Enhancing MAC Coordination to Boost Spatial Reuse in IEEE 802.11 Ad Hoc Networks

Fengji Ye, Hua Yang, and Biplab Sikdar
Department of ECSE, Rensselaer Polytechnic Institute
Troy, New York 12180

Abstract— **In a wireless ad hoc network where multi-hop traffic dominates the network, spatial reuse has an enormous impact on the network performance in terms of end-to-end throughput and delay characteristics. In this paper, we investigate the MAC coordination of persistent flows in 802.11 ad hoc networks and point out that the aggressive behavior of 802.11 MAC can throttle the spatial reuse and reduce bandwidth efficiency. We thus propose an adaptive layer-2 pacing scheme fully compatible with 802.11 MAC using explicit MAC feedback to balance the transmissions on adjacent nodes. By promoting MAC coordination, our scheme can assist the MAC to operate around its saturation state while minimizing resource contention. Experiment results demonstrate that our scheme significantly outperforms the original 802.11 MAC by boosting the throughput while still maintaining latency at a low level.**

I. INTRODUCTION

Over the past few years the success of IEEE 802.11 technology has led to rapid proliferation of wireless LANs and also made IEEE 802.11 MAC protocol [1] the *de facto* multiple access standard for wireless ad hoc networks. As an essential performance benchmark in ad hoc environment, *spatial reuse* dominates the number of simultaneous communications allowed in a given region, which in turn strongly affects the aggregate throughput and delay characteristics of the network. This paper investigates the spatial reuse efficiency in static or low-mobility ad hoc networks in the presence of multi-hop traffic, and further proposes to promote coordination among neighboring nodes on the link layer to leverage spatial reuse.

Prior research has suggested that 802.11 MAC does not function well in multi-hop environment since it has intrinsic flaws in combating the hidden and exposed terminal problems [2], [3]. In an ad hoc network, when a persistent data flow travels from source to destination over multiple hops, the pipeline efficiency along the path becomes a dominant factor affecting the throughput and latency as the hop count grows. The *pipeline efficiency*, characterized by simultaneous use of the same spectrum along the path of the data flow, relies on the coordination of transmissions at each relay node. However, current wireless MAC protocols only govern single-hop packet delivery based on the interference information collected within the scope of a single hop, thus lacking support of concerted transmissions among relay nodes in a larger area. As a result, directly applying the existing MAC protocols in multi-hop networks usually leads to sub-optimal spatial reuse and performance degradation.

In this paper, we shed light on the spatial reuse of persistent multi-hop traffic in 802.11 ad hoc networks. The ideal spatial reuse should operate around its saturation state, where bandwidth efficiency is maximized while incurring minimum resource contention. Our study reveals that the 802.11 MAC tends to over-utilize the spectrum by attempting more simultaneous transmissions than being allowed in a neighborhood, and thereby is prone to collisions and involuntary packet drops on the MAC layer. Specifically, the aggressive behavior of 802.11 MAC leads to unnecessary RTS failures and ensuing spectrum wastage. We further show that by properly tuning its transmission rate, a "polite" relay node who yields channel access to others can nevertheless deliver better overall performance. These observations motivate us to employ adaptive pacing to orchestrate the transmissions of relay nodes in a more balanced manner, which is verified by simulations showing performance improvements over original 802.11 MAC.

According to our scheme, for an active node in the network, the traffic information in a multi-hop neighborhood is collected by its MAC via a newly introduced MAC signal. The multi-hop traffic inference is then conveyed to a link layer traffic shaper to assist dynamic adjustment of transmission rate. As opposed to the TCP pacing in literature [10], [11], [12], which is a mixture of rate-based transmission control and congestion control on a per flow basis, we propose to apply adaptive pacing directly on the link layer. Our layer-2 pacing scheme is transparent to upper layer protocols, involving no cross-layer design issues and thus representing a lightweight pacing scheme. It makes use of instant MAC feedbacks for immediate traffic coordination, resulting in faster convergence.

The rest of the paper is organized as follows. In Section II we examine the spatial reuse of persistent multi-hop traffic in 802.11 ad hoc networks. Our layer-2 adaptive pacing scheme is presented in Section III and then evaluated by simulations in Section IV. We give a literature survey in Section V and conclude the paper in Section VI.

