Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2011, Vol. 3, No. 1

Distributed Bottleneck Flow Control in Mobile Ad Hoc Networks

Congzhou Zhou

Dept. of Electrical Engineering, Columbia University

New York, NY 10027, USA

Tel: 1-917-441-8756 E-mail: zhoucongzhou@gmail.com

N. F. Maxemchuk

Dept. of Electrical Engineering, Columbia University

New York, NY 10027, USA

IMDEA Networks, Madrid, Spain

Tel: 1-212-854-0580 E-mail: nick@ee.columbia.edu

Received: January 26, 2011 Accepted: April 5, 2011 DOI: 10.5296/npa.v3i1.576

Abstract

Flow control in mobile ad hoc networks (MANETs) encounters more challenges than flow control in conventional wired networks, such as channel bandwidth variation, medium contention, and frequent re-routing. Without proper flow control, serious network performance degradation in MANETs has been reported in recent studies. In this paper, we introduce a distributed bottleneck flow control technique in MANETs. The proposed scheme uses a distributed flow control mechanism that has been used in both bottleneck flow control and bandwidth balancing in Distributed-Queue-Dual-Bus (DQDB) in wired networks. It extends bandwidth balancing to operate in a mobile wireless environment. Extensive simulations demonstrate that our flow control scheme is effective and can provide max-min fairness as well as improve Quality of Service (QoS) for flows in MANETs.

Keywords: Flow control, bottleneck, bandwidth balancing, max-min fairness.

1. Introduction

The demand for real-time applications in wireless networks grows as handheld wireless devices, such as iPhone and iPad, gain popularity. Providing proper flow control is essential to support such applications. Controlling the flows that can enter the network results in QoS guarantees, such as delay, packet delivery ratio, etc., that are required for real-time applications, such as voice and video communication.

Much research has been conducted and many solutions have been proposed to perform flow control in wired networks. However, because of the differences between wired and wireless communication and the frequently changing network topology, those techniques perform poorly in MANETs. C. Lochert et al. [1] provide a comprehensive survey of current congestion control techniques for MANETs. In that paper, existing proposals are grouped into different categories based on the problem they solve. For example, OPET [2] deals with problems caused by a shared medium and uses a MAC layer scheduling scheme to provide network layer flow control; RBCC [3] limits TCP's packet output in order to adapt TCP to MANETs; and EXACT [4] provides an alternative protocol design which uses a rate-based technique for MANETs. There are some other techniques that are not based on flow control that provide QoS in MANETs. Y. Yang and R. Kravets [5] introduce a contention-aware admission control algorithm; L. Chen and W. Heinzelman [6] propose a QoS-aware routing protocol for multi-hop ad hoc networks. Both of these mechanisms use the full channel information within a node's contention area to make admission or routing decision.

In our previous research [7], [8], [9], we introduced a macro model to perform flow and access control in frequently changing mobile networks. The macro model represents a wireless network as a collection of super nodes and links that resemble a wired network. Through the macro model, flow control algorithms that are designed for wired networks are applied to the wireless networks. In our previous research [10], [11], we show how to use our macro model to achieve max-min fairness [12] in MANETs. However, a centralized intelligence is used to construct our macro model. And, in a mobile environment frequent model reconstruction is required and it will consume a large percentage of network resource.

In this paper, we introduce a distributed bottleneck flow control scheme for MANETs. To perform flow control, each node monitors its channel to determine the residual capacity and only assigns a portion of that capacity to its active flows. The rate constraints are collected along the path from the source to the destination and rate on the bottleneck node along a path is communicated to the source node. As a result, the source node changes its sending rate as the flows on the nodes on the path change or as the nodes on the path change. By constraining nodes to acquire only a fraction of the available bandwidth, our flow control scheme can adapt to the dynamic nature of MANETs and provide fair bandwidth access.

The rest of this paper is organized as follows: in section 2, we describe the work on bottleneck flow control and bandwidth balancing that motivated our distributed flow control scheme. We show that both strategies are the same, and that they can be extended to wireless networks. Section 3 introduces our distributed bottleneck flow control algorithm that uses the bandwidth balancing technique. The detailed implementation based on an IEEE 802.11

network is also presented. Section 4 evaluates the performance of our scheme through extensive simulations. To measure fairness, we introduce a modified version of Jain's fairness index that can be used to provide a max-min fair environment. Finally, our conclusions are presented in section 5.

2. Related Work

In this section, we briefly describe the two previous flow control mechanisms that have motivated our research, namely bottleneck flow control and bandwidth balancing. We find that these two techniques are similar to each other and show how the bandwidth balancing parameter can be applied to bottleneck flow control. Using the bandwidth balancing parameter, instead of the parameter originally used in bottleneck flow control, clarifies the operation of bottleneck flow control and shows how to apply it to the more general mobile wireless networks.

2.1 Bottleneck Flow Control

In Jaffe's bottleneck flow control [13], a user can acquire a flow up to $X_k R_k$ at each node on its path and is constrained to the smallest flow on the path. At each node on a path a flow is constrained to:

$$\gamma_k = X_k R_k . \tag{1}$$

where γ_k is the user's largest allowed flow on link k, X_k is a non-negative constant between [0, ∞], and R_k is the residual, unused, capacity on link k. The parameter X_k is used to achieve different throughput-delay tradeoffs. A larger value of X_k results in a higher network utilization and a larger network delay; a smaller value of X_k results in smaller network utilization and a smaller network delay.

User r learns γ_k for each link on its path and adjusts its flow to

$$\gamma_r = \min_{k \in p}(\gamma_r^0, \gamma_k) , \tag{2}$$

where γ_r^0 is the maximum rate that the r^{th} user requests. For example, a voice source may set γ^0 to the bit rate that it would like to use, and a data source may set γ^0 to ∞ to get as many bits/sec as possible.

For user r on link k, $f_k = f_{k,other} + \gamma_r$, where, f_k to is the total flow on link k. The residual capacity is:

$$R_k = C_k - f_k = C_k - f_{k,other} - \gamma_r , \tag{3}$$

where C_k is the link capacity. When γ_r^0 is set to ∞ user r is constrained on link k to a flow

$$\gamma_r = \frac{X_k}{1+X_k}(C_k - f_{k,others}). \quad (4)$$

Bottleneck flow control is fair in that it guarantees that a user gets as much flow through its bottleneck link as any other user on that link. This technique is different from the max-min fair water filling algorithm because it only allocates a fraction of the bandwidth on a link, rather than all of the bandwidth. Leaving a small amount of unused bandwidth allows the flow allocations to change as flows are added or deleted. The algorithm operates by giving a new flow a fraction of the residual capacity and reducing the capacity of the other flows because the residual capacity has been reduced. The flows eventually converge to a fair operating point. This is similar to the operation of bandwidth balancing.

2.2 Bandwidth Balancing

Bandwidth balancing [14], [15] solves the fairness problem in long DQDB networks by allowing each node to take only a fraction α of the available slots. Bandwidth balancing converges to a fair operating point where all sources acquire the same bandwidth. The steady state throughput of user r is

$$\gamma_r = \alpha(C_k - f_{k,others}). \quad (5)$$

α, the fraction of the residual bandwidth that a source takes, is between [0, 1]. If α is small, a large fraction of slots are wasted but the network converges to a fair operation fast; if α is large, a small fraction of slots are wasted but the network convergence time increases.

