Routing Cost and Latency in the VillageTelco Wireless Mesh Network

Michael Adeyeye
Department of Information Technology
Cape Peninsula University of Technology
Cape Town, South Africa
adeyeyem@cput.ac.za

Antoine Van Gelder
7 degrees
Cape Town
South Africa
antoine@7degrees.co.za

Sunday Ojo
Faculty of Information and Communication Technology
Tswane University of Technology
Pretoria, South Africa
ojoso@tut.ac.za

ABSTRACT

Afrimesh is the Network Management System (NMS) used in the VillageTelco (VT) project. The NMS uses both Simple Network Management Protocol (SNMP) and Internet Control Message Protocol (ICMP) as its network management protocols. It provides a thorough report on network health, such as interference, noise level and latency. In addition, it displays the current network topology and can run at a node thereby making it efficient to identify compromised or isolated node(s). This article presents the performance of a typical mesh network. The performance metrics include signalling overhead and network latency of a small scale mesh network with few nodes. Experiments showed that an additional hop in a network increases the network latency by averagely 35ms. In addition, the signalling overhead of a VT network increases with time at every node.
However, the footprint is small and would not impede the performance of a network in a large-scale deployment.

Categories and Subject Descriptors

C.2.2 [**Computer Systems Organisation**]: Computer-Communication Networks
C.2.3 [**Network Operations**]: Network Management

General Terms

Experimentation, management, performance, verification

Keywords

Afrimesh, Wireless Mesh Networks, Network Management

1. INTRODUCTION

A WMN can provide network capacity for community activities, such as content distribution, education, health care, games, file sharing, peer-to-peer applications and services and resource sharing. VillageTelco (VT) is another WMN project. It is a local and wireless infrastructure that can provide a local telephone network and Internet services for personal use and make a sustainable business for interested parties. Its wireless nodes, called Mesh Potatoes (MPs), form a WMN with nearby wireless nodes to provide wireless links. An MP (Mesh Potato) includes the Afrimesh Network Management System (NMS). An NMS is a service that employs a variety of tools, applications and devices to assist human network managers in monitoring and maintaining networks. Its concepts in wireless mobile network include mobility management, power management and network monitoring and profiling.

Traditional solutions for monitoring wireline networks provide poor performance in mesh networks due to several reasons. First, the network is being maintained by combined effort of the hosts that make the wireless network. Second, the hosts often operate under severe constraints, such as limited bandwidth. Third, mesh networks experience significant signal quality fluctuation, such as interferences, hidden hosts and unreliable physical medium. While the previous paper [2] reported the power consumption, antenna gain and the hardware and software architecture of an MP (Mesh Potato), this paper presents Afrimesh and the performance of a typical VT network using it.

Network Management tools are required for network monitoring and profiling. MPs, Afrimesh and other related WMNs are discussed in this paper. The paper is arranged as follows: Section II presents a background of WMNs; section III presents related WMN works; section IV presents Afrimesh routing and network management protocols; section V presents experiments that were carried out; and section VI discusses the results of the experiments. The conclusion is presented in section VII.

iiWAS2013, 2-4 December, 2013, Vienna, Austria.

2. BACKGROUND

NMS and its tools are required to speed up development and testing of novel protocols and architectures. Functions of the tools include initial network planning, frequency allocation, fault management, performance management, band-width management and accounting [3]. One of the tools used in NMS is Captive Portals, which provide access control and bandwidth sharing between clients. An example of a captive portal solution is the Nocat system [4]. The WiFi-Dog system is another popular captive portal system. Another project dedicated to developing software for WMNs is CUWIN, which is used to create new solutions and protocols. Open source software programs used to manage user authentication for multiple wireless hotspots include NoCatAuth. Groups, such as freifunk, CUWIN and Open Mesh, have focused on developing open source software that enables meshing. An example of the software is CUWiNware. As an alternative to proprietary solutions, Wi-Fi (Wireless Fidelity) equipment can be used with modified firmware using a Linux distribution, such as OpenWRT and DD-WRT. The solution is aimed for embedded devices and enables the easy sharing of network resources. In addition, it reduces the total cost of deploying a WMN [5].