II. SPATIAL REUSE AND PIPELINE EFFICIENCY IN MULTI-HOP NETWORKS

We begin our discussion with an investigation of IEEE 802.11 *Distribution Coordination Function* (DCF) in terms of its medium reservation mechanism. We will show that it suffers from low pipeline efficiency in the presence of persistent multi-hop traffic and thus throttles the spatial reuse.

A. Diagnosing 802.11 DCF

The IEEE 802.11 MAC accomplishes contention-based access using DCF which distributes the channel reservation information by announcing the impending use of the medium via RTS/CTS frames. The exchange of RTS/CTS frames between sender and receiver prior to the actual data frame intends to identify hidden terminals. However, it is shown that there are effectiveness issues in 802.11 DCF when it reserves the channel in the single-hop neighborhood of sender and receiver [4], [5]. In this section, we further examine the behavior of 802.11 DCF in multi-hop scenarios where a persistent flow traveling across multiple hops attempts to maximize its end-to-end throughput.

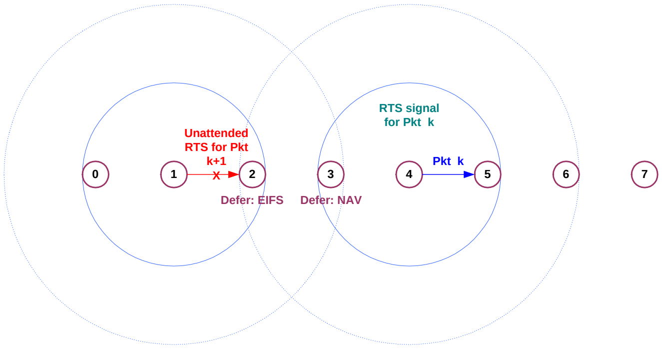

Fig. 1. A persistent flow traverses a chain and generates unattended RTS at node 1. Dotted line circle stands for carrier sense range while solid line circle stands for transmission range. Node 2, in its deferral state when hearing the RTS from node 1, deliberately ignores the RTS and causes its failure.

We use the example in Figure 1 in which a persistent flow traverses a chain from node 0 to node 7. At some time node 4 receives the k-th packet of the persistent flow and continues to forward it to node 5, with a preceding RTS. Let us assume in the neighborhood of node 4, node 3 and 5 are within its transmission range (solid line circle centered at node 4 in Figure 1), while node 2 and 6 are out of its transmission range but within its carrier sense range (dotted line circle). According to 802.11 DCF, node 3 sets its *Network Allocation Vector* (NAV) according to the duration field embedded in the RTS frame and starts to defer until NAV counts down to zero. While node 2 senses the signal in the channel but cannot correctly decode it, it still defers for a duration equal to the extended interframe space (EIFS). Suppose that the flow is uninterrupted and packet $k+1$ arrives at node 1, who would sense the channel to be idle and thus initiate an RTS to node 2. Since node 2 is in its deferral state and cannot reply with a CTS, node 1 would attempt the RTS repeatedly until the RTS/CTS handshake eventually gets through.[1] We refer to these vain RTS attempts as *unattended RTS*, since its intended receiver purposely ignores the RTS owing to the deferral.

In 802.11, repetitive unsuccessful RTS is harmful because any node that overhears it may defer access unnecessarily. From a MAC layer perspective, unsuccessful RTS commonly takes place in three scenarios:

- RTS collides with other packets at the intended receiver.
- Returned CTS collides with other packets at the sender.
- Receiver intentionally ignores the RTS (unattended RTS).

In any of the scenarios above, the sender fails to receive the responding CTS and would retransmit the RTS. Although they are not involved in collisions, unattended RTS frames still result in RTS failures and may prompt the sender to repeat unsuccessful RTS attempts, making the situation even worse. According to our observations, unattended RTS can take up as much as 70% RTS failures and may thus account for significant performance degradation.

