EEC 687/787 Mobile Computing (Spring, 2007)

Ns-2 Laboratory #1

Chansu Yu

Cleveland State University

Contents

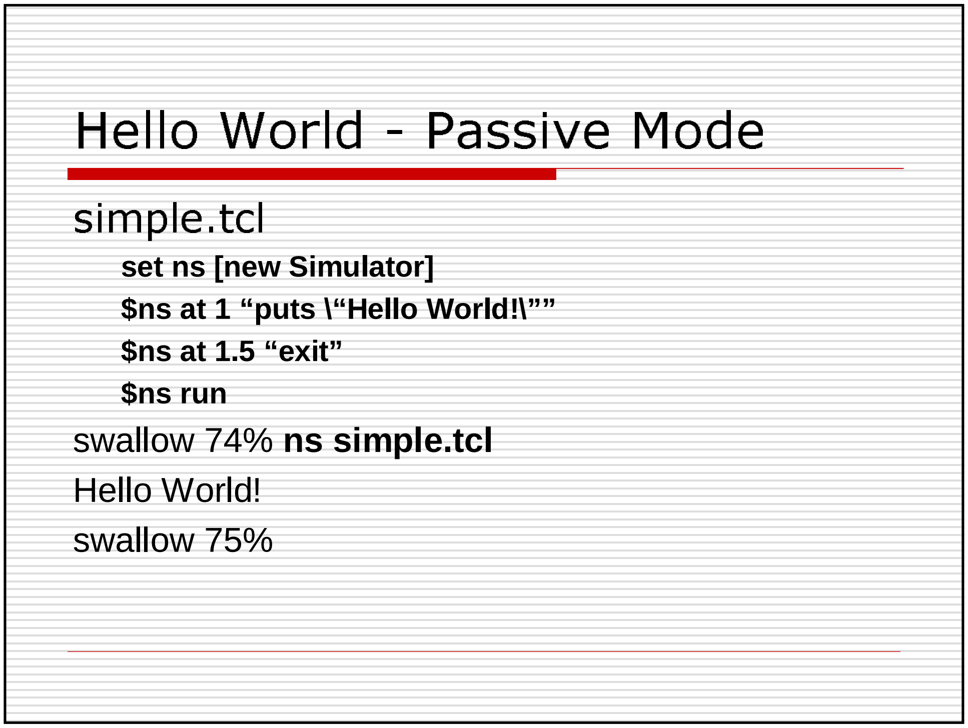
- □ Download & Install
- □ Basic TCL
- □ Introduction to ns-2
  - ■ ns-2 overview
  - ■ What can ns-2 do?
  - ■ ns-2 structure
  - ■ How to use ns-2?
- □ Simulation in MANETs
  - ■ What is MANETs?
  - ■ Simulation scenario configuration
  - ■ Simulation results analysis

Download and Install

- □ Reading
  - ■ Wireless and Mobility Extensions to ns-2 (http://www.isi.edu/nsnam/ns/tutorial/nsindex.html).
  - ■ Ns2 manual, "Mobile networking in ns," Ch. 16
- □ Download the latest ns-2 (version 2.30) with additional components source from http://www.isi.edu/nsnam/dist/ns-allinone-2.30.tar.gz and put it to your desirable folder, say /home/student.
- □ Run the following commands at /home/student :
  - ■ % gunzip ns-allinone-2.30.tar.gz
  - ■ % tar –xvf ns-allinone-2.30.tar
- □ Run the script at /home/student/ns-allinone-2.30 : "% ./install"
  - ■ This installation script will check your Linux environment, compile and install your ns-2 system.

Hello World - Interactive Mode

```
swallow 71% ns
% set ns [new Simulator]
_o3
% $ns at 1 "puts \"Hello World!\""
1
% $ns at 1.5 "exit"
2
% $ns run
Hello World!
swallow 72%
```

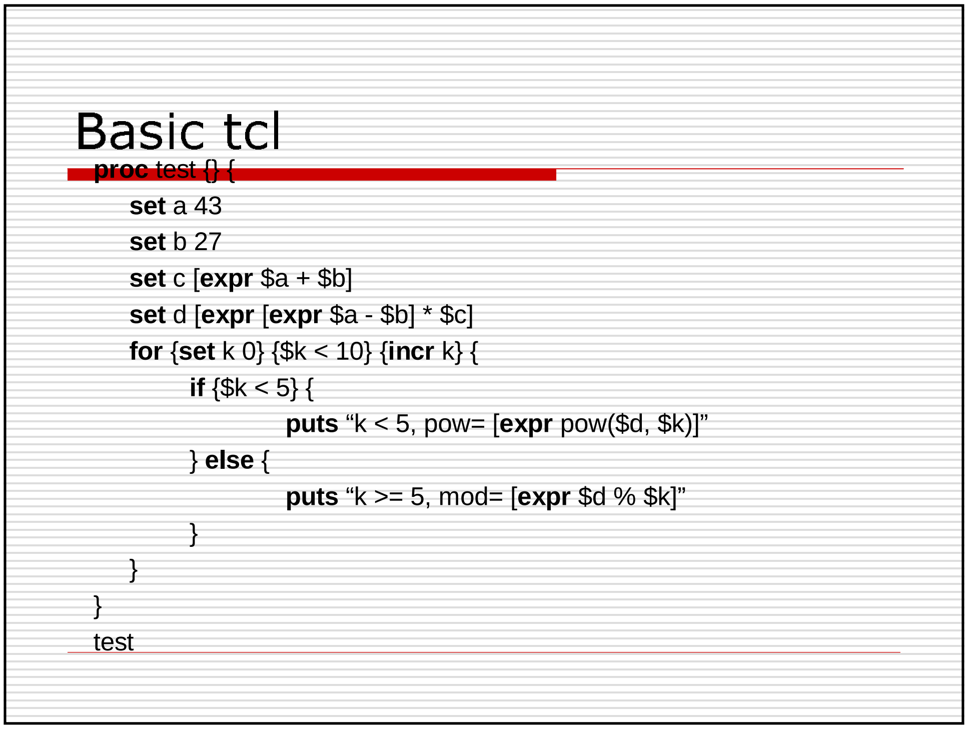


Hello World - Passive Mode

simple.tcl

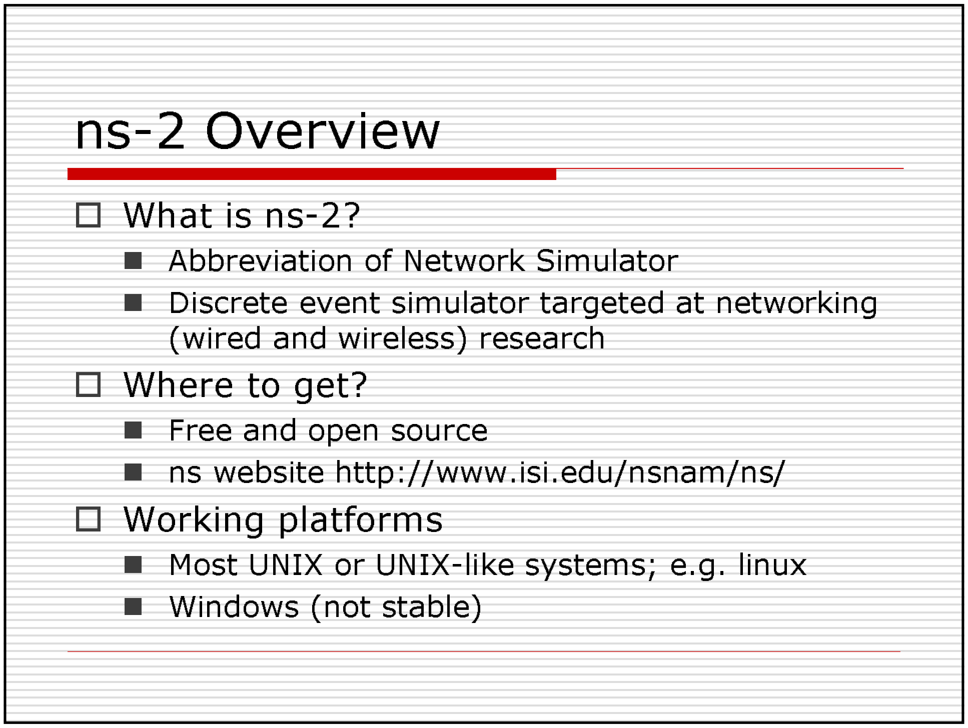
```
set ns [new Simulator]
$ns at 1 “puts \“Hello World!\””
$ns at 1.5 “exit”
$ns run
```

```
swallow 74% ns simple.tcl
Hello World!
swallow 75%
```



