Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 12, December 2013, pg.394 – 400

RESEARCH ARTICLE

Order Matters: Transmission Reordering In Mobile Adhoc Networks

P.Pradheepa[1], M.Baskar[2]

[1] Research Scholar, Department Of Computer Science, Vivekanandha College for Women, unjanai village, Tiruchengode, Tamil Nadu, India

[2] Assistant Professor, Department Of Computer Science, Vivekanandha College for Women, unjanai village, Tiruchengode, Tamil Nadu, India

[1] pradheepa0@gmail.com; [2] baskerm2u@gmail.com

ABSTRACT

A physical layer capability Message in Message was supported by modern wireless interface, MIM has capability to engage to strong signal reception from week signal reception. The proposed routing scheme jointly addresses the issues of learning and routing in an opportunistic context and the opportunity in MIM-aware reordering, characterizes the optimal improvement in throughput, and designs a link-layer protocol for enterprise wireless LANs to achieve it. Congestion in point-to-point transmissions in a wireless mesh network or underwater acoustic network can have debilitating effects on data transport through the network. The effects of jamming at the physical layer resonate through the protocol stack, providing an effective Denial-Of-Service (DoS) attack on end-to-end data communication. The simplest methods to defend a network against jamming attacks comprise physical layer solutions such as spread-spectrum or beam forming, forcing the jammers to expend a greater resource to reach the same goal. However, recent work has demonstrated that intelligent jammers can incorporate cross layer protocol information into jamming attacks, reducing resource expenditure by several orders of magnitude by targeting certain link layer and MAC implementations as well as link layer error detection and correction protocols.

Index Terms— Wireless LAN; wireless networks

1. INTRODUCTION

Congestion in point-to-point transmissions in a wireless mesh network or underwater acoustic network can have debilitating effects on data transport through the network. The effects of jamming at the physical layer resonate through the protocol stack, providing an effective denial-of-service (DoS) attack on end-to-end data communication. The simplest methods to defend a network against jamming attacks comprise physical layer solutions such as spread-spectrum or beam forming, forcing the jammers to expend a greater resource to reach the same goal. However, recent work has demonstrated that intelligent jammers can incorporate cross layer protocol information into jamming attacks, reducing resource expenditure by several orders of magnitude by targeting certain link layer and MAC implementations as well as link layer error detection and correction protocols. Hence, more sophisticated anti-jamming methods and defensive measures must be incorporated into higher-layer protocols, for example channel surfing or routing around jammed regions of the network.

The majority of anti- jamming techniques make use of diversity. For example, anti-jamming protocols may employ multiple frequency bands, different MAC channels, or multiple routing paths. Such diversity techniques help to curb the effects of the jamming attack by requiring the jammer to act on multiple resources simultaneously. In this project, we consider the anti-jamming diversity based on the use of multiple routing paths. Using multiple-path variants of source routing protocols such as Dynamic Source Routing (DSR) or Ad-Hoc On-Demand Distance Vector (AODV), for example the MP-DSR protocol, each source node can request several routing paths to the destination node for concurrent use. To make effective use of this routing diversity, however, each source node must be able to make an intelligent allocation of traffic across the available paths while considering the potential effect of jamming on the resulting data throughput. In order to characterize the effect of jamming on throughput, each source must collect information on the impact of the jamming attack in various parts of the network.

The Extent of jamming at each network node depends on a number of unknown parameters, including the strategy used by the individual jammers and the relative location of the jammers with respect to each transmitter-receiver pair.

The impact of jamming is probabilistic from the perspective of the network1, and the characterization of the jamming impact is further complicated by the fact that the jammers' strategies may be dynamic and the jammers themselves may be mobile. In order to capture the non-deterministic and dynamic effects of the jamming attack, we model the packet error rate at each network node as a random process. At a given time, the randomness in the packet error rate is due to the uncertainty in the jamming parameters, while the time-variability in the packet error rate is due to the jamming dynamics and mobility. Since the effect of jamming at each node is probabilistic, the end-to-end throughput achieved by each source-destination pair will also be non-deterministic and, hence, must be studied using a stochastic frame work.

