A Passive Approach to Rogue Access Point Detection

Lanier Watkins
Department of Computer Science
Georgia State University
Atlanta, GA, USA

Raheem Beyah
Department of Computer Science
Georgia State University
Atlanta, GA, USA

Cherita Corbett
Sandia National Laboratories
Livermore, CA USA

Abstract—Unauthorized or rogue access points (APs) produce security vulnerabilities in enterprise/campus networks by circumventing inherent security mechanisms. We propose to use the round trip time (RTT) of network traffic to distinguish between wired and wireless nodes. This information coupled with a standard wireless AP authorization policy allows the differentiation (at a central location) between wired nodes, authorized APs, and rogue APs. We show that the lower capacity and the higher variability in a wireless network can be used to effectively distinguish between wired and wireless nodes. Further, this detection is not dependant upon the wireless technology (802.11a, 802.11b, or 802.11g), is scalable, does not contain the inefficiencies of current solutions, remains valid as the capacity of wired and wireless links increase, and is independent of the signal range of the rogue APs.

***Keywords*—rogue access point, wireless security, insider threat**

I. INTRODUCTION

The flexibility and portability of wireless communications has afforded organizations many benefits such as increased productivity and lower installation cost. The benefits are not without cost, by using airwaves as the transmission medium, an entirely new set of security issues are born. Some of the most common risks associated with wireless networks are: 1) the loss of confidentially – unauthorized users may gain access to systems and information; 2) loss of integrity – network data may be corrupted or altered; consequently many corporations and government entities still do not allow wireless networks. If allowed, normally these entities will carefully deploy their wireless network ensuring a separation (firewall) between their wireless and wired networks. However, many employees decide to deploy their own wireless network by connecting a rogue access point directly to the corporate network. Rogue access points are typically inexpensive wireless access points (WAP) deployed on a network port behind the organization's (corporation, university, etc.) firewall by a disgruntled or mischievous member of that organization. The existence of this rogue access point in an organization's network is a severe security breach and can serve as an unmonitored point of entry, not only for the mischievous member but also for any other unauthorized person that may stumble upon it.

Rogue access points are very difficult to detect, the mischievous member could have hidden it in a very esoteric location or could have simply disabled the SSID broadcast of the access point; for these reasons alone there is a grave need for rogue access point detection. This paper presents a simple yet robust method capable of passively detecting rogue access points using the RTT of traffic on the local area network.

The remainder of the paper is organized as follows: In Section II we discuss current approaches. We discuss the background of our scheme in Section III. In Section IV we describe our experimental setup. Section V gives the results and performance analysis of our scheme. In Section VI we conclude the paper and discuss future work.

II. RELATED WORK

Detecting rogue APs presents a challenging problem since one has no cooperation from the rogue AP or from the offending system using the rogue AP. The approaches to detecting rogue APs fall into three categories: wireless approaches, hybrid wired and wireless approaches, and wired-only approaches. Other than two wired only approaches [2] & [17], and the work done in [9] this area has been overlooked by the academic community. Below we discuss other tangentially related academic work and previous academic-based and industry-based related work.

A. Tangentially Related Academic Work

In [3] the authors consider RTT as a method of determining if one or more wireless links are in the communication path. The goal is to address QoS issues in hybrid wired and wireless environments. They use UDP packets as probes to determine the RTT. They develop a fuzzy reasoning technique as opposed to a set threshold to help distinguish between wired and wireless links. Though this technique adds overhead and is active, it can be considered appropriate for QoS applications since the node being probed is trusted and will respond to probes. However, malicious users can easily disallow responses to probes, further, it is in the network administrator's interest to not let the malicious user know he/she is being monitored.

In [4] the authors considered calculating RTT to support improved protocols that are optimized for certain link characteristics. In particular, they look at distinguishing between different access networks including: Ethernet, 802.11b, ADSL, cable modem, and dialup. This technique requires that the node requesting the RTT of a path between itself and another node to send a request to the end node. That end node returns a sequence of packet pairs and the requesting node determines the connection type based on the

1930-529X/07/$25.00 © 2007 IEEE

median and entropy of the inter-arrival times of the packet pairs. This scheme is also active, and is not appropriate for detecting malicious devices on networks.