2.3 Relationship between the two Techniques

Bandwidth balancing operates on a single link in a DQDB network, while bottleneck flow control operates on each link on a routing path. In bottleneck flow control a single source accesses the capacity on a link, while in a DQDB network the capacity on the link is shared by several sources that are distributed along the link. However, the underlying flow control strategy is the same for both techniques. The system does not allocate all of the bandwidth on a channel; each user is constrained to a fraction of the bandwidth that is not being used; by adjusting their flows, each of the users eventually obtain the same bandwidth; users are added and achieve parity with the other users by acquiring part of the residual bandwidth, which forces the other users to reduce their bandwidth; and, when users leave the residual bandwidth increases and is fairly redistributed among the other users.

From (4) and (5), we note that the parameters in bottleneck flow control and bandwidth balancing are related:

$$\alpha = \frac{X_k}{1+X_k}. \quad (6)$$

Moreover, by replacing the parameter X_k with a new parameter α, it becomes clear that

bottleneck flow control is actually allocating a fraction of the remaining capacity on each link along a path.

We apply that same technique to wireless networks and use parameter α because it clearly shows the fraction of the remaining bandwidth that a user acquires. The wireless network has several links on a path, as in bottleneck flow control, but the capacity on a channel is shared by several sources at different nodes, as in bandwidth balancing. Each flow at each node in a transmission region takes a fraction of the remaining capacity.

2.4 Flow Control in Wireless Networks

In a wireless network, we replace the residual capacity of a node with the residual capacity in a transmission region. In wireless networks, due to the nature of a shared medium, a node has to compete with other nodes that are within its contention area for the access to the channel. By monitoring the channel, a node can calculate the available bandwidth, and each flow in the node can contend for a fraction of the residual bandwidth in the transmission region. The residual capacity already excludes other nodes' transmission. Therefore, flow control based on residual capacity avoids the complicated interference issue in MANETs. By making decisions based on local information, flow control based on residual capacity is implemented in a distributed fashion without explicitly exchanging information between the nodes. Furthermore, since each flow continuously adjusts its bandwidth to account for changes in the other flows, it is suitable for the dynamic environment in a MANET.

However, in wireless networks, the residual bandwidth at nodes that interfere with one another may not be the same, because a different set of nodes interferes with each of the nodes. This differs from wired networks where all of the flows in a node see the same residual bandwidth.

As shown in Fig. 1, node *A* and node *B* are separated by a distance of *d*. The common contention area is the overlapping area of the two dotted circles. It can be calculated as:

$$INTC(d) = 4\int_{d/2}^{2R} \sqrt{(2R)^2 - x^2}\, dx\,, \tag{7}$$

where *R* is the transmission radius and the interference radius is *2R*. Fig. 2 shows the relationship between distance *d* and the percentage of common area, which is *INTC(d)* divided by contention area, $\pi(2R)^2$. As *d* increases, the percentage of the common area decreases, and vice versa as *d* decreases. If *A* and *B* are at the same location, they see the same channel and can operate exactly as in a wired network; if *A* and *B* are separated by a distance of more than *4R*, they do not have any common contention area. When nodes see similar channel, i.e. they have large percentage of common area, flow control based on residual capacity still works. However, it may not provide perfect fair bandwidth access as in wired networks. Sharing channel information helps, but it imposes a high control overhead. If *A* and *B* are outside each other's transmission radius but within each other's interference range, the control overhead will be even higher because it requires multi-hop communications. Our simulation shows that without information exchange, we can still provide effective flow control in MANETs using the bandwidth balancing technique and

achieve reasonably fair bandwidth allocation.

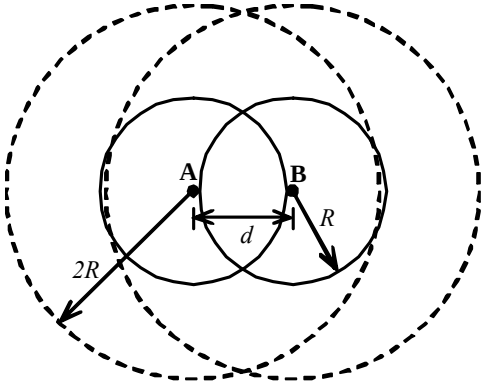

Figure 1. Common contention area of node A and B.

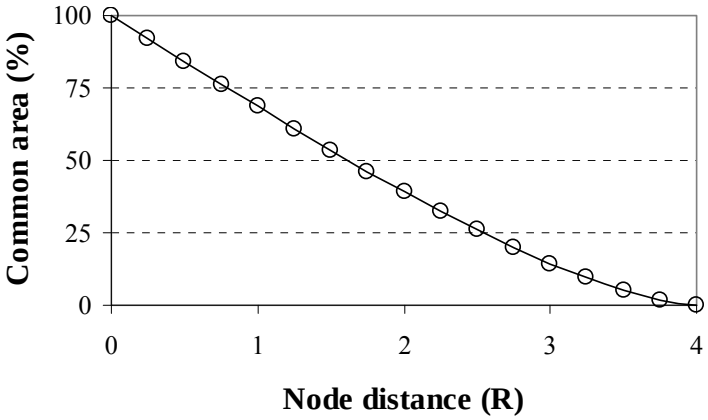

Figure 2. Node distance versus common area.

3. Distributed Bottleneck Flow Control

In this section, we describe our distributed bottleneck flow control algorithm in MANETs. In our scheme, nodes do not maintain flow information, which is different from other techniques. Every node in a routing path monitors its channel status and calculates its residual capacity, the capacity available for use.

Each packet that passes through a node contains the current bandwidth that a flow is using and the minimum flow that the packet is assigned at the previous nodes that it has visited. Each node on a path uses the residual capacity and a flows current assignment to determine the flows new allocation. If the new rate is smaller than the minimum rate allocated by the previous nodes on the path, the node changes that field in the packet. The destination returns the minimum rate to the source. The source node adjusts its sending rate accordingly to avoid congesting its bottleneck.

As a result, any bandwidth change resulting from changes in node positions, the routing path, or newly added/removed flows is returned to the source node after one round trip time (RTT). In this way, our distributed bottleneck flow control scheme can adapt to the dynamic nature of MANETs and provide reasonable max-min fairness.

3.1 Residual Capacity

Residual capacity at a node is the difference between the node's channel capacity and the sum of the bandwidth consumed by all contending flows of that node. It denotes the channel capacity that is not used and it constrains the rate that contending flows can acquire. To measure the residual capacity, each node in the network monitors the channel activity. The fraction of channel idle time during the past measuring period and the channel capacity, determine its residual capacity. The residual capacity at node k is:

$$R_k = \frac{T_{idle}}{T_p} C_k . \tag{8}$$

Here C_k is the channel capacity at node k; T_{idle} is the channel idle time during the last measuring period T_p. Larger T_p will give more accurate channel view but also longer response time for the source node to react to changes in the network. In this paper, T_p is set to 0.5 second.

3.2 Flow Bandwidth Consumption

A flow passing a node's contention area may consume more bandwidth than the flow's rate. Finding flow bandwidth consumption is very important when applying the bandwidth balancing technique in wireless networks. As shown in Fig. 3, flow r has a route of $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow \ldots$ Node A, B, C, and D are all within node K's contention area. Therefore, flow r consumes a bandwidth that is 4 times its rate at node k. How to convert flow rate into corresponding channel bandwidth can be found in [5]. In this paper, we assume the flow rate is already translated.