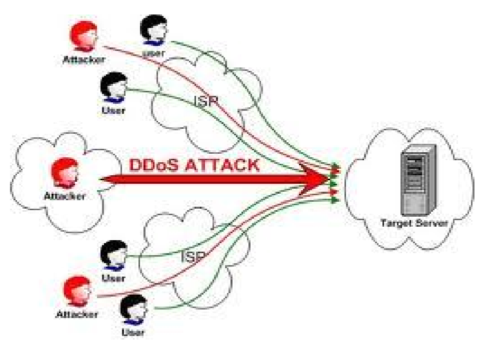

Fig 1.1.2 DDOS attack

2. RELATED WORK

We present practical models for the physical layer behaviors of packet reception and carrier sense with interference in static wireless networks. These models use measurements of a real network rather than abstract RF propagation models as the basis for accuracy in complex environments. Seeding our models requires N trials in an N node network, in which each sender transmits in turn and receivers measure RSSI values and packet counts, both of which are easily obtainable. The models then predict packet delivery and throughput in the same network for different sets of transmitters with the same node placements. We evaluate our models for the base case of two senders that broadcast packets simultaneously. We find that they are effective at predicting when there will be significant interference effects. Across many predictions, we obtain an RMS error for 802.11a and 802.11b of a half and a third, respectively, of a measurement-based model that ignores interference.

We analyze the carrier sensing and interference relations between the two wireless links and measure the impact of these relations on link capacity in two indoor 802.11a mesh network testbeds. We show that asymmetric carrier sensing and/or interference relations happen frequently in wireless networks; these asymmetric relations affect not only the level of performance degradation, but also the fairness of channel access. We then propose a new methodology that predicts the relation of carrier sensing and interference based on radio signal strength measurements. The measurement complexity increases only linearly with the number of wireless nodes.

To our knowledge, the proposed methodology is the first trial that considers physical layer capture, and detects the source of interference that is out of the communication range. We validate the prediction methodology on an 11-node wireless mesh network testbed.

We propose an enhancement to the IEEE 802.11 distributed coordination function (DCF). The enhancement improves the level of channel spatial reuse; thus, it improves overall network data throughput in dense deployments. Our modification, named the location-enhanced DCF (LED), incorporates location information in DCF frame exchange sequences so that stations sharing the communication channel are able to make better interference predictions and blocking assessments. Hence, more concurrent transmissions can be conducted in densely deployed wireless LANs. The potential performance enhancement of LED is studied both analytically and via ns-2 simulations. The results show that the LED method achieves significant throughput improvements over the original DCF.

In wireless networks, a frame collision does not necessarily result in all the simultaneously transmitted frames being lost. Depending on the relative signal power and the arrival timing of the involved frames, one frame can survive the collision and be successfully received by the receiver. Using our IEEE 802.11a wireless network testbed, we carry out a measurement study that shows the terms and conditions (timing, power deference, bit rate) under which this capture takes place. A recent measurement work on the capture in 802.11 networks argues that the stronger frame can be successfully decoded only in two cases: (1) The stronger frame arrives earlier than the weaker frame, or (2) the stronger frame arrives later than the weaker frame but within the preamble time of the weaker frame.

However, our measurement shows that the stronger frame can be decoded correctly regardless of the timing relation with the weaker frame. In explaining various capture cases we observe that the successful capture of a frame involved in a collision is determined through two stages: the preamble detection and the frame check sequence (FCS) check.

The combination of unlicensed spectrum, cheap wireless interfaces and the inherent convenience of untethered computing has made 802.11-based networks ubiquitous in the enterprise. Modern universities, corporate campuses and government routinely deploy scores of access points to blanket their sites with wireless Internet access. However, while the _ne-grained behavior of the 802.11 protocol itself has been well studied, our understanding of how large 802.11 networks behave in their full empirical complexity is surprisingly limited. In this paper, we present a system called Jigsaw that uses multiple monitors to provide a single view of all physical, link, network and transport-layer activity on an 802.11 network. To drive this analysis, we have deployed an infrastructure of over 150 radio monitors that simultaneously capture all 802.11b and 802.11g activity in a large university building (1M+ cubic feet). We describe the challenges posed by both the scale and ambiguity inherent in such an architecture, and explain the algorithms and inference techniques we developed to address them. Finally, using a 24-hour distributed trace containing more than 1.5 billion events, we use Jigsaw's global cross-layer viewpoint to isolate performance artifacts, both explicit, such as management interreferencies, and implicit, such as co-channel interference. We believe this is the first analysis combining this scale and level of detail for a production 802.11 network.

