

# A simulation design for link connection-oriented mesh networks

R. Luo, D. Belis, R. M. Edwards AMIEE, G. A. Manson

The University of Sheffield

**Abstract:** Forthcoming Packet Data Communications systems will require associated robust QoS. A Core of such systems is the MAC, In general, recent radio LAN (rLAN) implementations have used contention-enabled MACs that make use of statistical multiplexing algorithms for access control while regular channel division-based MACs based on TDM and FDM, are used less often. The latter types of access regimes have been shown to be less efficient at flexible and dynamic bandwidth distribution between nodes. However, both TDM and FDM are well suited to the QoS function that becomes topical if rLANs are to be used to extend 3/4G implementations. As a result, link connection-oriented based networks combined with mesh topologies are considered to be good research theme. In this paper we present a novel simulation design for link connection-oriented networks that will be germane to research in this area.

## 1 Introduction.

To aid simplicity from this point forward we use CENs as an abbreviation for contention-enabled networks and CBNs for connection based networks. CENs, having been successful in wired Local Area Networks (LANs), now dominate access control regimes used in Radio LANs (rLANs). In both pure ALOHA and S-ALOHA [3], packets that fail due to collision will affect overall throughput. Throughput can be increased using small control packets to reserve contention-free slots for data transmissions as in CSMA/CA, MACA [5], MACAW [6] and FAMA [7]. A second issue with a CEN is the difficult control in the network density since all nodes have the freedom to compete for a common medium. Conversely, nodes in CBNs make possible strict admission controls. Discussion of end-to-end QoS is also relevant here since we see that whilst CEN terminals will arrive on an ad-hoc basis each of which may degrade performance across the whole of the shared channel. In essence this last point promotes CBN mesh networks as being more able to allocate bandwidth. CBNs perform well in areas constituted of more than one segments because they use traditional routing algorithm. Each link of a mesh network is peer-to-peer mode and packets in the link are isolated from any interference from other links. Fewer contentions mean more successful deliveries, which is a prerequisite for wireless QoS.

Based on the previous discussion, we are interested in research related to the CBN mesh topology networks. Simulation software for mesh node networks does not yet exist or at least is not widely available. In this paper we introduce an example over a link connection-oriented mesh networks to discuss how to implement a simulation design. Our paper will be organised as follows. In section II we will explain the basic entities considered in network simulation. In section III we will design the implementation of simulation on mesh networks. In section IV we will discuss the performance of mesh networks. The paper concludes with section V where results will be presented for comparison between CENs and CBNs.

## 2. Entities in simulation model for mesh network

### 2.1 Queue in a mesh node

A queue causes the delay when packets are passing through. In a node with multiple interfaces each node may have several queues for different peers. We may use infinite queues in the discussion. But practical queues are of limited size. A drop-tail policy can be used to discard the excessive amounts of packets. We assume that the retransmission of the dropped packets is a task performed by the higher layer.

### 2.2 Node

A node is the basic entity and the centre-point of other entities. A node may contain queues, include the traffic generator/sink, connect links and understand routing tables. We show the structure of a node in Figure 1.

The outer parts of the node are links to other nodes. When traffic flows into a node via a link, the routing module checks the packets. If the node happens to be the destination, then packets dump into the sink module. Otherwise packets continue to the corresponding queue. Meanwhile the node itself probably generates new packets in traffic generator module. Various queues point to different outgoing links so the routing module must decide to which queue the incoming packet should be sent. The Selector module is used to distribute the queues according the policy defined in the division-based access control algorithms, e.g. TDM or back-off timers. A queue can pop the data out to the outgoing link only after has obtained the token from the selector.

### 2.3 Traffic generator/sink

The traffic generator and the sink determine the start and stop time of a packet. The traffic generator is often combined with a distribution, e.g. exponential Poisson process. Another entity, the sink, is the receiver that responds to the packet arrival and then dumps the data and marks the packet with a stop time. We can use the following sample code to generate traffic.

```
return ((rand() < p*RAND_MAX)); //p is packet generated probability
```

### 2.4 Link

The following conf code represents links of a mesh node, which here means that *node0* connects to *node1*, *node3* and *node2* in sequence. It is noted that the sequence is important since it determines the action in a slot, e.g. "link 0 1 3 2". Furthermore, links could be represented in a matrix *LinkMatrix[src][dst]* that contains information about: 1) connection condition of a node pair, 2) The quality of the link.

