ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Efficient and Secured Application with Passive Measurement in WIFI Network

M. Kiruba [#1], Mrs. K. Akilakumari M.E.,[#2]

Embedded System Technologies, Anna University, India.

Department of Electronics and Communication Engineering , Raja College of Engineering and Technology Madurai, India,

ABSTRACT— Co-operative communication is one of the fastest growing areas of research, and it is likely to be a key enabling technology for efficient spectrum use in future. The key idea in user-cooperation is that of resource-sharing among multiple nodes in a network. The reason behind the exploration of user-cooperation is that willingness to share power and computation with neighboring nodes can lead to savings of overall network resources. Wi-Fi networks provide an enormous application space for user-cooperation strategies to be implemented. In traditional communication networks, the physical layer is only responsible for communicating information from one node to another. In contrast, user-cooperation implies a paradigm shift, where the channel is not just one link but the network itself. It demonstrates an application based on selfish carrier sense behavior. The experimental and simulations result demonstrates that the proposed approach to estimate interference relationship is more accurate and quite competitive. It can be implemented in real wireless LAN environment as well as NS-2 simulation. The metric of selfishness is used to estimate selfish behavior matches closely with actual degree of selfishness observed.

KEY WORDS—802.11 protocol, hidden Markov model, MAC layer misbehavior, interference

I. INTRODUCTION

To understand the wireless interference between network nodes and links in realistic Wi-Fi network deployments. The goal is to do this in the most unobtrusive fashion possible: 1) Without installing any monitoring software on the network nodes. This is motivated by practicality as many APs are often closed devices, and clients may not be always be privy to new software; 2) Using a completely passive technique. This is important as active measurements impact (and are impacted by) network traffic.

To achieve these goals, our approach uses a distributed set of "sniffers" that capture and record wireless frame traces. We then analyze the trace to understand the interference relations. While this is true that this approach requires additional hardware for measurement, this can be viewed as a form of third-party solution. Such independent third-party solutions for wireless monitoring are not uncommon in industry. The research community has also provided similar approaches. While these approaches provide many monitoring solutions, they still do not provide fundamental understanding of interference relations between network nodes and links. Aside from understanding interference relationships, there are other applications of the technique we develop. Certain types of selfish behaviors can be detected via this approach—an example we will demonstrate.

A selfish node can gain unfair share of the available bandwidth by manipulating different MAC protocol parameters, such as the clear channel assessment (CCA) threshold, or the back off window size. This can deliver an unfair bandwidth advantage to a selfish node and can be used to even launch a denial of service attack. A node, for example, can be selfish by raising the CCA threshold.

This can effectively disable its carrier sensing and creates more transmission opportunities for the selfish node. This can also cause collisions, and thereby force the other transmitters in the vicinity to perform back off. While the selfish node itself may also undergo a collision, the back off period will be shorter as it will not freeze its back off counter when carrier sensing is disabled. We can detect the selfish carrier-sense behavior using the pair wise

interference relationships discovered by the proposed technique. In our knowledge, this problem has been explored only in one paper that provides a limited solution using a non passive technique.

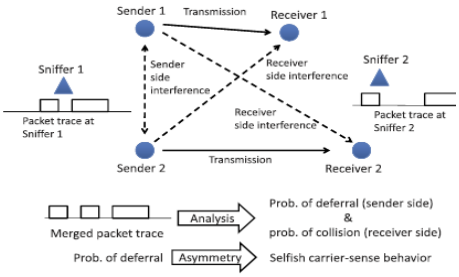

Fig.1 Overview of the approach

II. PREVIOUS WORK

2.1 Detecting MAC-Layer Misbehavior

This paper manipulation of the carrier-sense behavior is harder to detect. This is because normal fluctuations of wireless channel must be distinguished from manipulated carrier sensing.The technique proposed in relies on a strong assumption that the selfish node that has increased its CCA threshold is unlikely to correctly recognize low power transmissions from the AP as legitimate packets. Thus, by sending low power probes, the AP can potentially detect such nodes. This assumption implies that packet reception with power lower than CCA threshold is not possible, as such packets are treated as noise. the attacker can avoid detection by simply changing the CCA threshold only when it transmits a packet and reverting back to the normal threshold right after the transmission.[3] Also, depending on how the radio transceiver is designed, packet reception success may not be dependent on the CCA threshold. Also, this technique is not passive.

