CHAPTER 3

EFFECTIVE ADMISSION CONTROL MECHANISM IN
WIRELESS MESH NETWORKS

Introduction

Measurement-based scheme, that constantly monitors the network, will incorporate the current network state in the decision making process. Centralized, measurement-based admission control scheme is effective in controlling traffic load in WMNs, while incurring very little communication overheads. So, this chapter discusses about the basic concepts, throughput maximization in WMN, Centralized Admission Control Method, Scale the admission control and QoS provisioning, Leaky Bucket Algorithm Based Transmission, measurement of Delay in Admission Control, Overheads during Transmission, performance Evaluation related with admission control in wireless mesh networks.

3.1 THROUGHPUT MAXIMIZATION IN WMN

To obtain the maximum throughput in WMN suitable routing is done. In Interference-Aware routing, blocking metric of a route is considered. To overcome this limitation and also to take the number of packets into account by defining the blocking metric of the node v to be the number of blocked nodes multiplied by the number of packets at the node v. Hence the modified blocking metric for a node v is: B(v) = (Number of nodes blocked by v) × (number of packets at v). Finally the path with the minimum blocking metric is selected as P₀ = arg min B(P). Problem of routing based on centralized scheme in which the Mobile Base Station (MBS) acts as a centralized scheduler for the entire network is considered.

Following constraints are used for transmission:

1) A node cannot send and receive simultaneously.
2) There may be only one transmitter in the neighborhood of a receiver.

3) There may be only one receiver in the neighborhood of a transmitter.

Given \( G = (V, E) \), where set \( V \) consists of MBS \( v_0 \) and \( v_1, v_2, \ldots, v_n \), such that \((v_i, v_j)\) belong to \( E \) if and only if \( v_i \) and \( v_j \) are within the transmission range of each other. The objective is to find a feasible routing tree and a schedule for the packets such that the number of timeslots required is minimized. In the proposed system model, the routing tree (scheduling tree) is constructed under two conditions. First, when a new node enters the network, the scheduling tree is updated according to broadcast messages MESH-NCFG (Mesh Network – Network Configuration) and MESH-NENT (Mesh Network Entry) from the new node. Then MBS recalculates the routing node and reconfigures the network by broadcasting the MESH-CSCH message to the Subscriber Station (SS). Second, the MBS also periodically recomputes the routing tree by considering new updated throughput requirements, and changing the routing tree if required.

3.2 CENTRALIZED ADMISSION CONTROL METHOD

Centralized schemes are apt for small size networks. A distributed approach may have the advantage of not having a single point of failure, but suffers from large messaging over-heads among the nodes. Note that several of such WMNs can collectively form a larger mesh network, where a hybrid approach to resource management is feasible i.e., the gateways will make centralized resource control decisions for the local clusters, while coordinating in a distributed manner with other gateways, for grained load-balancing.

3.3 SCALE THE ADMISSION CONTROL AND QOS PROVISIONING

The goal is to maintain tolerable end-to-end delays for each client under heavy load conditions [38, 39]. It has been noted that for most web applications, the ideal round trip delay should be less than 800 ms for good performance. RTT
(Round Trip Time) of more than 1000 ms will affect the performance of the application. Subsequently, decided to use 1000 ms as the maximum tolerable delay for each client that associates with the network. Again, it should be noted that we do not aim to provide hard-delay guarantees’ needed for real-time applications like VoIP, but only soft delay assurance. Let $D_{user}$ have the maximum tolerable delay ($= 1000$ ms), objective is to make a Yes/No admission control decision for that client and determine the best path to carry traffic from the client to the Internet gateway. Let $RTT_i$ be the round trip delay on hop $i$ of the network. Then the admission control decision will be $yes$ if the algorithm can find the shortest path $P$ from the source to the destination such that for all $I$ hop on path $P$:

$$\sum_i RTT_i + \beta < D_{user}$$  (1)

Where $\beta$ is a hysteresis parameter. If the above condition is not satisfied, it means that the network does not have enough resources to support the new client and the request will be rejected. $\beta$ is introduced to make our scheme more conservative and minimize the potential impact of the new flows on the delay of pre-existing flows in the network.

### 3.4 CONGESTION CONTROL

When one part of the subnet, e.g. one or more routers in an area, becomes overloaded, congestion results [40]. Because routers are receiving packets faster than they can forward them, one of two things must happen:

- The subnet must prevent additional packets from entering the congested region until those already present can be processed.
- The congested routers can discard queued packets to make room for those that are arriving.
Factors that Cause Congestion [41, 42]

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets
- Bussy traffic
- Slow processor
- Congestion Control is concerned with efficiently using a network at high load.

Techniques: Several techniques can be employed. These include:

- Warning bit
- Choke packets
- Load shedding
- Random early discard
- Traffic shaping

The first three deal with congestion detection and recovery. The last two deal with congestion avoidance.

Traffic Shaping

- Traffic shaping reduces congestion and thus helps the carrier live up to its guarantees
- Traffic shaping is about regulating the average rate (and bussy) of data transmission
- Traffic shaping controls the rate at which packets are sent (not just how many)
Used in ATM and Integrated Services networks.

- At connection set-up time, the sender and carrier negotiate a traffic pattern (shape).
- Two traffic shaping algorithms are:
  - Leaky Bucket
  - Token Bucket

**Leaky Bucket and Token Bucket:**

- LB discards packets; TB does not. TB discards tokens.
- With TB, a packet can only be transmitted if there are enough tokens to cover its length in bytes.
- LB sends packets at an average rate. TB allows for large bursts to be sent faster by speeding up the output.
- TB allows saving up tokens (permissions) to send large bursts. LB does not allow saving.