B. Wireless Approaches

In [5] the authors designed the Distributed Wireless Security Auditor (DWSA). Their Linux- and Windows-based (functional on both 2000 and XP) implementations provide continuous wireless network assessments by harnessing the power of available, trusted wireless clients as distributed anomaly sensors throughout a company's network infrastructure. Using periodic security reports, a back-end server detects rogue and misconfigured APs and subsequently locates them via 3D trilateration, a location-finding algorithm used in systems such as GPS.

Most of the current industry-based wireless approaches for detecting rogue APs are rudimentary and easily evaded by hackers. Some organizations have equipped IT personnel with wireless packet analyzer tools (e.g., sniffers) on laptops and handheld devices (e.g., AirMagnet [6] and NetStumbler [7]), forcing IT personnel to walk the halls of the enterprise or campus searching for rogue APs. This method is generally ineffective because manual scans are time-consuming and expensive – and, therefore, are conducted infrequently. Also, with 802.11 hardware operating at separate frequencies (802.11a - 5Ghz and 802.11b/g - 2.4Ghz), IT personnel must upgrade their detection devices to accommodate multiple frequencies. Moreover, scans are easy to elude, since a rogue AP can easily be unplugged when the scan takes place.

Another approach is to initiate an enterprise-wide scan from a central location by using separate hardware devices [5,8,9,10,11,16] (e.g., sensors) or using APs to detect beacons from surrounding APs [11], and transmitting this information back to a central management platform containing the wireless network policy for analysis [12]. This method becomes costly, considering that one must place sensors or APs throughout the entire enterprise to monitor the airwaves. This technique is also completely impractical for the networks that do not have wireless APs. Much like the drawback of the "walking the halls" solution, each sensor/AP must operate at both frequencies to be completely effective. Moreover, with sensors deployed throughout the network, one still may not be able to detect the rogue AP. The clever employee could have used a directional antenna, or reduced the signal strength to cover the small range within his/her office. Another drawback of wireless-based solutions is that they will falsely report the wireless network in the coffee house next door as a rogue.

C. Hybrid Wireless and Wired Approach

Taking a step in the right direction, Wavelink [11] combined the previously mentioned techniques for detecting rogue APs with listening at network layers 2 and 3 and querying switches and routers to determine what devices are connected to them. This combination attempts to provide a hybrid wired and wireless approach to detecting rogue APs. This fails for the same reasons that the wired-only solutions discussed in detail below fail.

D. Wired-Only Approaches

In [2] we propose the use of inter-packet spacing to determine whether the traffic originated on a wireless link or a wired link. Though this scheme is passive and proved successful at differentiating types, inter-packet spacing can be a result of the load on the switch and may lose accuracy as the number of switches increases between the assailant and the monitoring device.

In [17] the authors propose the use of ACK-pairs to determine if packets originate from a wired or wireless node. They use an iterative Bayesian interference algorithm to obtain a maximum likelihood estimate. Though this approach proves effective, it requires time to converge, consequently making it difficult to be deployed inline. Further, their analysis was conducted on 802.11b traffic which we show is drastically different from wired traffic and thus significantly easier to distinguish from wired traffic.

Cisco offers a scalable and comprehensive approach using a suite of tools [13] that are not limited by signal range. They attempt to detect APs by querying routers and switches for company MAC address assignments (i.e., if the MAC address belongs to Linksys, the MAC address cannot belong to a PC and becomes suspicious). This is ineffective because MAC addresses can be spoofed or cloned easily by an AP. Another approach in the suite is the use of http query to communicate with the web server residing on the AP. This is a good approach, but the node must already be suspected as being an AP (maybe using one of the aforementioned methods), or every node on the network must be queried. This approach assumes that the wireless router responds to http queries. Additionally, this invasive approach is considered active, adding significant unwanted traffic on the network and can also alert an advanced rogue AP user of a scan for the AP. The suite also has an application which allows the viewing of html code (assuming unencrypted) generated when configuring AP settings. Though this approach will work in theory, the window of opportunity is limited since this data is only transmitted when the AP's configuration is updated. Additionally, as signature-based IDSs can attest, reassembling application-level data becomes more difficult and impractical as network speeds increase.