3. METHODOLOGY

3.1 MIM

We validate the existence of MIM capabilities in commodity hardware using a testbed of Soekris embedded PCs, equipped with Atheros 5213 chipsets running the MADWiFi driver. The experiment consists of two transmitters with a single receiver placed at various points in between. This subjects the receiver to varying SINRs. To ensure continuous transmissions from the transmitters, we modify the MADWiFi spacing's. To time-stamp transmissions, a collocated monitor is placed at each transmitter. Each monitor is expected to receive all packets from its collocated transmitter, while the in-between receiver is expected to experience some collisions.

Merging time-stamped traces from the two monitors and the receiver, we were able to determine the relationship between transmission order and collision.

3.2 MIM: OPTIMALITY ANALYSIS

A natural question to ask is how much throughput gain is available from MIM. Characterizing the optimal gain will not only guide our expectations, but is also likely to offer insights into MIM-aware protocol design. Toward this end, we first prove that MIM-aware scheduling is NP-hard and use integer Programming methods to characterize the performance bounds for a large number of topologies. We compare the results with an MIM-incapable model.

Theorem 1:

Optimal MIM scheduling is NP-hard. Proof: Consider the problem of Optimal Link Scheduling with MIM-capable nodes. An optimal schedule consists of a link selection and a corresponding MIM-aware ordering that together maximize the network throughput. Assume that a polynomial-time algorithm exists to provide the optimal MIM link scheduling from known network interference relationships. Conventional (no-MIM) link scheduling is a known NP-complete problem, reduced from Maximum Independent Set .Therefore, if our assumption is true, then it would be possible to find the optimal.MIM-incapable link schedule in polynomial time just by restricting the SoI-last SINR threshold to infinity in our algorithm (i.e., ensuring later-arriving signals are never decoded). This contradiction proves that optimal MIM-aware link scheduling is NP-hard.

A. Optimal Schedule with Integer Program

To quantify the performance gains from MIM, we model wireless networks with MIM-capable and MIM-incapable receivers and compare their optimal throughput over a variety of topologies. The networks consist of multiple access points (APs), each associated to a number of clients. Each transmission produces an interference footprint derived from a path loss index of 4.With MIM-capable receivers, the SF SINR requirement is 4 dB, while the SL requirement is 10 dB [5]. With MIM-incapable receivers, the SINR requirement for reception is uniformly 4 dB, and later-arriving packets cannot be received. We construct linear (binary integer) programs to compute the maximum number of concurrent links meeting the required SINR thresholds. The linear program also produces the order. Fairness is not considered in this analysis. To make our model solvable within reasonable execution time.

Theorem 2:

Any 0/1 solution to the above integer program satisfies the following.

- Encodes the active links.
- The variables encode a total ordering on the active links, where is made active after Constraints (2) and (3).
- The set of active links along with their ordering satisfies the interference constraints and is hence feasible [Constraints (4) and (5)].

The optimal solution to the integer program is therefore precisely the optimal solution of interest.

Proof: Consider constraints (2) and (3). Suppose first that all Then, constraints (2) and (3) are equivalent to and .We interprets the variable as follows: if follows in the ordering and 0 otherwise. Note that the constraints exactly encode the following information: In any ordering, for every , either appears after or the other way around; for every , it cannot happen that follows , follows , and follows. It is shown in literature that these constraints are necessary and sufficient to encode a complete ordering. Now, suppose they are not all 1.

In that case, only if follows in the ordering and both and are active, so that. In this case, the constraints (2) and (3) are meaningful only if all the corresponding variables are all 1, which means all the corresponding links are active. For these links, the constraint (2) and (3) encode a total ordering. The constraints (4) and (5) encode the interference constraints. For any link , the only that contribute to constraint (4) are those with , which are precisely the that are active and precede . Furthermore, the left-hand side of the constraint is nonzero only if itself is active. A similar reasoning shows the validity of constraint (5).

MIM-aware link-layer solution that reorders transmissions to improve concurrency. Shuffle targets enterprise WLAN (EWLAN) environments, such as universities, airports, and corporate campuses. In EWLANs, multiple APs are connected to a central controller through a high-speed wired backbone The controller coordinates the operations of APs. The APs follow the controller's instructions for transmitting packets to their clients. The rationale for targeting EWLAN architectures is twofold.