### 2.5 Routing

Dynamic link data structures holding the traversal path between a source node and a destination node are used to represent one of the *n* items in a routing table. We route packets in a shortest path using *Dijkstra* algorithm [1].

### 2.6 Packet

A packet is an entity that keeps a record of its own traversal path. It should contain the attributes: 1) Source node. 2) Destination node. 3) Start time at the sender. 4) Stop time at the receiver and 5) drop information.

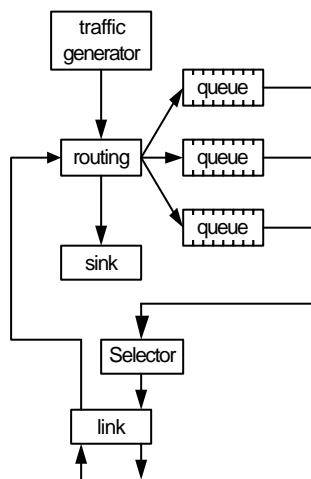


Figure 1 Architecture of a node in a mesh network

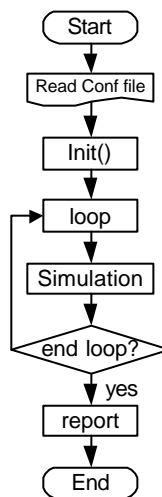


Figure 2 Flowchart in a simulation

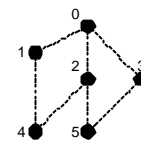


Figure 3 A simple mesh network

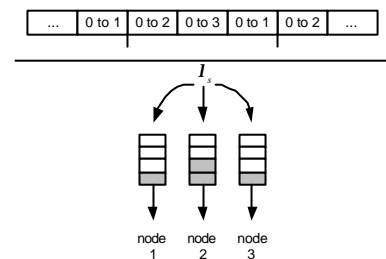


Figure 4 queuing in proposed model

## 3. Simulation implementation of mesh networks

Before we discuss the simulation implementation, we must define the structure style. Currently we are not developing a complete simulation package that provides full functionality to all types of networks, but only for CBN mesh topology. Thus, we decided not to use Object-oriented programming, although it might help to write a complete software package. As a result, the simple code brings advantages in the speed (faster) and resources (less consumption). The flowchart of the simulation design looks like this (Figure 2):

### 3.1 Configuration File "mesh.cfg"

Besides the main compiled routine, we put the configuration data in a separated file "mesh.cfg". Thus, we can conveniently adjust the data without recompiling the source code. When the simulation routine starts it automatically loads "mesh.cfg", reads the data and dynamically creates the corresponding data structures. Common variables exist in the configuration file like *NodeNumber*, *Link*, *Pgen*, *SimSlots* and *QueueLen*.

### 3.2 Init

The procedure *Init* is responsible for computing space allocation, data structure construction and origin routing table calculation. The number of nodes determines the computing space consumption. Because this variable is

retrieved from the configuration file, Init has to dynamically allocate the space for nodes. Before the actual simulation starts, Init should know whether the existing simulation host could satisfy the resource consumption or not. One of the tasks in Init procedure is to construct the data structure *LinkMatrix[src][dst]* that we mentioned before and defines the link connections and their quality. It should be noted that Init is not the only place that includes routing table calculation for if we want to see the impact of dynamic wireless routings algorithms such as AODV, the routing calculation module may exist inside the simulation body.

### 3.3 Loop

The loop controls the time elapsed during a simulation. For a simulation designer, two aspects are often hard to decide. The first is the number of slots in the simulation. In general, for convergence, more than one simulation with various values of simulation slots is needed. If the convergence is in our expected threshold, the value of loop slots is acceptable. The second aspect is the capacity of the simulation host. Considering the simulation time related to the host capacity, we couldn't increase the simulation period unlimitedly. Besides the loop that represents the simulation slots, it is possible to add an outer loop to see the results over some parameters such as link quality (BER, PER and etc), node transmission possibility, packet generating rate and number of nodes.