Another wired-only approach is presented by Wimetrics [14]. Their product has a wired-only approach, but is ambiguous with details. The basic premise of their work is that they probe the network to identify the profile of a wireless AP. While the details were unclear, Wimetrics' general approach proves not scalable since it requires a PC to sit on each segment of the network. Their approach unjustly assumes that the network is a shared network. As discussed in the previous section, APs can be configured to ignore network queries.

The wireless-only solution is by far the most widely-used method and the most costly, but the least effective for the aforementioned reasons. Accordingly, we propose a wired-only solution that is inherently independent of wireless protocol, frequency, signal range, that cannot be easily evaded by attackers, and that can be performed at a central location.

1930-529X/07/$25.00 © 2007 IEEE

Table I. Relevant combinations of stages of TCP from which RTT information can be gleaned.

Conn. Initiation	Conn. Termination	SYN - SYN/ACK	SYN/ACK - ACK	Incoming Data - ACK	FIN - ACK
Internal	Internal	_	√	√	√
Internal	External	_	√	√	√
External	Internal	√	_	√	√
External	External	√	_	√	√

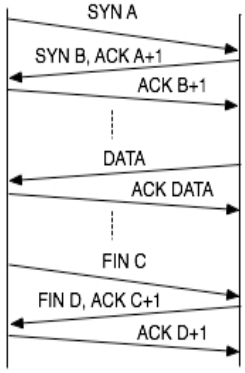

Figure 1. TCP timing diagram.

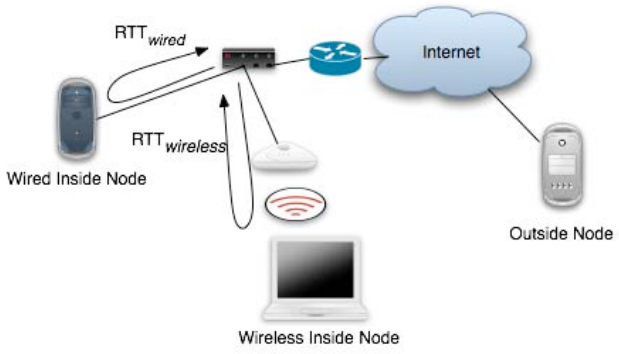

Figure 2. Network with at least one wireless and wired link.

III. EFFECTS OF THE PHYSICAL LAYER OBSERVED AT THE TRANSPORT LAYER

The time it takes for packets to traverse the link is a composite of propagation delay (d_{prop}), queuing delay (d_{queue}), transmission delay (d_{trans}), processing delay (d_{proc}) (1). However, in a LAN environment with nodes with an average processing capacity, the transmission delay is likely to be the dominant factor in the total delay. That is, $d_{trans} >> (d_{prop} + d_{proc} + d_{queue})$. Thus in the aforementioned environment (1) reduces to (2).

$$\textbf{\textit{Total delay}} = d_{prop} + d_{trans} + d_{proc} + d_{queue} \quad (1)$$

$$\textbf{\textit{Total delay}} = \boldsymbol{d_{trans}} = packetsize \,/\, capacity \quad (2)$$

As shown in (2) the capacity of the link directly impacts the speed at which packets travel across the link. Further, when comparing links, if we assume that the higher–layer protocols are the same on each link, the rate at which a link can support traffic depends on link-layer characteristics (e.g., MAC protocol) and the physical-layer characteristics (e.g., encoding scheme, type of link, loss rate, etc.). For the current discussion we will refer to the effects of both layers together as the physical-layer characteristics.

The total delay in the transmission will contain characteristics of a combination of all the protocols in the TCP/IP stack (at least the ones that affect throughput). However, if the higher-layer protocols are the same network-wide, then the delay seen in the network is directly correlated to the physical-layer characteristics. Thus, if we determine the delay across links, we can effectively determine what type of link is being traversed.

The total delay in a network can be measured by the round trip time. There have been many approaches in the literature to measuring round trip time, some active and some passive; however, the majority are interested in wide area network (WAN) RTT and are not focused on detecting malicious activity.

Accordingly, we now focus our attention on the RTT from router to end node within a local area network. The basic idea is simple, packets that have a significantly longer RTT, are likely to have traversed a network with greater delay or lower capacity, namely a wireless network.

