Scalable Max-Min Fairness in Wireless Ad Hoc Networks

Congzhou Zhou and N.F. Maxemchuk

Department of Electrical Engineering
Columbia University
New York, USA
cz2110@columbia.edu, nick@ee.columbia.edu

Abstract. Our previous work proposes a macro model to perform flow and access control in wireless ad hoc networks. In this paper, we demonstrate specifically how to apply the model to achieve max-min fair rate allocation. Our proposed scheme is simple and scalable when comparing to other techniques in the literature. Moreover, it has the ability to provide stability in mobile environment. Simulation results show that our new method provides a good max-min fair flow assignment, and with that assignment, quality of service guarantees can be achieved for real-time applications.

Keywords: macro model; max-min fairness; scalability; quality of service.

1 Introduction

In this paper, we consider the fair rate allocation problem in wireless ad hoc networks. More specifically, we propose a simple and scalable flow control algorithm that can achieve a max-min fair [1] rate allocation. With that rate allocation, we can maximize the network utilization while maintain fairness to each end users. Since the network resource is properly managed, Quality of Service (QoS) guarantees can be provided. This extends the range of possible new services, such as Voice over IP, that can be used on wireless ad hoc networks.

Max-min fairness is one of the most widely used fairness notions, and hence has been studied in both wired [1], [2], [3], and wireless networks [4], [5], [6], [7], [8], [9]. For example, [6], [7] present algorithms for achieving max-min fairness at the link layer in Aloha random access networks; [8] introduces collision domain to calculate the max-min fair capacity in wireless mesh networks; and [9] proposes an algorithm to maximize the network utilization while ensuring a slightly *weaker* notion of max-min fairness in wireless sensor networks.

Our proposed scheme differs from them because our method uses a macro model [10], [11] to control sets of flows only across naturally occurring bottlenecks in the network. Unlike other schemes that apply max-min fairness at every node for all flows, our method concentrates on the critical junctions in the network, and allows us to control a much larger number of flows even in areas where individual nodes are moving. Because our method does not depend on the number of nodes in the network, it scales better than the previous techniques. In addition, our method can reserve some

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 79–93, 2010.

capacity to cope with nodes' mobility in the network at the cost of network utilization. Therefore it is more stable than the previous techniques in mobile environment. The stability of our technique will be investigated in a future paper.

The rest of this paper is organized as follows. Section 2 briefly describes our macro model. Section 3 shows how to apply our macro model to achieve max-min fairness in wireless ad hoc network. We compare our proposed method to other methods for max-min fairness in the literature, and demonstrate the scalability of our approach in section 4. Section 5 presents simulation results to highlight the performance of our approach. Finally, conclusions and future work are presented in section 6.

2 The Macro Model

The topologies of real networks are not uniform. Instead, they have naturally occurring bottlenecks where a large number of flows must squeeze through a narrow area. This observation provides the incentive for constructing our macro model to perform flow control in wireless ad hoc networks. If we control the flows through these areas to meet a quality of service objective, the flows in the rest of the network are lower and will also meet this objective.

In [10], we present our macro model to perform flow control and access control in mobile ad hoc networks. Our objective is to determine the bottlenecks that are likely to constrain flows and the capacities of those bottlenecks. To construct the macro model, the geographic area of the network is partitioned into virtual grids first. We then approximate obstacles on the grid structure for its simplicity and stability in mobile environment. A network partition rule [11] is introduced to identify bottlenecks in the network and to form super-nodes that are surrounded by obstacles and bottlenecks. Therefore, a wireless network can be transformed into a collection of super-nodes and bottlenecks that resemble a wired network. Flow control techniques and other control algorithms that are designed for wired networks can be applied to wireless networks directly through our macro model. Our previous work shows that the bottlenecks we identified are the vulnerable places in the network. Moreover, controlling the flows across those naturally occurring bottlenecks can avoid congestion and hence provide quality of service guarantees to real-time applications in multi-hop wireless networks.

To illustrate the process, we apply our macro model to the main portion of Columbia University Morningside campus, between Amsterdam Avenue and Broadway, as shown in Fig. 1. The network is divided into 15x17 virtual grids. In Fig. 2, there are 16 obstacles approximated on the grid structure, including the perimeter obstacle of the network. Our network partition procedure identifies 10 bottlenecks and forms 8 super-nodes. According to the network partition result, Fig. 3 shows the macro model of the network that resembles a wired network. Super-node N_i in the model represents all individual nodes that are located in super-node area A_i; and bottleneck-link B_i represents bottleneck B_i we identified in Fig. 2. [10], [11] provide more details for interested readers. In this paper, we show how to implement max-min fair flow control in wireless ad hoc networks through our macro model.

