, & \text{if } load(d) < C_d, \\ load(d) - C_d, & \text{otherwise.} \end{cases} \tag{2}$$

where C_d is the capacity of domain d. We seek D that minimizes:

$$OV(D) = \sum_{d_i \in D} overload(d_i) \tag{3}$$

The objective is not to balance load in the sense that all domains carry approximately the same, but to minimize the excess load of congested domains, ideally removing it. In this paper we maintain that, as long as the overload of the domains is minimized, it is preferable to maintain a solution as close as possible to the NGW solution, even if load imbalance persists. Load should not be balanced arbitrarily between gateways with the only objective of evening the load between them. Contention plays a major role in multi-hop wireless networks and can severely degrade performance. The capacity of a chain of nodes and the locality of traffic are key factors influencing network capacity [13], [14]. If flows are routed along very long paths, or interference between flows is not taken into account, throughput can be much lower than expected. For example, in Fig. 1 it is obvious that assigning $R8$ to $GW1$ and $R5$ to $GW2$ is not a good decision, because it will increase contention in most of the network. In summary, domains will mostly resemble the NGW solution, because this maintains traffic locality. If congestion occurs, some sinks, preferably border sinks (sinks close to the border between domains), will switch from their nearest gateway to another, preferably close. Simulation results of section V validate this claim.

Our algorithm defaults to the NGW solution if it results in uncongested domains, i.e. if $OV(NGW) = 0$. From the above discussion we conclude that, even if domains don't carry the same load, if they are under-utilized, using the NGW will be the best solution. If all domains with NGW are overloaded, the algorithm also returns this solution, because no solution with $OV(D) < OV(NGW)$ exists in this situation. Note that not all solutions are considered feasible. From the above discussion, we can deduce that there is a cost associated with rerouting flows that must not be exceeded.

The capacity of a domain d_i is estimated as the average aggregate throughput achievable inside the domain if d_i were obtained using shortest path routing. In other words, it is the capacity of the region that surrounds a gateway, encompassing all sinks that are nearest to this gateway. For this paper, we will assume that a network administrator has determined the capacity of each domain.

The association of sinks to gateways is done only for the purpose of receiving traffic from the Internet (i.e. download traffic). Traffic sent by nodes to the Internet will be routed to the nearest gateway. This way, all knowledge of sink-gateway association and decisions can be kept at the gateways.

IV. Gateway Load-balancing protocol

The load-balancing protocol is a distributed protocol where gateways exchange information about their associated sinks and their demands. Gateways communicate with themselves through the wired network. Every time a demand changes, the serving gateway notifies the rest of the gateways. And all gateways reapply the load-balancing algorithm GWLB, thus calculating a new set of domains D. Download traffic is routed based on this solution, i.e. a serving gateway is responsible for injecting the traffic of its sinks inside the WMN. To derive the demand of a sink, a gateway measures the throughput of flows destined to it. Since download traffic passes through the gateway, it can measure the throughput of the flows.

To execute the load-balancing algorithm, gateways must know the distance of every node to every gateway. Distance is measured as the length of the shortest path, with respect to a certain routing metric (e.g. hop-count). Since in a WMN nodes are typically static (or quasi-static) this knowledge can be obtained easily by the gateways. Routing inside the WMN should be performed by shortest path routing using the same metric. This means that when packets are injected inside the network by a gateway, those packets follow the shortest path to reach their destination. We do not impose any particular routing protocol.

A. The load-balancing protocol

Each gateway has the list D of domains in the WMN. $D = \{d_i\}$ where d_i is the set of sinks served by $GW_i \in G$.

The following protocol is executed at each gateway GW_i:

At gateway startup

1) GW_i advertises itself to each gateway GW_j.
2) GW_j sends GW_i the demands of sinks in d_j.
3) All gateways apply the GWLB algorithm.

During gateway uptime

All gateways apply GWLB when:

- A new sink appears. The gateway that detects it informs all gateways of its demand.
- The demand of a sink in d_i changes. GW_i informs all gateways of the new demand.
- A gateway shuts down or ceases to function.

Traffic reception

- When a gateway receives traffic not destined to a sink in its domain, the gateway forwards it to the gateway calculated in the solution (via a tunnel through the wired network). If there is no domain calculated yet for the sink, the gateway sends the traffic to the sink's nearest gateway (by tunnel).
- If the traffic is destined to a sink in its domain it routes it through the WMN via the shortest path.