Consider Figures 1 & 2, TCP's inherent feedback mechanisms provide a means for passively capturing the round trip time on the network. Depending on the direction of traffic flow, we can capture the RTT on the local network using all or a combination of the following: 1) TCP three-way connection establishment handshake; 2) incoming data packets and corresponding outgoing ACKs; and 3) the connection termination sequence. The scenario depicted in Figure 1 assumes the connection is internally initiated. Thus, the SYN/ACK – ACK combination of the three-way handshake can be used to compute a RTT associated with the connection establishment (RTT_{CE}). Another source of an internal RTT is the incoming data and corresponding ACKs (RTT_{DA}). Finally, assuming an internal termination, we can use the combination of the FIN (or FIN/ACK) sent from the external node and its corresponding ACK sent from the internal node to obtain a RTT associated with the connection termination sequence (RTT_{CT}). The relevant combinations of the different stages of a TCP connection are listed in Table I.

1930-529X/07/$25.00 © 2007 IEEE

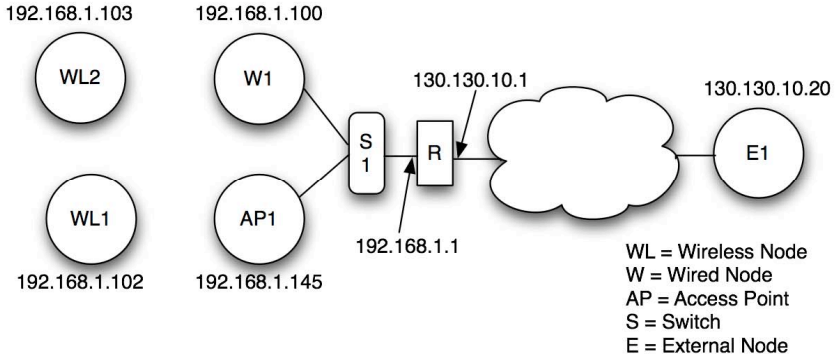

Figure 3. Experiment Testbed.

Table 2. Algorithms Used to Compute TCP RTT.

Algorithm 1-Computing RTT using TCP connection establishment

```
for every packet do
  if packet is a SYN and source IP is external
        record header of SYN packet and timestamp TS_S
        locate corresponding SYN/ACK
        record header of SYN/ACK and timestamp TS_SA
        RTT_SSA = TS_SA - TS_S
  else if packet is a SYN/ACK and source IP is external
        record header and SYN/ACK timestamp TS_SA
        locate corresponding ACK
        record header and timestamp TS_A
        RTT_SAA = T_A – TS_SA
  end if
end for
RTT_CE = RTT_SSA combined with RTT_SAA
```

Algorithm 2 – Computing RTT on Data packets

```
for every packet do
  if packet is Data and source IP is external
        record header of Data packet and timestamp TS_D
        locate corresponding ACK
        record header of ACK and timestamp TS_A
     RTT_DA = TS_A - TS_D
  end if
end for
```

Algorithm 3 – Computing RTT on TCP connection termination

```
for every packet do
  if packet is a FIN and source IP is external
        record header of FIN packet and timestamp TS_F
        locate corresponding ACK
  record header of ACK and timestamp TS_A
     RTT_CT = TS_A – TS_F
  end if
end for
```

IV. EXPERIMENTAL SETUP AND PROCEDURE

To test the RTT theory in a controlled environment, we built an experimental testbed. The testbed was comprised of Lenovo 3000 C100 laptops running Fedora Core 4, kernel ver. 2.6.11-1.1369 with 512 MB RAM as clients and servers. The Lenovo laptops use an Intel PRO/Wireless IPW2915a/b/g internal wireless card. A Linksys router was used to connect the external network to the internal network. Several Netgear ProSafe 16 Port 10/100 Switches were used to create different network segments. A Linksys Wireless G Access Point was used to bridge the wired and wireless networks. The wired links are 100Mbps. Also, not shown in Figures 2 and 3 is the Netgear Prosafe 4-port 10/100 hub with a monitoring node connected on the LAN segment of the router between *R* and *S1*.

The only traffic on the link was the traffic generated from our client. This is an appropriate setup considering most institutions have switched networks. In our experiments we consider several different scenarios. Several different permutations were executed per each configuration with each permutation deemed an execution sequence. The network traffic monitoring software used was Ethereal and traffic was generated using *sock* [15].

