ENSC 833 Network Protocols And Performance

Final Project Presentations - Spring 2001

Performance of TCP Protocol Running over Wireless LAN Network using the Snoop Protocol

Chi-ho Ng and Jack Chow
cng@sierrawireless.com
chowj@tycoelectronics.com

The Roadmap

- Introduction (Problem and Scope of Project)
- TCP retransmission and window size
- Wireless LAN using TCP
- The Snoop Protocol
- Opnet Implementation

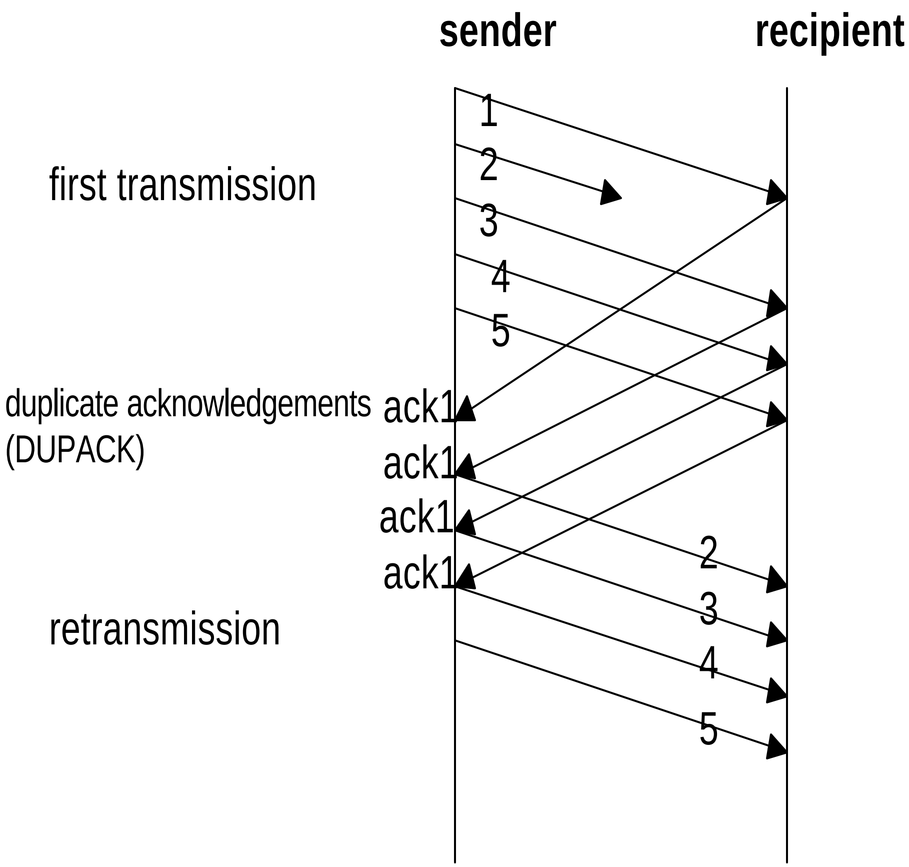
The Problem

- TCP is a reliable protocol with packet retransmission and congestion control (transmission window adjustment)
- Loss packets seen as network congestion; packets are re-sent with a smaller window size
- Scheme works well in wired network
- However, in wireless networks, with the high bit error rate, TCP reduces window size excessively and under-utilizes the bandwidth available