B. Pipeline Efficiency and Link Layer Coordination

In 802.11 DCF, a backlogged node always attempts to transmit whenever it considers the channel is clear in its vicinity, through either physical carrier sensing or virtual carrier sensing achieved by RTS/CTS handshake [1]. As we have seen in Figure 1, this best-effort strategy at an individual node, however, may lead to aggressive behavior along a pipeline. Ironically, lower pipeline efficiency may result from this, because the unattended RTS, a potential performance killer, makes false channel reservation and thus wastes the spectrum.

Our solution to improve the pipeline efficiency is to enhance link layer coordination among the neighboring nodes. We propose adaptive pacing to help distribute persistent traffic among relay nodes in a more balanced way. We show that this approach is effective at boosting the spatial reuse and end-to-end throughput. Another approach is to adjust the carrier sense range [7], which is beyond the scope of this paper.

III. ADAPTIVE PACING FOR IMPROVED SPATIAL REUSE

In this section, we present our adaptive pacing scheme in detail. It makes necessary changes to the link layer architecture and introduces a simple modification to the 802.11 MAC frame. Our scheme is fully compatible with the original 802.11 MAC, so a smooth migration is possible for deployment.

A. Locking Traffic in Pace

We begin with a simulation study which paces the CBR (constant bit rate) traffic in a chain topology to gain insights into the link layer coordination. We use the same 8-node chain topology as in Figure 1 with UDP/CBR traffic pumped continually from the left to the right in a network of 2Mbps bandwidth. Each UDP packet has 512 bytes, carrying CBR traffic. Figure 2 shows the end-to-end measurements in terms of transmitted and received packets in *ns-2* with regard to CBR traffic interval.

It is seen that when CBR traffic has a fast pace, i.e., with a small interval, it suffers a considerable packet loss. This happens when packet delivery gets saturated along the pipeline and cannot grow even if more packets are injected into the network. On the other hand, as the CBR traffic slows down, the received packets can pick up at some point and eventually

---

[1] In IEEE 802.11 DCF, an upper limit is imposed on maximum allowed RTS retransmissions.

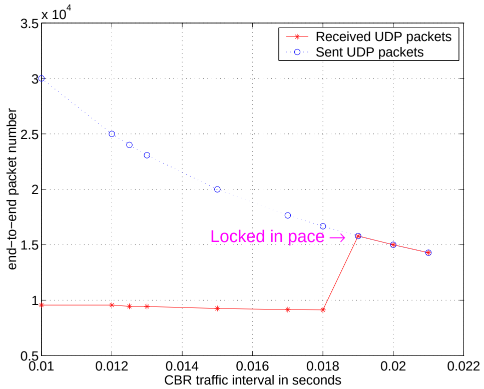

Fig. 2. The CBR traffic is pumped into an 8-node chain. The x-axis represents the CBR interval, and the y-axis represents end-to-end measurements in terms of transmitted and received UDP packets. At some point the CBR traffic rate is locked with maximum received packets and no packet drops in the pipeline.

get locked to the transmission rate. The end-to-end throughput reaches its peak when pace is locked, where the best tradeoff is achieved between hidden terminals and exposed terminals. It is interesting to notice that a persistent flow may acquire higher throughput if it properly yields channel access and maintains its pace in harmony, which in turn suggests that aggressive behaviors usually receive penalties.

In real world scenarios, however, not all kinds of flows can inherently achieve the desired pacing. TCP, as a common example, is known to generate bursty traffic and has poor interactions with the underlying 802.11 MAC in multi-hop environment [11], [12]. According to its window based congestion control algorithm, a single TCP ACK may trigger multiple transmissions of TCP packets from the sender. The sender's MAC then performs in a best-effort manner and pumps data into the pipeline regardless of the proper pace. As we have explained, any aggressive transmission attempt along the pipeline is penalized due to the spatial reuse and may be subject to packet drops, causing substantial performance degradation. We identify the above problem as *pipeline syndrome* for TCP. The congestion control algorithm in TCP also leads to other deficiencies in 802.11 ad hoc networks, e.g., false indication of link congestion, which we do not cover in this paper. To sum up, TCP brings burstiness to multi-hop networks and may potentially hurt the spatial reuse. We will show that the pipeline syndrome can be fixed using our pacing scheme without affecting the end-to-end semantics of TCP.