Note that this protocol is distributed and adapts to the gateways present in the network at any time. If for example a gateway fails, flows of its sinks will be rerouted through the remaining gateways. Another advantage is that it doesn't depend on a centralized control point.

B. GWLB: Gateway Load-Balancing algorithm

Gateways execute Algorithm 1 to calculate D. The algorithm is heuristic and deliberately kept simple in order to reduce execution time at the gateways. Because the algorithm is executed periodically and adapts to load changes, execution must be kept fast. Therefore, it is not guaranteed to find the optimal solution that minimizes eq. 3, but we will see in section V that it offers good results.

GWLB starts by assigning all sinks to their nearest gateway. As explained in III-A, it is important to build upon the NGW solution. Next steps consist in trying to reroute flows from an overloaded domain d_1 to an uncongested domain d_2 such that the *overload* of both domains is reduced. If a domain is overloaded, its sinks are checked in descending order of distance to their serving gateway. This is done to give preference to border sinks. The farther a sink is from its serving gateway the less it will harm other flows of its domain if it is rerouted. And its path to other domains will be shorter, thus improving performance. For the same reason, when a sink is chosen, domains are checked in ascending order of distance to the sink. Next, to perform the switching of domains, the overload after the switch must be less than the overload before the switch (lines 13-15). Lastly, the cost of switching is checked. nGW_s is the gateway nearest to s. Only if the cost is less than the switching threshold Δ_s will it be performed (line 16). This rule takes into account the existence of contention, because it prevents the establishment of long paths, which suffer from intra-flow interference and increase

Algorithm 1 GWLB: Gateway load-balancing algorithm

for each gateway GW_i **do**
  d_i := {} // domain of GW_i is empty

for each sink s **do** // Add sinks to nearest domain
  Add sink s to d_i where $\min_i\{distance(s, GW_i)\}$

for domain d_1 in D **do**
  if $load(d_1) > C_{d_1}$ **then** // domain is overloaded
    for sink s in d_1 **do** // iterate in descending order of distance to GW_1
      for domain d_2 in D **do** // iterate in ascending order of distance to s
        if $d_1 = d_2$ **then**
          continue
        $ovl_{before} := overload(d_1) + overload(d_2)$
        $ovl_{after} := overload(d_1 - \{s\}) + overload(d_2 \cup \{s\})$
        if $ovl_{after} < ovl_{before}$ **then**
          if $distance(s, GW_2)/distance(s, nGW_s) < \Delta_s$ **then**
            $d_1 := d_1 - \{s\}$
            $d_2 := d_2 \cup \{s\}$
            break
      if $load(d_1) <= C_{d_1}$ **then**
        break

inter-flow interference in the network, and gives preference to border sinks.

Some remarks on the algorithm: (i) GWLB defaults to NGW solution when it results in no domains being overloaded or all domains being overloaded; (ii) The problem of *route flapping* does not occur with GWLB, because all gateways generate the same solution; (iii) The protocol converges quickly. As soon as gateways exchange information and apply GWLB, the solution converges. Because gateways communicate via the wired network and the execution time of GWLB is low, the protocol adapts quickly to the current network situation; (iv) GWLB is load-dependent, and can find solutions better than NGW even in topologies that are balanced. In balanced topologies load imbalance can still occur due to heterogeneous traffic demands.

V. Performance Evaluation

We have used the ns-2 [15] simulator. Our goal is to compare the performance of GWLB with the NGW solution and with the load-balancing algorithms in [4]. Simulations run for 500 seconds. The MAC layer employs 802.11b with a data rate of 11Mbps. We have generated ten random topologies. They consist of 100 static nodes placed in a 2000 meter x 2000 meter area. Of these, five act as *gateways* and the rest as mesh *routers*. Gateway placement is fixed for all topologies: there is a gateway in each corner of the square area and one in the center. Mesh routers are placed randomly, but with the following restrictions: there is a minimum distance of 160 meters between every pair of nodes, and the generated topology must be connected.