Our project

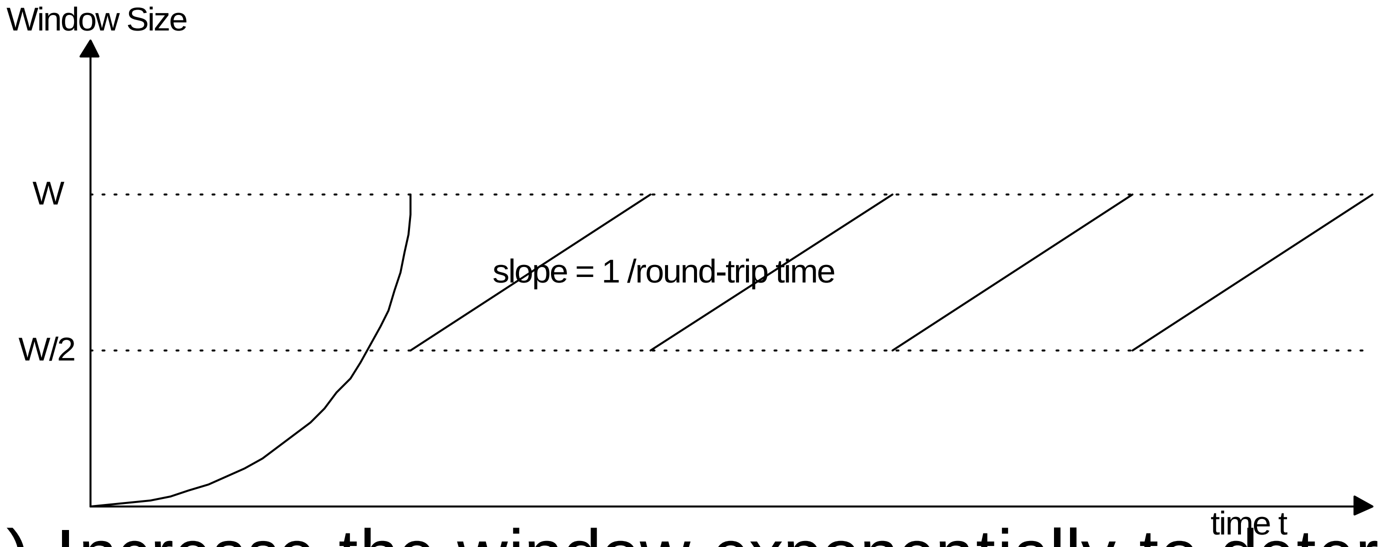
- Investigate the TCP congestion control policy and its problem on wireless LAN networks
- Research for possible enhancing algorithms
- Investigate the Snoop Protocol
- Implement Snoop on Opnet
- Compare results with and without Snoop

TCP retransmission policy



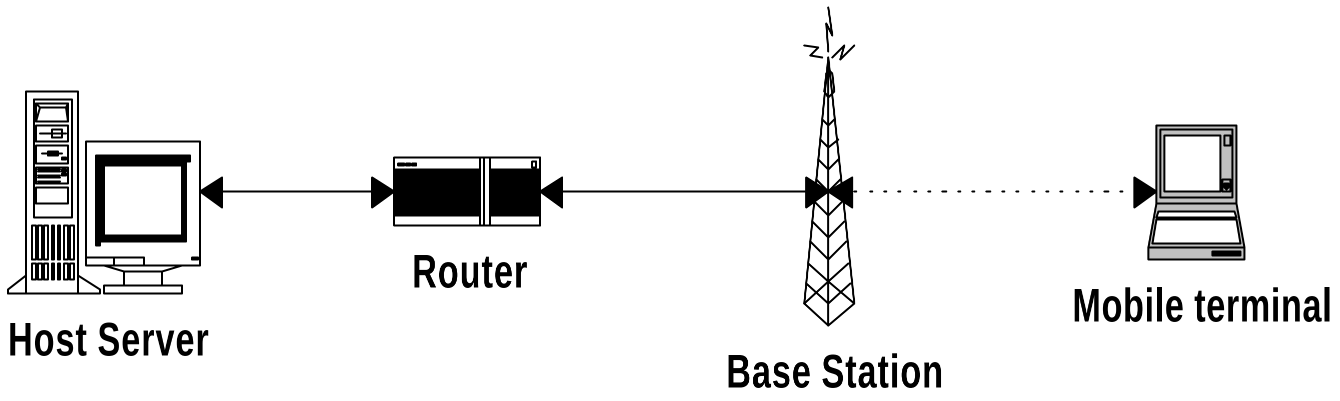
- For every packet received, the recipent returns an ACK
- Recipient sends duplicate ACK if a packet is lost
- Sender re-transmits the lost packets