The experiments were conducted using the setup shown in Figure 3. The algorithms used to calculate RTT are listed in Table 2. There were ten 10MB transfers for both the internal connection initiation and the external connection initiation. We compared a 100Mbps link with an 11Mbps 802.11b link and a 54Mbps 802.11g link. Encryption was turned off to reduce delay on the wireless network.

V. RESULTS

Most of the results presented in this section follow a general pattern. Each figure shows a cumulative distribution function (CDF) of the RTT of the wired and wireless traffic. As such, the y-axis of each graph peaks at one, representing the cumulative maximum value, while the x-axis is time in µs. The data shown below assumes that the monitoring for a rogue AP would be performed at the immediate switch connecting the rogue AP. This one hop from switch to desktop configuration is extremely common. Of course, the switch would be located in a wiring closet located several meters away from the offending host.

1930-529X/07/$25.00 © 2007 IEEE

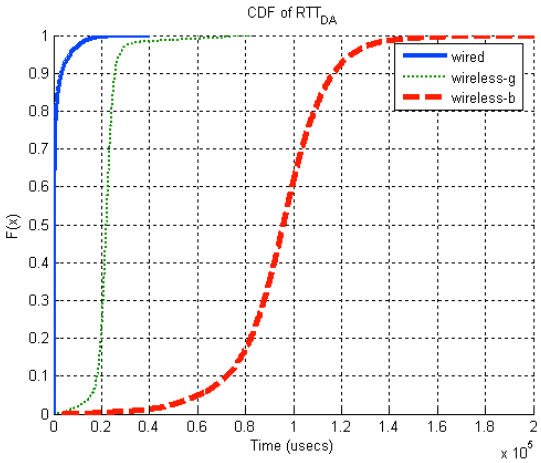

Figure 4. RTT of the path on the internal network.

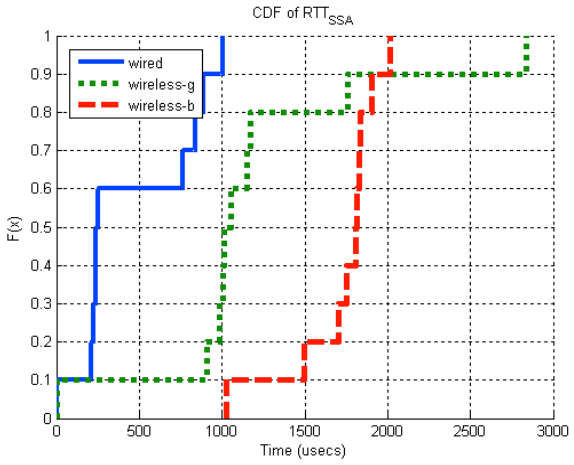

Figure 5. RTT of SYN-SYN/ACK portion of connection establishment.

This scenario is certainly far from a worst case scenario given the loads on the network. However, because most networks are switched and have a dedicated link, if we capture traffic by port (as they arrive at the switch), we should be able to have similar results with the accuracy degrading as the load on the switch introduces unpredictable variable delays to the traffic.

A. Using Magnitude of RTT to detect Rogue APs

Figure 4 shows the RTT of the path on the internal network. As expected, due to the wireless channel variability and lower link capacity compared to a wired link, the RTT for the traffic traversing the wired link is noticeably less than that of the wireless links. Further, the 802.11g RTT is significantly less than the 802.11b. Specifically shown in Figure 4, approximately 100% of the RTT values for the wired link are less than .02s, while 88% of the RTT values for the wireless-g link range from .02s - .03s, and 98% of the RTT values for the wireless-b link are greater than .03s.

As mentioned in Section III, the connection establishment and termination sequences (SYN/ACK-ACK and FIN-ACK scenarios are not shown due to space constraints but are similar) are also a potential location to extract underlying characteristics of node traffic. Figure 5 shows that there is a clear distinction between wireless and wired traffic with 100% of the wired traffic having an RTT below 1ms, where 80% of the wireless-g traffic and 100% of the wireless-b traffic are greater than 1ms. It is interesting to note here that the connection establishment RTT is more than an order of magnitude smaller than the RTT associated with the data-ACK scenarios as a result of the smaller packet sizes (40 bytes versus 1500 bytes).

