Paper 25 Survey

The paper presented by Dr. Timothy Brown, et al., is proof of concept on how to measure performance of physical mesh networking hardware and protocols. Rather then relying solely on simulation software as many authors have done in the past, this team used a combination of off-the-shelf hardware and in-house developed software to measure dynamic source routing (DSR) mesh networking performance in real-time. While their “test bed,” as it is called throughout the paper, has the potential to facilitate the monitoring of 10’s of nodes, they tended to only test six to eight nodes. The mismatch between potential nodes and the actual number of nodes used is understandable as there are real costs associated with physical hardware and available open space for testing a large number of nodes. Dr. Brown, et al., also admit that their monitoring hardware was significantly under-powered and would have to be upgraded in order to capture the real-time behavior of a larger DSR network.

The test bed used to conduct the research was located at a federal facility near Boulder, Colorado. The facility consisted of an unobstructed, flat 7 km^2 area. The unique open area of the facility allowed for different types of nodes ranging from hand-held radios to typical drones. Normally a flat area such as this would be “too perfect” to do any sort of real-world performance measurement. Fortunately, the test bed focused on measuring the resiliency of the DSR algorithm itself, rather then on radio interference. In order to capture real-time DSR radio traffic, each node must report wirelessly to a fixed node in the facility. The fixed node takes in all the individual reports and relays them over TCP/IP to a remote monitoring location. It is at the remote monitoring location where the data from all the nodes is compiled and metrics are calculated in real-time. It is also at this location in which one of the many choke points in the system exists. Consider figure 1:

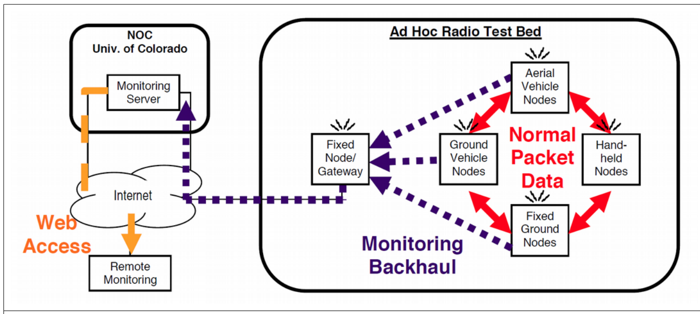

Figure 1 shows the overview of the entire system. Inside the “Ad Hoc Radio Test Bed” box is where the DSR traffic is created and captured. It is unknown what frequencies are used to produce DSR traffic and to produce reports for the fixed node/gateway. The team did report a 10% decrease in performance when DSR monitoring was activated. The performance decrease and subsequent packet loss between nodes is most likely a result of wireless noise and the fixed node/gateway choke point.

One of the four major goals of Dr. Browns research team was to accomplish sufficiently complete, real-time data presentation. They accomplished this goal with little packet loss and reasonably good results considering the hardware and resources at their disposal. In summary, the team used combinations of off-the-shelf computing hardware.

Each node consisted of:

*Soekris Single Board Computer
*Orinoco 802.11b PCMCIA card
*1W Fidelity-Comtech Bidirectional Amplifier
*GPS Receiver

The Gateway consisted of a Click Modular Router.

The Monitoring Server is assumed to be a university rack-mounted data center.
The Remote Monitoring CPU is assumed to be a Pentium-III class computer.

Each individual node consists of a communication framework and an additional system monitoring module. The framework is called the Mesh Network Radio (MNR) framework. The MNR framework consists of a real-time operating system, various hardware drivers, communication interconnects, and a dedicated DSR routing subsystem. Within the MNR framework also lies the Monitoring Module (MM). The purpose of the MM is to sample all incoming and outgoing communications of the host node. Once sampled, the MM uses the internal data structures and various MNR framework services to send real-time metrics directly to a listening gateway. As can be seen from the provided illustration, it's a rather interesting relationship between the MNR and the MM. Instead of having it's own dedicated framework, the MM piggybacks on the MNR which maximizes the usage of scarce computing resources.

Figure 2 to be placed here.

The next major part of the test bed is the university hosted monitoring server. The server consists of a single 450Mhz machine running open-source software such as MySQL. It is the job of the server to not only archive all incoming metric packets, but also to sort them out into three distinct categories: per-packet data, per-node data, and application messages. Once sorted, the three different packets can be queried to produce graphical output on any browser or end-user client.

The separation of metric data into three types of packets allows for a deep understanding as to the behavior of the nodes using the DSR protocol. The per-packet data type, for instance, allows for an explicit tracing of a single packet throughout the network. At each hop, the MM reports to the server the location of the packet. This will allow analytical tools to determine the health of the overall network in regards to dropped packets. The next type of packet, the per-node data packet, consists of metrics regarding the physical location of the node and it's relative distance from other nodes. At various time intervals, the MM simply samples a GPS sensor and reports it to the server. The last packet, the application messages packet, allows for the capturing of incoming text messages and other non-DSR related communications. It should be noted that DSR path maintenance packets and other such DSR protocol specific packets are not captured and reported to the server by the MM.

The last major part of the test bed is the PC-hosted graphical client. The graphical client is programmed in Java and connects to the remote server over TCP/IP. Depending on the use case, the graphical client can either display the states and location of the nodes in real-time, or allow the user to "play back" a time-lapse recording of previous node states. The researchers are particularly excited about the pluggable-map feature of their client. Their PC client can be configured for any environment if given the correct map and GPS coordinates. The usage of maps with embedded GPS coordinates allows users to accurately follow packet and node traversals and see the actual distance between each node. Another major feature of the PC client is the displaying of various graphs. One such graph is the Packet Graph. The Packet Graph displays metrics regarding packet traversals throughout the DSR network. Statistics such as packets sent, packets received, and logical links between nodes are easily obtained through the graphical client.

The test bed developed by Dr. Brown et. al., is a refreshing change of pace from other ad-hoc mesh networking papers. Instead of primarily relying on simulations and assumptions, this team built their own nodes using off-the-shelf components and tested them in a physical environment. While it is true the test bed environment was ideal as it appeared to have very little in the means of radio interference or obstacles, the paper is still useful in the fact that it is every easy to understand and follow the logic of the paper and naturally understand many of the results presented. In addition, the paper allows for a lot of future work opportunities because it not only gives a great starting point, but also discusses the technologies used to create a great data collection and analyzing tool for any number of mesh networking protocols.