Introduction to Network Simulator

Mouhamad IBRAHIM and Giovanni NEGLIA
mibrahim@sophia.inria.fr, gneglia@sophia.inria.fr

www-sop.inria.fr/maestro/personnel/Giovanni.Neglia/ns_course/ns_course.htm

Maestro team

INRIA Sophia-Antipolis - France

IEEE 802.11 overview

- Regardless of medium type, MAC protocols regulate access to the shared medium by defining rules that allow the nodes to communicate in an orderly manner.
- Similarly to wired LAN MAC protocols, there are different MAC protocols that have been proposed for wireless LANs: ALOHA, CSMA, IEEE 802.11, etc.
- Since its standardization in 1997, IEEE 802.11 became the de facto standard in wireless LANs

IEEE 802.11 overview (cont'd)

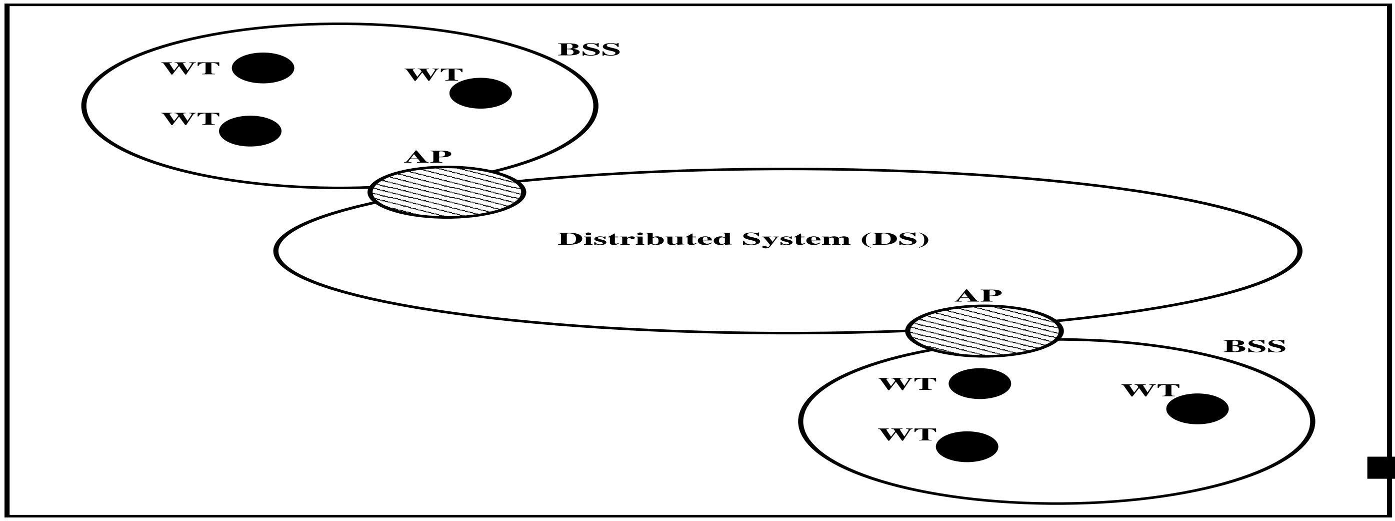
- 802.11 can operate in two modes:
  - Infrastructure or centralized mode: Centralized wireless networks have `base stations (BSs)`, also called `access points (APs)`, which act as interfaces between the wired and the wireless parts of the network. Communications between two nodes need to go through BSs. The centralized nature of these networks make them able to support QoS. However, they are less robust and more complex to deploy than ad-hoc networks.
  - Ad-hoc or distributed mode: Distributed wireless networks, also called ad-hoc wireless networks, have no centralized coordinators. There are no BSs and nodes `can communicate directly among them`.

IEEE 802.11 overview (cont'd)

- 802.11 defines two techniques to access the channel, which are related to the mode of operation of the network.
  - `Point Coordination Function (PCF)`: Centralized access to the channel. It can be used only in Infrastructure mode. It is an optional technique defined to support delay sensitive transmissions and can be used in combination with DCF.
  - `Distributed Coordination Function (DCF)`: Primary access technique in 802.11, where nodes access the channel in a distributed way. It is used in Infrastructure mode as well as ad-hoc mode. DCF is based on the CSMA/CA with slotted time scale. DCF supports delay insensitive data transmissions (e.g. email, FTP).