To determine the exact bandwidth consumption at a specific location, we need to know the number of nodes on the route of the flow that contend for bandwidth at that location. This number is called *contention count* N_{ct} in [5]. The same flow can have different N_{ct} at each node. For example, in Fig. 3, flow r's N_{ct} at node k is 4 while it's N_{ct} at node A is 3. A simple estimation is provided in [6], [16], which relates the contention count to flow's hop count and sets N_{ct} to *min(hop count, 4)*. A more accurate estimation can be found in [5], [17]. In this paper, we choose the simple estimation method, and fix N_{ct} to 4. Through simulation, we find that it is adequate to use a fixed N_{ct}.

3.3 Allowed Rate Calculation

In this section, we show how to calculate the new allowed rate for flow r at node k. Given the residual capacity measured by node k and flow r's current rate, the new allowed

rate for user r at node k is:

$$\gamma_r^{'} = \alpha(R_k + \gamma_r), \tag{9}$$

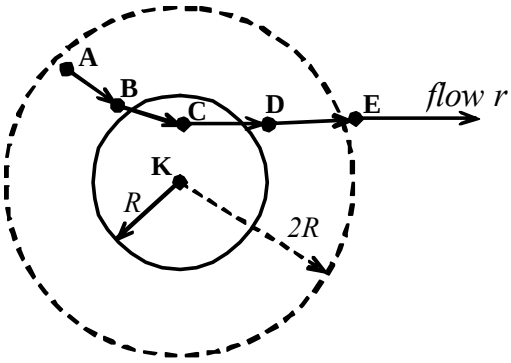

Figure 3. Flow bandwidth consumption.

according to (3) and (5). Here $\gamma_r^{'}$ is the new allowed rate and γ_r is the current sending rate. Now adding contention count for wireless networks, we get

$$N_{ct} \cdot \gamma_r^{'} = \alpha(R_k + N_{ct} \cdot \gamma_r). \tag{10}$$

The new rate becomes

$$\gamma_r^{'} = \alpha(R_k / N_{ct} + \gamma_r). \tag{11}$$

If node k is the bottleneck for user r, the steady state rate for flow r is

$$\gamma = \frac{\alpha}{1-\alpha} \cdot \frac{R_k}{N_{ct}}. \tag{12}$$

In order to calculate the fraction of the residual capacity assigned to a flow we first subtract the flows previous capacity from the measured channel utilization. When nodes move and a flow enters a new node, the node may provide too high an initial assignment. The adaptive nature of the algorithm corrects this error when subsequent packets are transmitted.

3.4 Rate Adjustment

This section describes the mechanism the source node uses to adjust its sending rate. The flow information is carried in the IP header of data packet. There are two fields called *CR (Current Rate)* and *AR (Allowed Rate)*. CR is set by the source node to record the current sending rate and cannot be modified by any other node. AR is initially set by the source node as $min(\gamma_r^0, \gamma_r^{'})$, where γ_r^0 is the maximum flow limit of the r^{th} user and $\gamma_r^{'}$ is the new allowed rate calculated by the source node according to (11). The intermediate nodes and the

destination node all use their local information R_k and the information CR in the packet to calculate the new allowed rate. If the new rate is smaller than AR in the packet, they modify the AR field with the newly calculated rate. The destination node returns AR to the source in an acknowledgement packet. The source node adjusts its sending rate accordingly.

3.5 Discussion

In this section, we discuss the effect of the system parameter α, in a single-hop network. The effect in a multi-hop network is complex and requires extensive simulations to analyze. Next, we discuss the max-min fairness adopted by our flow control scheme.

In our flow control scheme, the new allowed rate depends on the value of α (11). To study the effect of α, we use a single cell network where every node can hear each other. In that environment, the view of the channel at each node is the same. Larger α results in larger flow rate. To achieve fairness, α has to be the same for flows constrained in the same bottleneck area. The rate of convergence to a fair operating point is also affected by α. As shown in [14], [15], the smaller the value of α, the faster the convergence. Parameter α can also affect the network utilization in the bottleneck area. For example, if there are N flows in the single transmission region, in the steady state each will get a flow rate of

$$\gamma = \frac{\alpha \cdot C_k}{(1+\alpha(N-1)) \cdot N_{ct}}. \tag{13}$$

Therefore, the network utilization of the bottleneck area is:

$$\eta = \frac{\alpha \cdot N}{1+\alpha(N-1)}. \tag{14}$$

Fig. 4 shows the relationship between η, α, and the number of flows, N. For a certain number of flows, increasing α increases the network utilization. To maintain a specific utilization, when the number of flows increases, α must decrease, and vice versa. As shown in [14], [15], α can also be used to create priorities for different type of traffic.

The multi-hop network is much more difficult to understand than the single-hop network. It requires extensive simulations to determine the exact effect of α. However, a relationship similar to the single-hop network can be expected. Our simulation results in the next section verify this hypothesis. To achieve the same network utilization in all bottleneck areas, different α is needed for flows constrained by different bottleneck area. In this paper, we use the same α for all flows. Therefore, different channel utilizations are expected in different bottlenecks.

Our flow control scheme adopts max-min fairness [12] to allocate resources to all flows in the network. Flows that are constrained by the same bottleneck area should get the same rate. A flow has a smaller rate than other flows only if it reaches its maximum flow limit or is constrained by another bottleneck area. In a wired network, Jaffe's bottleneck flow control [13] produces a max-min fair rate allocation within a small fraction of residual bandwidth.

With the same technique, our flow control scheme should achieve reasonable max-min fairness in a wireless network.

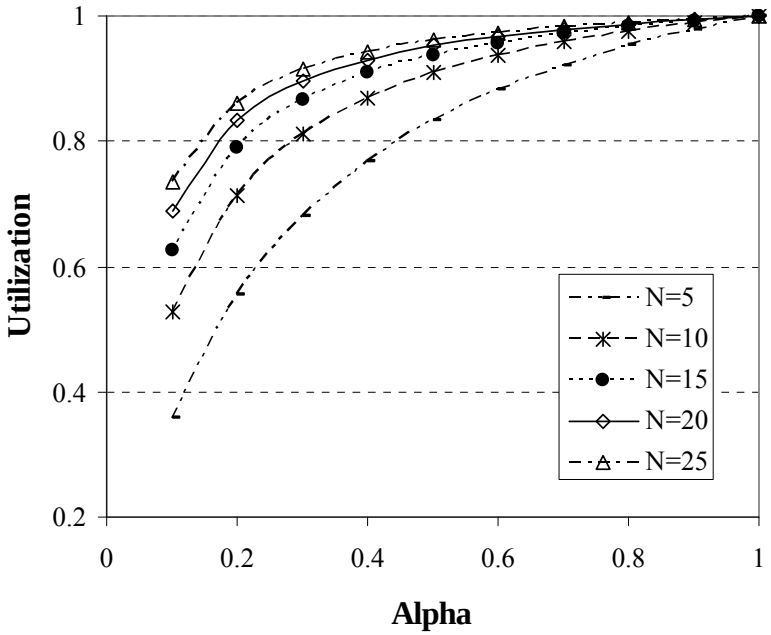

Figure 4. Utilization versus α under different N.

4. Simulation Results

In this section, we evaluate the performance of our proposed distributed bottleneck flow control algorithm by simulations in GloMoSim [18], [19]. In the simulations, 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 transmission radius and a 38.8 meters sensing radius. The radio propagation model we use at these short distances is the free-space attenuation model. The MAC layer is IEEE 802.11 [20] and the maximum number of retransmissions is set to seven. The packet size is set to 1024 bytes.

