Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks

Summary presented by Rubeel M Iqbal

Abstract

- This paper presents an architecture for detecting and diagnosing faults in IEEE 802.11 infrastructure wireless networks
- Proposes and evaluates a technique called Client Conduit which enables bootstrapping and fault diagnosis of disconnected clients
- Present an approach for detecting unauthorized access points

Wireless Networks & Problems

- IEEE 802.11 networks are implemented widely across homes, offices, Universities and etc.
- There are problems such as
  - Intermittent Connectivity
  - Poor Performance
  - Lack of Coverage
  - Authentication Failures

Problems at a leading Corp

- "dead spots" or "RF holes"
  - due to a weak RF signal
  - lack of a signal
  - changing environmental conditions
  - obstructions.
- Locating an RF hole automatically is critical for wireless administrators;
  - can be resolved by
    - relocating APs
    - increasing the density of APs in the problem area
    - by adjusting the power settings on nearby APs for better coverage.

Connectivity Problems

- Presenters
- Presented at DACM MOBICOM 2004 by
  - Atul Adya
    - Microsoft Research
    - adya@microsoft.com
  - Paramvir Bahl
    - Microsoft Research
    - bahl@microsoft.com
  - Ranveer Chandra
    - Department of Computer Science, Cornell University
    - ranveer@cs.cornell.edu
  - Lili Qiu
    - Microsoft Research
    - liliq@microsoft.com

- Presenters
- Presented at DACM MOBICOM 2004 by
  - Atul Adya
    - Microsoft Research
    - adya@microsoft.com
  - Paramvir Bahl
    - Microsoft Research
    - bahl@microsoft.com
  - Ranveer Chandra
    - Department of Computer Science, Cornell University
    - ranveer@cs.cornell.edu
  - Lili Qiu
    - Microsoft Research
    - liliq@microsoft.com
Performance problems

- Users observe degraded performance, e.g.,
  - Low throughput or high latency

Reasons

- Traffic slow-down due to congestion
- RF interference due to a microwave oven or cordless phone
- Multi-path interference
- Large co-channel interference due to poor network planning
- Poorly configured client/AP
- Due to a slow server or proxy (not related to Wireless part of network)

Network security

- “Rogue AP Problem”
  - Are most common and serious breaches of wireless network security.
  - Can allow external users access to the resources on the corporate network
  - Rogue APs can cause interference with other access points in the vicinity
  - Detecting Rogue APs in a large network manually is expensive and time consuming

Authentication problems

- A number of complaints are related to users’ inability to authenticate themselves to the network
- Authentication failures are typically due to missing or expired certificates
- Detecting such authentication problems and helping clients to bootstrap with valid certificates is important
- Future work will investigate diagnosis of authentication problems as well

System Architecture

- System Requirements
- System Components
- System Scaling
- System Security

System Requirements

- Software Modifications to AP Software
- Client software needs to be augmented
  - Ability to control Beacons and probes
  - Ability to start an Infrastructure Network or ad-hoc network
- Availability of Database to keep track of locations of AP’s
System Components

- **Diagnostics Client (DC)**
  - Runs on wireless client machine

- **Diagnostics Access Point (DAP)**
  - Optional component

- **Diagnostics Server (DS)**
  - Runs on backend server of organization

Diagnostics Client (DC)

- Monitors the RF Environment
- Monitors traffic flow from neighbouring clients
- Collected data is used by DC to perform local fault diagnostics
- Summary of data sent to DAP or DS
- DC sends MAC and channel info every 30 sconds
- DC also accepts commands from DAP and DS if required
- If wireless is disconnected then the DC collects data and stores it locally

Diagnostic Access Point (DAP)

- Main function to accept diagnostic messages from DC
- Merge DC data with its data and send summary it to DS
- Not a requirement for the proposed architecture but can help offloading work to DS

Diagnostic Server

- Accepts data from DC and DAP
- Performs analysis on data and diagnose faults
- DS has access to the data that contains Location of AP's, MAC Address, channel info and etc.
- DS interacts with other servers to perform client and user authentication

Proactive vs. Reactive Monitoring

- **Proactive Monitoring**
  - DS and DAP's monitor the system continuously
  - If an anomaly is detected, the Network administrators or system administrators are notified

- **Reactive Monitoring**
  - Is used when the support personnel want to diagnose a user complaint

System Scaling

- System designed to be scaled as the number of clients increased
- Work offloaded from DS and shared with DAP's and DC's.
- DS are only used if the DAP's and DC's are not able to diagnose the problem and require global analysis.
System Scaling cont..