TCP transmission window



- 1) Increase the window exponentially to determine the available bandwidth
- 2) When the source fails to receive an acknowledgement, the window size is reduced by half.
- 3) The source increases its window size by one unit every average round trip time

Wireless LAN using TCP

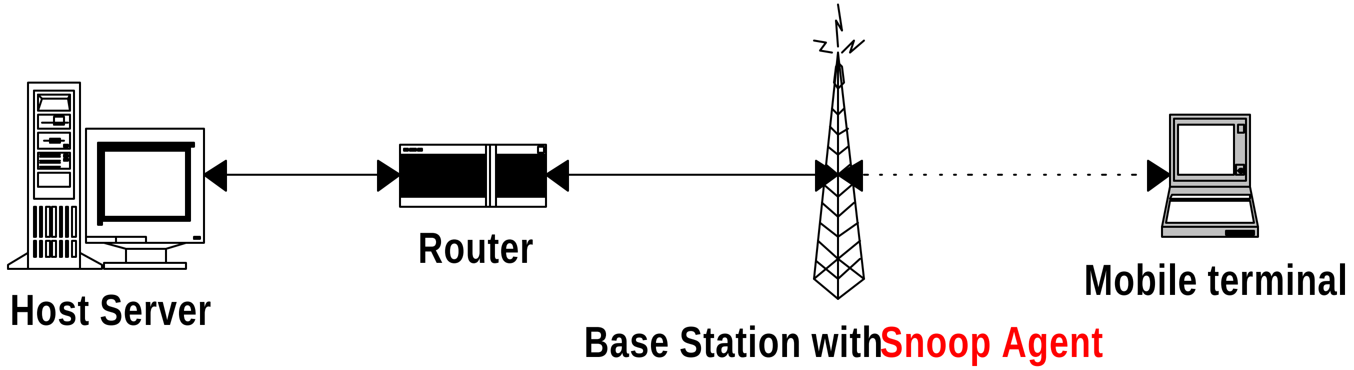


(1) Host server establishs a TCP connection with mobile terminal and starts to send data

(3) Missing acknowledgements trigger congestion control at host server (transmission window is reduced)

(2) High bit error rate in the wireless channel

Adding the Snoop Agent



(1) Host server establishes a TCP connection with mobile terminal and starts to send data