The simulations are performed in the following order. We first validate our scheme in a simple chain topology in section *4.1*. To measure max-min fairness in our simulations, we modify Jain's fairness index. In section *4.2*, we compare our scheme with a system without flow control in random static topologies, and demonstrate the effectiveness of flow control. We show the performance of our scheme in mobile environment in section *4.3*. In section *4.4*, we use the Columbia University Campus network and show that our distributed approach performs better than our earlier centralized approach.

4.1 Simple Scenarios

To illustrate the effectiveness of our flow control protocol, we first conduct a simple

simulation on a three-chain topology. As shown in Fig. 5, there are three 9-node chains and there is a one-way CBR/UDP flow on each chain. We set α to 0.6 in this simulation according to (14) for a reasonable operation point. The information about each connection along with the starting/ending time is listed in Table 1.

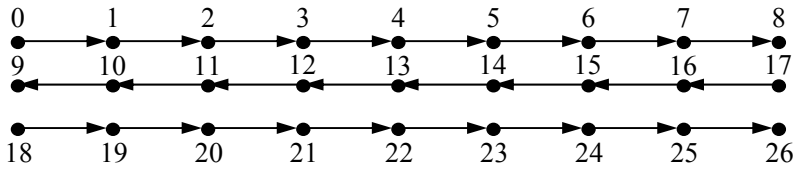

Figure 5. Three-chain topology.

Table 1. Configurations of three CBR flows.

Flow No.	Source Node	Destination Node	Starting Time (s)	Ending Time (s)
1	0	8	0	200
2	17	9	50	250
3	18	26	100	300

We first set these three chains close to each other so that a node in a chain and its competitors in the other two chains can see the same channel utilization. For example, node 12 in flow 2 has the same channel utilization as node 3 in flow 1 and node 21 in flow 3. Fig. 6 shows the throughput of the three flows. At 0 seconds, flow 1 starts and it quickly settles to 500 Kbps. At 50 second, flow 2 starts. It increases its rate while flow 1 decreases its rate both to around 350 Kbps. After flow 3 starts at 100 second, all three flows adjusts their rates to about 250K bps. The results show that with our flow control, a flow can adjust its rate when new flows are introduced. Moreover, when multiple flows share the same bottleneck area and measure the same channel utilization, they get the same bandwidth.

Next, we separate the three chains so that flow 2 can see flow 1 and flow 3 while flow 1 and flow 3 cannot see each other. In that case, flow 2 will measure a more congested channel than flow 1 and flow 3 when all three flows are active. Fig. 7 depicts the throughput of three flows in separated chains. When compared to Fig. 6, the three flows get unequal shares of the channel capacity during 100~200s period. The reason is that when using the same α, a flow that sees a congested channel will get a smaller rate allocation than a flow that observes a lighter channel condition. Therefore, flow 2 achieves a smaller throughput than the other two flows. Outside 100~200s period, the two scenarios show the same results.

Fig. 8 shows the total end-to-end throughput in two different chain topologies. It verifies one of our conclusions from Fig. 4. The network utilization and throughput increases with N, the number of flows. For example, during 0~50s, there is only one active flow in the network and the end-to-end throughput is about 500 Kbps; while in 50~100s period, two flows are

active and the end-to-end throughput is almost 700 Kbps. When all three flows are active during 100~200s period, separated chains topology achieves a higher throughput than adjacent chains topology. This is because separated chains topology uses a larger physical space than adjacent chains topology.

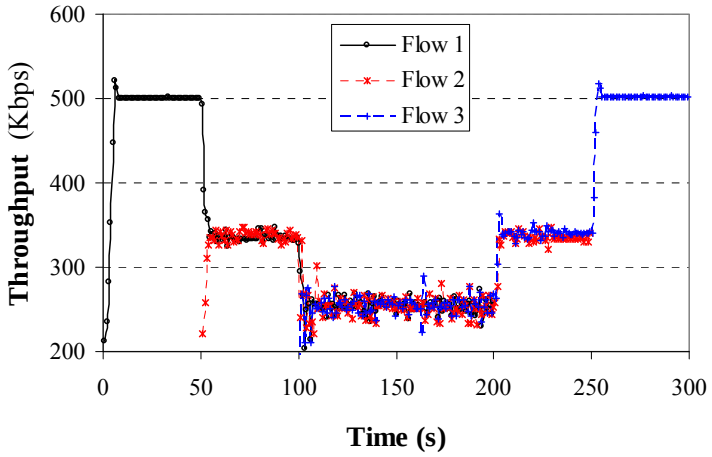

Figure 6. Throughput in adjacent chains.

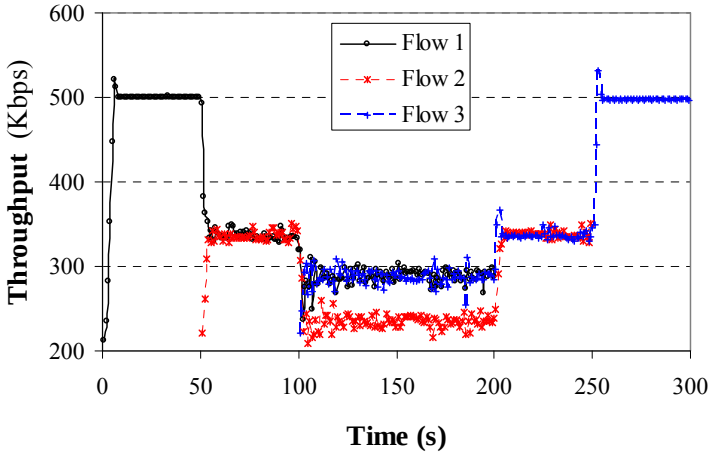

Figure 7. Throughput in separated chains.

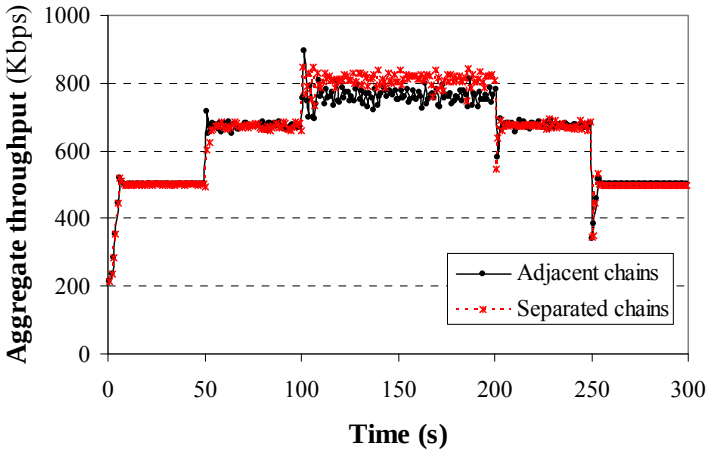

Figure 8. Total end-to-end throughput in three-chain topology.

In Table 2, we list the delay and delivery ratio of three flows in both adjacent chains and separated chains. All flows have delays that are less than 0.03 second and delivery ratios that are greater than 99.99%, which are adequate for many real-time applications.