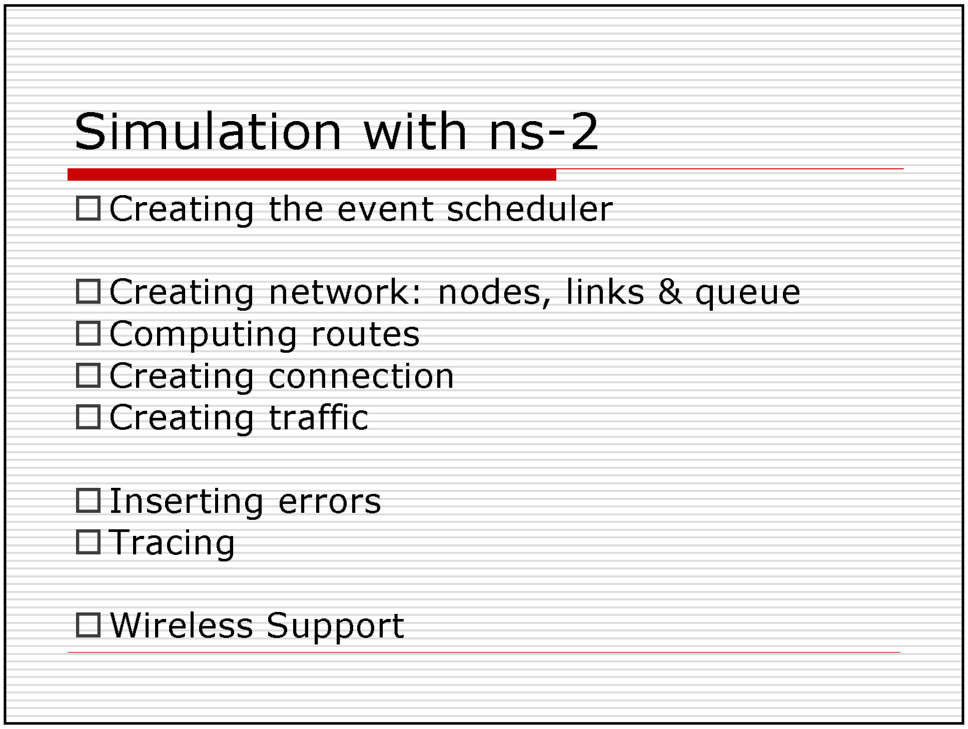
Basic tcl

```
proc test {} {
    set a 43
    set b 27
    set c [expr $a + $b]
    set d [expr [expr $a - $b] * $c]
    for {set k 0} {$k < 10} {incr k} {
        if {$k < 5} {
                puts “k < 5, pow= [expr pow($d, $k)]”
        } else {
                puts “k >= 5, mod= [expr $d % $k]”
        }
    }
}
test
```



ns-2 Overview

- □ What is ns-2?
  - ■ Abbreviation of Network Simulator
  - ■ Discrete event simulator targeted at networking (wired and wireless) research
- □ Where to get?
  - ■ Free and open source
  - ■ ns website http://www.isi.edu/nsnam/ns/
- □ Working platforms
  - ■ Most UNIX or UNIX-like systems; e.g. linux
  - ■ Windows (not stable)



Simulation with ns-2

- □ Creating the event scheduler

- □ Creating network: nodes, links & queue
- □ Computing routes
- □ Creating connection
- □ Creating traffic

- □ Inserting errors
- □ Tracing

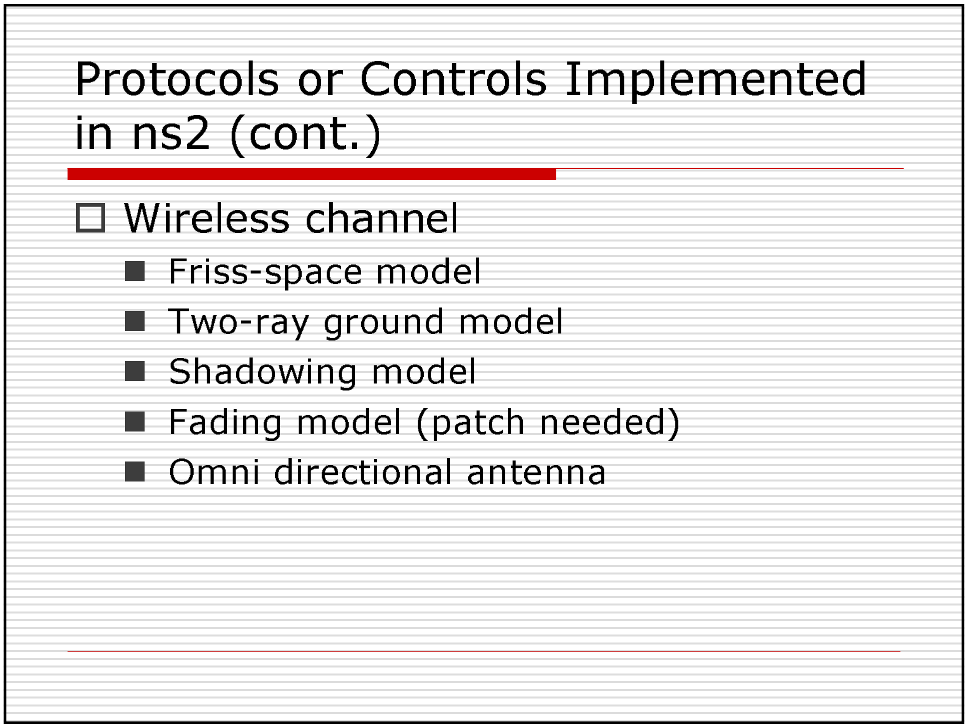
- □ Wireless Support

Protocols or Controls Implemented in ns2

- □ Transport layer (traffic agent)
  TCP; UDP
- □ Network layer (routing agent)
  - ■ Wired
    Distance vector; Link state (patch needed)
  - ■ Wireless
    AODV; DSR; DSDV; TORA
- □ Interface queue
  FIFO queue; DropTail queue; Priority queue; etc.
- □ Logic link control layer
  IEEE 802.2; ARP