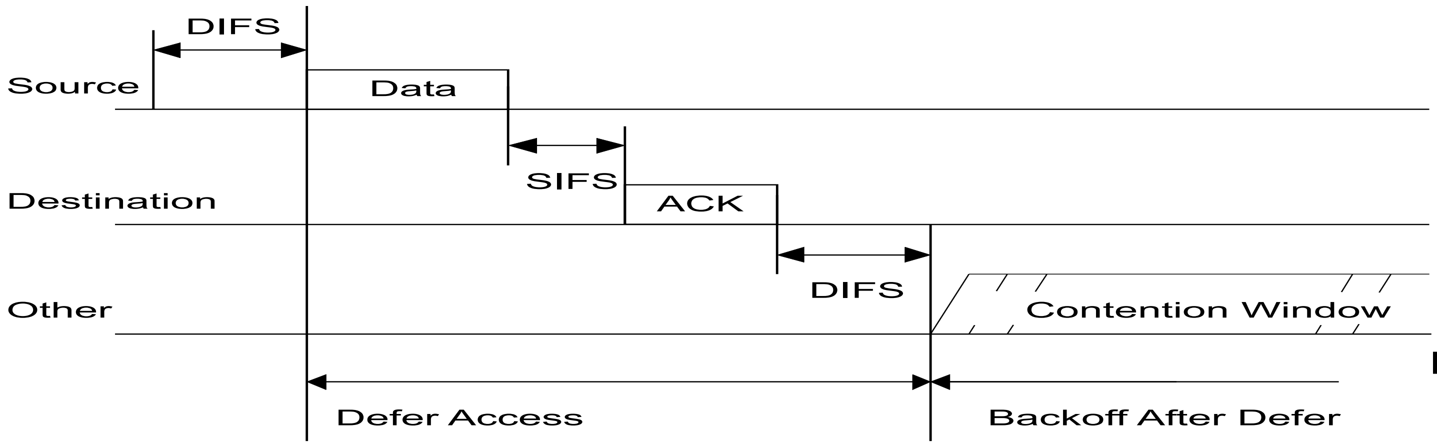
Snapshot of 802.11 network

- A group of `fixed or mobile` Wireless Terminals (WTs) forms a Basic Service Set (BSS). A BSS can operate either in an ad-hoc or in an infrastructure mode.Access Point links the WTs to a Distribution System (DS), and hence to other BSSs. DS can be any kind of wired or wireless LAN.
- In an BSS, when two fixed WTs are out of range of each other, multi-hop paths may be formed and intermediate WTs would route frames from source to destination.



802.11 DCF mechanism

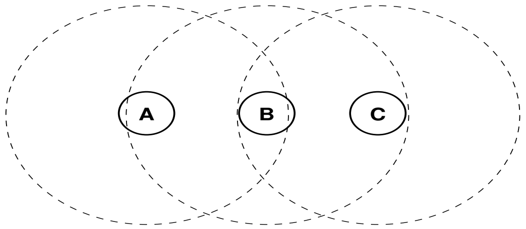
- DCF is based on a `Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism with a slotted time scale`.
- As in CSMA, a node listens to the channel before starting a new transmission for a period of time `DIFS`.
- As opposed to wired transmissions, when a node transmits its frame, it should wait for an ACK from the receiving end. After an `ACK timeout`, the sender assumes that the transmission failed and retransmits the frame again.