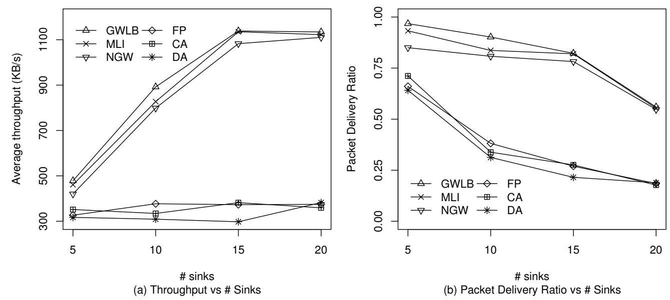

Fig. 2. Performance comparison of load-balancing protocols.

Gateways are connected to the wired network (Internet) by 100Mbps links. A subset of routers act as *sinks*. Traffic is generated at a Internet server and sent to a varying number of randomly chosen sinks in the WMN, between 5 and 20. There is one flow per sink. Flows start after the first 50 seconds of simulation and stop 5 seconds before the simulation ends. Flows are CBR/UDP and their rate roughly follows a log-normal distribution with $\mu = \log(67)$ and $\sigma = 0.4$. With this distribution half of the flows will be in [0,67] KB/s interval and half of them in [67,200] KB/s. For lack of an accepted rate model for download traffic in WMNs, we choose to use a log-normal distribution, as studies [16] suggest that the rate of Internet flows follows this distribution. Note that this distribution generates heterogeneous traffic demands.

Routing is static for all the algorithms tested. Traffic is routed through the shortest path (using hop-count) from gateways to sinks. Because we are only interested in evaluating the effectiveness of the load-balancing approach of the algorithms, we eliminate the overhead of routing protocols.

We have tested and compared GWLB, NGW and the following protocols of [4]: Minimum load-index (MLI), Centralized assignment (CA), Distributed assignment (DA) and Fully probabilistic (FP). We have implemented these protocols in ns-2 as close as possible to their descriptions, but emulated them in a centralized manner, eliminating message exchange in the WMN (e.g. gateway advertisements). Because we have eliminated control overhead, these protocols should in fact behave better than their actual specification. The switching threshold of GWLB is $\Delta_s = 1.8$. For the other protocols, we have used the same parameters as in [4].

We compare the network throughput and packet delivery ratio of all schemes. In the figures shown each point represents an average of ten runs (corresponding to the ten different topologies). Traffic pattern and sinks are the same across all protocols tested.

Fig. 2 presents the results of the simulations. As we can see in Fig. 2 (a), GWLB outperforms all other schemes. FP, CA and DA perform very poorly. This is because contention is not taken into account. In CA and DA, sinks associate with gateways considering only their load. They don't take into account their distance to gateways or interference with other flows. In FP, a sink receives its traffic simultaneously from multiple gateways (traffic is split proportional to gateway capacities and sent to the sink). This translates into five times more flows inside the network in our simulations, because there are five gateways. This, as well as the length of paths, greatly increases contention.

MLI builds upon the NGW solution, improving it. With MLI, every node is initially associated with its nearest gateway. Periodically, border nodes switch to another domain to even the load of the domains. The service range of gateways is extended at most one hop each time. As we have stated throughout the paper, due to the potential risks of switching a sink from its nearest gateway to another, switching should only occur if overload can be minimized. And even then, the negative effect of switching has to be taken into account. Because MLI switches every time there is an imbalance between gateways, it can generate poor solutions. Still, on average, it improves NGW. Another drawback of MLI is that convergence time will depend on the frequency of gateway advertisements, which we have emulated by applying MLI every second. Also, note that because MLI relies on gateway advertisements, it should perform worse than shown, due to the overhead of these messages. GWLB outperforms MLI because it only switches to minimize overload, i.e. when a domain is congested and others are not. And the switching threshold Δ_s avoids the negative effects of interference.

Fig. 2 (b) shows the packet delivery ratio of the protocols. Congestion builds up as the number of sinks increases. Note that, as the network becomes more congested, GWLB is more limited in the amount of switching it can perform. The less uncongested domains, the less options GWLB has. And if all domains are congested, it will default to the NGW solution. That is why the throughput and delivery ratio of GWLB and NGW converge as sinks increase. Also of note is the fact that GWLB only generates a different solution than NGW in 4 out of 10 cases with 5 sinks. This is because GWLB defaults to NGW if it results in uncongested domains, which occurs in most of these scenarios due to low traffic load. The average throughput gain with respect to NGW in these 4 scenarios is 48%, with a maximum gain of 80% in scenario 7.