2.2 Use of distributed sniffers

Techniques based on using distributed sniffers can be found in a number of measurement studies for the purpose of learning various properties of live network such as conges-tion, protocol behavior in a hotspot setting etc. The DAIR system also uses such an approach for troubleshooting and security.

2.3 Discusions

To estimate the interference relations between a given pair of nodes, our technique needs to have instances when simultaneous transmissions are attempted by the two nodes. The conjecture here is that if one observes the live network traffic for a long enough period, enough of such instances will be available for each node pair. Our goal is to 1) identify such instances, and 2) infer the deferral behaviors during such instances. There are several challenges here. First, creating a complete and accurate trace is itself a difficult problem.

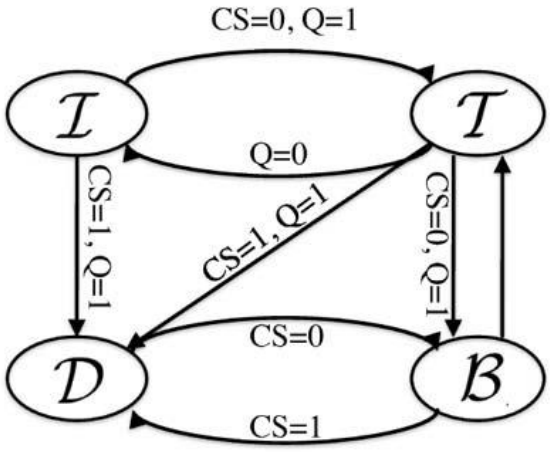

Fig 2. State transition diagram for a single sender.CS=0(CS=1) means that the carrier is sensed idle(busy) Q=0(Q=1)means that the interface packet queue is empty(nonempty).

There are many approaches proposed in literature to create a complete trace. But for our technique, incomplete trace may suffice as long as it is statistically similar to the complete trace.Second; unknown load of the nodes makes it hader to estimate the deferral behavior.In our approach we utilaize the statergy of analyzing inter packet times which can provide certain confidence.third heuristics can be used to infer the deferral behavior. But straight forward heuristics may have limited power.

III. PROPOSED WORK

The key idea in user-cooperation is that of resource-sharing among multiple nodes in a network. The reason behind the exploration of user-cooperation is that willingness to share power and computation with neighboring nodes can lead to savings of overall network resources. Wi-Fi networks provide an enormous application space for user-cooperation strategies to be implemented. In traditional communication networks, the physical layer is only responsible for communicating information from one node to another. In contrast, user-cooperation implies a paradigm shift, where the channel is not just one link but the network itself.

A.Hidden Markov Model For Sender-side Interactions

A hidden Markov model represents a system as a Markov chain with unknown parameters. Here the states of the Markov chain are not directly visible, but some observation symbols influenced by the states are visible. The unknown parameters (such as the state transition probabilities of the Markov chain) can be learned using different standard methods with the help of the observed sequence of observation symbols. Various machine

learning applications such as pattern, speech, and handwriting recognition have used HMM technique. We will be using the HMM approach for modeling interactions between a pair of senders in an 802.11 network and inferring sender-side interference relations (deferral beha-vior) between them.

B Markov Chain

In Fig. 2. A sender node, say X, is found in one of the following four states—"idle," "back off," "defer," and "transmit." The essence of the 802.11 MAC protocol lies in these four states. We intentionally ignore interframe spacing's (e.g., DIFS) to keep the chain simple. In the rest of the paper, we call the four states I, B, D, and T , respectively for the sake of brevity.

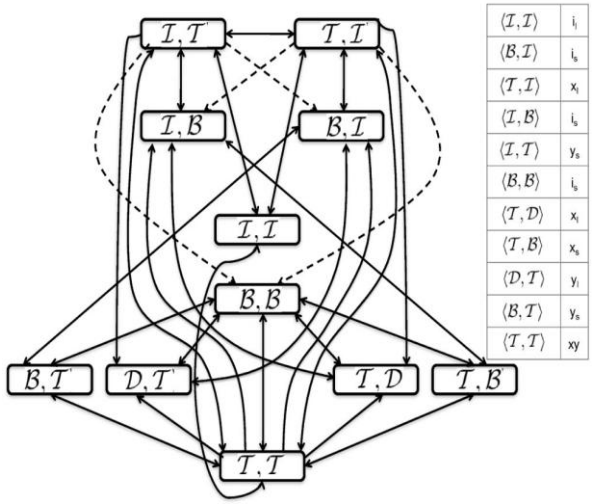

Fig. 3. Markov model of the combined MAC Layer behavior of two nodes (sender side only). Note that some arrows are bidirectional

Note that the state transition probability between B and D of the corresponding sender node is influenced by the states of other nodes (i.e., transmitting or not) in the network, and the deferral probabilities between the sender and these nodes. Similar argument applies for the transition prob-abilities from I to D and T , and transition probabilities from T to D and B.