To measure fairness among flows in the network, we choose the commonly used Jain's fairness index [21]:

$$F = (\sum_{i=1}^{n} \gamma_i)^2 / (n \cdot \sum_{i=1}^{n} \gamma_i^2) . \quad (15)$$

However, this metric only measures equality of the flows and is not suitable in max-min fairness, where the objective is to obtain as much bandwidth as possible without taking bandwidth that a lower rate user might use. The Jain's equality measurement achieves fairness by constraining the bandwidth of all users to the lowest rate user's bandwidth. To measure fairness rather than equality, we apply it only to flows that are constrained by the same bottleneck area, instead of all flows in the network. According to Fig. 2, when d equals R, the common area is more than 2/3 of the contention area and we consider that node A and Node B measure similar channel utilizations. When d is greater than R, node A and node B see a slightly different channel. In this paper, we set a node's bottleneck area with a radius of R, i.e. for that node we apply Jain's fairness index to all flows that are constrained in that bottleneck area.

Here is how we calculate the modified Jain's fairness index. For each flow, we first find the constrained node, and then we find which other flow is also constrained at nodes in that node's bottleneck area. Once we determine the set of flows, we apply Jain's fairness index to those flows. Therefore, we have one fairness index for each flow and take the average as the final fairness index for our flow control scheme. Fig. 9 shows the fairness index in the two topologies. In the adjacent chains topology, the fairness index appears to be 1 most of the time, except at 50 and 100 second when new flows are first added, and are in the process of

adjusting their rates. The separated chains topology shows similar results except in 100~200s period, where the three flows have unequal rates. However, the index is above 0.98 and the system is considered to be reasonably fair.

Table 2. Delay and delivery ratio of three flows.

Flow No.	Adjacent chains		Separated chains	
	delay (s)	delivery ratio (%)	delay (s)	delivery ratio (%)
1	0.023	100.00	0.021	100.00
2	0.030	99.99	0.027	100.00
3	0.024	99.99	0.021	99.99

4.2 Static Topology

In this simulation, we evaluate our flow control scheme in a random topology. 100 stationary nodes are randomly deployed in a 100m x 100m area. 30 CBR flows are randomly established. α is set to 0.15. We pre-compute the shortest path and there is no routing overhead in this scenario. Each simulation runs for 300 seconds.

Fig. 10 shows the aggregate end-to-end throughput for our flow control scheme and for a system without flow control. When the total offered load exceeds the network capacity, congestion occurs in the system without flow control which results in an increase in the number of collisions and packets that are discarded because of excessive collisions, and a decrease in the end-to-end throughput. Our flow control scheme limits the sending rates of the flows under heavy traffic and significantly improves the end-to-end throughput.

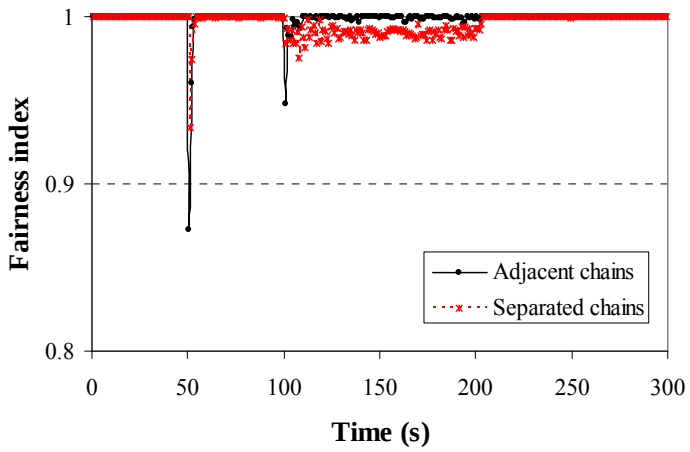

Figure 9. Fairness index in three-chain topology.

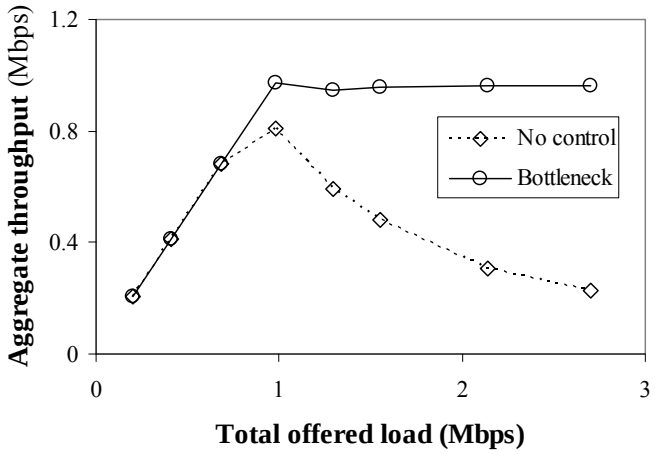

Figure 10. Aggregate throughput in static topology.

From Fig. 11 and Fig. 12, we observe that our scheme performs much better than the scheme without flow control in terms of packet delivery ratio and end-to-end delay. Without flow control, congestion occurs when the offered load exceeds the network capacity. Many packets are dropped and excessive queuing delays occur. On the other hand, our flow control scheme limits the rate of each flow according to the bottleneck capacity and hence maintains a small packet loss ratio and a small end-to-end delay for all traffic status.

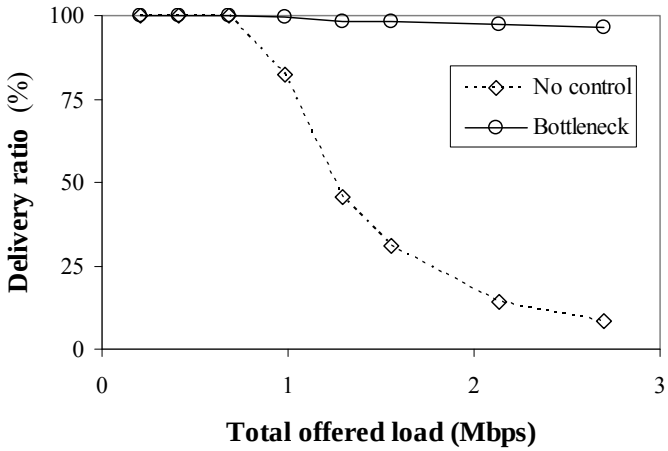

Figure 11. Delivery ratio in static topology.

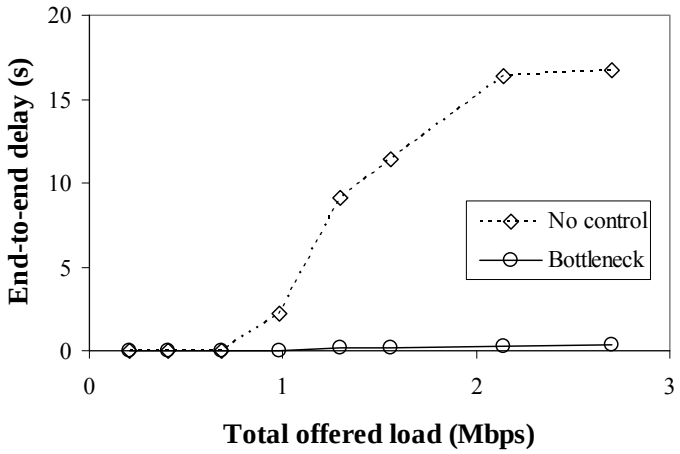

Figure 12. Delay in static topology.

Fig. 13 depicts the mean and 95% confidence interval of the fairness index for our rate allocation in the static random topology. The fairness indexes are all above 0.95 which shows that the rate allocation for flows constrained by the same bottleneck area is fair. Under light traffic, every flow obtains its requested rate and the fairness index is 1; when traffic becomes heavier, our flow control mechanism limits the rates of some of the flows, but still provides fairness in bottleneck areas.