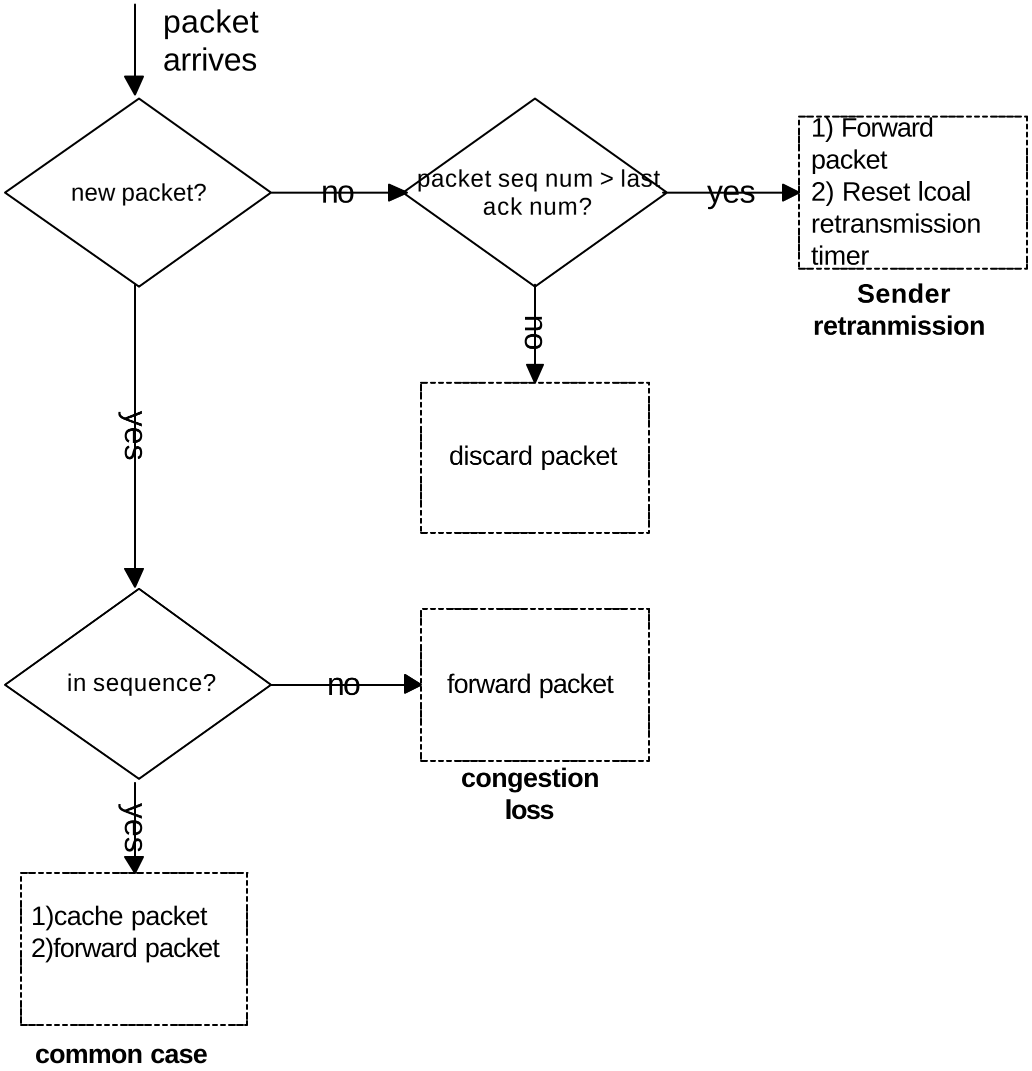
(3) Snoop Agent re-transmits lost packets locally