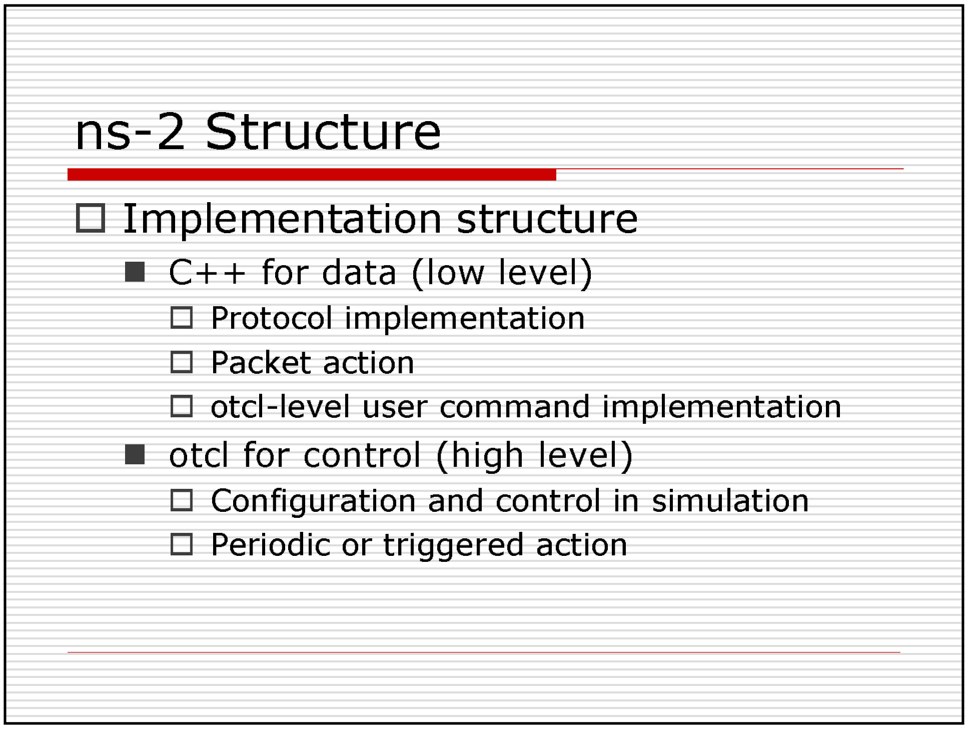
Protocols or Controls Implemented in ns2 (cont.)

- □ MAC layer
  - ■ Wired
    - □ IEEE 802.3 (CSMA/CD)
  - ■ Wireless
    - □ IEEE 802.11 (CSMA/CA)
      - ■ DCF
      - ■ PCF (partially implemented)
- □ Physical layer
  - ■ Wired
    - □ IEEE 802.3
  - ■ Wireless
    - □ IEEE 802.11
      - ■ DSSS (Direct Sequence Spread Spectrum)
      - ■ FHSS (Frequency-Hopping Spread Spectrum); not implemented
      - ■ IR (Infrared); not implemented



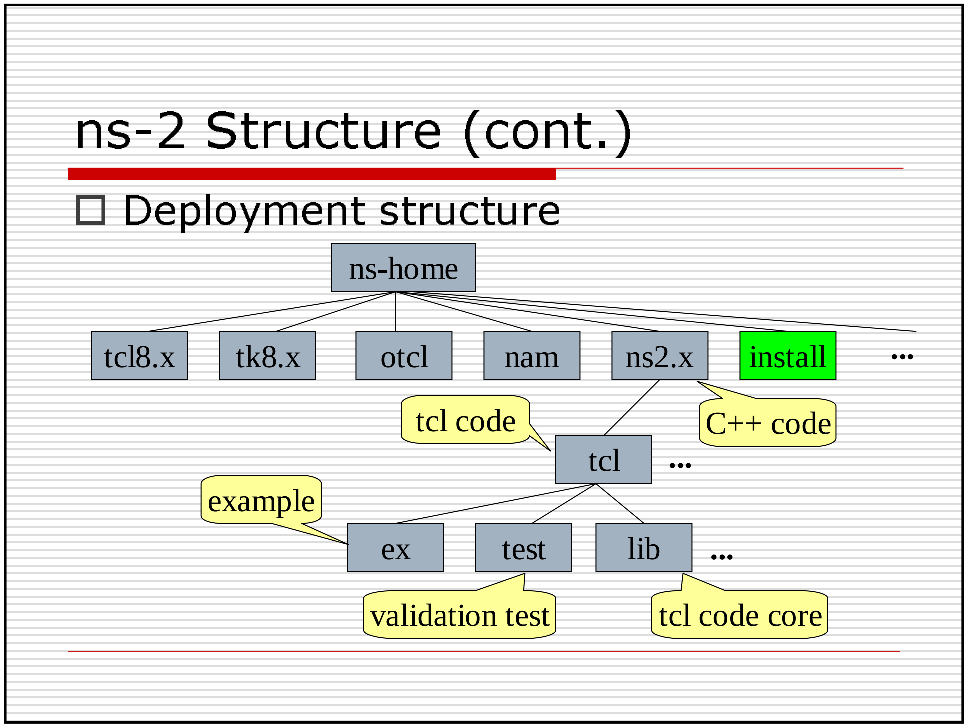
Protocols or Controls Implemented in ns2 (cont.)

- □ Wireless channel
  - ■ Friss-space model
  - ■ Two-ray ground model
  - ■ Shadowing model
  - ■ Fading model (patch needed)
  - ■ Omni directional antenna



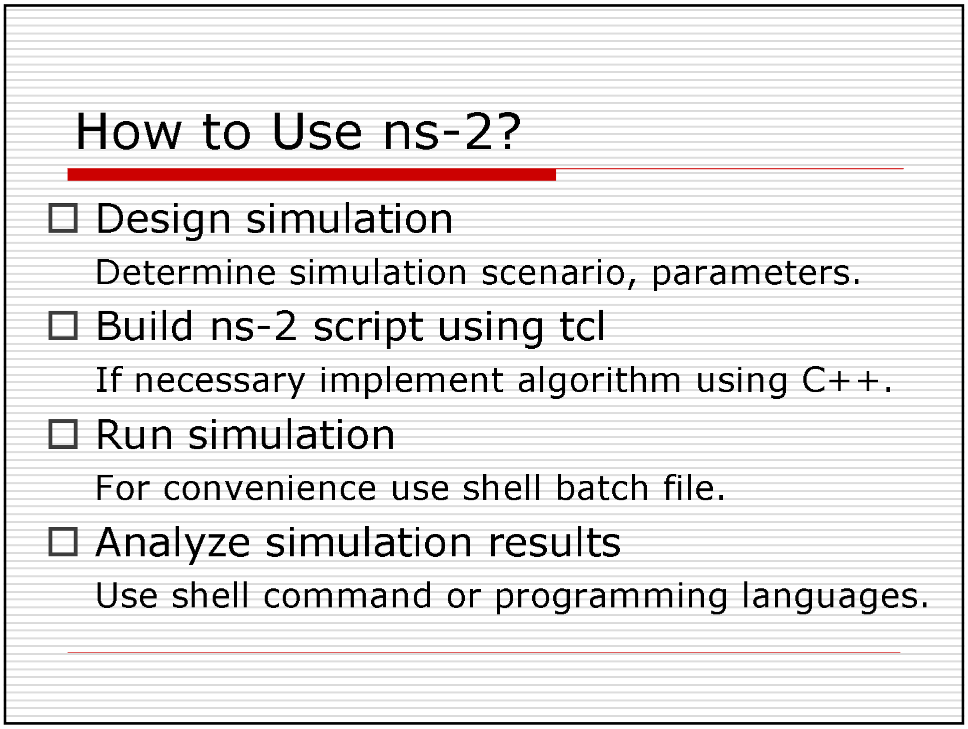
ns-2 Structure

- □ Implementation structure
  - ■ C++ for data (low level)
    - □ Protocol implementation
    - □ Packet action
    - □ otcl-level user command implementation
  - ■ otcl for control (high level)
    - □ Configuration and control in simulation
    - □ Periodic or triggered action



ns-2 Structure (cont.)