802.11 DCF mechanism (cont'd)

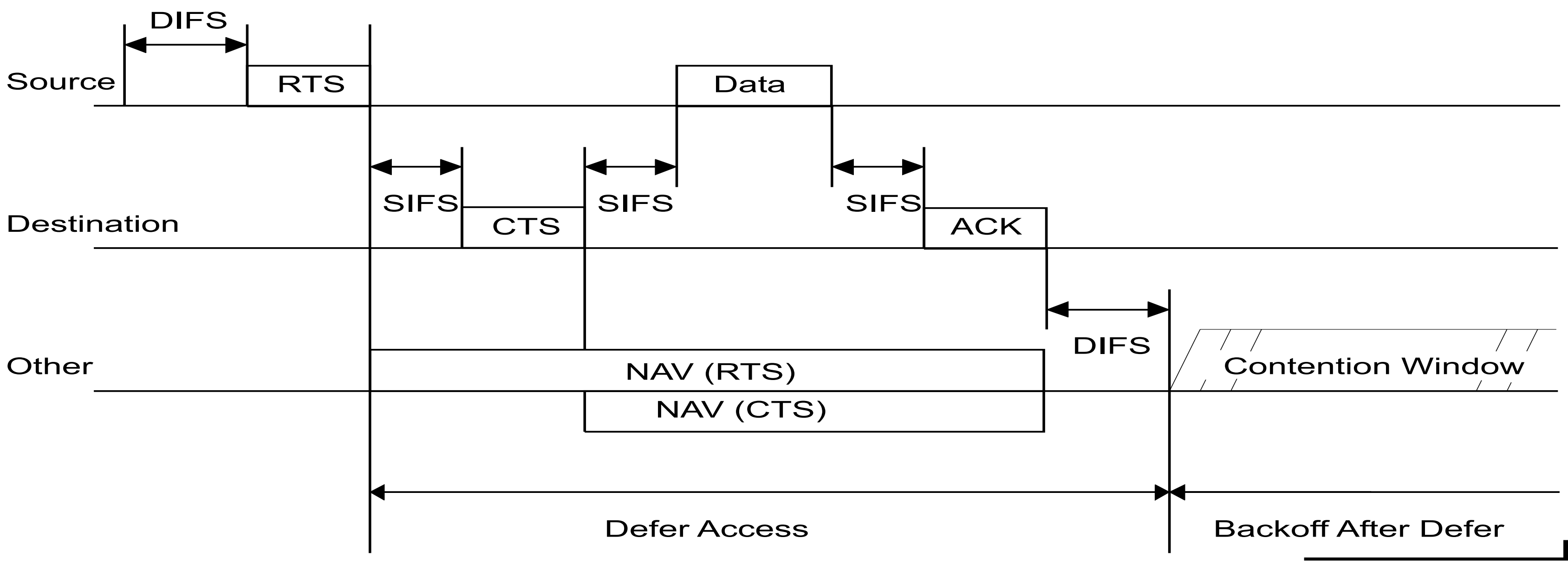
- Collision avoidance part is based on exchanging control frames to deal with the hidden node problem.
- A hidden node is one that is within the range of the receiver, but out of range of the sender:

  `Node A is transmitting to node B. When node C has a packet to transmit to node B, it will sense an idle channel as it is out of range of A. Therefore it starts its transmission which causes collision at B which is in the range of A and C. Similarly, node A is hidden to node C.`



802.11 DCF mechanism (cont'd)

- To avoid hidden nodes, a node with a packet to send transmits a short `Request to Send (RTS)` frame and waits for the corresponding `Clear to Send (CTS)` frame from the destination. All stations within the ranges of the sender and the receiver defer their transmission upon hearing the RTS and CTS respectively.