- DAP are shared resource
- Different fault diagnosis mechanisms can use a technique called Busy AP Optimization
  - An AP doesn’t perform active scanning if a client is associated with it

System Security

- Communications between DC, DAP and DS are secured using Extensible Authentication Protocol Transport Layer Security (EAP-TLS) certificates
- Malicious activity by legitimate users is beyond the scope of this paper

Introducing Client Conduit

- It is a mechanism to allow disconnected users to convey messages to system and network admins.

Step 1

- The DC on the disconnected client D configures the machine to operate in promiscuous mode.
- Scans all channels to determine if any nearby client is connected to the infrastructure network.
- It starts a new infrastructure or an ad hoc network on the channel on which it detected the client’s packets.

Step 2

- This newly-formed AP at D now broadcasts its beacon like a regular AP, with an SSID of the form “SOS HELP <num>” where num is a 32-bit random number to differentiate between multiple disconnected clients.

Step 3

- The DC on the connected client C detects the SOS, C needs to inform D that its request has been heard and it can stop beaconing.
- Rather than connecting “active scanning” mechanism of IEEE 802.11 networks is used and a Probe Request of the form “SOS ACK <num>” to D.
Step 3

- Probe Request is sent with a different SSID than the one being advertised by the AP running on D.
- This prevents some other nearby client that is not involved in the Client Conduit protocol from inadvertently sending a Probe Request to D (as part of that client’s regular tests of detecting new APs in its environment).

Step 4

- D Stops being an AP after it hears the probe request
- In response to the Probe Request, a Probe Response is sent out by D; so no more probe request from C
- D’s Probe Response indicates if D would like to use client C as a hop for exchanging diagnostic messages with the DS.

Step 5

- D starts an ad hoc network and C joins this network via Multi-Net.
- At this point, C becomes a conduit for D’s messages and D can exchange diagnostic messages with the DS through C.

Client Conduit Security and Attacks

- **Performance Degradation of Helping Clients**
  - When a connected client helps a disconnected client, we need to make sure that the helping client doesn’t feel any adverse performance issues.
  - These steps not only consume negligible resources on C but they also do not result in any security leak or compromise on C.

- **As part of the authentication step, client C obtains the EAP-TLS machine certificate from the disconnected client and validates it (for ensuring mutual authentication, client D can perform these steps as well).**
  - If the disconnected client has no certificates or its certificates have expired, client C acts as an intermediary for running the desired authentication protocol.
  - E.g., C could help D perform Kerberos authentication from the backend Kerberos servers and obtain the relevant tickets.
Client Conduit Security and Attacks

- If the disconnected client D still cannot authenticate, C asks D to send the last (say) 10 KBytes of its diagnosis log to C and C forwards this log to the DS.
- To prevent a possible DoS, the connected client can limit the total amount of unauthenticated data that it sends in a fixed time period, e.g., C could say that it will send at most 10 KBytes of such data every 5 minutes.

Fault Detection and Diagnosis

Locating Disconnected Clients
- Double Indirection for Approximating Location or DIAL is used to locate the clients.
- DIAL: when a client D discovers that it is disconnected, it becomes an AP, starts beaconing.
- Nearby connected clients hear D’s beacons and record the signal strength (RSSI) of these packets.

DIAL

- They inform the DS that client D is disconnected and send the collected RSSI data.
- The DS uses any known location-determination technique to locate the connected clients.
- The DS uses the locations of the connected clients as “anchor points” and the disconnected client’s RSSI data to estimate its approximate location.

Network Performance Problems

Detection
- Network performance problems can manifest themselves in different ways, such as:
  - low throughput,
  - high loss rate, and
  - high delay
- Detection component triggers the diagnosis component if a connection is very lossy or it experiences high delay.

Network Performance Problems

- A connection is detected as experiencing high delays if the RTT of a particular packet is more than 250 msec or is higher than twice the current TCP RTT.
- To avoid invoking our diagnosis algorithm for high delays that occur temporarily, we flag a connection only when D or more packets experience a high delay.
- A connection is classified as lossy if its loss rate (for transmitted or received packets) is higher than 5%.