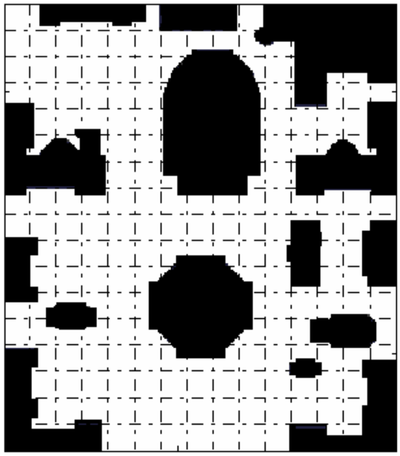

Fig. 1. Columbia University campus

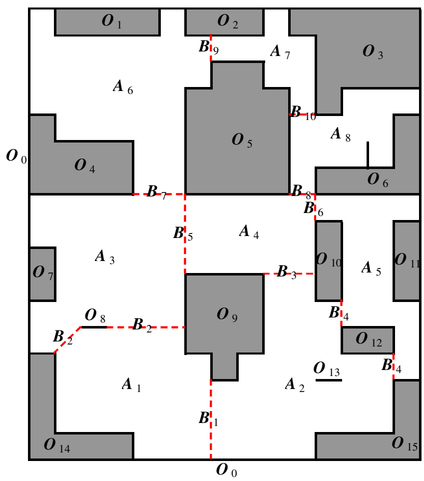

Fig. 2. Network partition

3 Applying the Macro Model to Max-Min Fairness

The algorithm to achieve max-min fairness for wired networks [1] can be applied to our macro model with a simple modification. The reason for the change is that in wireless networks flows not only consume bandwidth on the links they traverse but also consume bandwidth on links they interfere with. If we apply wired algorithm directly without considering interference, we over-assign bottlenecks' capacity and cause congestion in the network. Before describing the modification, let us first define two terms to be used later. When considering bandwidth consumption at a bottleneck, we need to consider flows that actually traverse that bottleneck (let's call them

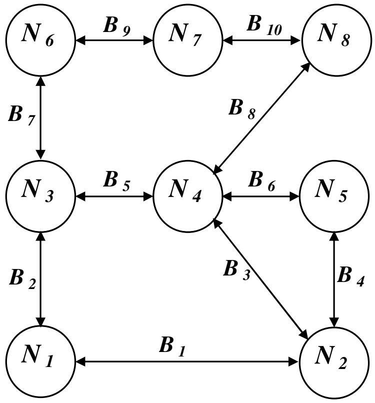

Fig. 3. Macro model

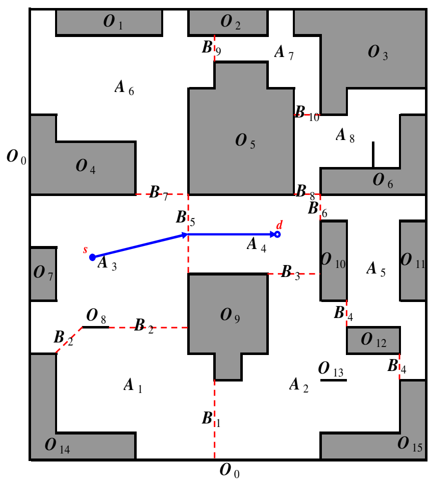

Fig. 4. An example flow

passing flows to the bottleneck). We also need to consider flows that do not traverse that bottleneck but interfere with the bottleneck (let's call these flows *interfering flows* to the bottleneck).

Now, instead of only putting passing flows on each bottleneck, we also put on interfering flows. This is the only change we need to make when applying wired max-min fair algorithm. Here is an example to show how to consider interfering flows when applying our macro model for max-min fairness. In Fig. 4, node s, in super-node N_3, starts a flow to node d, in super-node N_4. From this figure and our macro model in Fig. 3, we can see that this flow passes only one bottleneck, B_5. However, due to a shared wireless medium, the multi-hop transmission from s to d will consume some bandwidth on bottlenecks surrounding super-nodes N_3 and N_4. Therefore, in this case, we not only need to consider the bandwidth requirement on bottleneck B_5, but also the bandwidth consumption caused by interference on bottlenecks B_2, B_3, B_6, B_7, and B_8.

