Do You See What I See (DYSWIS)

Introduction:

DYSWIS is a system that leverages distributed network nodes to gather network management information for fault isolation and diagnosis. This project extends this functionality to wireless home networks.

It is quite common to observe low download speeds in a home network which may be due to abundant reasons. The causes may be busy network, packet drops, poor signal quality or RF interference. The identification and isolation of these causes is quite non trivial and is the inspiration behind the project.

The central idea is to monitor data reagrding all the metrics related to the wireless interface card storing the data and probing peers for the same information when fualt diagnosis is in progress.

Taking the DYSWIS Approach:

The fault detection mechanism is modelled on the medical diagnosis system and the cooperation between peer nodes is modeled on human social network system. A medical diagnosis of a patient by a doctor involves analysis of symptoms, certain diagnostic tests to isolate or confirm possible causes, in addition to leveraging knowledge already available. Identification of the root cause of a failure in wireless networks requires right tools and logic that comes from observed scenarios and simulations.

Challenges:

The major challenge was to identify a reliable and sustainable source to gather the network manangement information. Microsoft Native Wifi API was the most suitable choice to monitor network metrics.

Microsoft Native Wifi:

The Native Wifi application programming interface (API) functions have two purposes: to manage wireless network profiles and to manage wireless network connections. For this purpose the API has abundant structures and enumerations where wifi metrics are stored and can be queried. It is this feature of the API that this project leverages.

The structure below is by far the most relavant to current project

```
typedef struct _WLAN_PHY_FRAME_STATISTICS {
        ULONGLONG ullTransmittedFrameCount;
        ULONGLONG ullMulticastTransmittedFrameCount;
        ULONGLONG ullFailedCount;
        ULONGLONG ullRetryCount;
        ULONGLONG ullMultipleRetryCount;
        ULONGLONG ullMaxTXLifetimeExceededCount;
        ULONGLONG ullTransmittedFragmentCount;
        ULONGLONG ullRTSSuccessCount;
        ULONGLONG ullRTSFailureCount;
        ULONGLONG ullACKFailureCount;
        ULONGLONG ullReceivedFrameCount;
        ULONGLONG ullMulticastReceivedFrameCount;
        ULONGLONG ullPromiscuousReceivedFrameCount;
        ULONGLONG ullMaxRXLifetimeExceededCount;
        ULONGLONG ullFrameDuplicateCount;
        ULONGLONG ullReceivedFragmentCount;
        ULONGLONG ullPromiscuousReceivedFragmentCount;
        ULONGLONG ullFCSErrorCount;
} WLAN_PHY_FRAME_STATISTICS, *PWLAN_PHY_FRAME_STATISTICS;
```

Simulation and Testing:

Modelling the real world scenarios is quite a non trivial task since the wireless networks behave in a more unpredictable way as compare to the wired world. There are a lot of factors to consider if a proper fault isolation model is to be built. The RF rnvironment, distance from the access point, received signal strenght and the number of peers.

The initial idea was to use the network simulation tools however the tools did not provide the kind of historical data that was required. Thus a simple home network with 2 nodes and an access point was chosen as the simulation environment.

Requirements:

The fault diagnosis system was intended to support the following scenarios

Busy Netowork:

When a large number of network nodes are present in the network it is understandable that the netwlok would experience slow downloads. Thus the number of bytes exchanged by the network nodes would be recorded to see if the wireless network was busy