(2) High bit error rate in the wireless channel

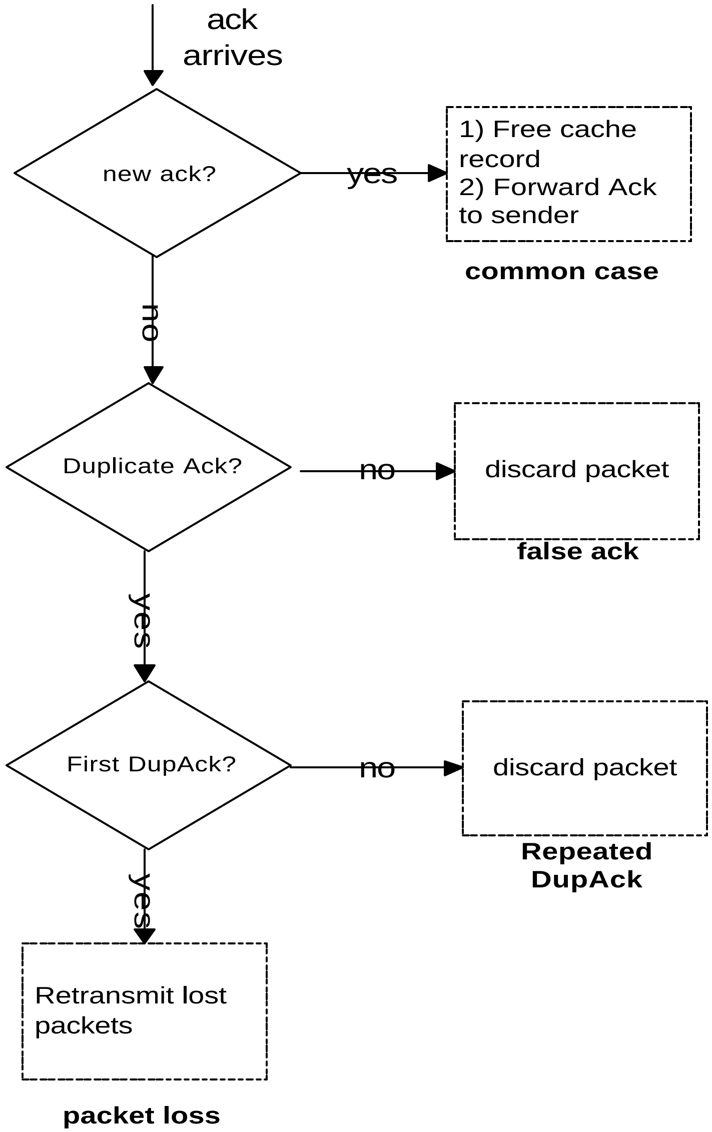
The Snoop Protocol

- Snoop copies packets to its cache
- Starts a retransmission timer
- Re-transmits the packets if a DupAck is received or the timer pops
- When an Ack is received, deletes the cache entry

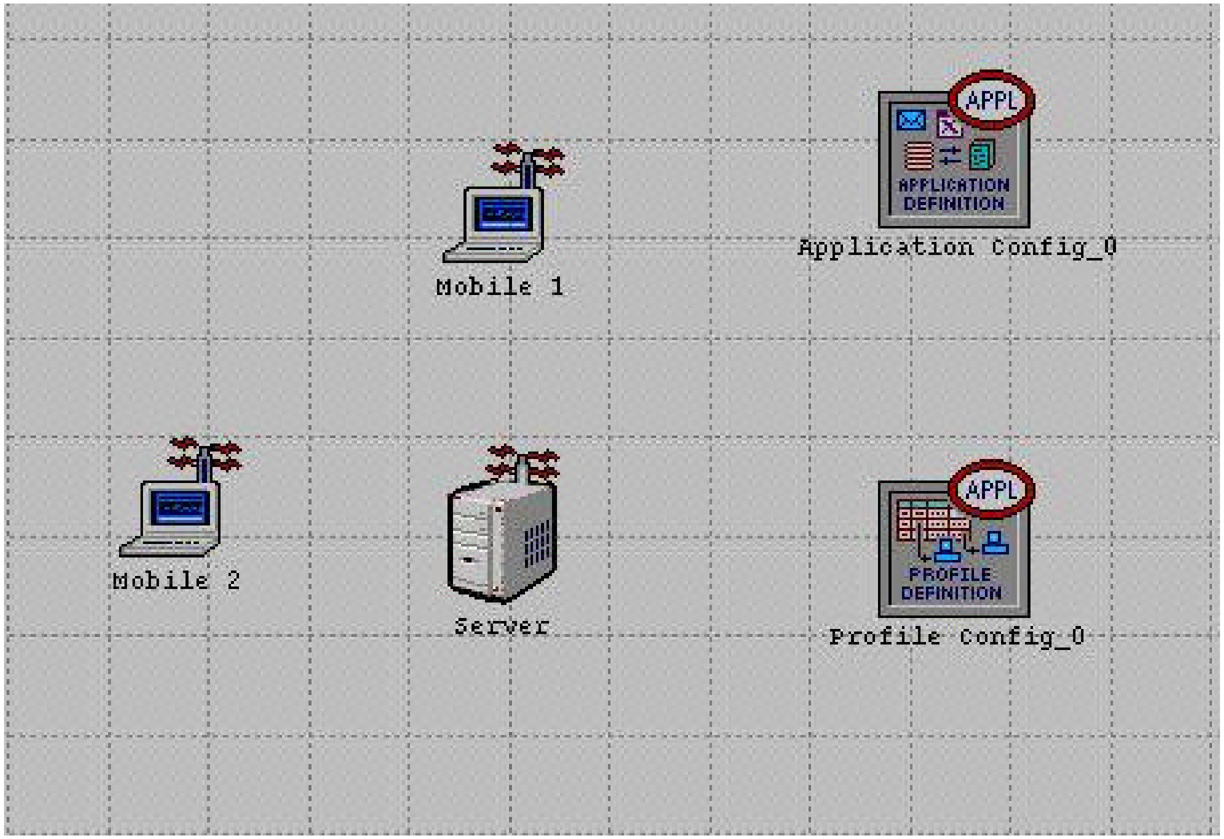
Snoop_Data()