The bandwidth consumption on a bottleneck by *useful* transmission and *wasteful* interference can be calculated by using the contention counter as described in [12], [13]. The bandwidth consumption, B_c, of a flow at a bottleneck can be expressed as:

$$B_c = N_{ct} \times W \ . \tag{1}$$

where N_{ct} is contention count, which is the number of nodes on the route of the flow to have interference on the bottleneck, and W is the flow's rate allocation. With the bandwidth consumption calculated, we can put flows on the macro model and apply the wired technique to find max-min fair rate allocation in wireless ad hoc networks.

The above method is accurate but requires a lot of information to calculate the bandwidth consumption. Moreover, once nodes start to move in the network, the information, such as N_{ct}, will change and the bandwidth consumption needs to be re-calculated. The frequency of re-calculation increases as nodes' mobility increases. In this paper, we adopt a simple approach to estimate the bandwidth consumption. The simple approach also reserves some bottleneck capacity to cope with nodes' mobility at the cost of network utilization. In other word, the capacity of a bottleneck is not fully assigned to flows that traverse it and flows that interfere with it. Some of its capacity is assigned to flows that currently have no consumption on it but may have in the near future due to mobility.

In the simple estimation model, each passing flow requires 1 unit of bandwidth on the bottleneck; while an interfering flow consumes α unit of bandwidth of the bottleneck ($0 \leq \alpha \leq 1$). When α equals zero, we only consider flows that pass a bottleneck; while α equals one, we treat passing flows and interfering flows the same. When we set α to zero, we over-assign the capacity of a bottleneck and cause congestion in the network, and, when we set α to one, we reserve too much capacity on a bottleneck for interfering flows and decrease the network utilization. This is because interfering flows generally have smaller contention counter on a bottleneck than passing flows. There is a trade off between network utilization and model stability when selecting α value. In this paper, we simply set α to 0.5. How to choose α value for a better max-min fairness flow assignment bears further study.

Passing flows are easy to find. They traverse the bottleneck. Interfering flows, on the other hand, are harder to determine because they depend on the interference patterns of nearby transmissions. To cope with a nodes' mobility, we consider a flow that traverses a super-node an interfering flow to all bottlenecks surrounding that super-node. In this way, no matter where the node moves within the super-node, its flow assignment can still be supported. Using the above example, flow from s to d requires 1 unit of bandwidth on B_5 and also consumes α unit of bandwidth on B_2, B_3, B_6, B_7, and B_8. By reserving bandwidth on those bottlenecks, our max-min fair rate allocation does not change even when node s and node d move within their own super-nodes.

In the Columbia University network, the radio transmission range is about $\sqrt{2}$ times the grid size. This guarantees that a node can reach any other node in the same grid element if no obstacle is in between. The interference range is about twice the transmission distance. From Fig. 4, we can see that all bottleneck sizes are relatively small when compared to the interference radius. Therefore, the bottlenecks can probably only support one transmission at a time. In this paper the capacity of each bottleneck is the same. Our simulation results in section 5 show that this assumption is valid. Estimating the capacity for different sized bottleneck, and studying the effect of α on flow assignment, network utilization, and model stability are currently under investigation.

We apply the algorithm in [1], [3] or other flow assignment approaches for max-min fairness in wired networks to our macro model. We put fractions of the interfering flows on the appropriate bottlenecks and apply the max-min fairness algorithm as in a conventional wired network.

4 Comparison of Three Methods

In this section, we list and compare three methods to achieve max-min fair rate allocation in wireless ad hoc network. The first approach is an experiment model; the second one is an analytical model; and the third one is our proposed simple estimation model. Although our model provides a more conservative flow assignment than the other two models, it is the simplest approach among all three methods and scales well.

4.1 Experiment Model

In the experiment model, we implement experiments using the *progressive filling* algorithm presented in [1]. Similar approaches are used in [2] for wired networks and in [8] for wireless networks to ensure max-min fairness. We increase all flows with the same amount of capacity until one or more flows reach their maximum throughput. Then we increase each individual flow to see which one can be increased without decreasing other flows' throughput. This is where our progressive filling algorithm differs from that used in [8]. Since in wireless networks, it is hard to find the exact place of congestion and the exact flows that are constrained by the congested area, we use an experimental technique to determine which flows can be increased and which flows can not. Once we find the flows that can be increased, we increase them all with the same amount of capacity until one or more of those flows reach their maximum throughput. We repeat the process until all flows reach their maximum throughput. To acquire the max-min fair rate allocation, the amount of trials we need to experiment depends on the number of flows in the network.