B. Transmission Rate Control on the Link Layer

In our scheme, pacing is directly applied on layer-2, as opposed to the TCP pacing in [10], [11], [12]. The reason is threefold. First, this implementation is transparent to upper layer protocols. As we have seen, pacing proves to be helpful to persistent traffic, which may be composed of TCP traffic or UDP traffic, or both. As far as spatial reuse is concerned, we do not need to differentiate the flows on the link layer. Our approach is capable of handling hybrid traffic in a unified framework and does not involve cross-layer design issues.

Second, TCP pacing routinely works with congestion control algorithm on an end-to-end basis. In reality, however, a TCP flow may travel across both wired and wireless networks. This is often seen in a multi-hop wireless access network, e.g., a wireless mesh network, where traffic is skewed and often directed to Internet gateways. Since TCP calls for different end-to-end control strategies in wired and wireless environment, TCP pacing [11] does not offer a universal solution effective to heterogeneous networks. However, layer-2 pacing only applies to the wireless stations and thus evades this problem.

Finally, layer-2 pacing can make use of instant single-hop feedback for timely MAC coordination among neighboring nodes, which results in faster convergence as compared to any transport layer implementation. Therefore, adaptive pacing on the link layer is a more desirable solution than end-to-end rate control in wireless ad hoc networks.

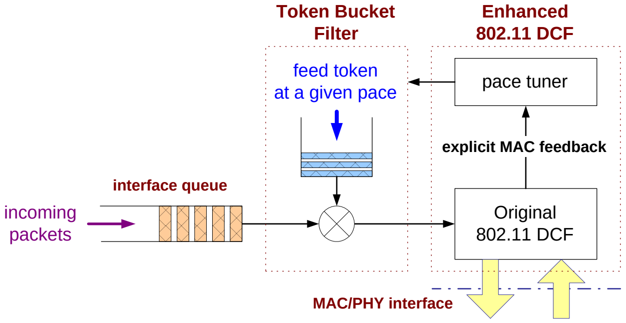

Fig. 3. This is a link layer diagram in which token bucket filter is used as a traffic shaper to enforce adaptive pacing. Tokens are issued at a dynamic pace set forth by the pace tuner which works with 802.11 DCF as an enhanced feature. Explicit MAC feedback is introduced to assist pace adjustment on the fly.

To implement pacing on the link layer, we propose to use *Token Bucket Filter* (TBF) to smooth traffic and provide support to MAC coordination, as shown in Figure 3. In this link layer diagram, a TBF sits between the interface queue and the MAC function. TBF is a pure traffic shaper to filter the traffic based on the expenditure of tokens and can effectively alleviate burstiness. In our scheme, adaptive pacing is accomplished by issuing tokens at a dynamic pace set forth by the pace tuner, which coordinates the transmission rate with neighboring nodes through explicit MAC feedback (to be discussed in Section III-C). In this way, the link layer at each node can work distributively towards balanced traffic in the node's vicinity.

It is worth mentioning that the optimal pace for persistent traffic usually exceeds the maximum backoff window allowed in 802.11. For example, in Figure 2 optimality is achieved when CBR traffic interval is approximately 19 milliseconds, indicated by the receiving peak value. Therefore we do not implement pacing inside the 802.11 DCF by adding extra backoff time for outgoing packets.

C. Introducing Explicit Feedback to 802.11 DCF

In Section II-A, we noted that unattended RTS represents an early indication of throttled spatial reuse in the sense that aggressive transmission attempts can overwhelm a pipeline.

Thus we can use it as a feedback for triggering a MAC sender to adjust its pace so that the MAC may probe for the optimal transmission rate in the neighborhood. This is the mechanism of adaptive pacing. We further enhance the 802.11 DCF to incorporate this explicit MAC feedback by making moderate modifications on the receiver and sender side, respectively.