- EWLANs are becoming popular in single-administrator environments. Developing this platform on sound physical and link layer technologies can further drive its proliferation.
- MIM-aware scheduling is hard, and a systematic approach to solving it should perhaps start from a more tractable system. EWLAN presents a semi centralized platform, amenable

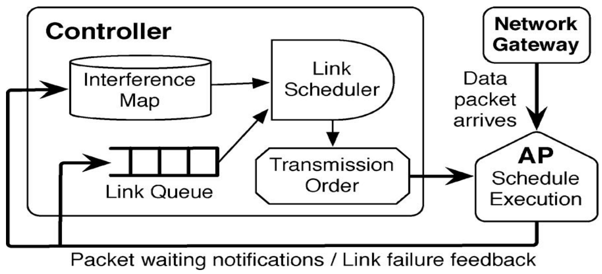

Fig 3.2.1 Optimal Schedule with Integer Program

B. Opportunistic Learning

To update the interference map more frequently, Shuffle takes advantage of opportunistic overhearing. For instance, a client C3 that overhears a packet from= AP5 at time can piggyback this information in an ACK packet that it sends in the near future. The controller has a record of which other APs were transmitting at Assuming AP7 was, the controller can immediately deduce that link (AP5 C3) can be concurrent with a transmission from AP7. The exact order for this concurrency can also be derived since the controller also remembers the relative transmission order between AP5 and AP7 from past time .Continuous overhearing of packets and piggybacking in ACKs can considerably increase the refresh rate of the interference map. Convergence time can reduce, facilitating better scheduling.

C. Packet Scheduling

Given the interference map of the network, the MIM-aware scheduler selects an appropriate batch of packets from the queue and prescribes their order of transmission. To maximize throughput, it should schedule the largest batch of packets that can be delivered concurrently without starving any client. As noted earlier, optimal MIM-aware scheduling is NP-hard, and graph-coloring approaches are not applicable because MIM conflicts are asymmetric in nature. Thus, new algorithms are required for effective MIM-aware scheduling. In this section, we consider packet scheduling with fixed rate.

3.3 Greedy Algorithm

A simple greedy algorithm would be to consider the packets in the first-in–first-out (FIFO) order for inclusion in the batch. Initially, the batch is set to empty. Then, each packet is attempted in turn to check if it is feasible to add it to the batch. Since the ordering of packets in the batch may affect its feasibility, a packet may need to be tried at different positions. Once a feasible ordering is found, the packet is inserted in that position in the batch. While this greedy scheme may not achieve optimal concurrency, it protects the clients against starvation. In every round, the first packet in the queue is guaranteed to get scheduled, and hence every conflicting packet will progress by at least one position in t he queue. As a result it is guaranteed to get transmitted within a bounded number of batches. A reasonable fairness is also achieved through this simple scheme. The worst-case time complexity of the basic greedy algorithm is where is the number of packets in the queue. Pseudocode is presented in,

Algorithm 1: Greedy

1: Let be the first packets in the queue.
2: **for all** Packets **do**
3: **for** to **do**
4: **if then**
5: Add to in position
6: Return

4. EXPERIMENTS AND RESULTS

We discuss some limitations with Shuffle implementation and identify avenues of further work. External Network Interference: We assume that all WiFi devices are associated to the same enterprise network. Put differently, no other WiFi transmission occurs that is not accounted for by the central controller. In reality, electronic devices such as microwaves may interfere in the 2.4-GHz band. Wireless devices from "neighboring" networks may interfere at the periphery of a Shuffle deployment.

Assuming no queuing at the AP or client, the added latency is only due to propagation, switching, and processing of two control messages. As a design alternative, this latency may be eliminated if the controller is collocated with the network gateway, so that schedules may be forwarded to APs in tandem with the outbound packet. While this provides no direct improvement for retransmissions of lost packets, recall that retransmissions get higher priority than new packets to the same client. This is expected to make the retransmission delay tolerable.

Client Mobility: As a client moves, interference relationships between links may change dramatically. While Shuffle's concurrent link selection, rate control, and transmission ordering mechanism do adapt to changes in channel conditions, we have not yet characterized convergence time for continuous-mobility scenarios.

Transport-Layer Interactions: We have not yet characterized TCP interaction behavior with Shuffle scheduling. A potential point of concern is division of time into upload and download periods, possibly impacting TCP round-trip time estimation and ACK timeouts. However,we believe that upload periods may be scheduled frequently enough (every few download packets) to limit this effect.

We discuss some limitations with Shuffle implementation and identify avenues of further work. External Network Interference: We assume that all WiFi devices are associated to the same enterprise network. Put differently, no other WiFi transmission occurs that is not accounted for by the central controller. In reality, electronic devices such as microwaves may interfere in the 2.4-GHz band. Wireless devices from "neighboring" networks may interfere at the periphery of a Shuffle deployment.