4.2 Node-Path Model

In the node-path model, each node collects on-going transmission information from all other nodes that are within its interference range. Three methods have been proposed in [12] for that purpose. Once a pair of nodes that are involved in a one-hop transmission (let's call it a *path*) knows the number of on-going one-hop transmissions there are within their interference range, it can assign its capacity equally to all transmissions. More accurate model can be found in [4]. In this approach, we consider each path as a bottleneck and apply max-min fairness at every path for all flows. Therefore, the node-path model depends on the number of nodes as well as the number of flows in the network.

4.3 Estimation Model

In our estimation model, we apply max-min fairness only at bottlenecks. The number of bottlenecks is fixed for a certain network topology. Because our model does not depend on the number of nodes/flows in the network, it scales better than the estimation model and the analytical model described above. Moreover, since we reserve some capacity at the bottleneck for all possible future interfering flows, our model is more stable than the other two methods in mobile environment.

5 Simulation Results

We first validate our model by comparing it to the experiment results and the Node-Path model results in a 600-node and 20-flow case. This is a manageable scenario for the first two methods. We then apply our model to larger numbers of flows and larger numbers of nodes to demonstrate its scalability. Finally, we apply our model to a different network topology, the Time Square area in NYC.

We use the GloMoSim simulator [14] to validate our model and evaluate its performance. In the simulation scenarios, all nodes communicate with identical half-duplex wireless radios with a bandwidth of 11 Mbps. We adjust the radio transmission power, receiving threshold and sensing threshold to achieve a 19.4 meters of transmission radius and a 38.8 meters of sensing radius in the Columbia University network. The radio propagation model is the free-space attenuation model at this short distance. The MAC layer is IEEE 802.11 [15]. We pre-compute the shortest path routes for every flow in the simulations. Geographical routing techniques, such as [16], [17], [18], are considered in mobile environment in our on-going work. The packet size is set to 1024 bytes.

5.1 600-Node and 20-Flow

In this scenario, 600 nodes are randomly deployed in Columbia University network and 20 flows are selected randomly. Fig. 5 is a snapshot of that scenario. We use all three models to get the max-min fair rate allocation and use the results as inputs to the simulator. Fig. 6 to Fig. 8 shows our model is comparable to the other two models in terms of network performance.

Fig. 6 shows the different flow assignments for three models. We see that all three models find similar rates allocated to the smallest flows. As we move beyond the smallest flows there are differences. The experiment results are the optimal flow assignment. The node-path model results do provide consistent match to the experiment results. Our estimation model provides a more conservative flow assignment than the other two models. This is reasonable, since the first bottleneck saturated in the network not only fixes the flows passing through it, but also flows that might interfere with it at a later time. As a result, more flows are fixed at the smallest value in our model than the other two models.

Fig. 7 shows the delay experienced by all 20 flows in all three models. Since the network is operated at the maximum capacity, some flows have a delay of more than

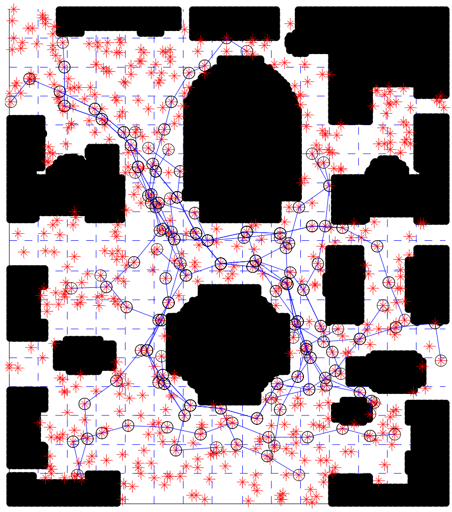

Fig. 5. A snapshot

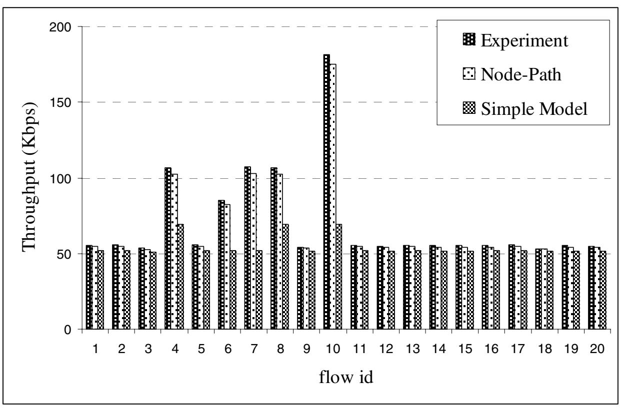

Fig. 6. End-to-end throughput for three models

0.2 second in the experiment model. The other two models all have delays that are less than 0.2 second, which is adequate for many real-time applications. This is because they have lower flow assignment than the experiment model. In general, a network will not operate at its full capacity. Therefore, the delay will be lower than that in Fig. 7.