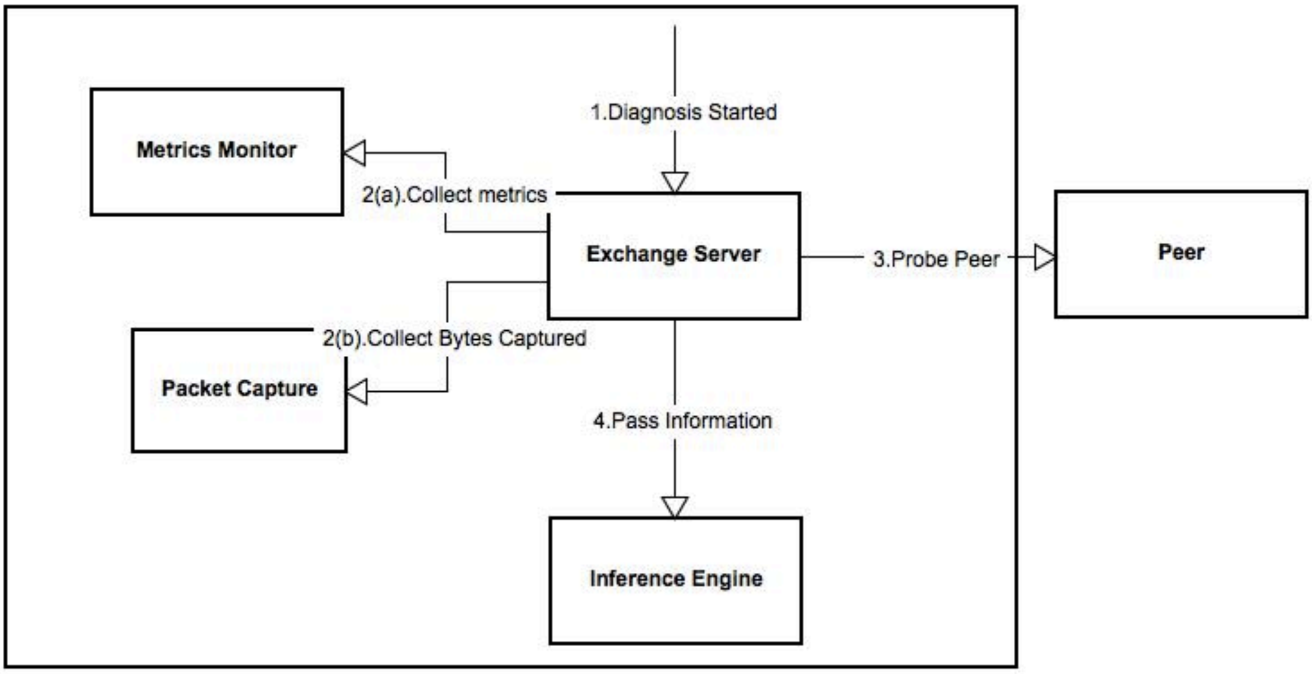
Poor Signal Quaity:

The signal quality varies directly with the distance between the nodes and the access point. The signal quality would be measured theough series of intervals of time to see if the value falls below a threshold where issues like slow download speed would occur.

RF Interference:

This one one of the most difficult scenario to predict. The RF interence could be caused by any device that operated on the same frequency as the wireless netowrk (~ 2.4MHz). The variations in signal in presence of different devices would be measured to identify if there is any RF interference in the wireless environement.

Design And Implementaion:



The architecture of the proposed system consists of different modules separated based on the function.

Metric Monitor:

This module interacts with the Native Wifi API and stores historial data. It is composed in Java, however the API only supports C/C++ thus a JNI bridge is built so that data and logic can be sandboxed into one engine.

Packet Capture:

The second module is the packet capture engine which snoops all the packets and stores the bytes that have been exchanged by the current host.

Exchange Server:

This module is responsible for exchanging the data with the peers. It is TCP server that listens for connections from peers.

Inference Engine:

This is the module where the algorithm resides. It applies all the observations to knwn facts to deduce the isolated fault and the cause of the fault.

Simulation and Testing:

As mentioned before the simulation of the real world in wireless environment is a non trivial task and the network simulators would not provide the metrics in the format that is required.