Isolating Wireless and Wired Problems

- If a DC detects a network problem at a client, it communicates the issue to DAP.
- The DAP starts monitoring the TCP data and ACK packets for that client’s connection.
- The DAP computes the difference between the received time of a data packet from the client to remote host and the TCP ACK packet sent.
Isolating Wireless and Wired Problems

- The ACK is the time difference estimate of the delay incurred in the wired network.
- To ensure reasonable estimates, various heuristics used by TCP need to be applied to these roundtrip measurements as well, e.g., Karn’s algorithm

Diagnosing Wireless Network Problems

- Estimating Delay using Eavesdropping Neighbors or EDEN, which leverages the presence of other clients to quantify the delay experienced by clients.
- When a client D performance diagnosis component is triggered by its problem detection component, it starts broadcasting packets asking for diagnosis help from nearby clients

EDEN Phase 1

- The DAP periodically (say every 2 seconds) sends Snoop request packets to client D.
- When D receives a Snoop request packet, it immediately replies with a Snoop response message.
- The eavesdropping clients log the time when they hear a Snoop request and the first attempt by D to send the corresponding Snoop response packet

EDEN Phase 1

- Timing values ignored if an eavesdropping client misses either of these packets.
- The difference between the recorded times is the client delay, i.e., application and OS delays experienced by D after receiving the request packet.
- For robustness, Snoop requests are sent a number of times.
EDEN Phase 2

- A similar technique is used to measure the AP delay, i.e., client D sends the Snoop request packets and the AP sends the responses.
- Client D also records the round trip times to the AP for these Snoop requests and responses along with the number of request packets for which it did not receive a response, e.g., the request or response was lost.
- In the end all the eavesdropping clients send the AP and client delay times to the client D.
- The difference between the round trip time reported by D, and the sum of the delays at the client and the AP, approximates the sum of the delay experienced by the packet.
- The client can then report the client/AP/medium breakdown to the network administrator.

Rogue AP Detection

- When the DS receives information for an AP from various clients, it uses DIAL to estimate the AP's approximate location based on these clients' locations and the AP's RSSI values from them.
- The DS classifies an AP as rogue if a 4-tuple does not correspond to a known legal AP in the DS's AP location database:
  - MAC address is not present in the database
  - AP is not in the expected location
  - SSID does not match any SSID(s) in org
  - AP is on a channel that does not overlap with the one on which it is expected

IMPLEMENTATION

- Kernel drivers:
  - A miniport driver
  - Intermediate driver (IM driver) called the Native WiFi driver
Miniport Driver

- The miniport driver provides basic functionalities such as sending/receiving packets, setting channels, etc.
- It exposes sufficient interfaces such that functions like association, authentication, etc. can be handled in the IM driver.

Intermediate Driver

- The IM driver supports a number of interfaces (exposed via ioctl()) for querying various parameters such as the current channel, transmission level, power management mode, SSID, etc.
- It allows the user-level code to request for active scans, associate with a particular SSID, capture packets, etc.

Addition to IM Driver

- Capturing packet headers and packets:
- Storing the RSSI values
  - RSSI value of every received packet and maintain a table called the NeighborInfo.
  - Maintain an exponentially weighted average with the new value given a weighting factor of 0.25.
  - The RSSI information is needed for estimating the location of disconnected clients and APs using DIAL.

Fault Diagnostic daemon

- This daemon gathers information and implements various mechanisms
  - Collect MAC addresses of APs for Rogue AP detection, perform Client Conduit,
  - If the device is an AP, it communicates diagnostic information with the DS and the DCs
- The Diagnostic daemon on the DC obtains the current Neighbor-Info table from the kernel every 30 seconds.

DC and DAP Components
SYSTEM EVALUATION

Cost of Individual Operations
- 4 machines, A, B, C, and D to determine
- If placing a machine in promiscuous mode has any effect on the machine’s incoming/outgoing bandwidth
- We setup the machines such that
  - machine A did a TCP transfer to C at full blast
  - B performed a full blast TCP transfer to D.

Bandwidth Experiment
- Experiment was performed three times;
  - machine C was placed in normal mode first and then in promiscuous mode.
  - We observed that C’s throughput was largely unaffected in promiscuous mode
    - 254.7 KB/sec (standard deviation of 63.7 KB/sec) in the normal mode case
    - bandwidth of 252.3 KB/sec (standard deviation of 21.7 KB/sec) in the promiscuous mode case.

CPU Experiment
- Full blast TCP transfer between two machines A and B;
  - during this process,
    - we first placed machine M in normal mode and then in promiscuous mode.
    - CPU overhead of placing it in promiscuous mode is quite low, below 10%
    - none of the packets were dropped
  - Thus, these results show that the CPU overheads on a machine due to promiscuous mode are reasonably low.