Fig. 8 shows the packet delivery ratio for all three models. All three models have packet delivery ratio that is great than 95%. This is pretty good when operating the network at its full capacity. Fig. 9 compares the total end-to-end throughput and total

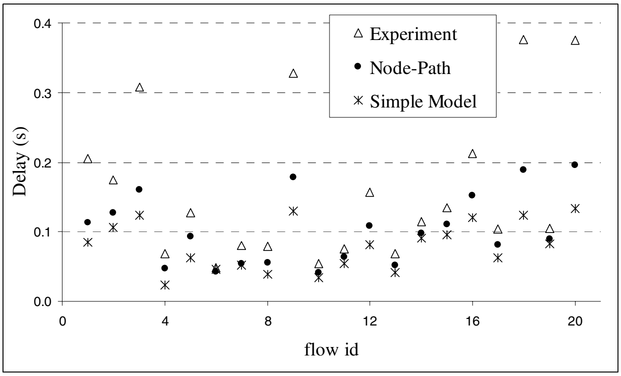

Fig. 7. Delay for three models

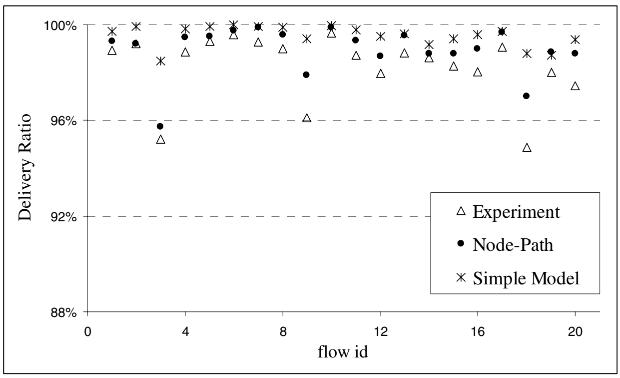

Fig. 8. Packet delivery ratio for three models

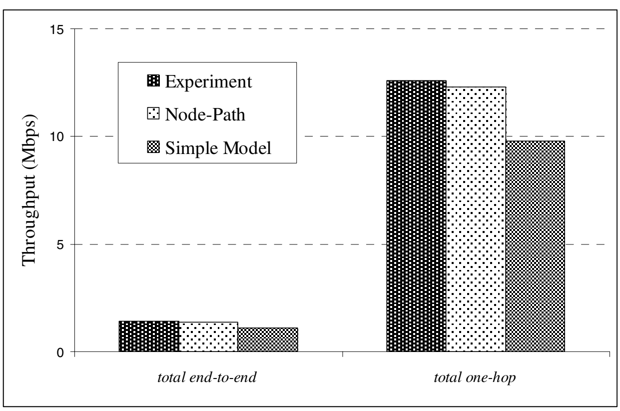

Fig. 9. Total end-to-end and total one-hop throughput for three models

one-hop throughput for all three models. The results are comparable. Our total one-hop throughput is 77.8% and our total end-to-end throughput is 76.98% of the experiment results. The remaining capacity in our estimation model is reserved for future flow violations in mobile environment. Therefore, we conclude that our simple estimation model provides a good max-min fair flow assignment in this scenario.

5.2 600-Node and Different-Flow

In this section we keep the number of nodes the same and increase the number of flows to 40 and 60. We only apply our estimation model because the other two models can not scale.

Fig. 10 and Fig.12 show the network performance for these flows. The 40-flow and 60-flow cases have a higher total end-to-end and total one-hop throughput than the 20-flow case. This shows that for 40-flow and 60-flow cases, our flow assignments do not under utilize the network capacity when compared with the 20-flow case [19], [20]. Although the 40-flow and 60-flow cases have a higher delay and a lower packer