In fig. 3 for the two-node combined Markov chain. (Only the solid lines indicate valid transitions. The dotted transition lines will be discussed later.)The state transition probabilities between certain states in this Markov chain are determined by the deferral prob-abilities between X and Y. For example, transition prob-abilities from state (B,B) to state (T,D) or (T,B) would depend on deferral probability of Y with respect to X. Let us explain this using an example. Assume that Y carrier senses X (or Y can sense X's transmission) perfectly. Then when X moves from B to T state (i.e., starts transmitting as soon as the backoff interval is over), Y must also move from B to D as it defers to X's transmission by freezing its back off countdown timer. If instead Y never carrier senses X, it will remain in the B state. The deferral probability of X and Y depends on the number of instances when either of the nodes moves to D state.

Note again that this combined Markov chain is specified for a node pair only, as we are interested in pair wise interference. This process can be repeated for all pairs to determine the all-pair sender-side interference. We filter out the packets of just the two senders under consideration for analysis, and ignore the other packets. This may misinter-pret an active node, deferring for a third node's transmis-sion, as idle, and we may miss an opportunity to interpret the interaction between the particular pair as interfering or non interfering. But, it is important to note that this does not create any incorrect interpretation. Recent studies show that the number of instances of three or more nodes simultaneously being active is much less than that of only a pair of nodes being active. Thus, we should get enough instances of just a pair of nodes being active in a long trace. An alternate but computationally expensive method could try to identify portions of the trace where only the senders in a node pair being considered are active.

IV. EVALUATION PROCESS

4.1 Simulation Based Evaluation

Simulations let us create arbitrary topologies and interference conditions easily. However, the physical layer (including interface behavior for carrier sense and packet capture) implementation is often idealized or unrealistic in simulations. To address this issue, we use an extended version of the ns2 simulator that includes realistic measurement based models.These models were validated against experimental results showing excellent accuracy.

For the sake of completeness, we note that the enhance-ments in ns2 in are done specifically in the following physical layer components—1) radio propagation model, 2) deferral or carrier sense model, and 3) packet reception model. For (1), models are derived from real measurements in a testbed. For (2) and (3), measurement-based profiles of a testbed are created where every value of RSS is mapped to a deferral probability and every value of SNR is mapped to receive probability, respectively.

4.2 Complete Evaluation on WLAN

Here, we provide a complete evaluation—both sender and receiver sides. These experiments are done on an active WLAN with seven APs spread over two floors of the Computer Science department building of Stony Brook University. Seven laptops

are used as clients. Each client fetches a large file via HTTP download using a unicast link for about 20 mins. This simulates real network traffic that is sniffed using nine sniffers (Soekris single board computers with 802.11 miniPCI cards with Atheros chipset and with external USB flash memory to store packet traces). The sniffers are deployed based on convenience, i.e., near a power outlet and in the rooms that we have regular access to. However, an attempt was made to keep them as close to the APs as possible.

Sixteen client laptop pairs are considered for evaluation. All of these pairs associate with two different APs. Unlike the micro benchmarking experiments, the default autorate control with 802.11b is used. Also, the 802.11 frames are now unicast with ACK. RTS/CTS is disabled. For each pair, the probability of interference between the pair of download links (AP to client) is "estimated" using (1). First, the probability of deferral (Pd) is estimated using the HMM-based method using the merged sniffed traffic traces from all sniffers. Second, the probability of collisions (Pc)is estimated by observing the retransmissions for overlapped packets as described.However, in all cases, retransmis-sions were rare, typically less than 1 percent of frames were retransmitted. This is consistent with prior experimental observations. Thus, p_c could be safely ignored with p_d alone determining the probability of interference.