Problems encountered in WMN solutions include bandwidth degradation, radio interference and network latency. Owing to this, the inherent multi-hop mesh networks only offer limited scalability and could be unsuitable for large-scale network deployments. Some of the challenges faced during deployment are ad-hoc partitioning, DNS route maintenance, link quality variability and inconsistent transmit power. There are also problems associated with the routing protocols. Routing protocols used by WMN solutions vary from one product to the other. Cisco solutions use a proprietary routing protocol named Adaptive Wireless Path (AWP), Nortel solutions use the traditional Open Shortest Path First (OSPF) wired routing protocol and VMesh uses Optimized Link State Routing (OLSR), which is a standard pro-active routing protocol. RoofNet uses a hybrid approach called Srcr, which combines link state and DSR-style on demand querying. Other protocols used in WMN solutions include Ad-hoc On-Demand Distance Vector (AODV) and Hazy Sighted Link State (HSLS) routing. There are also various protocols used in NMSs. Network Management protocols include SNMP , ICMP , and netconf [6]. They are used to support network and network devices management. An easy way to manage WMNs would be to extend the application layer protocol called Simple Network Management Protocol (SNMP) [2].

3. RELATED WORKS

The related WMN projects include Meshmon [1], Abare [5], Meshman [6], DAMON [8], and MTV [9]. Owing to the number of pages constraint, this section only presents MeshMan and MeshMon.

3.1 MeshMan

In Meshman, a lightweight query response mechanism was designed in place of SNMP using simplified encoding rules. The mechanism used simplified encoding rules as the network management protocol. The Optimal Link State Routing protocol (OLSR) [9] was used as the default mesh network routing protocol. Two considerations that motivated the design of MeshMan are WMNs intricacies, which include volatility of wireless channels, dynamic routes and limited bandwidth, and interoperability of wireless and wired networks [6].

Each mesh router in MeshMan is identified by a mesh ID, which is automatically configured in a hierarchical fashion. Mesh IDs are stored at a central location called a mesh manager. A mesh manager serves as routing directives for queries generated from the mesh manager to mesh routers. Unicast queries are routed using source routes encoded in the mesh ID and delivered over the OSI link layer. The layer eliminates the dependency on mesh routing protocols in use. Broadcast queries are flooded in the network for maximal resilience to packet losses. In events of missing or mismatched information at the mesh manager, MeshMan provides ARP-style resolution over multiple hops for mappings between MAC addresses and mesh IDs as fail-safe [6]. In the work, temporary unavailability of nodes in Meshman due to misconfiguration and/or slow convergence of routing table in the Meshman testbed were observed. In the case of network-level disconnection of a physically connected network, node state information (e.g. routing table entries) becomes unavailable when it is most needed. As a result, the design and implementation required that the solution be robust to network layer failure for the purpose of diagnosing and correcting network layer faults. In addition, it must be self-reconfigurable during failure of nodes and addition of new nodes.

3.2 MeshMon

Meshmon is a framework proposed and implemented in [1]. It performs network monitoring to help an administrator with his tasks. However, the framework is reduced to simply monitoring the network. It takes some actions automatically if the network behavior is different from a standard set statically by the administrator. Meshmon requires two sets of metrics.

The first sets of metrics, called baseline metrics, are sufficient when the network performance is satisfactory and can be used for coarse identification of potential problems. A comprehensive metric collection is however required when there are problems in the network. The speed of metric collection/generation is often the computational bottleneck.

As a result, visual rendering of the data could be slow. Owing to this shortcoming, there is a need for a methodology of monitoring and metric collection that is bandwidth-efficient and scalable with respect to the number of devices in the network.

4. AFRIMESH ROUTING AND NETWORK MANAGEMENT PROTOCOLS

Afrimesh makes it easy to create an IP network and is built on top of the BATMAN (Better Approach To Mobile Ad-hoc Networking) project to provide a simple network management dashboard. The dashboard enables network operators to create and sustain a resilient communications network. Afrimesh is a web application that provides node/client management, network maps maintenance, and network monitoring and bandwidth management.

The protocols supported by Afrimesh are SNMP (Simple Network Management Protocol), ICMP (Internet Control Message Protocol) and BATMAN. They report topology information and are responsible for route selection. All other management traffic utilizes TCP and is routed by the mesh network. BATMAN does have elements of classical routing protocols: It detects other BATMAN nodes and finds the best way (route) to deliver packets. It also keeps track of new nodes and informs their neighbours about their existence.