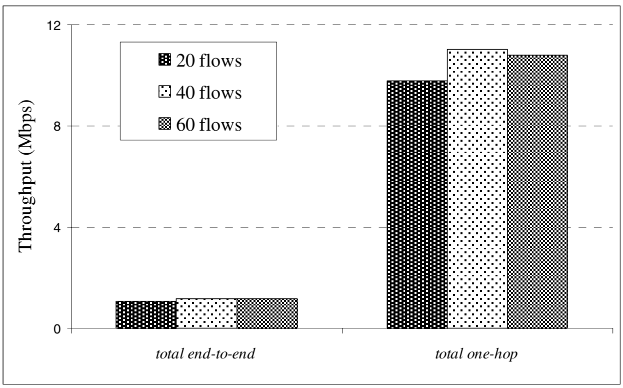

Fig. 10. Total end-to-end and total one-hop throughput for different flows

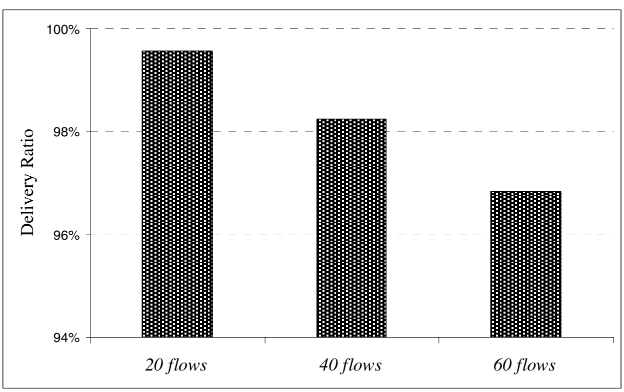

Fig. 11. Packet delivery ratio for different flows

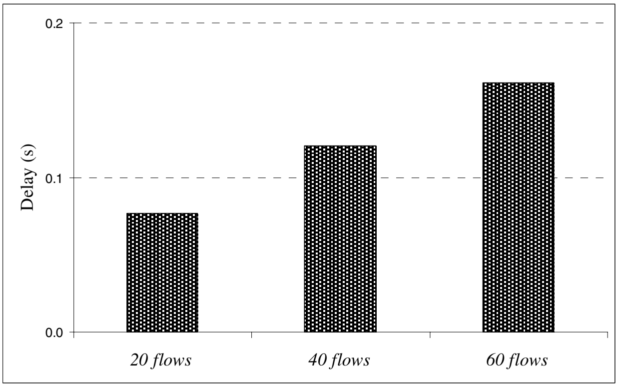

Fig. 12. Delay for different flows

delivery ratio than the 20-flow case, their flow assignments can still support real-time applications. This shows that the two flow assignments, for 40-flow and 60-flow cases, do not over utilize the network capacity. To summarize, our model demonstrates the scalability property as the number of flows in the network increases.

5.3 1000-Node and 100-Flow

In this section, we increase the number of nodes in the network to 1000 and the flow number to 100. We show that our max-min fair flow assignment in this case does not under or over utilize the network capacity.

Under-assign Capacity. It is difficult to show that all of our flow assignments do not under-assign the network capacity, since we can not determine the exact max-min fair flow assignment. However, we can use the experimental procedure to determine the smallest flows, "*minimum rate*," and compare these values with our assignments. Minimum rate is used to define max-min fairness in [9]. As shown in Fig. 13, the minimum assignment in our model is about 93% of that in the experimental model. Our model does not under-assign the network capacity in terms of the minimum rate.