4.3 Mobile Topology

In this simulation, 100 nodes are randomly distributed in a 100m x 100m network. 10 CBR flows are randomly established and are active throughout the simulation. α takes value from 0.2, 0.3, and 0.4. Nodes move according to the random waypoint mobility model [22]. The speed of nodes is randomly chosen between 0~3 m/s and the pause time is set to 5s. The routing protocol is AODV [23]. Each simulation runs for 300 seconds.

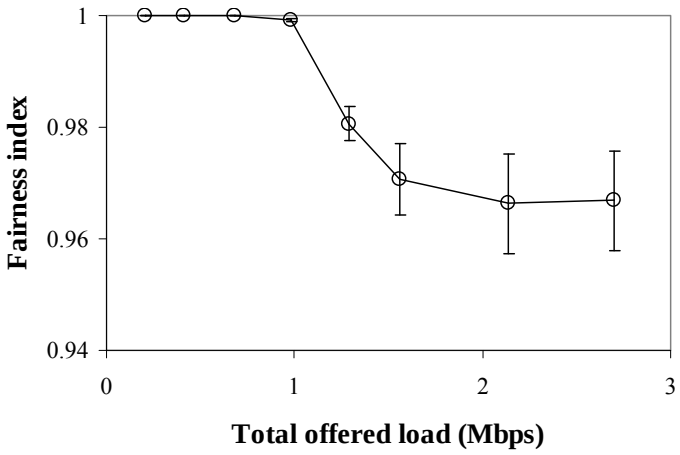

Figure 13 Fairness index in static topology.

Figures 14 to 16 depict the throughput, delivery ratio, and delay of the ten flows. In mobile networks, packet are dropped by congestion or broken paths. In all three cases with

different α, our flow control scheme achieves a delivery ratio higher than 90% and a delay less than 0.08s for all ten flows in the network. When α is smaller, the throughput and the delay are smaller but the delivery ratio is higher. This is because smaller α results in smaller network utilizations in the bottleneck areas. In Fig. 14, we observe that larger α results in bigger differences among ten flows. This is because flows that see lighter channel utilization acquire more bandwidth. In Fig. 17, the instantaneous throughput of flow 1 is plotted. During 50~200s period, flow 1 moves to a less congested area. With a bigger α, it gets a greater instantaneous throughput during that period and hence achieves a larger throughput in Fig. 14. Fig. 17 also demonstrates that with our flow control a user adjusts its rate continuously as node moves in the network.

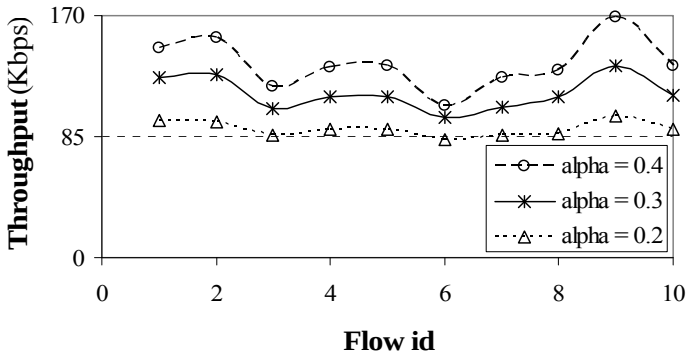

Figure 14. Throughput in mobile topology.

Next, we measure the effects of node mobility by changing the node speed. The maximum speed increases from 1 m/s to 11 m/s. Fig. 18 to Fig. 20 shows the throughput, delivery ratio, and delay of our flow control scheme. The results show that our scheme outperforms the scheme without flow control in all mobility scenarios. Moreover, mobility has little effect on the performance of our scheme.

We do not provide the fairness index in this section. The reason is that in a mobile environment, nodes move continuously and hence the bottlenecks change continuously which makes the metric difficult to calculate and interpret.

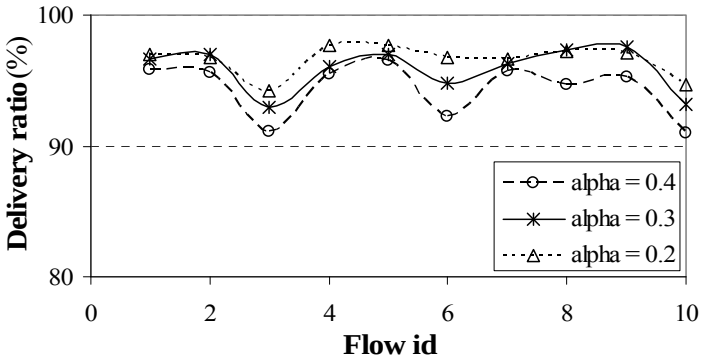

Figure 15. Delivery ratio in mobile topology.

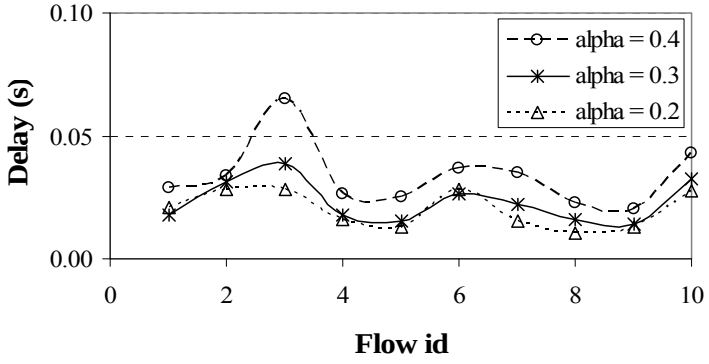

Figure 16. Delay in mobile topology.

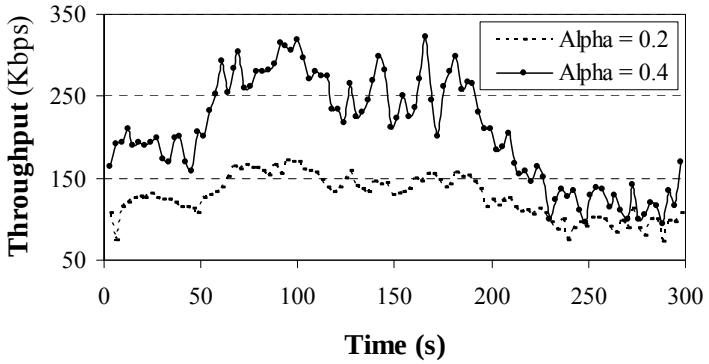

Figure 17. Instantaneous throughput of flow 1.

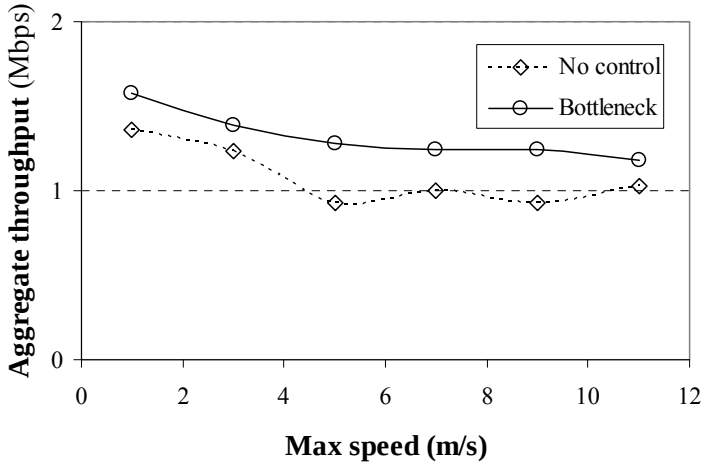

Figure 18. Throughput in different mobility.