4.3 Evaluating Large-Scale Wireless Traces

Encouraged by the strong validation results in the depart-mental WLAN trace analysis, we use the wireless network trace collected at the SIGCOMM 2004 conference for demonstrating powerful capabilities of our tool. The trace was obtained from the CRAWDAD archive. The SIGCOMM 2004 conference was four days long and was attended by more than 500 attendees. During busy periods, several simultaneously active flows were not uncommon. The WLAN under consideration in this trace had five APs—three on channel 1, one on channel 8 and the other one on channel 11. Five sniffers were used each with three wireless interfaces. Two of them listened on channel 1 and 11, respectively, and the third one listened either on channel 8 or 6. We consider only channel 1 in this work. First, we analyze the probability of interference between client-to-AP links where the clients are associated with the same AP. For this analysis, we pick random pairs of clients associated with the same AP and find a 20 mins long period when they are both simultaneously active.

V. SIMULATION

Ns2 simulations let us implement various degrees of selfishness, where the selfish node senses carrier with only a certain probability. We use the term degree of selfishness (Ps) to indicate that the selfish node senses carrier with probability equal to $1 \_ P_s$. Ns2 simulations also make it easier to investigate larger networks, where there are many nodes, possibly with more than one selfish node with varying traffic and degrees of selfishness.

In our simulated scenario, there are 40 network nodes distributed randomly in a square region. We chose a deployment typical of dense WiFi client distribution in indoor office environments, assuming that there is one node in 300 sq ft on average. The default ns2 wireless channel model is extended to include shadowing effects. This introduces randomness in the transmission range of a node instead of making it a perfect disk. Shadowing parameters are taken from where a set of measurements was done to model such parameters in an indoor environment. A set of feasible network links are chosen randomly and one-hop UDP flows are generated with randomly chosen loads (between 0.5-1 Mbps). Each flow is active (and then inactive) only for a random interval of time. Both intervals are chosen from an exponential distribution with a mean of 5 s. Note that the exact traffic parameters are not important for our work.

We deploy a set of 10 sniffers at random locations. Among the 40 network nodes, 1, 2, or 3 nodes are selfish. The degree of selfishness is varied. The similar plots for the scenarios with 2 and 3 selfish nodes using different heuristics. We instead show the overall statistics that summarizes how good our detection is. For each scenario and for each type of witness node identifica-tion technique, we evaluate for each node the "estimation error" as the algebraic difference between the computed

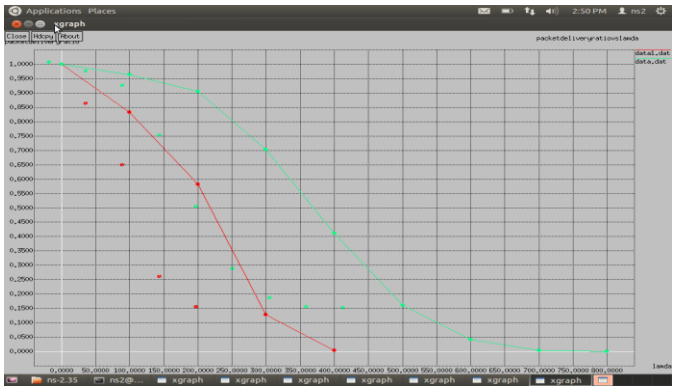

Fig .4 Packet delivery ratio vs. lamda

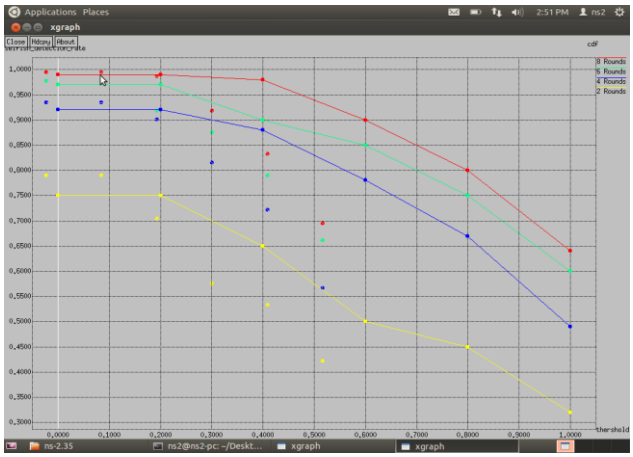

Fig.5 . CDF of "estimation error" for the selfishness metric. four different scenarios are presented where number of selfish nodes are varied (1, 2, 3 or 4) and witness nodes are identified in four different rounds.