SNMP, as an application layer protocol, operates on top of the UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) transport protocol. It relies on a functioning IP layer to route information. In SNMP, each managed node owns a Management Information Base (MIB), which is a collection of information that is hierarchically organized. MIBs are accessed using SNMP. The protocol is designed to work over heterogeneous link layer technologies, where network bandwidth and storage space are relatively abundant. Mesh routers are embedded devices with limited storage capacities, and their networks experience routing flaps and instability. Running SNMP in mesh networks can be heavy weight in bandwidth consumption. As a result, it can be resource-intensive if the network layer has not been configured properly. The reason is that SNMP relies on a functioning network layer to reach nodes that are more than one hop away.

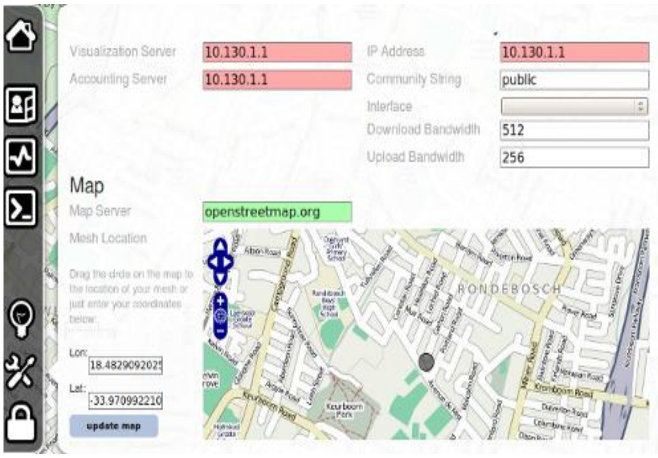

Figure 1: The Afrimesh Network Management Interface

The Internet Control Message Protocol (ICMP) provides a mechanism to send error messages, when services on routers and hosts are not available.

The UDP protocol lacks the reliability and data ordering that TCP has. Hence, datagrams may be out of order or lost in transit. However, it introduces no extra overhead to offer complex functionalities. It is faster and lightweight. Owing to this, it is optimally used in time-sensitive applications (e.g. network management). While BATMAN is used as the routing protocol, ICMP and SNMP are used as the network management protocols. Both ICMP and SNMP gather required network data, which are sampled and displayed by the visualization server. The protocols are selectively used based on network conditions.

Afrimesh look-and-feel on a PC and an MP are the same. Figure 1 shows the dashboard of Afrimesh on a PC, when it is running. The settings page, which can be found in both an MP and a PC running Afrimesh, enables an administrator to set the co-ordinates of the device and the IP addresses of the visualization and accounting servers. Other pages that can be found in the Afrimesh NMI include the people, the network health and the network logs pages. The people page shows the names of people with MPs, their email addresses and the IP addresses of the MPs. The network health page reports the overall signal strength, interference and noise level for the network. And lastly, the network log page gathers information about various activities taking place in the network, such as mesh nodes and nanostations joining and leaving the network. Critical tasks, such as change in ESSID, channel and BSSID, can be performed without rebooting a node.

5. EXPERIMENTS

For experimental purposes, the simplified testbed shown in Figure 2 uses three MPs (nodes 1, 2 and 4), one Ubiquiti nanostation as a SuperNode (SN) and the Afrimesh network management server (node 3). Nanostations are repeaters, which are typically used to bridge two or more mesh clouds to form a large network. In peer-to-peer networking, a super node is any node that acts as a network relayer, which handles data flow and connections for other users. Hence, a nanostation functions as a supernode. Hence, a nanostation functions as a supernode. Ideally, a supernode would contain three nanostations mounted on a single pole, each covering a 120 degree sector.