- □ Deployment structure



How to Use ns-2?

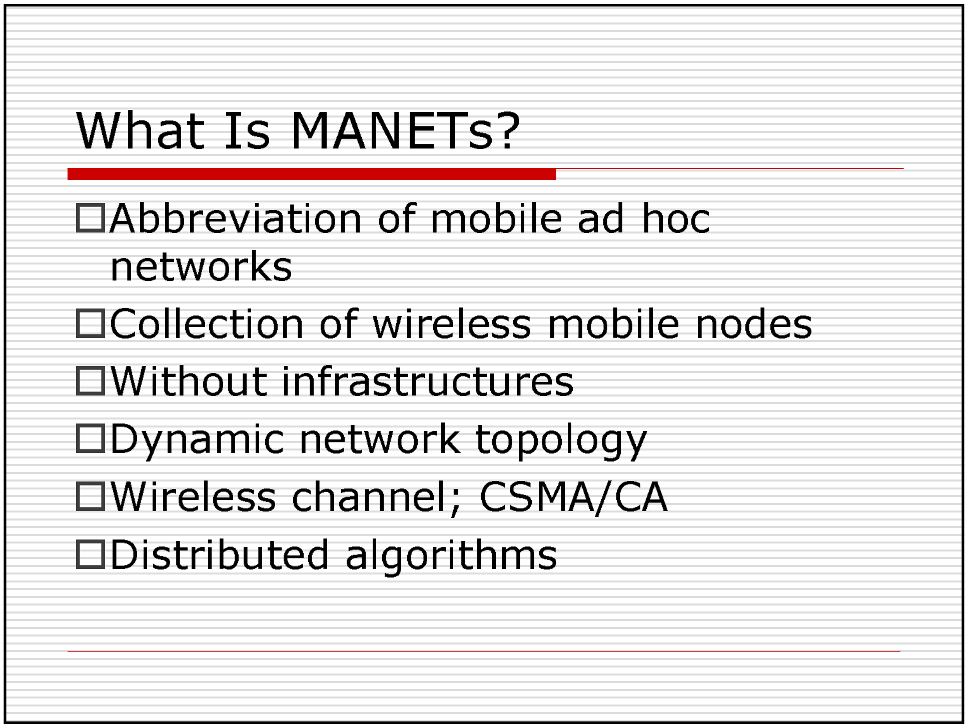
- □ Design simulation

  Determine simulation scenario, parameters.
- □ Build ns-2 script using tcl

  If necessary implement algorithm using C++.
- □ Run simulation

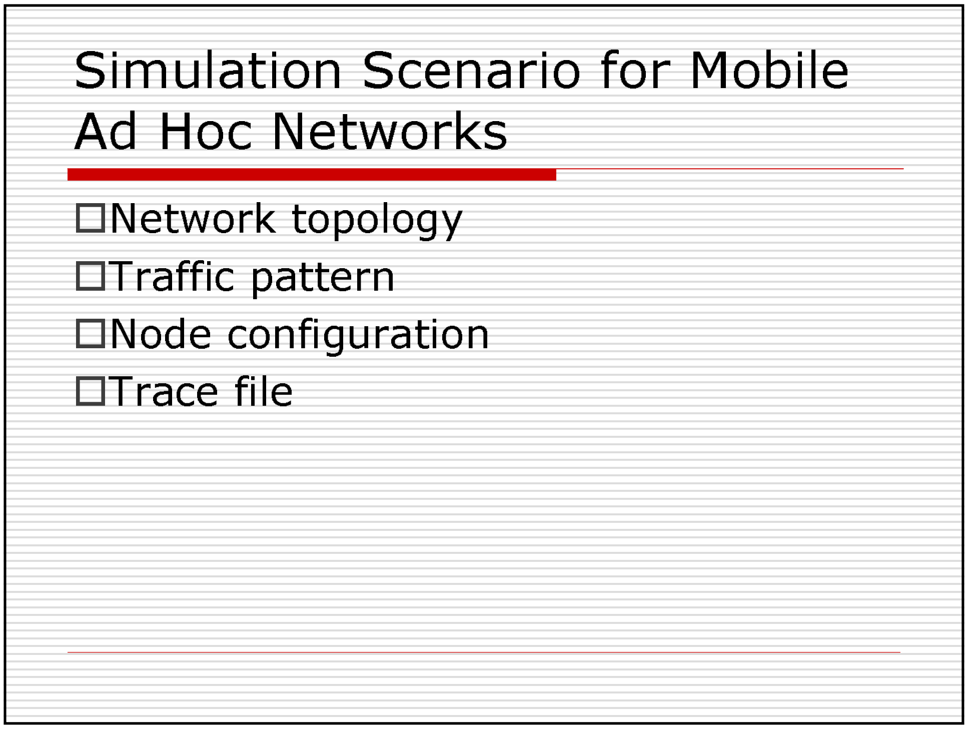
  For convenience use shell batch file.
- □ Analyze simulation results

  Use shell command or programming languages.



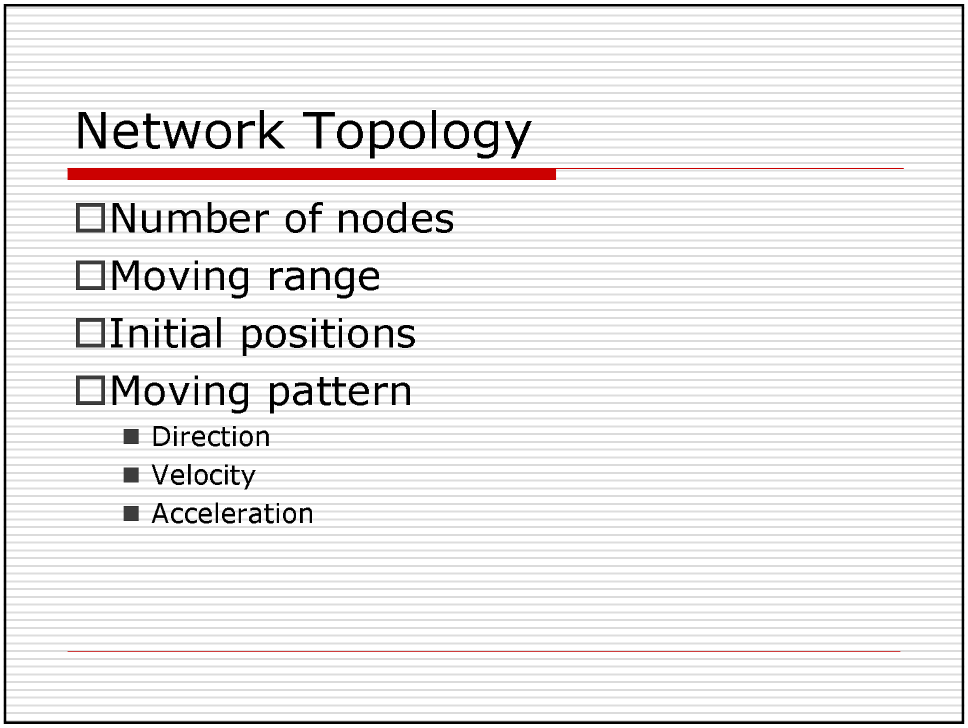
What Is MANETs?