Selfishness metric and the actual degree of selfishness of that node. All nodes (selfish and regular) are included. The estimation error is plotted as a CDF in Fig.5 . Nine plots are shown for three techniques used to identify the witness
nodes and for three different numbers of selfish nodes.The CDF shows that the estimation error is very small in general and heuristic H_2 performs somewhat better than the other two techniques in general.
In this scenario, the heuristics do not perform much better than the no heuristic case, because the no heuristic case itself performs very well. The reason for this is the high density of the network. To demonstrate the power of the heuristics we consider a sparser network with 40 nodes distributed randomly in squared region with one node in 1,500 sq. feet on average. Different scenarios are created by varying the number of selfish nodes (1, 2, 3 or 4) with degree of selfishness ¼ 1. Because of the sparsity of the network we now have to deploy more sniffers to capture all network traffic.

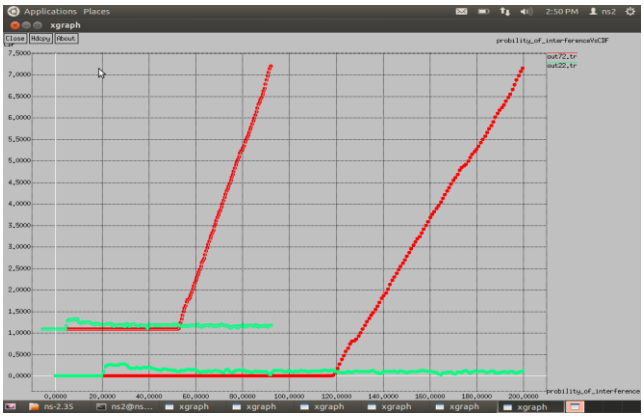

Fig .6 Probability of interference vs. CDF

VI.CONCLUSION

The technique uses a merged packet trace collected via distributed sniffing. It then recreates the MAC layer interactions on the sender-side between network nodes via a machine learning approach using the Hidden Markov Model. This coupled with an estimation of collision probability on the receiver-side is helpful in inferring the probability of interference in the network links. Significant asymmetry in the sender-side interaction in favor of a particular node witnessed by multiple other nodes indicates selfishness. The power of this technique is that it is purely passive and does not require any access to the network nodes. Co-operative communication is one of the fastest growing areas of research, and it is likely to be a key enabling technology for efficient spectrum use in future. The key idea in user-cooperation is that of resource-sharing among multiple nodes in a network. The reason behind the exploration of user-cooperation is that willingness to share power and computation with neighboring nodes can lead to savings of overall network resources. Evaluations show the effectiveness of the tool for both the applications.

REFERENCES

[1] A. Kashyap, U. Paul, and S.R. Das, "Deconstructing Inter-ference Relations in WiFi Networks," Proc. IEEE Seventh Comm. Soc. Conf. Sensor Mesh and Ad Hoc Comm. and Networks (SECON), 2010.

[2] U. Paul, S.R. Das, and R. Maheshwari, "Detecting Selfish Carrier-Sense Behavior in Wifi Networks by Passive Monitoring," Proc. IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN), 2010.

[3] J. Tang, Y. Cheng, Y. Hao, and C. Zhou, "Real-Time Detection of Selfish Behavior in IEEE 802.11 Wireless Networks," Proc. IEEE 72nd Vehicular Technology Conf. Fall (VTC-Fall), 2010.

[4] K. Pelechrinis, G. Yan, S. Eidenbenz, and S.V. Krishnamurthy, "Detecting Selfish Exploitation of Carrier Sensing in 802.11 Networks," Proc. IEEE INFOCOM, 2009.

[5] P. Bahl et al., "Enhancing the Security of Corporate Wi-Fi Networks Using DAIR," Proc. ACM/USENIX Mobile Systems, Applications, and Services (MobiSys), 2006.

[6] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-Based Characterization of 802.11 in a Hotspot Setting," Proc. ACM SIGCOMM, 2005.

[7] P. Bahl et al., "DAIR: A Framework for Troubleshooting Enterprise Wireless Networks Using Desktop Infrastructure," Proc. ACM HotNets-IV, 2005.

[8] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On Selfish Behavior in CSMA/CA Networks," Proc. IEEE INFOCOM, 2005.

[9] S. Radosavac, J.S. Baras, and I. Koutsopoulos, "A Framework for Mac Protocol Misbehavior Detection in Wireless," Proc. ACM Workshop Wireless Security, 2005.