### 3.4 Simulation

Simulation includes the management of *traffic generators*, *sinks* and *queues*. Because our attention has been focused on the MAC layer, we have assumed that the packet length is equal to the time-unit of a slot. This enables simplification of the model by avoiding use of control protocols such as ARQ [1]. Our traffic generator has responsibility for packet attributes and distributions. The sink records the session time for a packet and releases the trace resource for that packet. Operations of the queues are important and include "In\_queue" (i.e. select a correct queue to push packet due to a search of the routing table, forward the packet to the sink if it arrives at the destination), "Out\_queue" and queue buffer operations (i.e. judge if a buffer is full, drop tail packets and map virtual queue marks to the actual positions of queues).

### 3.5 Report

Output is given by the report module that looks up the packet trace including start time, hops, stop time and discards. With the information it gives further results like throughput, delay and variance.

## 4. Simulation and Discussion

We realise that a link connection-oriented network should perform better than contention-enabled networks in terms of traffic variance, because a link connection-oriented can deploy regular division multiplexing MAC such as TDM that helps to decrease collisions. It is sometimes useful because one aspect of QoS requires fluctuation-less networks. In our sample we use TDD plus FDM to implement multiplexing for a wireless link. Figure 3 is a sample network although its mesh topology has not yet been optimised.

For simplicity, in the analysis we only consider the outgoing traffic from a node without considering slots for the incoming traffic (it can be achieved by a full-duplex transceiver). Ignoring the incoming slots is not practical but reasonable. Though it deliberately decreases the variance of delay quantitatively, it will not produce qualitative effort in our analysis. Thus, we use the queuing model in Figure 4. Three independent queues are used for three outgoing paths, respectively. When a packet arrives the node will decide to which path it should go through, and be stored in the corresponding queue. For long run traffic we also assume the incoming traffic per node is stable and the outgoing traffic selects equally each of the three paths. Hence each queue sees the arrival rate in  $I_s/3$ .

The  $i^{th}$  arrival needs to wait at least  $j$  slots to be transmitted ( $j = 1, 2, 3$ ). We denote  $R_i$  this residual time, then we get  $E(R_i) = 2$  and  $E(R_i^2) = 14/3$ . We denote  $W_i$  the wait for  $i^{th}$  arrival in queue, so

$$W_i \approx R_i + (I_s/3) \times W_i \times 3 \quad (1)$$

The standard deviation of delay is  $s_T = \sqrt{E(X^2) - E(X)^2} = \sqrt{E\{(W_i + 1)^2\} - (E\{W_i + 1\})^2}$ . Finally we have

$$s_T = \sqrt{2/3} / (1 - I_s) \quad (2)$$

In a  $k$ -node multi-hops CBN, we estimate the average hop count per packet delivery  $h \approx \log_2 k$  when each node has a degree of 3.  $I_s$  contains two parts, one of which is produced by the node itself as  $I_{s-node}$ , and the other is

the traffic beard by the node as an intermediate. In case the congestion, the arrival rate produced by node itself  $I_{s-node}$  should be limited to  $1/h$ . We assume  $I_{s-node}$  around  $1/\log_2 k$  of  $I_s$ , and then we get

$$s_T = \frac{\sqrt{2\log_2 k/3}}{1 - I_{s-node} \log_2 k} \quad (3)$$

The variance of delay of a mesh CBN appears to be fairly small (Figure 5). It is nearly persistent within the threshold of arrival rate. It is beneficial for the network to carry fluctuation-sensitive traffic, which is of interest in multimedia QoS. From Figure 5, we also conclude that the network performance is slightly changed with the number of nodes. Therefore it is possible to design a predetermined jitter buffer for various sizes of networks.