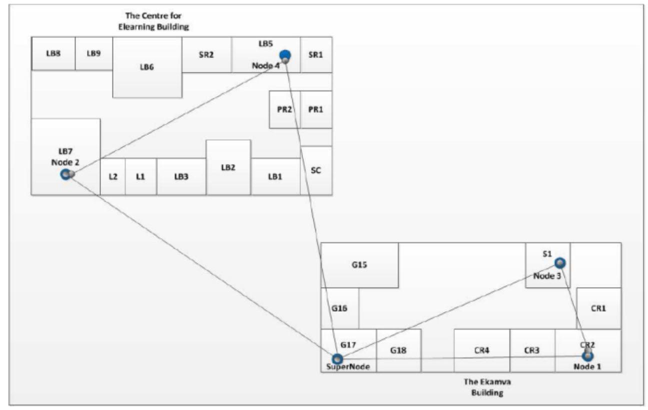

Figure 2: The VT Testbed at Cape Peninsula University of Technology, Cape Town

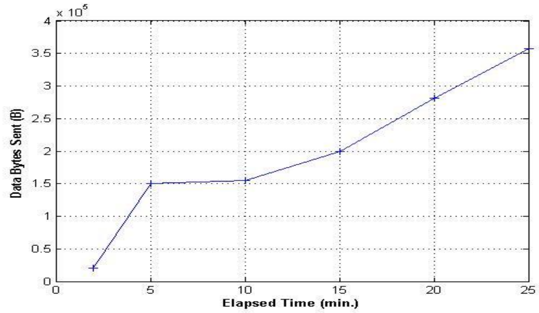

Figure 3: The Routing Protocol (Batman) Overhead

The directional antenna of the nanostations typically offers a 2 km radius of coverage to the VT. The management system is a PC with a 2.5GHz Intel processor, 2GB of RAM and an IEEE 802.11g Zyxel wireless USB card. It uses the Ubuntu 11.04 operating system. The MPs and nanostations were distributed in a linear topology, where each MP only reaches the other MPs in the same building with it. The management server (node3) is located in one of the buildings. While it is one hop away from node 1, it is two hops away from nodes 2 and 4. The nanostation in the testbed links the two autonomous networks to form one big network.

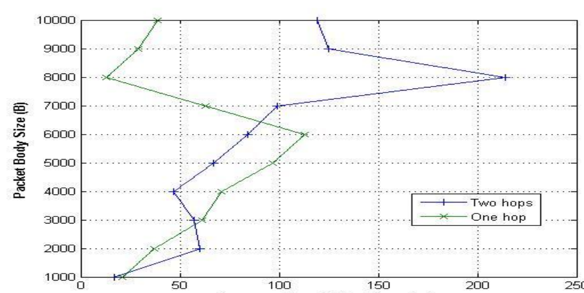

Figure 4: Transmission Delays for One and Two Hops

Two different cases were evaluated: (1) an unloaded scenario, where only traffic due to the BATMAN routing protocol is considered, and (2) a loaded scenario, where the traffic generated by an MP has to compete with several bandwidth- hungry TCP flows. Figure 3 represents the unloaded scenario, and Figure 4 represents the loaded scenario. Figure 3 shows that the routing protocol overhead increases with time when data flow from one node to another.

The first experiment required gathering the UDP traffic at a node, when the network was idle. The experiment was repeated over varying times, and the tcpdump utility was used to listen to the UDP traffic at the node. The traffic represents the routing protocol (BATMAN) signalling over-head.

The second experiment required determining the delay in transmission of data between nodes. In addition, the delay in transmission of data when packets traverse a supernode was determined. While the former (i.e. between two nodes) was a one-hop transmission i.e. between node 3 and the SN, the latter (i.e. traversing a supernode) was a two-hop transmission i.e. between node 2 and the SN and between the SN and node 3.

6. DISCUSSIONS

A graph of data bytes sent against the elapsed time was plotted as shown in Figure 3. The graph shows that the routing protocol overhead increases with time at a node; hence, there will be a gradual drop in throughput at every node with time. Figure 4 shows delays in the transmissions. The data set used in the figures represents average values from three iterations of each test. Results showed that latency is higher when transferring data over two hops and would increase as number of hops increases. It appeared in some cases that the RTT (Round Trip Time) in the two-hop graph was less than the RTT in the one-hop. It would likely be as a result of interference of other Access Points (APs) within the range of the experiment. However, the delay caused by an additional hop is averagely 35ms.