- □Abbreviation of mobile ad hoc networks
- □Collection of wireless mobile nodes
- □Without infrastructures
- □Dynamic network topology
- □Wireless channel; CSMA/CA
- □Distributed algorithms



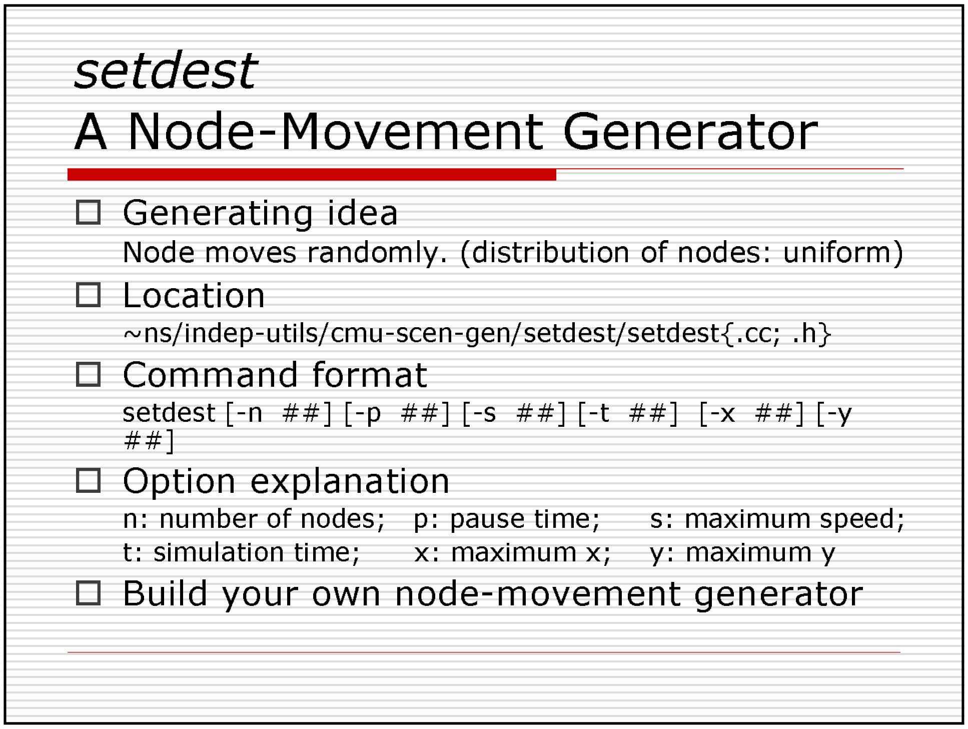
Simulation Scenario for Mobile Ad Hoc Networks

- □Network topology
- □Traffic pattern
- □Node configuration
- □Trace file



Network Topology

- □ Number of nodes
- □ Moving range
- □ Initial positions
- □ Moving pattern
  - ■ Direction
  - ■ Velocity
  - ■ Acceleration



setdest A Node-Movement Generator

□ Generating idea

Node moves randomly. (distribution of nodes: uniform)

□ Location

~ns/indep-utils/cmu-scen-gen/setdest/setdest{.cc; .h}

□ Command format

```
setdest [-n  ##] [-p  ##] [-s  ##] [-t  ##]  [-x  ##] [-y
##]
```

□ Option explanation

n: number of nodes;   p: pause time;   s: maximum speed;
t: simulation time;   x: maximum x;   y: maximum y

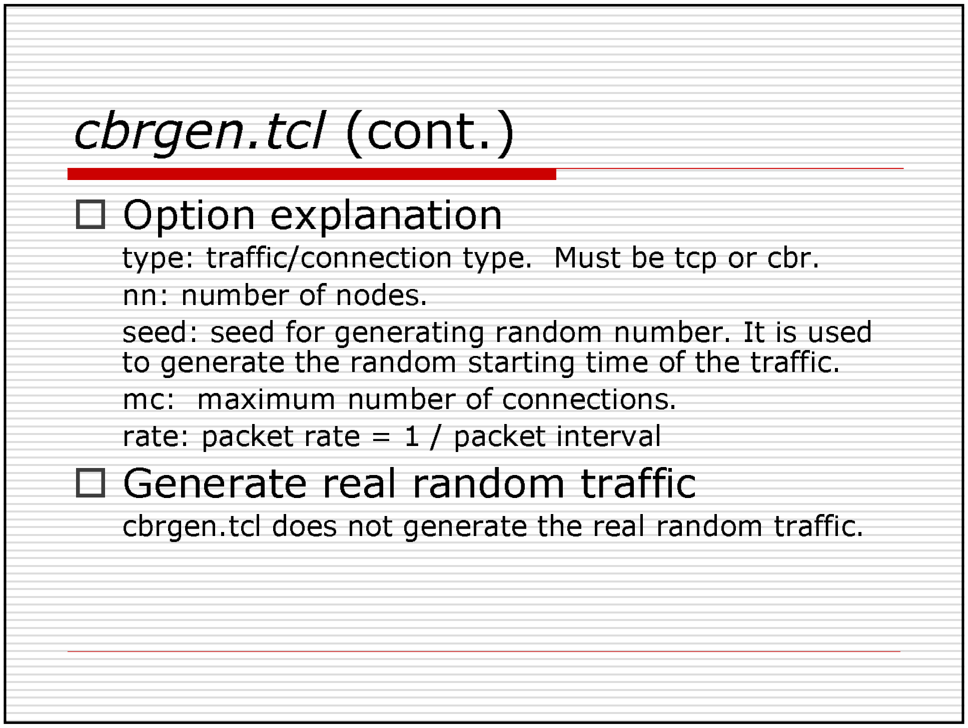
□ Build your own node-movement generator

Traffic Pattern

- Number of connections
- Traffic source/destination
- Connection type
  - TCP/UDP
- Packet size
- Packet rate

cbrgen.tcl
A CBR Traffic Generator

- What is CBR?
  - Constant Bit Rate.
- Generating idea
  - Randomly pick up node pairs as sources and destinations.
- Location
  - ~ns/indep-utils/cmu-scen-gen/cbrgen.tcl
- Command format
  - ns cbrgen.tcl [-type ##] [-nn ##] [-seed ##] [-mc ##][-rate ##]



cbrgen.tcl (cont.)

- Option explanation

  type: traffic/connection type. Must be tcp or cbr.

  nn: number of nodes.