Shuffle's packet loss recovery mechanism will be able to cope with sporadic interference. However, if the losses are frequent, carrier sensing may need to be selectively enabled for the peripheral APs, limiting the Shuffle controller's ability to schedule for those clients. Latency: Shuffle introduces some end-to-end delivery latency. When a packet is received at an AP, it cannot be forwarded to the intended client until the AP notifies the controller and receives a scheduled slot for transmitting the packet.

Assuming no queuing at the AP or client, the added latency is only due to propagation, switching, and processing of two control messages. As a design alternative, this latency may be eliminated if the controller is collocated with the network gateway, so that schedules may be forwarded to APs in tandem with the outbound packet. While this provides no direct improvement for retransmissions of lost packets, recall that retransmissions get higher priority than new packets to the same client. This is expected to make the retransmission delay tolerable.

Client Mobility*:* As a client moves, interference relationships between links may change dramatically. While Shuffle's concurrent link selection, rate control, and transmission ordering mechanism do adapt to changes in channel conditions, we have not yet characterized convergence time for continuous-mobility scenarios.

Transport-Layer Interactions*:* We have not yet characterized TCP interaction behavior with Shuffle scheduling. A potential point of concern is division of time into upload and download periods, possibly impacting TCP round-trip time estimation and ACK timeouts. However,we believe that upload periods may be scheduled frequently enough (every few download packets) to limit this effect.Compatibility*:* Shuffle is not immediately compatible with existing deployments. Clients must be protocol-compliant so as to remain silent during download periods and provide ACKs during upload windows. Small-Scale Testbed*:* We tested our Shuffle implementation on topologies consisting of up to three concurrent links. Though more experiments with larger topologies would be desirable to confirm the scalability of Shuffle, our simulation results indicate that Shuffle scales well.

5. CONCLUSION

Message in Message (MIM) in modern wireless cards allows receiver to disengage from an ongoing reception and engage onto a new stronger signal. The rewards from this physical layer Capability cannot be fully realized unless link-layer protocols are explicitly designed with MIM-awareness. Specifically, we have shown that links that conventionally conflict with each other may be made concurrent if they are initiated in a specific order. We then presented Shuffle, a system that reorders transmissions to improve spatial reuse. Theoretical analysis has shown that the optimal improvements with MIM can be significant. A functional test bed validated that MIM-awareness is practical, while results of experimental evaluation confirm consistent Performance improvements.

REFERENCES

[1] "NS2," [Online]. Available: http://www.isi.edu/nsnam/ns/
[2] "Qualnet v2.6.1," Scalable Network Technologies, Los Angeles, CA [Online]. Available: www.scalable-networks.com
[3] A. Kochut, A. Vasan, A. U. Shankar, and A. Agrawala**, *"Sniffing out the correct physical layer capture model in 802.11b,"*** in Proc. IEEE ICNP, 2004, pp. 252–261.
[4] T. Nadeem and L. Ji, **"***Location-aware IEEE 802.11 for spatial reuse Enhancement*,**"** IEEE Trans. Mobile Comput., vol. 6, no. 10, pp. 1171–1184, Oct. 2007.
[5] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi, **"An Experimental study on the capture effect in 802.11a networks,"** in Proc. ACM WinTECH, 2007, pp. 19–26.
[6] **"Soekris Net4521 and Net4826-Series,"** Soekris Engineering, Inc., Santa Cruz, CA [Online]. Available: http://www.soekris.com/
[7] ***"Madwifi-ng driver,"*** Linux-Consulting, Reno, NV [Online]. Available: http://madwifi.org/
[8] S. Ramanathan and E. L. Lloyd, **"***Scheduling algorithms for multihop radio networks***,"** IEEE/ACM Trans. Netw., vol. 1, no. 2, pp. 166–177, Apr. 1993.
[9] Solver CPLEX. Ilog, Inc., Armonk, NY, 2003.
[10] P. Bahl, J. Padhye, L. Ravindranath,M. Singh, A.Wolman, and B. Zill,

AUTHOR PROFILE

P.Pradheepa M.Sc (IT). I have completed my B.Sc degree from one of the affiliated college under Periyar University and M.Sc degree from one of the college under Anna University and currently pursuing M.Phil in computer science one of the college under Periyar University. I have done two projects in Networking during my M.Sc. Recently I presented a paper in Mobile Computing.