Cost of Client Conduit
- The Figure shows total time taken along with a breakdown of the
  - Connection Setup part of the protocol. “User time” indicates the end-to-end time taken by our user-level implementation
  - “Kernel time” indicates the time taken by the relevant ioctls (input/output control) for the same functionality.
Breakdown of costs for Client Conduit. The protocol steps are executed from the bottom entry in the legend to the topmost, i.e., starting at “Set channel”.

One can see that the Connection Setup and association time for the disconnected client is quite reasonable:
- It takes less than 5 seconds to run the setup
- 1.9 seconds to associate with a connected client \( C \) in ad-hoc mode so that the MultiNet protocol can be started on \( C \).

After MultiNet starts running on the connected client, the disconnected client can interact with the DS to diagnose its problems.

To evaluate the time taken to perform these transfers via MultiNet, we ran an experiment:
- A machine \( D \) sent files of different sizes (100KB, 500KB and 1MB).

In Multi-Net mode, file transfer speed is a direct function of the time a connected client stays in the ad hoc network.

We placed 3 connected clients in 3 offices on the same floor of our building.

We obtained the floor map, and applied the Cohen-Sutherland line-clipping algorithm to compute the number of walls between each of the three connected clients and the other rooms.
Location Determination

- Placed disconnected clients at 7 different locations
- Used AiroPeek to measure the RSSI of the disconnected clients.
- We then applied the equation specified in to compute the wall attenuation factor (WAF).
- Based on the WAF, we inferred that the disconnected client is in location $X$.

- RADAR algorithm was used on the collected RSSI data for locating the disconnected client $D$ using the precise location of the connected clients.
- Computed the error in $D$'s predicted location with respect to its actual location.
- Ran the algorithm again by assuming that there was an error in estimating the location by a distance of 3.3 meters; this distance corresponds to the average width of a room.

The results show that when there is no error in the known location of the connected clients,

- The median error is 9.7 meters.
- This error increases to at most 12 meters when the estimated location of one or more clients is one or two rooms off from its true location.

Estimating Wireless Delays

- Experiment to determine EDEN's accuracy in determining delays at an endpoint.
  - A client machine was associated with another machine running as an access point.
  - Both machines had Netgear MA521 802.11b cards and the corresponding Native WiFi drivers.
Estimating Wireless Delays

- Delays were injected in the path of all packets at the client (varying from 30 to 300 msecs).
- To emulate the EDEN protocol:
  - the AP sent 20 ping packets to the client; the ping packets and replies emulate the Snoop request and response messages in EDEN.
  - A third machine running AiroPeek was used to snoop on these ping packets;
- Figure shows that EDEN is reasonably accurate in estimating the delays at a client: EDEN can estimate client delays with an error less than 5% of the actual introduced delay.

Estimating Wireless Delays

- Delays were injected in the path of all packets at the client (varying from 30 to 300 msecs).
- To emulate the EDEN protocol:
  - the AP sent 20 ping packets to the client; the ping packets and replies emulate the Snoop request and response messages in EDEN.
  - A third machine running AiroPeek was used to snoop on these ping packets;

Rogue AP Detection

- Overlapping channels in IEEE 802.11 not only interfere with one another but it is sometimes possible for a NIC on one channel to decode packets from another overlapping channel
- Experiment was performed:
  - in which an AP was placed on channel 1 and a nearby client checked for the AP’s beacons on all 11 channels. Experiment was repeated by placing the AP in all channels from 2 to 11 and document where it could be heard.

Overlapping channels on which an AP is overheard
The results show that even though one cannot rely on overlap as a guaranteed mechanism for detecting Rogue Aps, it does reduce the need of performing frequent active scans. This observation also implies that there are more opportunities for detecting Rogue Aps: for a Rogue AP to go undetected, it must be far away from any client that is on an overlapping channel.

The proposed implementation performs an active scan every 5 minutes; this period is referred as the Active Scan Period. 2 sec out of 300 sec is a small fraction. Clients should perform the scans at appropriate times; otherwise, network traffic can get disrupted; packets sent to client may drop.

The maximum idle time duration available during every 5-minute period at different times of the day.
In this paper, we are presented with a novel solutions for detecting a variety of faults proposed approaches for analyzing performance problems experienced by end-users. The initial results show that the mechanisms of locating RF holes, detecting Rogue APs, and diagnosing performance problems are effective and impose low overheads.

A novel mechanism called Client Conduit can be used for assisting disconnected clients in real-time. These techniques in conjunction with our general architecture that uses clients, APs, and backend servers together for diagnosing wireless networks make our system unique and practical.