VI. Conclusions and Future Work

Load-balancing between gateways in a WMN is important, where the limited wireless link capacity and nature of traffic flows make gateways a potential bottleneck. In this paper we propose GWLB, a distributed protocol where gateways exchange information about their associated nodes and their demands, and coordinate to balance the load among them. It achieves improvements in scenarios with load imbalance (in both balanced and skewed topologies) over shortest path routing. It is designed to be directly implemented in real scenarios, taking into account the effects of interference. The simulations conducted in ns-2 prove its effectiveness. It outperforms all schemes tested, and in particular, shortest path routing, which is the most extended approach for associating nodes with gateways.

As future work, GWLB can be further enhanced. Specifically, with an improved capability of measuring domain capacity and overload situations, the protocol can yield better results. Other possible extensions include applying GWLB to multi-radio mesh networks, and employing more advanced routing metrics such as ETX and ETT.

Acknowledgment

This work has been partially funded by excellence research group funds from the Spanish CARM and projects TIN2008-06441-C02-02 and TSI-020100-2008-354.

References

[1] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *MobiCom ’03: Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 134–146.

[2] R. Draves, J. Padhye, and B. Zill, “Routing in multi-radio, multi-hop wireless mesh networks,” in *MobiCom ’04: Proc. of conference on Mobile computing and networking*. ACM, 2004, pp. 114–128.

[3] Y. Yang, J. Wang, and R. Kravets, “Interference-aware Load Balancing for Multihop Wireless Networks,” University of Illinois at Urbana-Champaign, Tech. Rep.

[4] C.-F. Huang, H.-W. Lee, and Y.-C. Tseng, “A two-tier heterogeneous mobile ad hoc network architecture and its load-balance routing problem,” *Mobile Networks and Applications*, vol. 9, no. 4, pp. 379–391, 2004.

[5] L. Ma and M. K. Denko, “A Routing Metric for Load-Balancing in Wireless Mesh Networks,” in *AINAW ’07, Advanced Information Networking and Applications Workshops*, 2007, pp. 409–414.

[6] K. Ramachandran, M. Buddhikot, G. Chandranmenon, S. Miller, E. Belding-Royer, and K. Almeroth, “On the Design and Implementation of Infrastructure Mesh Networks,” in *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[7] D. Nandiraju, L. Santhanam, N. Nandiraju, and D. P. Agrawal, “Achieving Load Balancing in Wireless Mesh Networks Through Multiple Gateways,” in *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2006, pp. 807–812.

[8] B. Xie, Y. Yu, A. Kumar, and D. P. Agrawal, “Load-balancing and Inter-domain Mobility for Wireless Mesh Networks,” in *Global Telecommunications Conference, 2006. GLOBECOM ’06. IEEE*, 2006, pp. 409–414.

[9] Y. Bejerano, S.-J. Han, and A. Kumar, “Efficient load-balancing routing for wireless mesh networks,” *Comput. Netw.*, vol. 51, no. 10, pp. 2450–2466, 2007.

[10] P.-H. Hsiao, A. Hwang, H. Kung, and D. Vlah, “Load-balancing routing for wireless access networks,” in *INFOCOM 2001*, vol. 2, 2001, pp. 986–995.

[11] A. Raniwala and T.-C. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, 2005, pp. 2223–2234.

[12] V. Mhatre, H. Lundgren, F. Baccelli, and C. Diot, “Joint MAC-aware routing and load balancing in mesh networks,” in *Proc. of the 2007 ACM CoNEXT conference*, 2007.

[13] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Transactions on Information Theory*, pp. 388–404, 2000.

[14] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” in *MobiCom ’01: Proc. 7th ACM Int. Conf. on Mobile Computing and Networking*, 2001.

[15] K. Fall and K. Varadham, “The ns Manual,” http://www.isi.edu/nsnam/ns/ns-documentation.html.

[16] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, “On the characteristics and origins of internet flow rates,” in *SIGCOMM ’02: Proc. of conference on Applications, technologies, architectures, and protocols for computer communications*, 2002.