COMPSCI 742 S2 C   Assignment 3

Department of Computer Science

The University of Auckland

Due Tuesday 11 October 05, 11:59 pm

This assignment will contribute 1/3 of your coursework mark,
and 10% to your overall course mark.

Submit your assignment via the DropBox, either in PDF (preferred), or in MS Word format.
Assignments in other formats will *not* be accepted or marked!

All the references cited in this assignment – shown in sans serif font – are available from the 742 *resources* page. The aodv-2nodes.tcl script can be downloaded from the 742 *assignments* page.

Wireless Networking

1. **WLANs: Communication between Access Points** [8 marks]

   Consider an 802.11 wireless LAN (WLAN). A Station (STA), e.g. a laptop with an 802.11 wireless card, can only communicate with other stations via an Access Point (AP). APs are connected to an infrastructure LAN, allowing them to communicate among themselves and with the wider world via that LAN.

   From time to time an STA may move from one AP to another; doing that requires a *handoff* from the 'old' AP (where the STA was communicating earlier) to a 'new' AP (where the STA is communicating 'now.'

   (a) A description of 802.11 and its handoffs is given in draft-ietf-mipshop-80211fh-04.txt. Read this Internet Draft so that you're sure of what is involved in an 802.11 handoff.

   When does a station perform a scan to see what APs are available?
   Why do APs need to communicate with each other during a handoff? [2 marks]

   (b) Inter Access Point Protocol (IAPP) is emerging as a standard method for APs to communicate. http://web.it.kth.se/ ren/iapp3/iapp3-1.html is a good introduction to IAPP.

   Briefly (two or three paragraphs) explain what IAPP is and why it is needed.
   What do IAPP's *Announce* and *Handover* protocols actually do? [3 marks]

   (c) In An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process, Mishra, Shin and Arbaugh consider three delays that occur during an 802.11 handoff.
   What are those three delays? Comment on how much they are due to propagation of wireless messages, and/or to other (external) causes. [3 marks]

CONTINUED

2. **Simulating an ad hoc Network** [5 marks]

   (a) Run the aodv-2nodes.tcl script (without making any changes to it). Observe what happens, by looking at nam and noting the approximate times when cbr packets start and stop being sent.

   Interpret the .tr file from time 0.0 until the first cbr packet is received by node 1. In other words, write a brief commentary on what happens during the simulated time. You should give the times (as reported in the aodv-2nodes.tr file) when cbr transfer starts and stops.

   Hint: *you may find it easier to see what's going on if you turn off some of the tracing.* [5 marks]

3. **AODV: ad hoc Wireless Routing** [7 marks]

   Routing Protocols for Ad Hoc Mobile Wireless Networks by Padmini Misra gives a good overview of routing protocols designed for wireless ad hoc networks. ns has agents for four of these, but we look only at Ad hoc On-demand Distance Vector (AODV) routing. *(I couldn't get any of the others to work reliably in ns)!*

   (a) Compare AODV routing with Bellman-Ford routing (which you studied in 314 as an example of *distance vector* routing). Your answer should emphasise differences between the wireless (AODV) and fixed (Bellman-Ford) environments. [2 marks]

   (b) Read the brief description of how to simulate wireless nodes with ns at the end of this note. Modify the aodv-2nodes.tcl script to have four nodes, as follows:

   - Node 0 starts at (10,450), node 1 at (150,250), node 2 at (350,250) and node 3 at (490,50)
   - At time 0.1s, node 0 starts moving at 200 m/s towards (490,450)
   - The CBR agent starts sending from node 0 to node 3 at 0.2s
   - At time 0.4s, node 3 starts moving at 200 m/s towards (10,50)
   - At time 3.01s the simulations stops (i.e. don't change the stop time)

   Write a brief commentary (as you did for question (2) above) on what happens during the simulated time. You should give the times when cbr transfer starts and stops, but finding them using nam is sufficient here (you don't need to analyse the trace file). Does your script demonstrate that AODV worked properly? What behaviour of your simulation would support your conclusion? [5 marks]

Simulating wireless networks with ns

For a detailed description, see the *NS Manual*, section 16 (you can find it easily from the 742 *Resources* page). Meanwhile, here's a brief overview . . .

The simulator uses several separate objects in a node, one for each layer. This assignment's example script, aodv-2nodes.tcl, uses a CBR agent to send a stream of UDP packets at a constant rate between two nodes, i.e. from node 0 to node 1 as follows:

- Packets originate node 0's CBR agent (AGT)
- They may prompt node 0's AODV router to send packets
- After that, the router (AODV) passes the cbr (UDP) packet to the MAC layer
- The MAC layer sends the packet (after sending an RTS message
  and getting a CTS reply) to node 1
- Node 1's MAC layer receives the packet, and passes it to node 1's UDP agent (AGT)

aodv-2nodes.tcl starts by setting values that configure the wireless nodes; those values are used in the $ns node-config statement. The $ns node-config statement also specifies which of the agents (layers) in each node you want to trace.

Trace formats are given in section 16.1.6 of the ns manual. As an example, here is a trace line in which node 0's CBR agent has sent a UDP packet to its link/router layer. (Note that ns uses the node numbers as IP addresses).

```
s 0.400000000 _0_ AGT  --- 0 cbr 210 [0 0 0 0] ------- [0:0 1:0 32 0] [0] 0 0

s/r/D/f                 |             |                 |             |
      time              |             |                 |             |
         nodeid         |             |                 |             |
           object type  |             |                 |             |
                      why             |                 |             |
                       eventid        |                 |             |
                          packet type |                 |             |
                            size of cmn header          |             |
      [dest ether_addr source ether_addr ether_type]    |             |
                           [source ip:port dest ip:port TTL next_hop]  |
                                              [cbr_seq] nbr_fwds opt_nbr_fwds
```

The space the mobile nodes move about in is the *grid*, it's dimensions are specified by the set val(x) and set val(y) statements, to 500 $\times$ 500.

The topography is set up and managed by the following three statements:

```
set topo    [new Topography]
$topo load_flatgrid $val(x) $ val(y)
create-god $ val(nn)
```

('god' here is ns's Global Operations Director).

CONTINUED

Initial positions (when simulation starts) are set by the $node(n) set X_, Y_ and Z_ statements.
The Z_ co-ordinate is not used by ns, set it to zero.
Random motion of nodes is disabled by the $node_(n) random-motion 0 statements.

Once the simulation is running, you make the nodes move using the setdest statement, e.g.

$ns at 1.0 "$node_(0) setdest 330.0 330.0 400.0"

setdest's parameters are: destination position, velocity
In this example, at simulation time 1.0 (seconds), node 0 starts moving at 400 units per second. It continues to do that until it reaches its destination, (330,330). *I'm not sure what the units are here, we'll assume they are metres (m).*

aodv-2nodes.tcl has only two nodes, they move as follows:

- Node 0 starts at (10,10), node 2 at (30,30)
- Both move along a diagonal at 400 m/s
  - Node 1 starts at 0.1s, stops at (350,350), and starts CBR sending (to node 1) at 0.4s
  - Node 0 starts at 1.0s, stops at (330,330)

After 3 seconds the simulation ends, and nam is called to display the simulated network behaviour.
The spreading circles on the nam display show you when packets are sent out from a node.