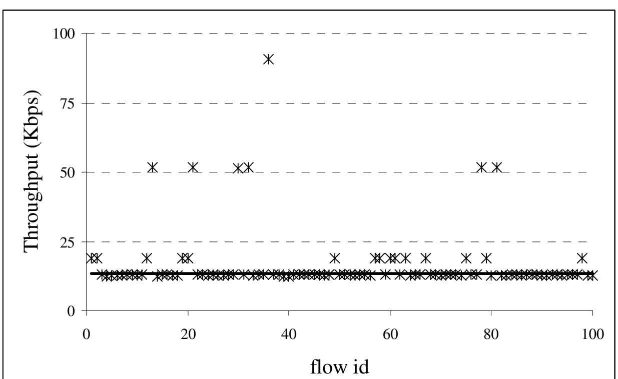

Fig. 13. Throughput for 1000-node and 100-flow case

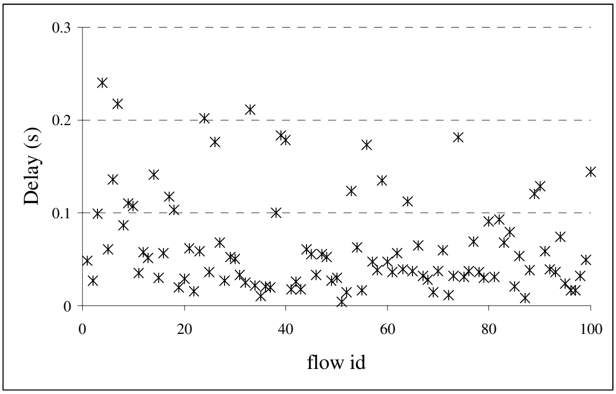

Fig. 14. Delay for 1000-node and 100-flow case

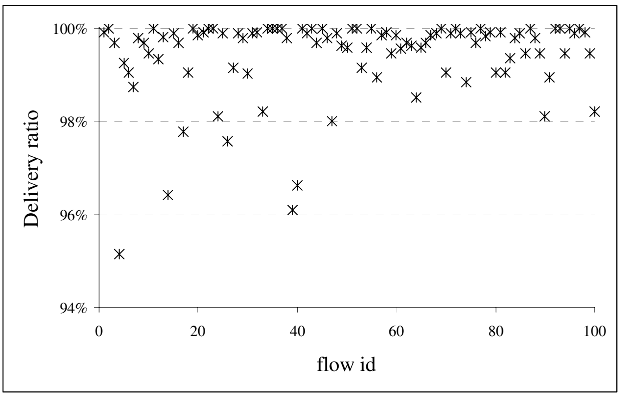

Fig. 15. Packet delivery ratio for 1000-node and 100-flow case

Over-assign Capacity. Fig. 14 shows the delay experienced by all 100 flows and Fig. 15 provides the packet delivery ratio for all 100 flows. Both of them demonstrate that our model has not over-assigned the network capacity. This shows that our model scales well as the number of nodes and the number of flows increase while providing a good max-min fair rate allocation.

5.4 Time Square

We apply our model to another network topology, Time Square between 40th Street and 44th Street, as shown in Fig. 16. In this case, 800 nodes are randomly deployed in the network and 20 flows are randomly generated. The results are consistent with Columbia University network as seen in Fig. 17, Fig. 18, and Fig. 19. The only difference is that when comparing to the experimental minimum rate allocated to any flow, the model finds the minimum rate that is about 80% of the experimental result, as seen in Fig. 17. This is much lower than the 93% in Columbia University network.

Fig. 16. Time Square (courtesy: Yahoo Map)

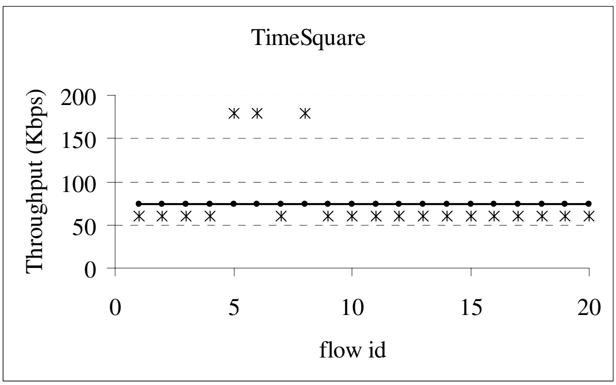

Fig. 17. Throughput for Time Square network

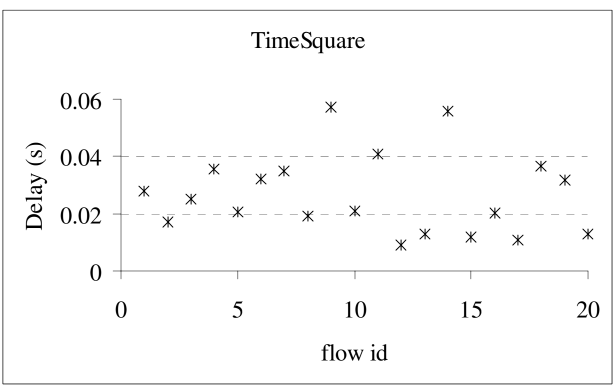

Fig. 18. Delay for Time Square network

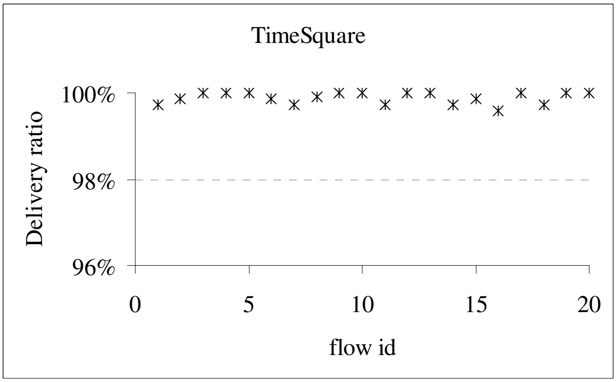

Fig. 19. Packet delivery ratio for Time Square network

The reason for that is we reserve too much capacity for interfering flows. Choosing a smaller value α will increase the percentage. The relationship between network topology and α value bears further study.