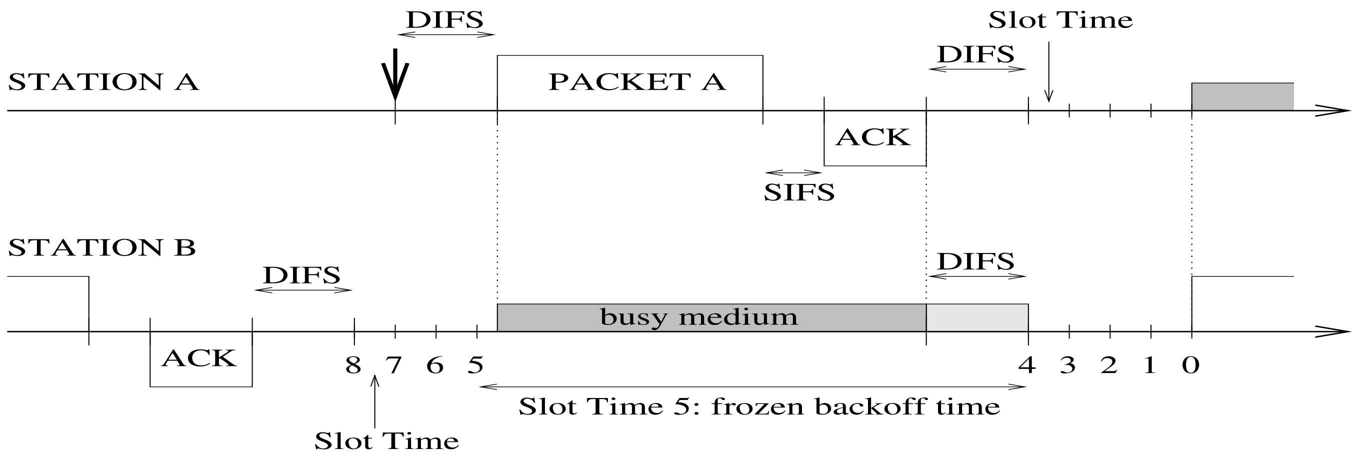


802.11 DCF mechanism (cont'd)

- To increase more the efficiency, the DCF employs a `binary exponential backoff algorithm`.
- A node with a data or RTS frame ready to be transmitted waits the channel to become idle for a DIFS time, then waits for an additional random time, called backoff time, after which it transmits its frame.
- The backoff time is an integer number of time slots between $[0, CW_i]$. CW is called `contention window`.
- At the first trial, $CW = CW_{min} = 32$, and at the last stage $CW = CW_{max} = 1024 = 2^m * CW_{min}$. (values depend on the PHY layer type).
- m is the number of stages.

802.11 DCF mechanism (cont'd)

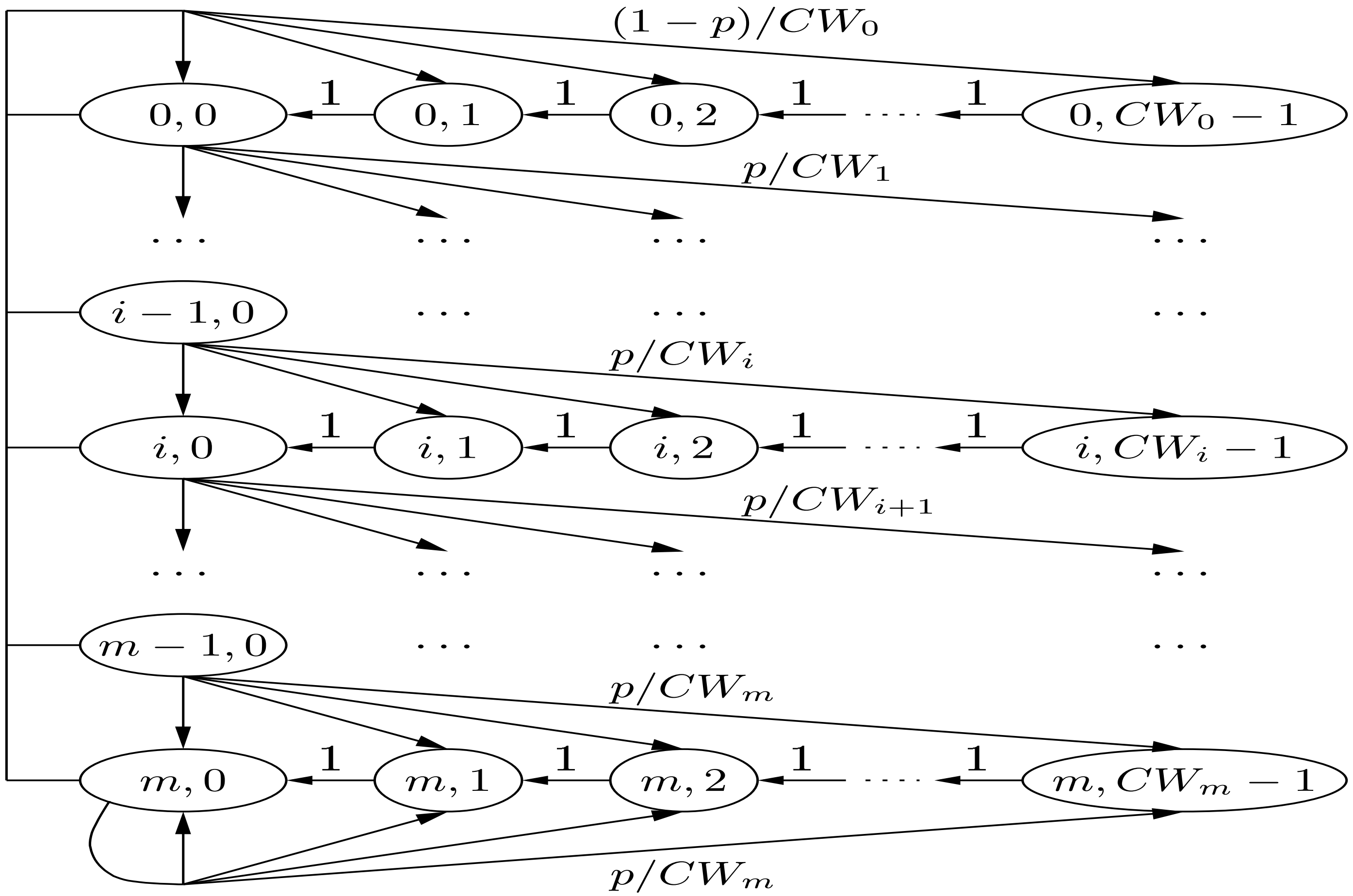
- The backoff is decreased as long as the channel is sensed idle, and frozen when the channel is sensed busy [a].
- When backoff time reaches zero, the node transmits its frame, then waits for the ACK.
  - If the packet is successfully transmitted (node receives an ACK), the node resets its CW to CW_{min}, and proceeds with next packet.
  - If the ACK times out, the node computes a new random backoff time with a higher CW: $CW_{i+1} = 2 * CW_i$