**Leaky Bucket Algorithm Based Transmission**

The **Leaky Bucket Algorithm** used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded. The leaky bucket enforces a constant output rate regardless of the burstiness of the input. Does nothing when input is idle. The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion. When packets are the same size, the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick.
There is a clear distinction between normal connection request and worm
connection request in the delay queue [43 - 45], that is to say, normal connection
requests show “Temporary emergency flow”, but frequent connection requests by
the worm will last for a period of time. Therefore, the delay queue of normal
connection request is shorter than the worm connection request, and the delay queue
with worm is blocked easily. For the normal connection, the two-stage leaky bucket
algorithm is used to output quickly from the delay queue, avoiding the delay in
normal connection. Network-flow-control technology based on the front buffer two-
stage leaky bucket algorithm is as shown below. The details are as follows: B is a
buffer; its capacity is $C_b$, Bucket1 and Bucket2 are leaky buckets; their capacities are
$C_k_1$ and $C_k_2$ respectively. S is token flow switch, K is a timer control by the timer
selector, the token speed $d$ decides the average transmission rate, and the size of the
$C_k_1$, $C_k_2$ decides the maximum sudden flow. Token bucket allows saving the token
in the spare time until it reaches maximum amount, token speed $d = 1$ is consistent
with the method of virus throttle dequeue speed to ensure that the delay of flow
connection requests.

Figure 3.1. A Leaky Bucket With Water and Packets
According to the quantity of tokens and the quantity of data packets in the delay queue, it decides the actual number to dequeue. The connection request to dequeue is the minimum between them. The schematic diagram is shown in Figure 3.2.

**Figure 3.2. The Schematic Diagram of Two Stage Leaky Bucket Algorithm**

**Working process:** When bucket1 and bucket2 are not empty, the packet queue enters into the network under the process of controlling K and Timer. Timer makes two packets go through K, the smallest interval less than \( \frac{1}{T} \), so the peak is under \( \frac{1}{\Delta T} \). The role of s is to control the token to infuse bucket1 or bucket2.

Two-stage leaky buckets are selected to guarantee that the normal connection requests dequeue quickly, and reduce the delay. For the worm connection requests, when the tokens in the bucket1 and bucket2 are used up, the speed of dequeuing is the same as that of the token, which is \( \frac{1}{s} \). After the emergent request within a period of time, the delay queue has lots of packets and the speed of producing packets is greater than that of producing token. If so, this port will be regarded as worm attacked port. At this moment, the speed of token generation for the delay queue will be set to 0, at the same time, shielding this port, giving up the packets.
3.5 MEASUREMENT OF DELAY IN ADMISSION CONTROL

(1) Time spent by the packet in the queue:

The queuing delay can be due to other packets ahead in the queue that are waiting to be transmitted. In addition, if the node senses the channel to be busy, the packet will remain in the queue for a longer time. The measurement captures this queuing delay that accounts for interference between and across flows. If one node is using the channel, then the other nodes would have to back off and the delay involved would be captured in our measurement.

(2) Transmission and propagation delay:

Once the packet is at the head of the transmit queue, it will be transmitted on to the channel and reach the destination. The corresponding delays involved in transmitting the packet and its propagation to the receiver are captured by our measurements.

(3) Retransmission delay:

If the packet reaches the receiver and is successfully decoded, an acknowledgement (ACK) is sent back and the receive time for the ACK is noted. However, if the packet is lost or corrupted, then the packet will be re-transmitted at the Medium Access Control (MAC) layer. The time involved in the successful delivery and decoding of the packet, including any re-transmissions, is captured in our measurements. Each node tracks per hop delay involved in transmitting a packet on a link and sends it periodically to the central controller. The central controller maintains a data base of the per-hop delays in the network and estimates the end-to-end delay for a given path in the mesh network.

3.6 OVERHEADS DURING TRANSMISSION

As a result, the measurement process itself does not introduce any overheads in the network. However, since our scheme is centralized, it involves periodic reporting of measurement data from the mesh nodes to the central controller. In addition, each node periodically sends out 'Keep Alive’ messages.
Also, when a client tries to associate with the network, client association request and reply messages are exchanged between the mesh Access Point (AP) and the central controller. All these messages contribute to communication overheads. It is important to make sure that the amount of overheads is kept at a minimum, the ratio of the control traffic to the data traffic. The measurements were carried out at four random nodes in the test bed. Clients were associated and disassociated at random. The amount of control traffic is very low as compared to the amount of data traffic in the network.

**Performance Evaluation**

In order to evaluate the effectiveness of our proposed scheme, we perform the following set of experiments:

Test the correctness of the delay measurement scheme. Figure 3.3 compares the delay measurements obtained from the driver against those obtained from a network monitoring tool. The network discovery protocol and the relaying of measurements data to the central controller involve certain messaging overheads.

![Figure 3.3 Delay Measurement](image)

Figure 3.3 Delay Measurement

Figure 3.4 shows the effectiveness of admission control scheme and evaluated in terms of improvement in throughput and delay for the clients.
The problem of provisioning QoS in wireless networks is not a trivial task, owing to their highly dynamic nature. A measurement-based scheme that constantly monitors the network, will incorporate the current network state in the decision making process. The key contribution of our work is that we evaluate our scheme experimentally using a ten node test-bed. Experiments show that a centralized, measurement-based admission control scheme is effective in controlling traffic load in WMNs, while incurring very little communication overheads. The measurement-based approach allows us to provide soft QoS guarantees, i.e. tolerable delay under heavy traffic load. Setting a more conservative delay threshold in the admission algorithm can help prevent performance degradation of admitted flows due to fluctuations in flow rates, channel quality and interference levels. However, it is not intended for providing hard delay bounds.