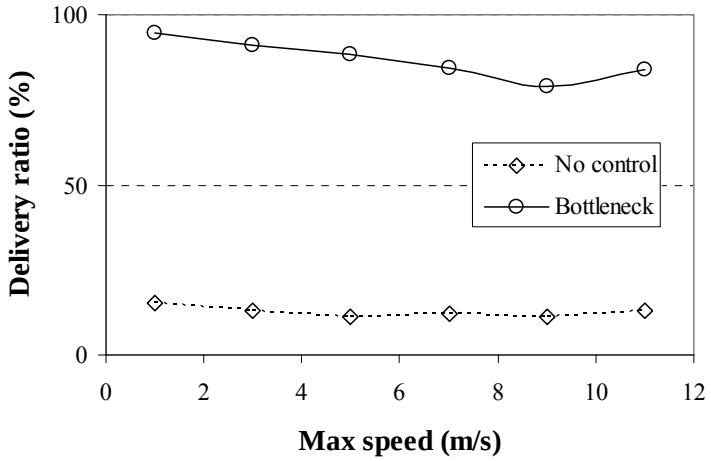

Figure 19. Delivery ratio in different mobility.

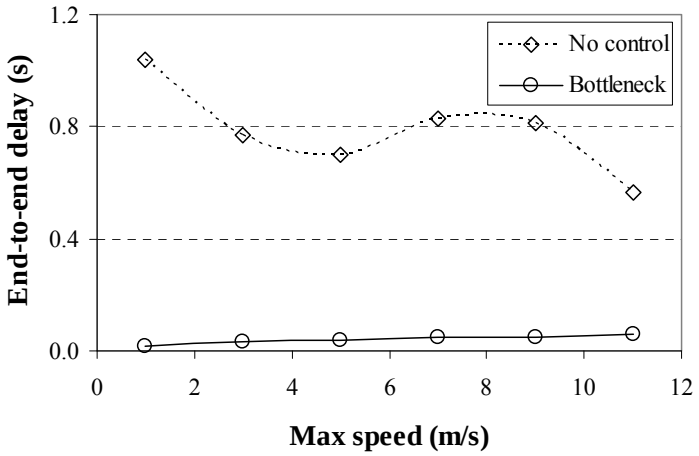

Figure 20. Delay in different mobility.

4.4 Columbia University Campus

In this simulation, we apply our flow control scheme to Columbia University Campus network and compare the result to our previous research that use a macro model of a wireless network to identify potential bottlenecks and control the flows only across those bottlenecks[10], [11]. 600 nodes are randomly deployed and 20 flows are selected randomly. α is set to 0.2. We pre-compute the shortest path routes for each flow in the simulation.

Fig. 21 shows the different rate allocations for three models. The experimental model implements a progressive water filling algorithm to determine a max-min fair rate allocation. Our earlier research constructs a macro model of the Columbia University network. The macro model identifies areas where flows are likely to be constrained as they pass between obstacles, and reduces the network to super nodes, that are clusters of mobile nodes without bottlenecks, and links, that are the capacity of the bottlenecks interconnecting the super nodes. The earlier work uses the model to apply max-min fairness only at the bottlenecks. A more

detailed description of these two techniques is in [10], [11].

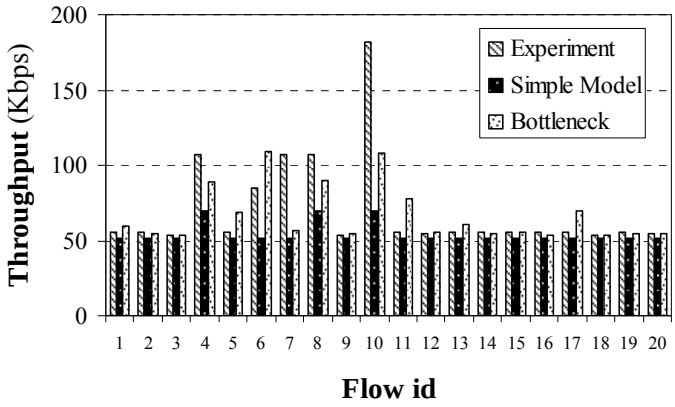

Figure 21. End-to-end throughput for three models.

All three models find similar rates allocated to the smallest flows. As we move beyond the smallest flows there are differences. Our distributed bottleneck flow control model is closer to the experimental model than our previous macro model. There are some flows that have larger rates than those determined by the experimental model. For example, flow 5, 6, 11, and 17. This is because at the bottleneck, nodes have different views of the channel and some flows get a larger fraction of the available capacity. The larger flow isn't necessarily fair, since it may take bandwidth from a user with a smaller flow. For instance, flow 11 may pass the edge of a bottleneck and experience less interference while other flows pass the center of that bottleneck and see a busier channel. Therefore, flow 11 may get a higher rate than other flows. This effect has been investigated in figure 7 and discussed in section *4.1*.

In [24], a max-min fairness index is proposed that measures the departure of a rate allocation from a max-min fair rate allocation. It is defined as:

$$F' = (\sum_{i=1}^{n} f_i^* f_i)^2 / (\sum_{i=1}^{n} f_j^{*2} \cdot \sum_{i=1}^{n} f_i^2). \quad (16)$$

Here f^* is the max-min fair rate allocation. The max-min fairness index value is between 0 and 1. This measure is different from the other fairness metrics that we use in this paper in that it requires that we know the true max-min fair values, rather than comparing the allocations obtained by a specific technique.

When we use the rate allocation obtained by the experimental model as the true max-min rate allocation, according to (16), our flow control scheme has a max-min fairness index of 0.921 while our previous macro estimation model has a max-min fairness index of 0.892. Our simple, distributed bottleneck flow control scheme achieves a better max-min fair rate allocation than our previous centralized approach that constructs a macro model.

The modified Jain's fairness index for our rate allocation in the Columbia University Campus network is 0.9937. [0.9918, 0.9956] is the 95% confidence interval. This shows our

scheme can provide fair bandwidth access in networks with obstacles.

In the Columbia network, each flow has a bottleneck node that constrains its rate. In Fig. 22, we plot the locations of all bottleneck nodes. According to our previous result in [9], [10], [11], most bottleneck nodes are located at bottlenecks identified by our network partition rule. When a bottleneck node is just one hop away from our bottleneck links, this node is called "on bottlenecks"; if it is two hops away, it is called "near bottlenecks"; and the remaining bottleneck nodes are called "far away". Table 3 lists the percentage of each category. The "on and near" nodes account for more than 90% of the total bottleneck nodes. For the remaining 10% nodes, most of them come from short flows that do not pass any bottleneck links and should be considered as local traffic. Therefore, the performance of our flow control scheme in the Columbia campus network verifies our previous work to predict the location of bottlenecks.

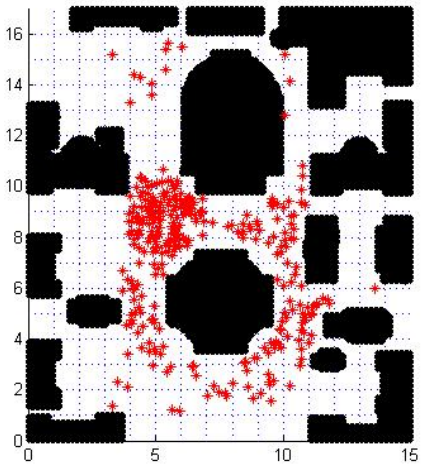

Figure 22. Bottleneck nodes' location.