The amount in bytes in the Y-axis for Figures 3 and 4 are absolute counts; the packet sizes were generated by hping3.
On one hand, Figure 4 shows that the routing protocol overhead increases with time a node. On the other hand, Figure 4 shows that latency is higher when transferring data over two hops and would increase as number of hops increases. It appeared in some cases that the RTTs (Round Trip Times) in the two-hop graph were less than the RTTs in the one hop. The reasons are either that the packets get delayed or are routed to another node before reaching the node they were intended for. It would likely be as a result of interference from other Access Points (APs) within the range of the experiment. The anomalies are possible since the network is susceptible to interference. The spikes (i.e. the abrupt drop in average RTT) can also be attributed to large packets that were sent in a single hop. The increase in the average RTT for large packets sent in two hops could however be acceptable as its reasons are the large file sizes and the number of hops in the transmission. On the other hand, the extremely large rise in RTT for two hops is likely a delay caused by a node scanning for best route. Considering both graphs and neglecting the sudden rise/fall in the RTT, the delay caused by an additional hop is averagely 35ms. An average change in RTT over change in packet size for each graph was calculated. The difference between the two values was determined, which gave the delay value.

7. CONCLUSIONS

Afrimesh simplifies the WMN monitoring and management processes by allowing critical changes to be made in a simple and efficient manner. Various experiments were carried out to provide additional performance metrics and information for the VillageTelco and Afrimesh projects. The experiments include determining the signalling overhead and latency in a network. While providing comprehensive reports on a network state, Afrimesh limits metrics so that users can take corrective actions on a node or network. Its metrics include network latency, radio signal, radio noise level and route link quality value. Lastly, it uses both SNMP and ICMP as the network management protocols thereby offering support for heterogeneous link layer technologies with small overhead for its complex functionalities.

8. REFERENCES

[1] Elizabeth M., Belding Ramya Raghavendra, Prashanth Acharya and Kevin C. Almeroth. "Meshmon: A Multi-tiered Framework for Wireless Mesh Network Monitoring". In Proceedings of the Tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09), pages 45-47, New Orleans, Louisiana, USA, 2009.

[2] Michael Adeyeye and Paul Gardner-Stephen. "The VillageTelco project: A Reliable and Practical Wireless Mesh Telephony Infrastructure". In EURASIP Journal of Wireless Communication Networks, 2011. doi: http://dx.doi.org/10.1186/1687-1499-2011-78.

[3] Hao-Chieh Chang Tein-Yaw Chung and Hsiao-Chih George Lee. "A Novel Cross-Layer Mesh Router Placement Scheme for Wireless Mesh Networks". In EURASIP Journal on Wireless Communications and Networking 2011, 2011. doi: 10.1186/1687-1499-2011-134.

[4] C. Sandvig. "An Initial Assessment of Co-operative Action in Wi- Networking". In Telecommunications Policy 28(7/8), pages 579-602, 2004.

[5] Waldir Moreira Billy Pinheiro, Vagner Nascimento and Antonio Abelem. "Abare: A Deployment and Management Framework for Wireless Mesh Network". In Proceedings of the IEEE Latin-American Conference on Communications, pages 1-6, Bogota, Columbia, 2010.

[6] V. Aseeja and R. Zheng. "Meshman: A Management Framework for Wireless Mesh Networks". In Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, pages 226- 33, 2009.

[7] Roberto Riggio, Nicola Scalabrino Daniele Miorandi and Imrich Chlamtac, "Janus: A Framework for Distributed Management of Wireless Mesh Networks." In Proceedings of the 3rd International ICST Conference on Testbeds and Research Infrastructures (TridentCom '07), pages 1-7, Lake Buena Vista, Florida, USA, 2007.

[8] E. M. Belding-Royer, K. Ramachandran and K. C.Almeroth. "Damon: A Distributed Architecture for Monitoring Multi-hop Mobile Networks". In Proceedings of the IEEE SECON, pages 601-609, Santa Clara, California, USA, 2004. doi: 10.1109/SAHCN.2004.1381963.

[9] Passos D. Albuquerque, C. De T. Valle, R. and D. Saade. "Mesh Topology Viewer (MTV):An SVG-based Interactive Mesh Network topology Visualization Tool". In Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2008), pages 292- 297, 2008. doi: 10.1109/ISCC.2008.4625602.