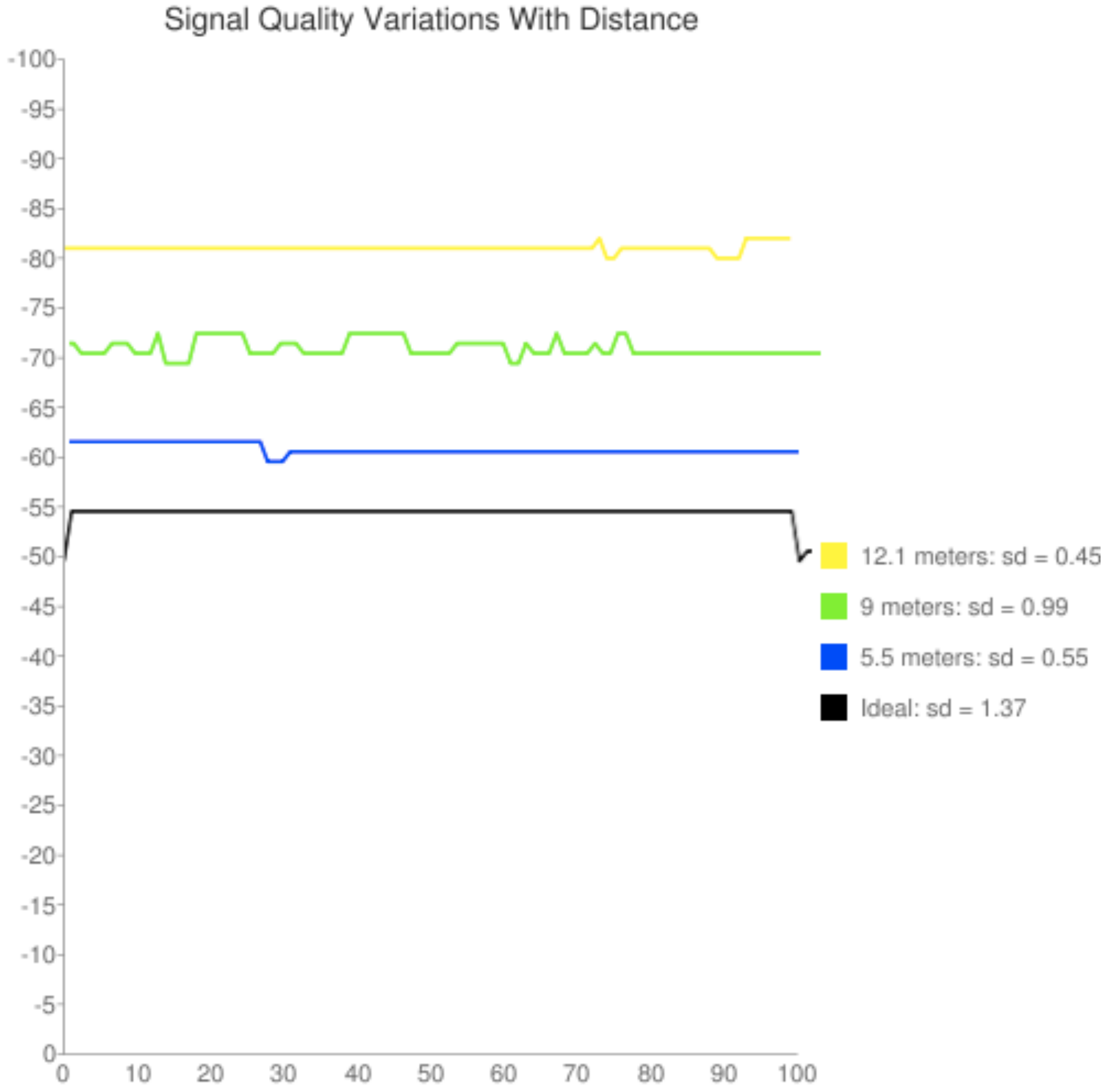
Thus the simple approach was taken that measures the metrics. Network node and an access point were used to create different scenarios to staright forward yet interesting results.

The network node operated on Windows 7 OS with an AMD 64 bit processor.