Table 3. Bottleneck nodes.

Total number	447	%
On bottlenecks	277	61.97%
Near bottlenecks	129	28.86%
Far away	41	9.17%
On & Near	406	90.83%

5. Conclusion

In this paper, we present a distributed bottleneck flow control algorithm in MANETs. Our scheme uses techniques from bottleneck flow control and bandwidth balancing in DQDB.

It extends bandwidth balancing to operate in a mobile wireless environment. Through simulation, we show that our protocol can effectively allocate network resources in a dynamic environment and hence can improve QoS for real-time applications.

Since our protocol can be implemented in a distributed way, it is more suitable than other centralized techniques in MANETs. The protocol uses a very simple distributed rule: every node monitors its channel condition and only uses local information to allocate a portion of its residual capacity to any flow that passes through the node.

Simulation results show that our flow control scheme significantly improves the end-to-end throughput compared to the scheme without flow control in static networks. Our scheme outperforms the scheme without flow control, in terms of throughput, delivery ratio, and delay, in all mobility scenarios. Our scheme also provides effective flow control and obtains a fairness index above 0.95 in all of the cases we have investigated.

References

[1] C. Lochert, B. Scheuermann, M. Mauve, “A survey on congestion control for mobile ad hoc networks”. Wireless Communications and Mobile Computing. Vol. 7, Issue 5, Pp 655-676. June 2007. http://dx.doi.org/10.1002/wcm.524

[2] H. Zhai, Y. Fang, “Distributed flow control and medium access in multihop ad hoc networks”. IEEE Transactions on Mobile Computing. Vol. 5, Issue 11, Pp 1503-1514. Nov. 2006. http://dx.doi.org/10.1109/TMC.2006.166

[3] H. Zhai, X. Chen, Y. Fang, “Rate-based transport control for mobile ad hoc networks”, The 2005 IEEE Wireless Communications and Networking Conference (WCNC'05). New Orleans, LA, March 13-17, 2005. http://dx.doi.org/10.1109/WCNC.2005.1424868

[4] K. Chen, K. Nahrstedt, N. Vaidya, “The utility of explicit rate-based flow control in mobile ad hoc networks”, The 2004 IEEE Wireless Communications and Networking Conference. Atlanta, GA, March 21-25, 2004. http://dx.doi.org/10.1109/WCNC.2004.1311847

[5] Y. Yang, R. Kravets, “Contention-aware admission control for ad hoc networks”. IEEE Transactions on Mobile Computing. Vol. 4, Issue 4, Pp 363-377. July/Aug. 2005. http://dx.doi.org/10.1109/TMC.2005.52

[6] L. Chen, W. Heinzelman, “QoS-aware routing based on bandwidth estimation in mobile ad hoc networks”. IEEE Journal on Selected Areas in Communications. Vol. 23, Issue 3, Pp 561-572. March 2005. http://dx.doi.org/10.1109/JSAC.2004.842560

[7] N. F. Maxemchuk, C. Zhou, “A macro model of frequently changing mobile networks to perform flow and access control”, The second International Conference on Broadband Networks (BroadNets 2005). Boston, MA, Oct. 7-7, 2005. http://dx.doi.org/10.1109/ICBN.2005.1589636

[8] N. F. Maxemchuk, C. Zhou, “A macro model of frequently changing mobile networks to perform flow and access control”. Mobile Networks and Applications. Vol. 11, Issue 5, Pp 649-659. Oct. 2006. http://dx.doi.org/10.1007/s11036-006-7793-x

[9] C. Zhou, N. F. Maxemchuk, "Applying a macro model of ad hoc networks to access control", The seventh International Conference on Networking (ICN 2008). Cancun, Mexico, April 13-18, 2008. http://dx.doi.org/10.1109/ICN.2008.50

[10] C. Zhou, N. F. Maxemchuk, "Scalable max-min fairness in wireless ad hoc networks", The first International Conference on Ad Hoc Networks (ADHOCNETS 2009). Niagara Falls, Canada, Sep. 23-25, 2009. http://dx.doi.org/10.1007/978-3-642-11723-7

[11] C. Zhou, N. F. Maxemchuk, "Scalable max-min fairness in wireless ad hoc networks". Ad Hoc Networks. Vol. 9, Issue 2, Pp 112-119. March 2011. http://dx.doi.org/10.1007/978-3-642-11723-7_6

[12] D. Bertsekas, R. G. Gallager, "Data Networks", 2nd ed.: Prentice Hall, 1992.

[13] J. Jaffe, "Bottleneck flow control". IEEE Transactions on Communications. Vol. 29, Issue 7, Pp 954-962, July 1981. http://dx.doi.org/10.1109/TCOM.1981.1095081

[14] E. L. Hahne, A. K. Choudhury, N. F. Maxemchuk, "Improving the fairness of distributed-queue dual-bus networks", The IEEE INFOCOM'90. San Francisco, CA, June 3-7, 1990. http://dx.doi.org/10.1109/INFCOM.1990.91247

[15] E. L. Hahne, A. K. Choudhury, N. F. Maxemchuk, "DQDB networks with and without bandwidth balancing". IEEE Transaction on Communications. Vol. 40, Issue 7, Pp 1192-1204, July 1992. http://dx.doi.org/10.1109/26.153363

[16] J. Li, C. Blake, D. D. Couto, H. Lee, R. Morris, "Capacity of ad hoc wireless networks", The seventh ACM International Conference on Mobile Computing and Networking (MobiCom'01). Rome, Italy, July 16-21, 2001. http://dx.doi.org/10.1145/381677.381684

[17] K. Sanzgiri, I. Chakeres, E. Belding-Royer, "Determining intra-flow contention along multihop paths in wireless networks", The first International Conference on Broadband Networks. San Jose, CA, Oct. 25-29, 2004. http://dx.doi.org/10.1109/BROADNETS.2004.32

[18] GloMoSim. Available at: http://pcl.cs.ucla.edu/projects/glomosim/

[19] X. Zeng, R. Bagrodia, M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks", The 12th Workshop on Parallel and Distributed Simulations (PADS'98). Banff, Canada, May 26-29, 1998. http://dx.doi.org/10.1109/PADS.1998.685281

[20] IEEE 802.11 WG. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In *IEEE*, 2003. http://dx.doi.org/10.1109/IEEESTD.2003.95617

[21] R. Jain, D. Chiu, W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems". DEC Research Report TR-301. Sep. 1984.

[22] D. B. Johnson, D. A. Maltz, "Dynamic source routing in ad hoc wireless networks". Mobile Computing. Vol. 353, Pp 153-181, 1996. http://dx.doi.org/10.1007/978-0-585-29603-6_5

[23] C. E. Perkins, E. M. Royer, S. R. Das, M. K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks". IEEE Personal Communications. Vol. 8, Issue 1, Pp 16-28, Feb. 2001. http://dx.doi.org/10.1109/98.904895

[24] B. Radunovic, J.-Y. Le Boudec, "Rate performance objectives of multihop wireless networks". IEEE Transactions on Mobile Computing. Vol. 3, Issue 4, Pp 334-349, Oct.-Dec. 2004. http://dx.doi.org/10.1109/TMC.2004.45