Infrastructure mode support for IEEE 802.11 implementation in NS-2

Ilango Purushothaman, Sumit Roy
{ilangop, sroy}@u.washington.edu
Department of EE, University of Washington,
Seattle, WA 98195-2500

The existing 802.11 implementation in ns-2.31 doesn't support infrastructure mode simulations. Beacon frames and Scanning/Association functions have not been implemented.

This report outlines the changes which were made to the existing ns-2.31 802.11 implementation.

Stage 1: Beacon Transmission and Reception

Beacon Transmission:

- A node can be configured as an AP by the following command.
  $mac_(ap_node) **ap** [$mac_(ap_node) id]
  Once an AP is set up, it will start transmitting beacons.

- A new timer, **BeaconTimer**, was added to facilitate periodic transmission of beacons (set for BeaconInterval). The BeaconTimer on expiry calls the beacon transmission function and sets the BeaconTimer again.

- The beacon packet is built in the **sendBEACON() function** and transmitted (depending on channel status). According to the IEEE standard, AP should send the beacon at every expiry of BeaconInterval. If the medium is found to be busy, AP should defer using basic CSMA deferral procedure.

- Currently, Beacon frames have the following fields.

    - Address and BSSID information.
    - Timestamp
    - Beacon Interval

- Multiple APs can now be setup and beacons can be successfully sent by all the APs and received.

Beacon Reception:

- On receiving the beacon, each STA's beacon reception function **recvBEACON()** is called, which basically stores the all the information in the beacon frame and also the received power levels from all APs.

Stage 2 – Association Request and Response Frames – Passive Scanning

- On receiving the beacon, each STA's **recvBEACON()** stores the received power level from each AP and the respective node ids, in a linked list. This list is updated every time a beacon is received, to account for mobility of the nodes.

- **Passive Scanning** – Each STA collects beacon information from all APs for a time which is the maximum of all beacon intervals. Once it has collected all the received power levels in the linked list, it picks out the "Strongest AP".

- Each STA sends out an Association Request frame to its Strongest AP, containing the specific BSSID. The function **sendASSOCREQ()** builds the Association request frame and the actual transmission function is **check_ pktASSOCREQ().**

- On receiving the Association request frame using **recvASSOCREQ()**, the AP sends out the Association response, containing the Association ID. The functions used are **sendASSOCREP() and check_pktASSOCREP()**. Basic CSMA rules are followed by the AP for sending the Association response.

- Right now, no decisions on Association, based on Supported Data Rates are made. Each AP builds its own list of associated nodes.

- On reception of Association response frames, using **recvASSOCREP(),** each STA sets its own "**associated**" flag and sends an acknowledgement to the AP. It can now transfer data packets in the BSS.

- Clients of the same BSS/different BSSs can exchange data, and the APs of the BSSs perform forwarding. Even if a connection (UDP, TCP etc) is setup between two clients of the same BSS, data will be forwarded through the APs only. Inter-AP communication has been implemented using the four different address fields and ToDs and FromDs bits (using the same channel used by the clients).

- Passive Scanning can be turned off by the following command
  $mac[$i] scanning OFF for all STAs.
  STAs will just receive beacons and will not initiate any association procedure.

Caveats:

- Inter-AP communication, using a separate distribution medium, has not been implemented yet. Inter-AP communication, for now, is achieved by using the same wireless channel used by the clients, but special address filtering makes sure that only APs can receive these "distribution frames". Typical scenarios that would work with the current implementation are: Clients can exchange data within the same BSS/ different BSSs. Client – Client communication, where the clients belong to different BSSs, works as: Client1 –> AP1, AP1 -> AP2, AP2 -> Client2.

- Authentication frames have not been implemented yet.

- Multiple BSSs can be set up in a single channel. Multiple BSSs can also be setup on different channels (using different channel objects) but communication across different channels is not possible for now. Due to lack of multiple interface support, an STA cannot scan all the channels used in the ESS. Hence, for now, passive scanning is limited to one channel only.
  An example script **"infra.tcl",** of two BSSs operating in a single channel is provided in **/tcl/ex/80211**. This sets up a single AP in a single channel and each STA uses the passive scanning procedure mentioned above to associate with the AP. This script provides an example of Client-Client communication, where the clients belong to the same BSS.

  Another example script provided is **"multiple_ap.tcl**" where three APs are set up in a single channel. This is an example of Client-Client communication, where the clients belong to different BSSs. Inter-AP forwarding of packets can be observed here.

  To see the beacons frames, please look for "BCN" packet type in the generated trace file. Association Request and Response frames are marked by "ACRQ" and "ACRP" respectively.
  Client-Client data exchange, facilitated by the APs forwarding, can also be observed in the trace file.

Stage 3:

- Inter-AP communication problem needs to be addressed more thoroughly. A separate wireless or wired distribution system needs to be implemented.
- Authentication will be implemented.
- Active Scanning needs to be implemented – Probe Request and Response frames.