1) Modifications on the MAC receiver side: MAC receiver is responsible for tracking any unattended RTS frames and conveying this information to MAC sender. In our scheme, we use CTS frame as the feedback carrier because it is a short control frame and has some vacant fields available in its frame format. According to the 802.11 standard [1], inside the 2-byte Frame Control field of the CTS frame, there are two unused subfields named More Fragments field and Retry field, respectively, with each taking up one bit and always set to zero. Therefore, we can make use of these two single-bit fields to deliver the pacing feedback to the sender while keeping our scheme compatible with the original 802.11 DCF. We thus introduce two new bits to replace unused old fields: *EPF bit* for backward compatibility, and *SLW bit* for pace tuning.

- **EPF (Explicit Pacing Feedback)**: This bit is set to 1 if explicit pacing feedback is enabled on the receiver node. For backward compatibility, it is set to 0 on non-pacing nodes.
- **SLW (Slow Pacing)**: This bit is set to 1 by the MAC receiver if it successfully receives but intentionally declines at least one RTS request due to deferral since its last CTS transmission; otherwise it is set to 0. It is used to inform the MAC sender whether its transmission rate is too fast causing unattended RTS and thus should be slowed down. The SLW bit is always set to 0 on non-pacing nodes.

2) Modifications on the MAC sender side: As soon as a MAC sender receives the CTS frame containing the pacing feedback, it uses the token bucket filter to update its transmission rate. The whole process is explained in Algorithm 1. Since our scheme assumes the same mechanism for contention-based access as in 802.11 DCF, all routine backoff or deferral operations are omitted in this algorithm.

As shown in Algorithm 1, for each outgoing RTS, the MAC sender starts a timer to wait for the corresponding CTS, and retransmits RTS in case of timeout.[2] The total retransmission attempts should not exceed $ShortRetryLimit$, set to 7 in 802.11 [1]. Once the expected CTS is received, the sender proceeds to retrieve its EPF bit to check if pacing feedback is carried in this CTS frame. Whenever feedback is available, the pace should decrease if SLW bit is set to 1, or increase if it is set to 0. Pace tuning is performed through changing the TBF token issue rate.

To approach the optimal rate promptly, the sender varies its pace whenever a new feedback is received. The pace update method can be either linear or multiplicative. In the latter case, the TBF token interval is multiplied by a ratio every time a new feedback is received. More sophisticated pace adjustment algorithms can be attempted by combining the linear and multiplicative methods. In general, fine tuning of transmission rate is subject to the degree of traffic fluctuation.

Algorithm 1 MAC sender pace tuning using explicit pacing feedback

```
$retryCount \leftarrow 0$
transmit RTS frame and initiate a timer
while CTS not received before timeout do
    if $retryCount < ShortRetryLimit$ then
        $retryCount \leftarrow retryCount + 1$
        retransmit RTS frame and restart the timer
    else
        abort transmission and notify upper layer
        QUIT the algorithm
    end if
end while
check the validity of the received CTS frame
if EPF bit is set to 1 then
    if SLW bit is set to 1 then
        decrease TBF token issue rate
    else
        increase TBF token issue rate
    end if
end if
proceed to transmit DATA frame
```

In our scheme, for any overwhelmed node that is not able to answer every incoming RTS, it should be allowed to send a slow-pacing feedback (SLW=1) to any aggressive sender in its neighborhood, whether this sender generates the latest unattended RTS or not. In this regard, we are not interested in the specific reason behind each feedback on which sender should be held accountable for the latest unattended RTS. In other words, the SLW bit in a feedback may be set by an unattended RTS that belongs to a different persistent flow from the one whose CTS carries this feedback. This should not compromise the effectiveness of our scheme because we only take actions on individual nodes and do not differentiate between flows on a specific node as long as spatial reuse is the sole concern.

It is seen that any node using our enhanced 802.11 DCF with adaptive pacing, whether it acts as a sender or receiver at some instant, can seamlessly work with any other non-pacing original 802.11 node, if any. This demonstrates full compatibility of our scheme and guarantees that it can have a smooth technical migration.