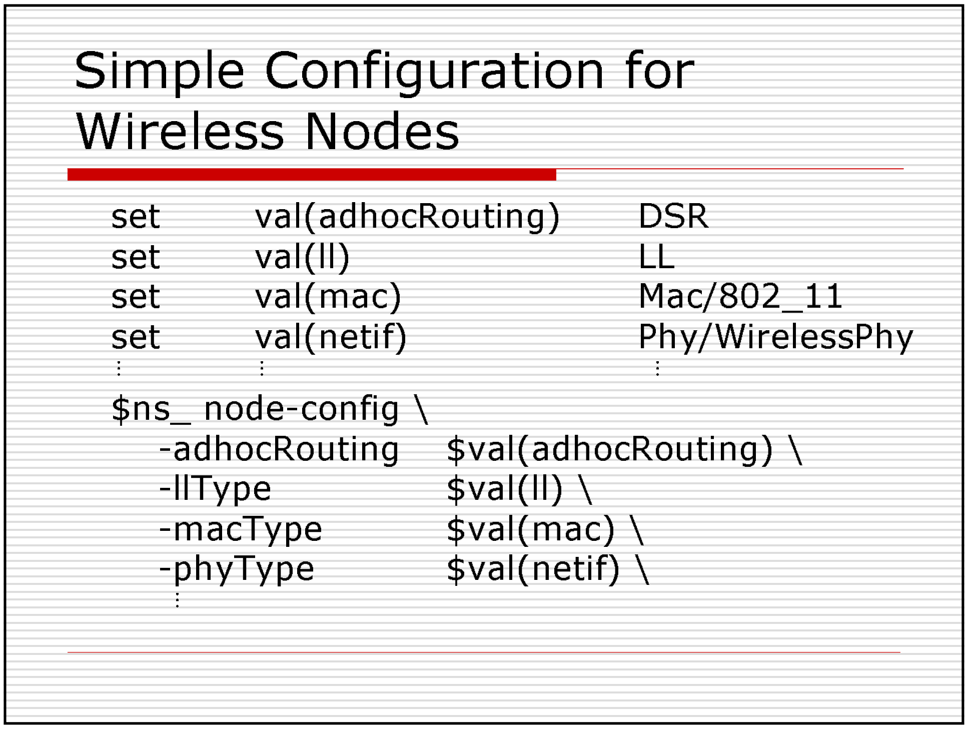
  seed: seed for generating random number. It is used to generate the random starting time of the traffic.

  mc: maximum number of connections.

  rate: packet rate = 1 / packet interval

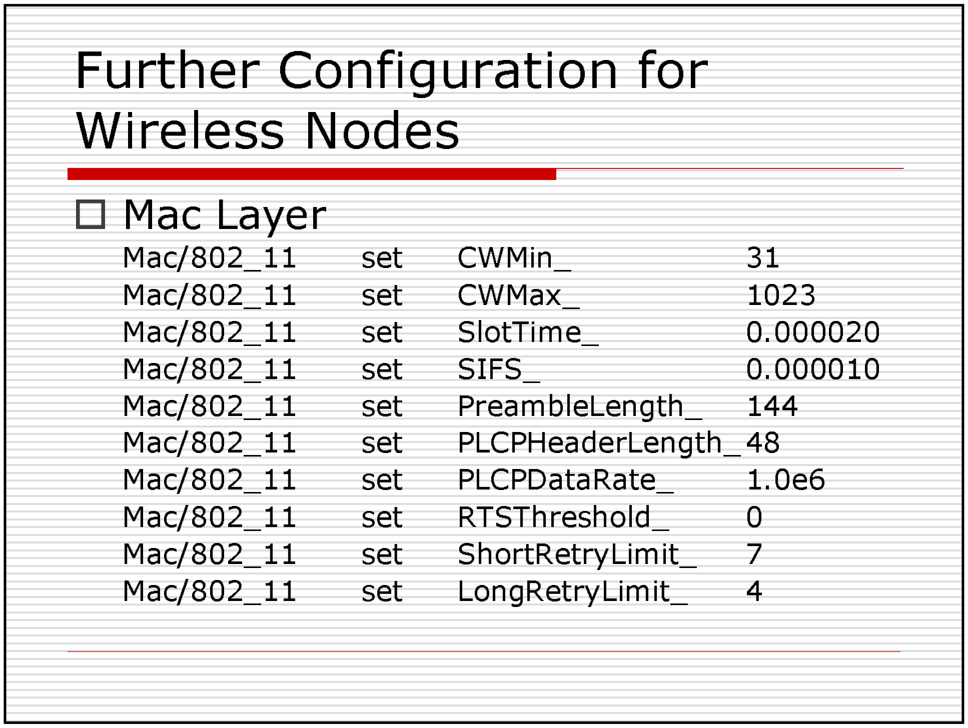
- Generate real random traffic

  cbrgen.tcl does not generate the real random traffic.



Simple Configuration for Wireless Nodes

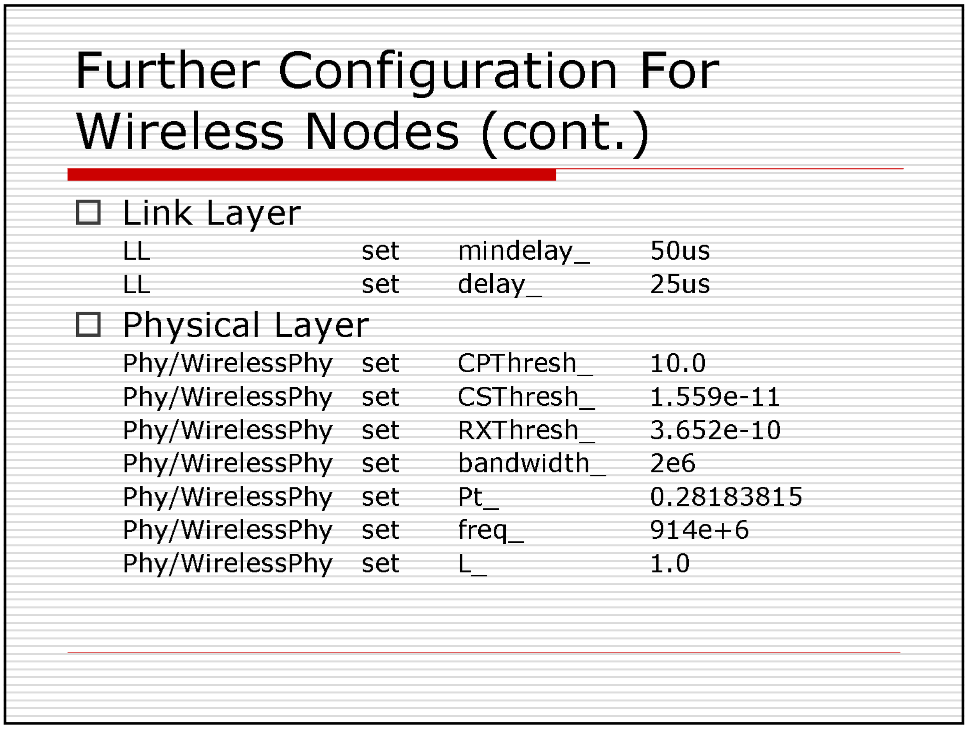
```
set      val(adhocRouting)      DSR
set      val(ll)                LL
set      val(mac)               Mac/802_11
set      val(netif)             Phy/WirelessPhy
⋮        ⋮                       ⋮
$ns_ node-config \
    -adhocRouting    $val(adhocRouting) \
    -llType          $val(ll) \
    -macType         $val(mac) \
    -phyType         $val(netif) \
     ⋮
```



Further Configuration for Wireless Nodes

□ Mac Layer

```
Mac/802_11      set     CWMin_              31
Mac/802_11      set     CWMax_              1023
Mac/802_11      set     SlotTime_           0.000020
Mac/802_11      set     SIFS_               0.000010
Mac/802_11      set     PreambleLength_     144
Mac/802_11      set     PLCPHeaderLength_ 48
Mac/802_11      set     PLCPDataRate_       1.0e6
Mac/802_11      set     RTSThreshold_       0
Mac/802_11      set     ShortRetryLimit_    7
Mac/802_11      set     LongRetryLimit_     4
```



Further Configuration For Wireless Nodes (cont.)

□ Link Layer

```
LL              set     mindelay_    50us
LL              set     delay_       25us
```

□ Physical Layer

```
Phy/WirelessPhy set     CPThresh_    10.0
Phy/WirelessPhy set     CSThresh_    1.559e-11
Phy/WirelessPhy set     RXThresh_    3.652e-10
Phy/WirelessPhy set     bandwidth_   2e6
Phy/WirelessPhy set     Pt_          0.28183815
Phy/WirelessPhy set     freq_        914e+6
Phy/WirelessPhy set     L_           1.0
```

Further Configuration For Wireless Nodes (cont.)