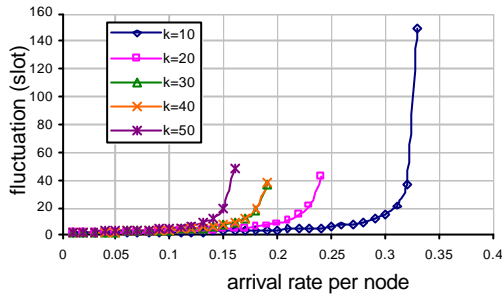


Figure 5 standard deviation of delay in proposed model

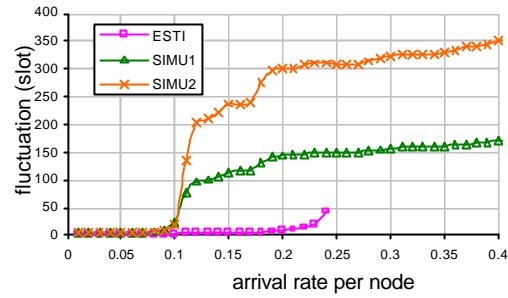


Figure 6 simulation results

We use our simulation model to see the standard deviation of the delay in a 20-node-CBN with a nearly balanced tree topology. Nodes have a maximum connectivity degree of 3. All the simulations take a period of 50000 slots, but two simulations are performed with different sizes of queue buffers at nodes respectively. Simulation 1 assigns to each queue the maximum size of 50 packets, where simulation 2 assigns 100 packets to each queue as the maximum size. Figure 6 shows the difference between analytical estimation and simulation results. Two reasons cause the gap. The first reason is the unbalanced traffic flows in the network during a simulation. Some nodes have to deal with more packets so that the delay produced by the queues brings up the fluctuation. It is reflected by faster increase from arrival rate per node at 0.1. The second reason is due to the drop-tail policy in finite queues. In analytical estimation we suppose each queue is infinite, which is hardly to implement in the simulation. In the simulations, if the arrivals go beyond the capacity of a queue, they will be discarded. That is why simulations show a weird slow increase when the traffic is high. But if we increase the buffer size in a queue, e.g. simulation2, a node will accommodate more packets in queues, which bring the curve up.

## 5. Conclusion and Future work

We have proposed a simulation design for mesh CBNs. Explaining the consideration on the entities, modules and simulation flows. It helps us to develop simulation models when the current simulation tools cannot meet the requirements of new research topics in a rapid development style. Additionally, we briefly introduce a simple analysis in the variance of mesh CBNs, and compare it with the simulation results. The discussion helps us to consider the topology layout and characteristics of queues.

The software code in this simulation design is related to a simulation software package development that is still undertaking, which focus on the research in wireless mesh CBNs. In the future, we will keep on working on the extension of the simulation design to accommodate multi-layers (e.g., physical channel features like distance, spectrum, fading, TCP/UDP, security), multi-protocols (e.g. NAT, QoS-related) and visualised RAD techniques.

## References.

- [1] Andrew S. Tanenbaum, "Computer Networks", third edition, Prentice-Hall, ISBN 0-13-394248-1
- [2] Dimitri Bertsekas, Robert Gallager, "Data networks" 2nd edition, Prentice Hall, 1992, ISBN 0-13-200916-1
- [3] N.Abramson, "The ALOHA system - another alternative for computer communications," Proc. Fall Joint Comput. Conf., AFIPS Conf., 1970, pp.281-285
- [4] L.Kleinrock and F.A. Tobagi, "Packet switching in radio channels: Part I – carrier sense multiple-access modes and their throughput - delay characteristics," IEEE Trans. Commun, Vol. COM-23, No. 12, Dec. 1975, pp. 1400-1416
- [5] P.Karn, "MACA – Anew Channel Access Method for Packet Radio," in ARRL/CRRL Amateur Radio 9<sup>th</sup> Computer Networking Conference, pp. 134-140, ARRL, 1990
- [6] V. Bharghavan, A. Demers, S. Shenker, and L.Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in SIGCOMM'94, pp.212 – 225, ACM, 1994
- [7] C.L. Fullmer, J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet radio networks," in SIGCOMM'95, pp.262-273, ACM, 1995