Multi-hop Wireless Network Emulation on StarBED

Lan Tien Nguyen[1], Razvan Beuran[2,1] and Yoichi Shinoda[1,2]

[1] Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1292 Japan
lannt@jaist.ac.jp,shinoda@jaist.ac.jp
[2] National Institute of Information and Communications Technology
2-12 Asahidai, Nomi, Ishikawa, 923-1211 Japan
razvan@nict.go.jp

Abstract. In this paper, we present an architecture to emulate multi-hop wireless networks on StarBED, a wired-network testbed at Hokuriku Research Center of NICT, Japan. The architecture uses a distributed approach, and it can effectively emulate in real time the properties of WLAN contention-based media access mechanism.

Key words: emulation, real-time, testbed, routing, wireless

1 Introduction

Network emulation is an experimental technique that intends to bridge the gap between simulation and real-world experiments, and thus, it delivers a significant impact on the wireless research community. There are two approaches for emulation, one is the centralized approach, and the other is the distributed approach.

In the centralized approach, all the nodes connect to a central server and direct their traffic to the server. Then, the sever forwards the traffic to the destination according to the parameters which characterize the current state of the emulated network, i.e. reachability, link quality, collision etc.

Opposed to the centralized approach, in the distributed approach, all the nodes are mutually connected via wired or wireless media. The nodes themselves are responsible for directing and forwarding traffic. Since all the nodes are mutually connected, network topology is created by using logical connectivities which are computed from geographical information, radio parameters, and medium information in a distributed fashion.

Due to the bottleneck that can occur at the central server in the centralized approach, only distributed emulators are able to support real-time evaluation of topology-related protocols. In the following sections, we will focus on the design and implementation of a distributed architecture, named AEROMAN (Architecture to Evaluate Routing PrOtocols for Multi-hop Ad-hoc Networks) which allows us to emulate multi-hop wireless networks on a wired network testbed, StarBED [2].

2 AEROMAN

2.1 How to emulate a wireless link

In order to emulate wireless links, we use Dummynet [3], a link emulation tool designed for FreeBSD. It works by intercepting packets on their way in the protocol stack, and passing them through its pipes, which simulate the effects of bandwidth limitation, propagation delay, and packet loss. These pipes can be either at the sending side or the receiving side.

In AEROMAN, pipes for unicast traffic are located at the sending side, while pipes for broadcast traffic are located at the receiving side. The reason for this is that dummynet classifies packets based on their IP addresses, and thus, in order to emulate multi-hop wireless networks, the incoming/outgoing link of a packet has to be determined from the addresses in its IP header. For unicast traffic, a sending node can easily find the link it will send a packet on by looking at routing table for the next-hop node. For broadcast traffic, it is impossible to locate pipes at sending side since links cannot be identified by using broadcast IP address. However, at the receiving side, a node can find out the source node of a broadcast packet by looking at the source IP address, and hence it knows the link through which the packet has passed. Being aware of the links used to forward packets, a node can direct a packet to the appropriate pipe configured with parameters (bandwidth, delay, packet loss rate) equivalent to parameters of the link through which the packet has traveled (broadcast) or will travel (unicast).

2.2 Design of AEROMAN

AEROMAN uses a two-stage approach to emulate multi-hop wireless networks. In the first stage, parameters of all wireless links are computed in contention-free conditions by using deltaQ library of QOMET [1]. This information is distributed to all the experimental nodes before the experiment. In the second stage, AEROMAN use the Adaptive Traffic (AT) model to adjust these parameters in a contention-aware fashion. Figure 1 shows AEROMAN node internals which includes six modules:

- *Pipes Controller*: Applying appropriate links parameters, which are generated by Adaptive Real-time Parameter Generator, to dummynet pipes.
- *Multicasting Module*: Exchanging traffic information between experimental nodes.
- *Local Node Real-time Traffic Collector*: Collecting traffic information of the current node.
- *Remote Nodes Real-time Traffic Collector*: Collecting traffic information of other experimental nodes.
- *Routing Support Module*: Identifying the link which will be used to forward a given IP packet that goes through the current node. Parameters of this link will be used for configuring the pipe which handles the packet.
- *Adaptive Real-time Parameters Generator*: Using Adaptive Traffic Model to adjust contention-free links parameters based on real-time traffic information of the wireless channel.

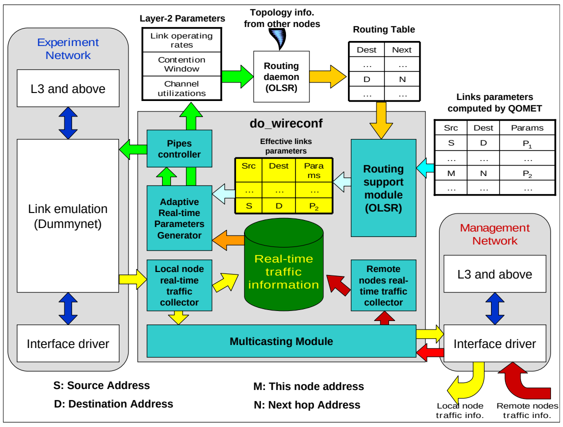

Fig. 1. AEROMAN Node Internals.

2.3 Adaptive Traffic Model

This model is used by the Adaptive Real-time Traffic Generator module to compute link parameters on the fly from both Effective Link Parameters and Real-time Traffic Information. Firstly, Frame Error Rate (FER) is recomputed based on current channel utilization. Secondly, both packet delay and bandwidth limitations are adjusted based on the value of FER computed in the previous step.

3 Conclusions

In this paper, we present the design and implementation of AEROMAN as an architecture for evaluating routing protocols for multi-hop wireless networks. AEROMAN follows a distributed approach, while still effectively emulating properties of the wireless environment, such as bandwidth limitation and shared media. Due to the lack of space, no experiment results are shown in this paper; however, such results will be displayed in the poster.

References

1. R. Beuran, L. T. Nguyen, K. T. Latt, J. Nakata, and Y. Shinoda. QOMET: A Versatile WLAN Emulator. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications*, pages 348–353, Niagara Falls, Ontario, Canada, 2007.
2. T. Miyachi, K. Chinen, and Y. Shinoda. StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software. In *Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools*, page 30, New York, NY, USA, 2006. ACM.
3. L. Rizzo. Dummynet FreeBSD Network Emulator.