□ Radio Propagation Model

```
Propagation/Shadowing   set   pathlossExp_   2.0
Propagation/Shadowing   set   std_db_        4.0
Propagation/Shadowing   set   dist0_         1.0
Propagation/Shadowing   set   seed_          0

Antenna/OmniAntenna     set   X_             0
Antenna/OmniAntenna     set   Y_             0
Antenna/OmniAntenna     set   Z_             1.5
Antenna/OmniAntenna     set   Gt_            1.0
Antenna/OmniAntenna     set   Gr_            1.0
```

Trace File Format

```
r 100.381997477 _1_ AGT  --- 82 tcp 1060 [13a 1 0 800] ------- [0:0 1:0 32 1]
   [32 0] 1 0
```

r: receive event; **100.381997477**: time stamps;
_1_ : nod:e 1; **AGT**: trace generated by agent;
82: event(pkt) id; **tcp**: tcp packet;
1060: packet size;
13a: expected duration of pkt transmission;
1 : sender mac id; **0**: transmitter mac id;
800: pkt type; **0:0**: sender address:port#;
1:0: receiver address:port#; **32**: TTL;
1: next hop address; **[32 0]** : TCP sequence #, ack #.

Summarize Trace File

□ Using simple linux commands

cat, grep, wc, |, >, >>, etc.

eg. Calculate packet delivery ratio from a trace file (aaa.tr)

- Number of sent packets

```
cat aaa.tr | grep AGT | grep cbr | grep ^s | wc -l
```

- Number of received packets

```
cat aaa.tr | grep AGT | grep cbr | grep ^r | wc -l
```

□ Simple programming

shell, awk, etc.

□ Advanced programming

C/C++, Java, VB, etc.

awk

□ Calling format

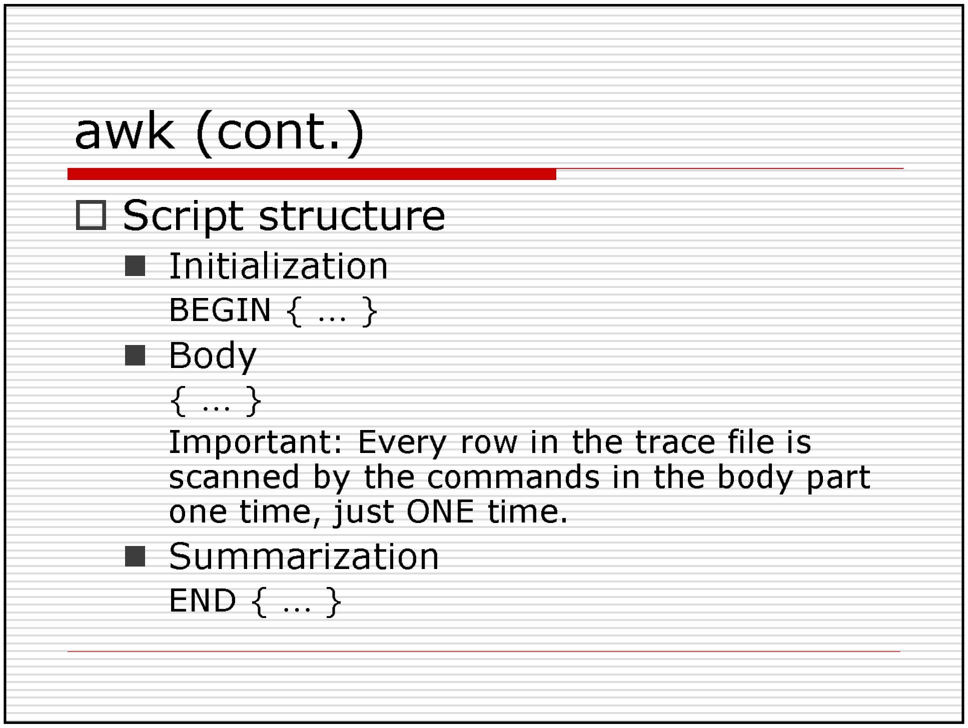
- awk \`/pattern-to-match/ {program to run}\` trace-file

eg. awk \`$1 == “s” {print}\` aaa.tr

- awk –f awk-script trace-file

□ Characteristics

- Flexible (C style)
- Simple (no pointers, no references)
- Powerful
  - □ Float calculation.
  - □ Automatic data type assignment and check.
  - □ Branch/Loop control.
  - □ Function call.



awk (cont.)

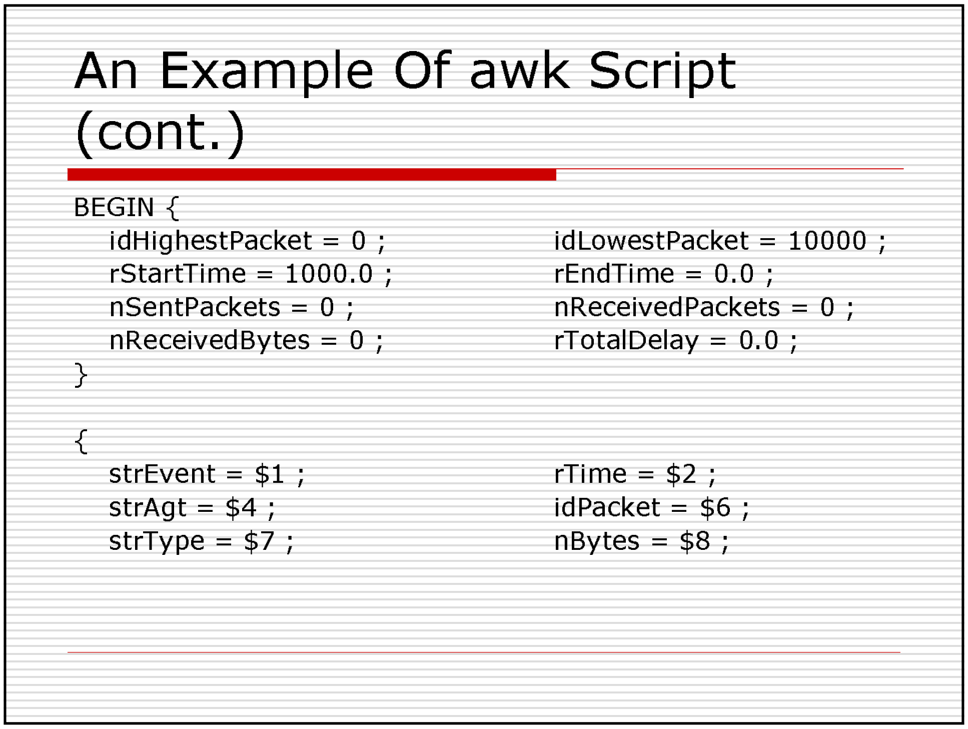
- □ Script structure
  - ■ Initialization