B. Effect of Sampling

Having to capture RTT values for all of the data becomes processing intensive. A better approach is to take a sample of the data and ideally maintain accuracy. Figure 7 shows the CDF of wired, wireless-g, and wireless-b flows with 5% of RTT values sampled. The results are promising and similar to those in *Section V-A* having a clear distinction between each flow type. Thus, with only 5% of the RTT values we can determine if a node is wireless or wired. This makes it plausible to deploy such a system inline.

C. Using Variability in RTT to detect Rogue APs

In the previous sections we discussed how the magnitude of the RTT can be used to distinguish between nodes that are connected to the network with a wired connection versus those that connect with a wireless connection. The CDF (Figure 4) that is discussed in that section not only revealed the distribution of the RTT values, it also alluded to the variability in the RTT values. Accordingly, even as speed of the wireless link approaches that of the wired link, the variation in RTT is a good indicator of link type. Essentially, the slope of the linear portion of the CDF gives an indication of the variability in the represented values. Accordingly, Figure 4 also shows a greater slope (closer to infinity) for the wired connection that is greater than the slope of the wireless-g connection, which is greater than the slope for the wireless-b connection. This allows us to conclude that there is greater variation in wireless RTT than wired RTT. This is a result of link-layer retransmissions, the Distributed Coordination Function, link contention, etc.

We now plot the standard deviation of the RTT for the wired and wireless links (Figure 6). The standard deviation for the wired link is the lowest, followed by the wireless-g link, followed by wireless-b link. The standard deviation of the RTT can clearly be used as a differentiator with zero probability of error. Figure 6 assumes a segment size of 1750 values (which is approximately half the number packets because TCP *normally* ACKs every other segment). If we use a smaller segment size the probability of accurate detection using RTT variance decreases but the time to detection increases. For example with a segment size of 92 RTT (approximately 184 packets) values the percentage of

1930-529X/07/$25.00 © 2007 IEEE

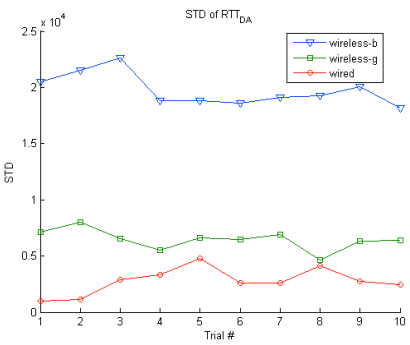

Figure 6. Standard deviation of the RTT.

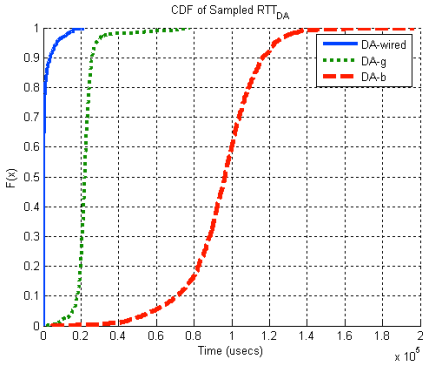

Figure 7. Effect of Sampling on RTT.

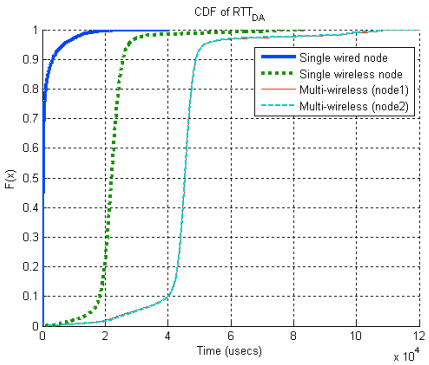

Figure 8. The effect of DCF on RTT.

accurate detection is approximately 33%, and if the segment size is increased to 500 packets, the percentage of correct classification increases to approximately 66%. The optimal segment size will vary based on congestion and network capacity but can be easily determined empirically.