IV. EXPERIMENT RESULTS

In this section we evaluate the performance of our scheme via simulations in *ns-2* with different network topologies. First we revisit the 8-node chain topology in Figure 1 which is traversed by a TCP flow from the left to the right. We assume fixed-rate pacing and use x-axis to represent the token issue interval for TBF, as shown in Figure 4. The end-to-end TCP transmitted/received packets are measured over 30sec and

---

[2] According to 802.11 DCF [1], the sender should increase the contention window and perform backoff before attempting RTS retransmission.

compared with the results from non-pacing original 802.11 protocol using the same topology, which are denoted by two horizontal lines in this figure. It is seen that at the optimal pace TCP throughput reaches its peak value, which is approximately 167% higher than the throughput without pacing. It suggests that pacing can significantly improve TCP performance by coordinating the traffic along the pipeline.

We then evaluate our adaptive pacing scheme using the grid topology in Figure 5. All the nodes, each equipped with a single transceiver with 2Mbps bandwidth, are placed on an 8 by 8 grid network with minimum separation distance of 200m (the transmission range in *ns-2* is 250m). k flows traverse across the network, while at the same time another k flows traverse down, with $k < 7$. All flows carry either TCP or UDP traffic over 7 hops, with packet size of 512 bytes. *AODV* is used as the ad hoc routing protocol. For the adaptive pacing algorithm at each node, we assume the depth of the token bucket is sufficiently large. We use multiplicative method for pace adjustment with 40msec as the initial pace, and at each pace adjustment we increase pace by 10% or decrease pace by 5%. All end-to-end measurements are made over 30sec on the MAC layer.

Figure 6 shows the number of transmitted/received TCP packets on a per flow basis. The x-axis represents the total number of flows in the given grid network. We see that due to the limited network capacity, per flow throughput goes down as more flows are added to the network. Our pacing scheme achieves considerably higher throughput than the original 802.11 MAC, while the loss rates stay comparable.

Figures 7 and 8 demonstrate the end-to-end throughput and delay measurements from the corresponding UDP experiment in the given grid network. CBR traffic is assumed and has a transmission interval of 10msec, which can sufficiently saturate the pipeline and thus suffers from worse packet drop. We observe that our scheme outperforms the original 802.11 MAC on the per flow UDP throughput, while its latency is not adversely affected despite the extra delay introduced by pacing on the link layer.

We also ran the simulation in a large random network with 228 nodes homogeneously spread across a 1600m by 1600m region. All the simulation settings other than the topology remain the same with the grid network. As shown in Figure 9, although the throughput advantage has shrunk as compared to that of grid networks, we still achieve higher TCP throughput when adaptive pacing is turned on.

V. RELATED WORK

Recently the spatial reuse efficiency in IEEE 802.11 ad hoc networks has attracted extensive research attention. In [2], Li *et al.* evaluate the influence of interference range on the network throughput from the perspective of spatial reuse. The paper examines the interaction of the 802.11 MAC and ad hoc forwarding via simulations and analysis by employing a simplified spatial reuse model, in which the transmission and interference ranges are assumed to be fixed. In [3], Xu *et al.* use TCP simulations to show that 802.11 MAC does not function well in multi-hop networks.

The *Signal to Interference and Noise Ratio* has been used to evaluate the scope of the interference in ad hoc networks, leading to a better understanding of the spatial reuse efficiency for 802.11 DCF [4], [5], [6], [7]. In [4], the authors show that the RTS/CTS handshake is not always effective because the power needed for interrupting a packet reception is much lower than that for delivering a packet successfully. The paper thus concludes that the virtual carrier sensing (VCS) implemented in the 802.11 DCF cannot prevent all the interference as expected in the design. The authors in [5] further investigate the effectiveness of the 802.11 VCS scheme through three scenarios in which the spatial reuse exhibits distinct characteristics. The paper also proposes a simple scheme to help mitigate the spatial reuse problem. By pointing out the spatial reuse in 802.11 ad hoc mode is sub-optimal due to the deficiency in its channel reservation mechanism, the authors in [6] propose to incorporate distance information in the decision making process for the channel reservation. In [7], Zhu *et al.* propose to enhance the 802.11 physical carrier sensing with tunable sensing threshold to improve the spatial reuse. By setting optimal threshold values which are derived from analytical estimations, the authors claim that the scheme can potentially achieve higher aggregate network throughput.