    BEGIN { … }
  - ■ Body

    { … }

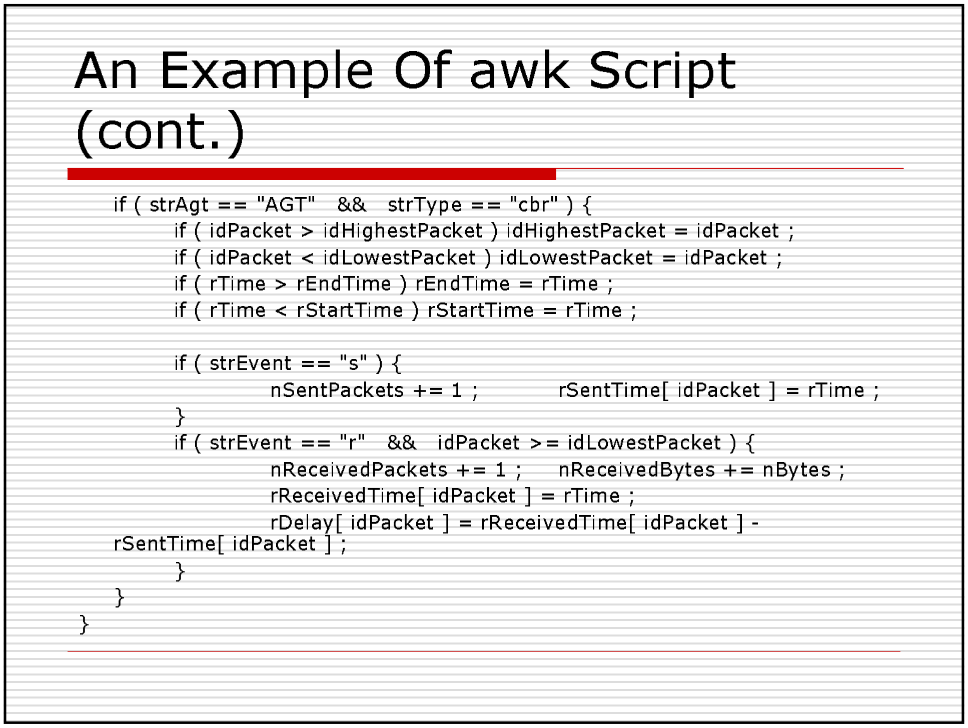
    Important: Every row in the trace file is scanned by the commands in the body part one time, just ONE time.
  - ■ Summarization

    END { … }



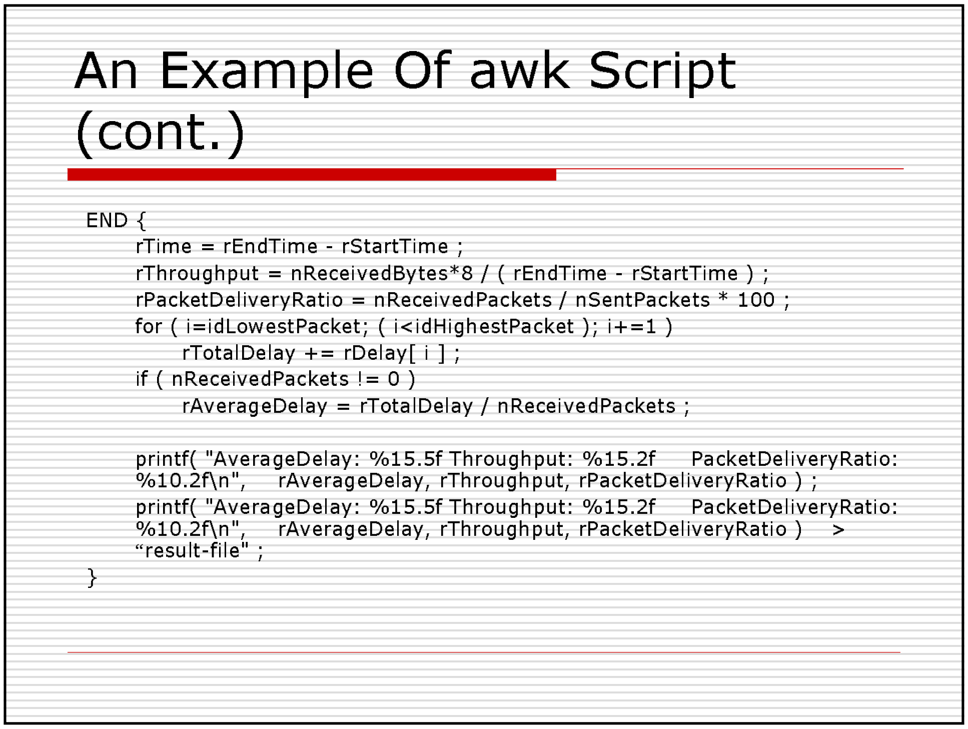
An Example Of awk Script (cont.)

```
BEGIN {
   idHighestPacket = 0 ;            idLowestPacket = 10000 ;
   rStartTime = 1000.0 ;            rEndTime = 0.0 ;
   nSentPackets = 0 ;               nReceivedPackets = 0 ;
   nReceivedBytes = 0 ;             rTotalDelay = 0.0 ;
}

{
   strEvent = $1 ;                  rTime = $2 ;
   strAgt = $4 ;                    idPacket = $6 ;
   strType = $7 ;                   nBytes = $8 ;
```



An Example Of awk Script (cont.)

```
    if ( strAgt == "AGT"   &&   strType == "cbr" ) {
        if ( idPacket > idHighestPacket ) idHighestPacket = idPacket ;
        if ( idPacket < idLowestPacket ) idLowestPacket = idPacket ;
        if ( rTime > rEndTime ) rEndTime = rTime ;
        if ( rTime < rStartTime ) rStartTime = rTime ;

        if ( strEvent == "s" ) {
                    nSentPackets += 1 ;          rSentTime[ idPacket ] = rTime ;
        }
        if ( strEvent == "r"   &&   idPacket >= idLowestPacket ) {
                    nReceivedPackets += 1 ;      nReceivedBytes += nBytes ;
                    rReceivedTime[ idPacket ] = rTime ;
                    rDelay[ idPacket ] = rReceivedTime[ idPacket ] -
    rSentTime[ idPacket ] ;
        }
    }
}
```



An Example Of awk Script (cont.)

```
 END {
      rTime = rEndTime - rStartTime ;
      rThroughput = nReceivedBytes*8 / ( rEndTime - rStartTime ) ;
      rPacketDeliveryRatio = nReceivedPackets / nSentPackets * 100 ;
      for ( i=idLowestPacket; ( i<idHighestPacket ); i+=1 )
            rTotalDelay += rDelay[ i ] ;
      if ( nReceivedPackets != 0 )
            rAverageDelay = rTotalDelay / nReceivedPackets ;

      printf( "AverageDelay: %15.5f Throughput: %15.2f     PacketDeliveryRatio:
      %10.2f\n",    rAverageDelay, rThroughput, rPacketDeliveryRatio ) ;
      printf( "AverageDelay: %15.5f Throughput: %15.2f     PacketDeliveryRatio:
      %10.2f\n",    rAverageDelay, rThroughput, rPacketDeliveryRatio )    >
      “result-file" ;
 }
```

ns Tutorials

- □ NS website http://www.isi.edu/nsnam/ns/
- □ NS Manual http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
- □ Marc Greis's Tutorial http://www.isi.edu/nsnam/ns/tutorial
- □ http://www.cs.virginia.edu/~cs757/slidespdf/cs757-ns2-tutorial-exercise1.pdf
- □ http://nile.wpi.edu/NS/
- □ http://nesl.ee.ucla.edu/courses/ee206a/2002s/guest_presentations/GP02_Park_ns2.ppt
- □ http://www.ece.ubc.ca/~elec565/ns2_tutorial.ppt

Run the Simulation

- □ex6sta.tcl (fil.awk, fil2.awk, fil4.awk, fil6.awk)
- □infra.tcl (fil-tcp.awk)
- □wireles-test.tcl