D. Effect of the Distributed Coordination Function (DCF)

The previous sections primarily focused on rogue access points that only have one node associated with the AP. However, there is no limitation to the number of users that use the rogue AP. If multiple nodes decide to simultaneously access the rogue AP the RTT of the wireless flows created are even further away than the RTT experienced by a wired node. Figure 8 gives a comparison of the RTT for a single wired flow, a single wireless-g flow, and multiple wireless-g flows showing that approximately 100% of the RTT values for the wired link are less than .02s, while 88% of the RTT values for the wireless-g link range from .02s - .03s, and 98% of the overlapping RTT values for the multiple wireless-g flows are greater than .03s. The CDFs of the two nodes that are using the rogue AP overlap indicating the fairness of the 802.11 Distributed Coordination Function (DCF) when two competing flows converge. The DCF is used to manage access to the shared wireless medium. As a result nodes must contend for the medium and each equally slowed as both are equally likely to secure the medium for any given transmission attempt. Accordingly, as more simultaneous nodes use the rogue AP, the effects of the MAC protocol (DCF) become pronounced and make the rogue AP easier to detect.

VI. CONCLUSION AND FUTURE WORK

We have shown that as a result of the lower capacity of the wireless link wireless nodes have greater RTT associated with their packets. As the capacity of wireless links increase it is likely that the RTT associated with the wireless link will come close to that of the current wired links. However, as the capacity of the wireless link increases so will (and has) that of the wired links so the scheme will hold as the links speed increase (e.g., the wired link tested was 100Mbps where 1Gbps could have been easily chosen). Further, the variation of the RTT as a result of the DCF, link-layer retransmissions, contention, etc. will continue to be a distinguishing factor. This information coupled with a list of authorized APs (and their respective switch ports) allows us detect rogue access points one hop away from the offender.

Although we only focused on TCP traffic (which is about 75% of Internet traffic), our approach can be applied to any protocol that has associated feedback. For example, many applications that use the User Datagram Protocol (UDP) use application layer acknowledgements. Accordingly, we plan to extend our work to these specific applications. Additionally, we plan to detect the wireless nodes multiple hops downstream.

REFERENCES

[1] Karygiannis,T., Owens, L. "Wireless Network Security 802.11, Bluetooth and Handheld Devices." National Institute of Standards and Technology, Special Publication 800-48.

[2] Beyah, R., Kangude, S., Yu, G., Strickland, B., and Copeland, J. "Rogue Access Point Detection Using Temporal Traffic Characteristics." In Proceedings of IEEE GLOBECOM 2004.

[3] Cheng, L. and Marsic, I. "Fuzzy Reasoning for Wireless Awareness." International Journal of Wireless Information Networks 8, 1, 15-26, 2001.

[4] Wei, W., Wang, B., Zhg, C., Kurose, J., and Towsley, D. "Classification of access network types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialup." In Proceedings of IEEE INFOCOM. 1060-1071, 2005.

[5] Branch, J., Petroni Jr., N., Van Doorn, L., and Safford, D. "Autonomic 802.11Wireless LAN Security Auditing." IEEE Security & Privacy, May/June 2004, pp. 56-65.

[6] Airmagnet. www.airmagnet.com

[7] Netstumbler. www.netstumbler.com

[8] Wavelink. www.wavelink.com/downloads/pdf/wlmobilemanager_w p_rogueap.pdf

[9] www.highwalltech.com/products.cfm?menu=hwsent&pa ge=hwsent

[10] Computerworld article. www.computerworld.com/mobiletopics/mobile/story/0,10801,72065,00. html

[11] Air Defense. www.airdefense.net

[12] Air Wave. www.airwave.com/airwave_rogue_detection.pdf

[13] Sourcefourge presentation. winfingerprint.sourceforge.net/presentations/APTools.ppt

[14] Wimetrics. www.wimetrics.com/WAPD.htm

[15] Stevens, W.R., TCP/IP Illustrated Volume 1: the protocols. Boston, MA: Addison Wesley, 1994.

[16] Bahl, P., Padhye, J. , Ravindranath, L. , Singh, M. , Wolman, A., and Zill, B. "DAIR: A Framework for Managing Enterprise Wireless Networks Using Desktop Infrastructure." In the Proceedings of ACM HOTNETS, 2005.

[17] Wei, W.; Jaiswal, S.; Kurose, J.; Towsley, D. "Identifying 802.11 Traffic from Passive Measurements Using Iterative Bayesian Inference." In the Proceedings of INFOCOM, 2006.

1930-529X/07/$25.00 © 2007 IEEE