[a]Figure taken from "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", by Giuseppe Bianchi

Markovian Model for DCF

- *Model taken from the paper:* `Performance Analysis of the IEEE 802.11 Distributed Coordination Function`, *by* `Giuseppe Bianchi`



Metrics of interest in wireless networks

- Throughput: reflects the efficiency of using channel resources. It is the fraction of the channel capacity used to transmit usefull data. Packet overhead, collisions and/or retransmissions decrease the throughput.
- Delay: is the amount of time spent by a packet to successfully reach its destination, taking into account queuing delays, retransmission delays etc.
- Fairness: it is the capability to distribute the available resources equally among the nodes. When we intend to support QoS, fairness is hence defined as the capability to distribute bandwidth in proportion to their intended allocation.
- Power consumption: Power consumption is critical since it is decreases battery life of the wireless nodes. Collision and subsequent retransmissions are the main factors influencing the energy consumption of wireless nodes.

Wireless networking in NS

- NS allows the simulation of fixed and/or mobile wireless LANs, multihop ad-hoc networks, combined wired and wireless networks.
- NS implements IEEE 802.11 mac protocol and different mobile routing protocols.
- Currently, NS implements four ad-hoc routing protocols which are Destination Sequence Distance Vector (DSDV), Dynamic Source Routing (DSR), Temporally ordered Routing Algorithm (TORA) and Adhoc On-demand Distance Vector (AODV).
- Also, NS supports node movement and traffic trace files for mobile wireless networks.

Wireless networking in NS

- Files related to this part are defined in `.../ns/mobile/,.../ns/mac/mac-802_11.cc,h, .../ns/tcl/mobility/dsr.tcl,tora.tcl,dsdv.tcl, .../ns/mac/wireless-phy.cc,h`
- Default values can be found and modified at `.../ns/tcl/lib/ns-default.tcl`
- Example scripts can be found in `.../ns/tcl/ex/`
- **Rule of thumb:** `Read carefully the documentation and the code of the protocol to be simulated to understand how it is defined and the values assigned to its different parameters.`

Configuring a wireless network

- To create a mobile node capable of wireless communication, one needs basically to change the default configuration parameters before creating the node with the classical command `$ns_ node`
- The configuration consist of defining the network components for mobile nodes, defining the type of antenna, the radio propagation model, turning on or off the trace options at Agent/Router/MAC levels, selecting the type of ad-hoc routing protocol for wireless nodes or defining the energy model.
- The default values for most of the available options are Null, which basically defines configuration of a simple node.
- Options that need to be changed may only be called.

Configuring a wireless network (cont'd

- To modify the configuration of a parameter, the command is `node-config` and it is used as follows:
  ```
  $ns_ node-config -adhocRouting dsdv
  ```
- Note that all node instances created after a `node-config` command will have the same property unless a part or all of the `node-config` command is executed with different parameter values.
  For instance, to turn on wired routing for a base-station node after configuring AODV for mobile nodes:
  ```
  $ns_ node-config -wiredRouting ON
  ```

Configuring a wireless network (cont'd

- Note that before configuring and creating mobile nodes, we need to define the topography for mobile nodes.
- Normally flat topology is created by specifying the length and width of the topography using the following primitive:

```
set topo [new Topography]
$topo load_flatgrid <X> <Y>
```

  This initializes the grid for the topography object. `<X>` and `<Y>` are the x-y coordinates for the topology and are used for sizing the grid.

Configuring a wireless network (cont’d

- Also, a `General Operations Director (god )` needs to be created. `god` is an object that is used to store the total number of mobile nodes and a table of shortest number of hops required to reach from one node to another.`god` accelerates the computation of the network routing/connectivity prior to the simulation run. The command is as follows:

  ```
  create-god num_nodes
  ```

  where the number of mobile nodes is passed as argument.