The Access point was a Cisco Linksys router working on 802.11 n

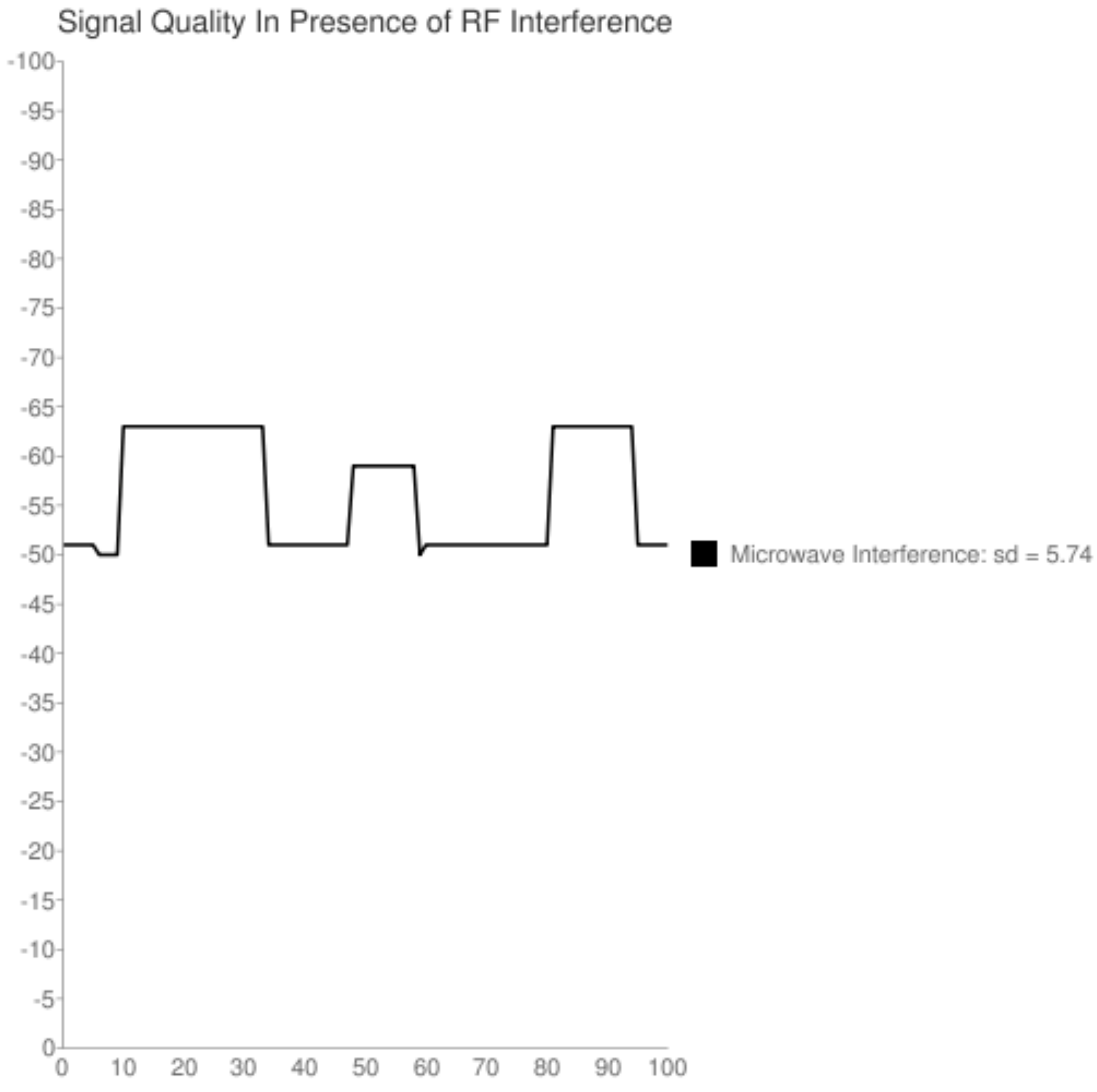
Test For Signal Quality:

The test for the variation in signal quality was quite straight forward. To simulate ideal condition the access point and the network node were taken to a remote location and the signal quality was measured for a series of 10 minute interval. The below figures show the variation of signal strength with distance.



RF Interference:

The signal quality in presence of RF interference had a lot of variations as compared to situations where there was no interference. To create the environment, bluetooth frequency and microwave frequency was used and the behaviour is shown below.



Observations:

As represented by the above graphs, it is quite clear that the signal strength when the environment remains static, remains constant to a certain extent. This behaviour quickly changes when the distance between the nodes and the access point increases. Signal variations are observed, however the standard deviation does not go beyond 2.

The experiment when repeated for same distance for a continous time periods of 1000 seconds over a period of 24 hours displayed similar results.

The Frame CheckSum error count was also taken into account to infer the presence of RF interfernce. However, the count of errors in a given period of time kept increasing linearly with time. This made it difficult to reach any conclusion with respect to the FCS error count.

RF interference created severe variations in the signal strength. The standard deviation was beyond 3 in such cases.