packet arrives
new packet?
no
packet seq num > last ack num?
yes
1) Forward packet
2) Reset lcoal retransmission timer
Sender retranmission
no
discard packet
yes
in sequence?
no
forward packet
congestion loss
yes
1)cache packet
2)forward packet
common case

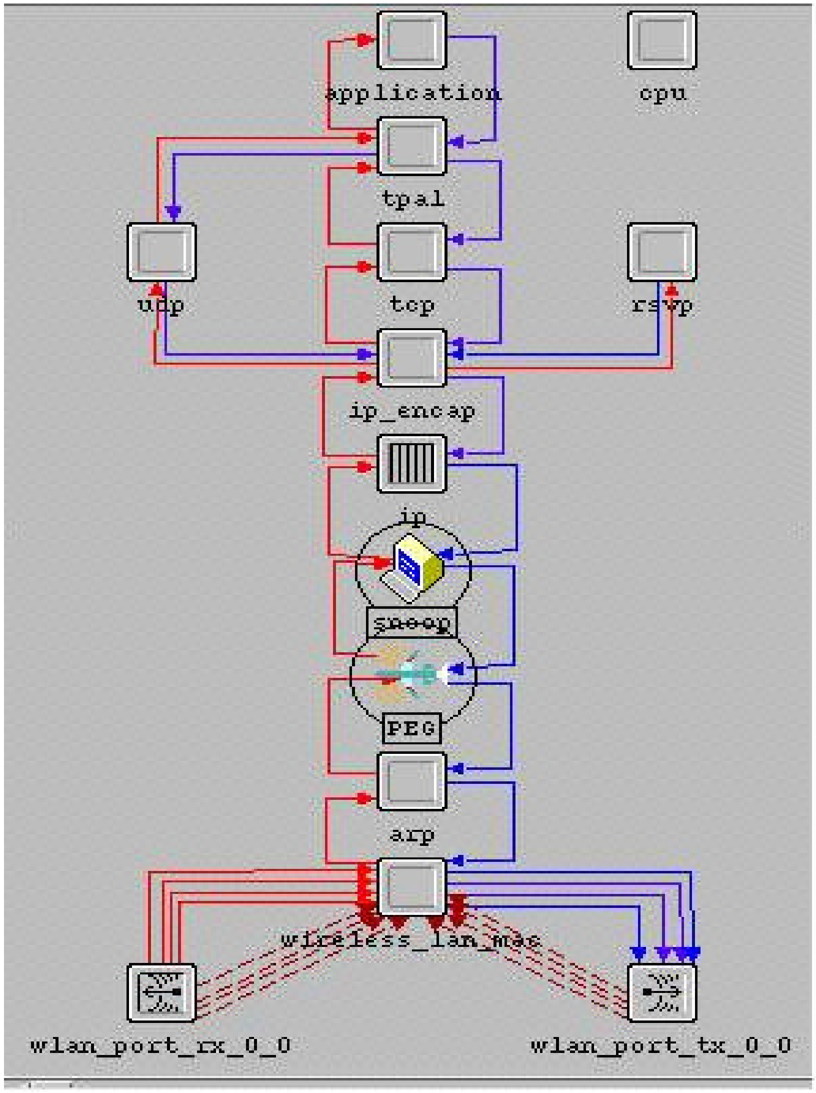
Snoop_Ack()


ack arrives
new ack?
yes
1) Free cache record
2) Forward Ack to sender
common case
no
Duplicate Ack?
no
discard packet
false ack
yes
First DupAck?
no
discard packet
Repeated DupAck
yes
Retransmit lost packets
packet loss