Configuring a wireless network (cont'd

- The following is a list of options needed to configure a mobile node:

```
$ns_ node-config -addressingType <usually flat or hierarchical
used for wireless topologies> \
              -adhocRouting <adhoc rotuing protocol like DSDV,
DSR, TORA, AODV, DumbAgent > \
              -llType <LinkLayer LL> \
              -macType <MAC type like Mac/802_11> \
              -ifqType <interface queue type like
Queue/DropTail/PriQueue> \
              -ifqLen <interface queue length like 50> \
              -antType <antenna type like Antenna/OmniAntenna>
\
              -propType <Propagation model like
Propagation/TwoRayGround or Propagation/FreeSpace> \
              -phyType <network inteface type like
Phy/WirelessPhy> \
              -channelType <Channel type like
Channel/WirelessChannel> \
```

Configuring a wireless network (cont'd

-
  ```
              -topoInstance <the topography instance> \
              -wiredRouting <turning wired routing ON or OFF> \
              -mobileIP <setting the flag for mobileIP ON or
  OFF> \
              -agentTrace <tracing at agent level turned ON or
  OFF> \
              -routerTrace <tracing at router level turned ON
  or OFF> \
              -macTrace <tracing at mac level turned ON or OFF>
  \
              -movementTrace <mobilenode movement logging
  turned ON or OFF>
  ```

Configuring a wireless network (cont'd

- Note that the config command can be broken down into separate lines like

```
$ns_ node-config -addressingType hier
$ns_ node-config -macTrace ON
```

- The option `-reset` can be used to reset all `node-config` parameters to their default values. If we want for instance to create a simple node later, we need to use:

```
$ns_ node-config -reset
```

- Next, we can create mobile nodes as usual:

```
for {set j 0} {$j < $nodesNbr} {incr j}{
    set node_($j) [$ns_ node]
    $node_($i) random-motion 0 ;# disable random motion
}
```

Configuring a wireless network (cont’d

- We can disable the RTS/CTS exchanges within the simulations. There are some functions in .../ns/mac/mac-802_11.cc which compare the size of the packet with a `RTSThreshold` variable. If the size of the packet is larger than the `RTSThreshold`, then perform the RTS/CTS exchange. A common way to disable RTS/CTS exchange is to set the value of `RTSThreshold` to some higher value:

```
Mac/802_11 set RTSThreshold_ 3000 # @ the beginning
of your Tcl script
```

Configuring a wireless network (cont'd

- Setting the communication range of a wireless node in NS is a little bite cumbersome. A wireless node does not have a transmission range, and it is not something we can set by a simple command.
- It depends on the propagation model, the power with which the signal is transmitted and other parameters like the sensitivity of the receiver, the antenna gain, etc. However, we can set the values of these parameters, either at the beginning of the script of in `ns-default.tcl`, by using a separate C program that is provided by NS at `...ns/indep-utils/propagation/threshold.cc`.

Configuring a wireless network (cont'd

- This tool computes these parameters corresponding to a given propagation model and a desired communication range. To run it, we need to compile it separately and to run the executable. Then, we can compute these parameters as follows:
  `./threshold -m <propagation-model>`
  `[other-options] distance`
  where `<propagation-model>` is either `FreeSpace`, T`TwoRayGround` or `Shadowing`, and the `distance` is the communication range in meter.
- The variable `Pt_` is the power with which the signal is transmitted. `RXThresh_` is the sensitivity of the receiver. A better sensitivity means more transmission range. These variables are defined in `ns/mac/wireless-phy.cc`

Simulating a wireless scenario

- There are six identical fixed wireless nodes uniformly distributed within a square area of $100 \times 100\ m^2$. The propagation model is FreeSpace and the queue length of each node is assumed set to 100. Using the default physical parameter values of the model, transmission range of each node is in the order of $250\ m$. There are three sources nodes working in saturation conditions which run `CBR` applications, and three receiving nodes. `CBR` sources are identical and send 500-$Bytes$ packets each $5\ ms$. The first source starts at $t_1 = 20s$, the second at $t_2 = 40s$, and the third at $t_3 = 60s$. All the sources stop at $t_4 = 100s$.
  `Plot on the same graph the maximal throughput in` $bits/s$ `of each source versus the simulation time in` $seconds$`.` `Conclude.`
  `Hint:` Trace all the events and turn `ON agentTrace` and let `OFF` the other object traces. Disable RTS/CTS to obtain the maximal throughput.