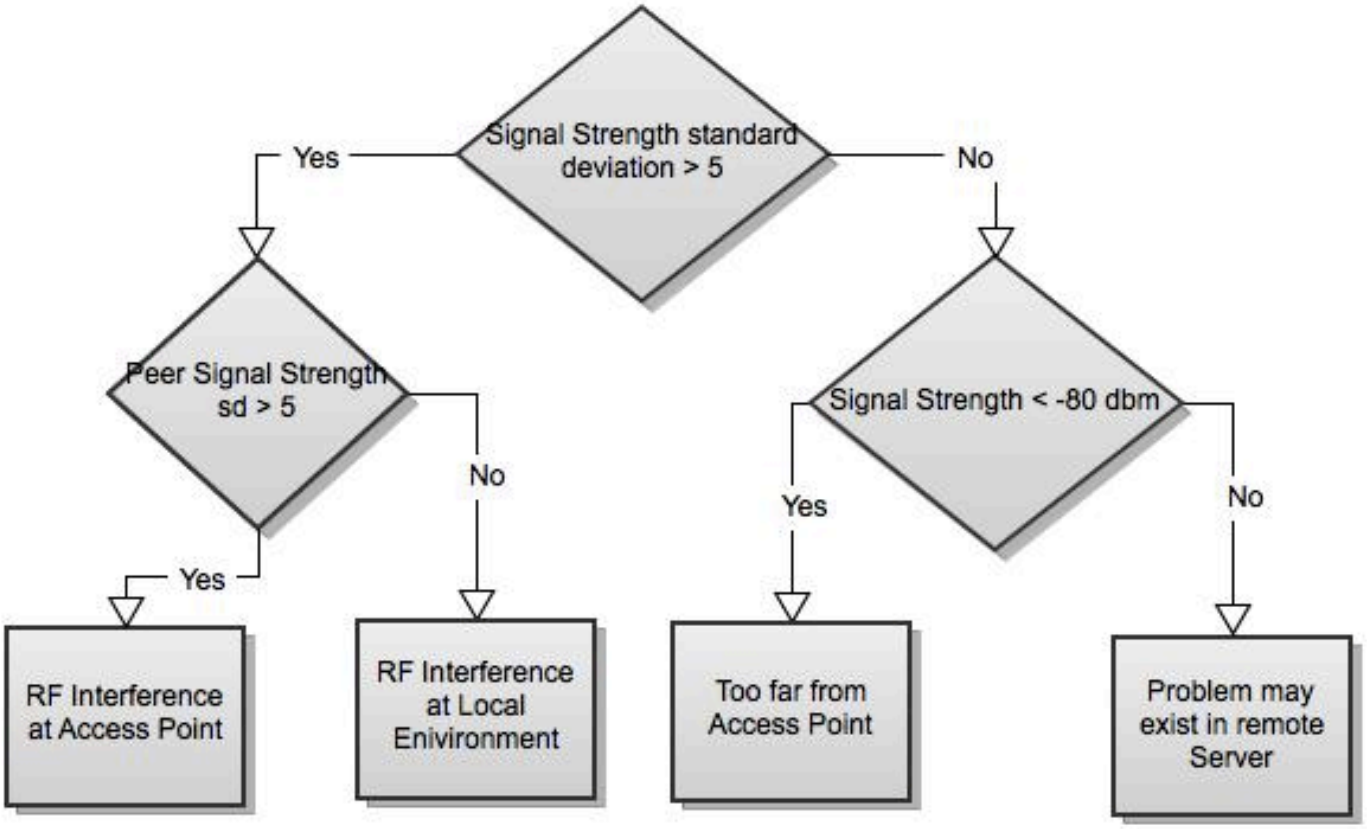
The packet capture engine captures the wirelength of the packets the node receives. Lower download speeds were observed at current node when another node was streaming or downloading a large file. However, no threshold values were deermined to conclude that the lower downloads could be because of bandwidth hogging by one node. In other words notion of busy network was in conclusive. However, the bytes exchanged by all the peers are kept track of.

A brief summary of the observations include the following facts

- Signal Quality remains constant for static environment
- Signal quality variations are too high in case of RF interference
- FCS grows linearly with time. No plausible conclusion reached
- No threshold on bytes transferred

Algorithm:

Based on the observations made above the following algorithm has been derived.


Signal Strength standard deviation > 5
Yes
No
Peer Signal Strength sd > 5
Signal Strength < -80 dbm
Yes
No
Yes
No
RF Interference at Access Point
RF Interference at Local Enivironment
Too far from Access Point
Problem may exist in remote Server

Future Work:

The scope of this project is limited yet has the potential to expand to various scenarios to derive more complicated conclusions based on the foundation created. The API is a well developed and well maintained by Microsoft. The packet capture uses the jnetPcap library based on Java and winPcap.

Few notable enhancements that can be possible include:

- Packet Capture from non participating nodes
  The packet capture is exchanged between nodes participating in the dywis project only. However, the pcap library can be expanded to capture all the packets in the wireless network.
- Monitor Received Packet counts and FCS errors to diagnose Packet drops
  The received packet counts and the recorded FCS error can be used to identify if there are a lot of packet drops. It may have to be used in conjuction with the packet capture to keep track of the sessions.
- Signal quality variations due to various RF interferences
  The variations were noted only in the presence of the mirowave frequency. It may be expaded to support the identification of different RF interfering frequencies.