6 Conclusions and Future Work

In this paper, we present a simple estimation model to achieve max-min fair flow assignment in wireless ad hoc networks. The proposed scheme works on our macro model and applies wired techniques to wireless networks with a simple modification. Simulation results show that our method produces a rate allocation that is comparable to the optimal max-min fair rate allocation in small-scale networks. Our method also works well in large-scale networks and provides rate allocations that can support real-time application. Since part of the network capacity is reserved, our model should work well in mobile environment.

We are currently investigating the effect of α value to the network utilization. α may also depend on the network topology and other network parameters, such as node density. We are also studying the stability of our proposed model in mobile environment. Improve the simple model to achieve more accurate max-min fair rate allocation and find a distributed implementation are two more ways to extend this paper.

References

1. Bertsekas, D., Gallagher, R.: Data Networks. Prentice Hall, Upper Saddle River (1992)
2. Nace, D., Pioro, M.: Max-Min Fairness and Its Applications to Routing and Load-Balancing in Communication Networks: A Tutorial. IEEE Communications Surveys & Tutorials 10(4), 5–17 (2008)
3. Jaffe, J.: Bottleneck Flow Control. IEEE Transactions on Communications 29(7), 954–962 (1981)
4. Huang, X.L., Bensaou, B.: On Max-Min Fairness and Scheduling in Wireless Ad-Hoc Networks: Analytical Framework and Implementation. In: 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, pp. 221–231 (2001)

5. Tassiulas, L., Sarkar, S.: Maxmin Fair Scheduling in Wireless Networks. IEEE Journal on Selected Areas in Communications 23(1), 163–173 (2005)
6. Wang, X., Kar, K.: Distributed Algorithms for Max-Min Fair Rate Allocation in Aloha Networks. In: 42nd Annual Allerton Conference, Urbana-Champaign (2004)
7. Wang, X., Kar, K., Pang, J.S.: Lexicographic Max-Min Fairness in a Wireless Ad-Hoc Network with Random Access. In: 45th IEEE Conference on Decision and Control, San Diego, pp. 1294–1300 (2006)
8. Aoun, B., Boutaba, R.: Max-Min Fair Capacity of Wireless Mesh Networks. In: 2006 IEEE International Conference on Mobile Adhoc and Sensor Systems, Vancouver, pp. 21–30 (2006)
9. Sridharan, A., Krishnamachari, B.: Maximizing Network Utilization with Max-Min Fairness in Wireless Sensor Networks. In: 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops, Limassol, pp. 1–9 (2007)
10. Maxemchuk, N.F., Zhou, C.: A Macro Model of Frequently Changing Mobile Networks to Perform Flow and Access Control. Mobile Networks and Applications 11(5), 649–659 (2006)
11. Zhou, C., Maxemchuk, N.F.: Applying a Macro Model of Ad Hoc Networks to Access Control. In: 7th International Conference on Networking, Cancun, pp. 445–453 (2008)
12. Yang, Y., Kravets, R.: Contention-Aware Admission Control for Ad Hoc Networks. IEEE Transactions on Mobile Computing 4(4), 363–377 (2005)
13. Sanzgiri, K., Chakeres, I., Belding-Royer, E.: Determining Intra-Flow Contention along Multihop Paths in Wireless Networks. In: 1st International Conference on Broadband Networks, San Jose, pp. 611–620 (2004)
14. GloMoSim Simulator, `http://pcl.cs.ucla.edu/projects/glomosim/`
15. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Computer Society (1999)
16. Karp, B., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: 6th Annual ACM/IEEE International Conference on Mobile computing and Networking, Boston, pp. 243–254 (2000)
17. Liao, W.H., Tseng, Y.C., Sheu, J.P.: GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks. Telecommunication Systems 18(1), 37–60 (2001)
18. Kim, D., Maxemchuk, N.F.: Simple Robotic Routing in Ad Hoc Networks. In: 13th IEEE International Conference on Network Protocols, Boston, pp. 159–168 (2005)
19. Gupta, P., Kumar, P.R.: The Capacity of Wireless Networks. IEEE Transactions on Information Theory 46(2), 388–404 (2000)
20. Li, J., Blake, C., De Couto, D.S.J., Lee, H.I., Morris, R.: Capacity of Ad Hoc Wireless Networks. In: 7th Annual International Conference on Mobile computing and Networking, Rome, pp. 61–69 (2001)