Some researchers have found that the end-to-end flow control in TCP has enormous impact on the MAC efficiency in 802.11 ad hoc networks. While some of them develop variants of TCP to control the traffic rate, others seek alternatives to TCP tailored towards the characteristics of ad hoc environment. Sundaresan *et al.* in [8] and Chen *et al.* in [9] propose two new transport protocols for wireless multi-hop networks using pure rate-based transmission control. Instead of sending new packets into the network upon receiving acknowledgments for old packets, both protocols transmit packets at a predetermined rate, which is decided on the feedback from intermediate nodes along the path.

By combining rate-based transmission control with TCP congestion control, TCP pacing is proposed as a solution to conquer the burstiness of TCP traffic which may reduce the spatial reuse efficiency in ad hoc networks [10], [11], [12]. In [10], Aggarwal *et al.* present a comprehensive evaluation of TCP pacing for the Internet. The authors consider an implementation of pacing based on a leaky bucket algorithm with evenly spaced packet transmissions within the congestion window. The paper shows that this scheme leads to significantly less goodput than regular TCP but with better fairness.

In [11], Fu *et al.* point out that the TCP throughput could improve significantly over multi-hop wireless networks if congestion window should operate around the optimal size that maximizes the spatial reuse. The paper further studies the packet losses under different load conditions and also proposes to use adaptive pacing to balance traffic among intermediate nodes. In a recent work, [12] introduces a congestion control algorithm for TCP over 802.11 multi-hop networks in which a TCP sender adaptively sets its transmission rate using an

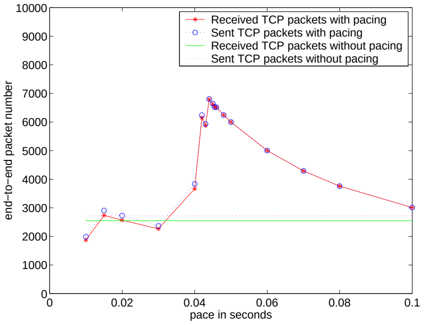

Fig. 4. The TCP traffic is pumped into an 8-node chain where fixed-rate pacing is enforced. The x-axis represents the token issue interval, and the y-axis represents end-to-end TCP measurements. For comparison we plot two horizontal lines representing transmitted and received TCP packets for the same topology but with pacing disabled.

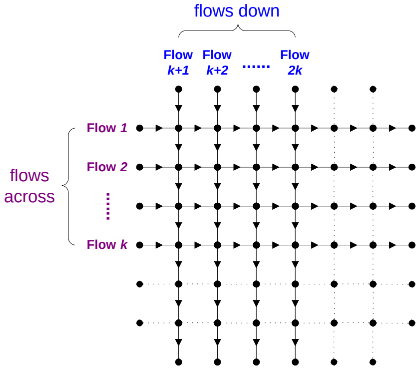

Fig. 5. This is an 8 by 8 grid topology with 200m as the minimum separation distance between neighboring nodes. At the same time k flows traverse across the network, while another k flows traverse down. Each flow has 7 hops and k is less than 7.

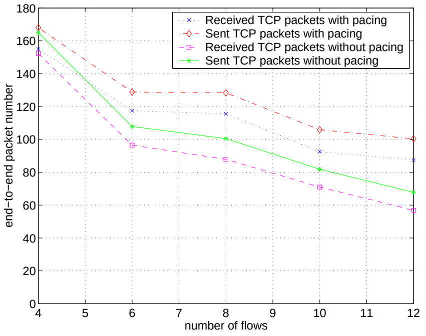

Fig. 6. Per flow end-to-end measurements for TCP traffic in the given grid network. The x-axis represents the number of flows in the network, and the y-axis represents transmitted and received TCP packets on a per flow basis.

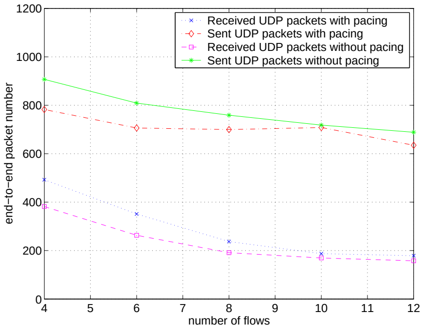

Fig. 7. Per flow end-to-end measurements for UDP traffic in the given grid network

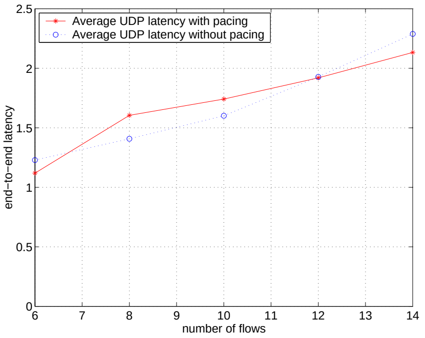

Fig. 8. Average end-to-end delay for UDP traffic in the given grid network

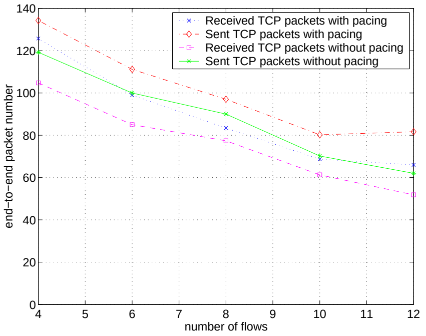

Fig. 9. Per flow end-to-end measurements for TCP traffic in a random network

estimate of the current four-hop propagation delay and the statistics of the recent round trip time.

VI. CONCLUSION

In this paper, we investigated the MAC coordination in ad hoc networks in the presence of persistent multi-hop flows from the perspective of spatial reuse. We proposed an adaptive pacing mechanism using a token bucket filter in 802.11 ad hoc networks to balance the transmissions on adjacent nodes for better spatial reuse. We introduced an explicit pacing feedback through a simple modification to the CTS frame format while maintaining its compatibility with the original 802.11 MAC. Simulation results demonstrated the performance improvements of our scheme over the original 802.11 MAC in terms of the end-to-end throughput.

REFERENCES

[1] ANSI/IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1999.

[2] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” *Proc. ACM MOBICOM 2001*, Rome, Italy, July 2001.

[3] S. Xu and T. Saadawi, “Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?” *IEEE Communications Magazine*, vol. 39, no. 6, pp. 130-137, June 2001.

[4] K. Xu, M. Gerla, and S. Bae, “How Effective is the IEEE 802.11 RTS/CTS Handshake in Ad Hoc Networks?” *Proc. IEEE GLOBECOM 2002*, Taipei, Taiwan, November 2002.

[5] F. Ye, S. Yi, and B. Sikdar, “Improving Spatial Reuse of IEEE 802.11 Based Ad Hoc Networks,” *Proc. IEEE GLOBECOM 2003*, San Francisco, CA, December 2003.

[6] F. Ye and B. Sikdar, “Distance-Aware Virtual Carrier Sensing for Improved Spatial Ruese in Wireless Networks,” *Proc. IEEE GLOBECOM 2004*, Dallas, TX, November 2004.

[7] J. Zhu, X. Guo, L. L. Yang, and W. S. Conner, “Leveraging Spatial Reuse in 802.11 Mesh Networks with Enhanced Physical Carrier Sensing,” *In Proc. IEEE ICC 2004*, Paris, France, June 2004.

[8] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar, “ATP: A Reliable Transport Protocol for Ad-hoc Networks,” *In Proc. ACM MOBIHOC 2003*, Annapolis, MD, June 2003.

[9] K. Chen, K. Nahrstedt, and N. Vaidya, “The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks,” *In IEEE WCNC 2004*, Atlanta, GA, March 2004.

[10] A. Aggrawal, S. Savage, and T. Anderson, “Understanding the Performance of TCP Pacing,” *In Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[11] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, “The Impact of Multihop Wireless Channel on TCP Throughput and Loss,” *In Proc. IEEE INFOCOM 2003*, San Francisco, CA, March 2003.

[12] S. M. ElRakabawy, A. Klemm, and C. Lindemann, “TCP with Adaptive Pacing for Multihop Wireless Networks,” *In Proc. ACM MOBIHOC 2005*, Urbana-Champaign, IL, May 2005.