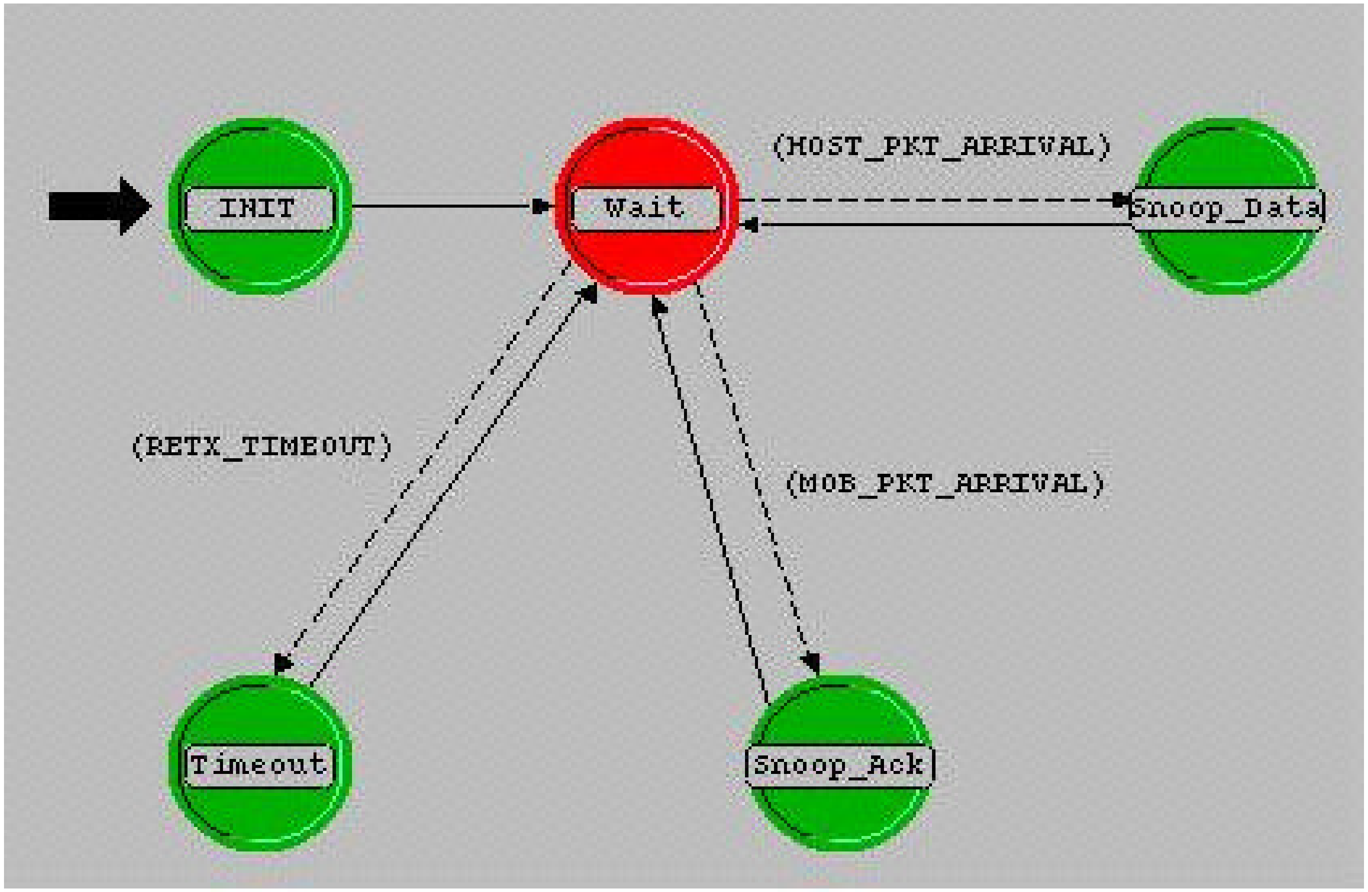
Opnet Implementation



2 extra protocol layers



Snoop State Transition Diagram



The Cache

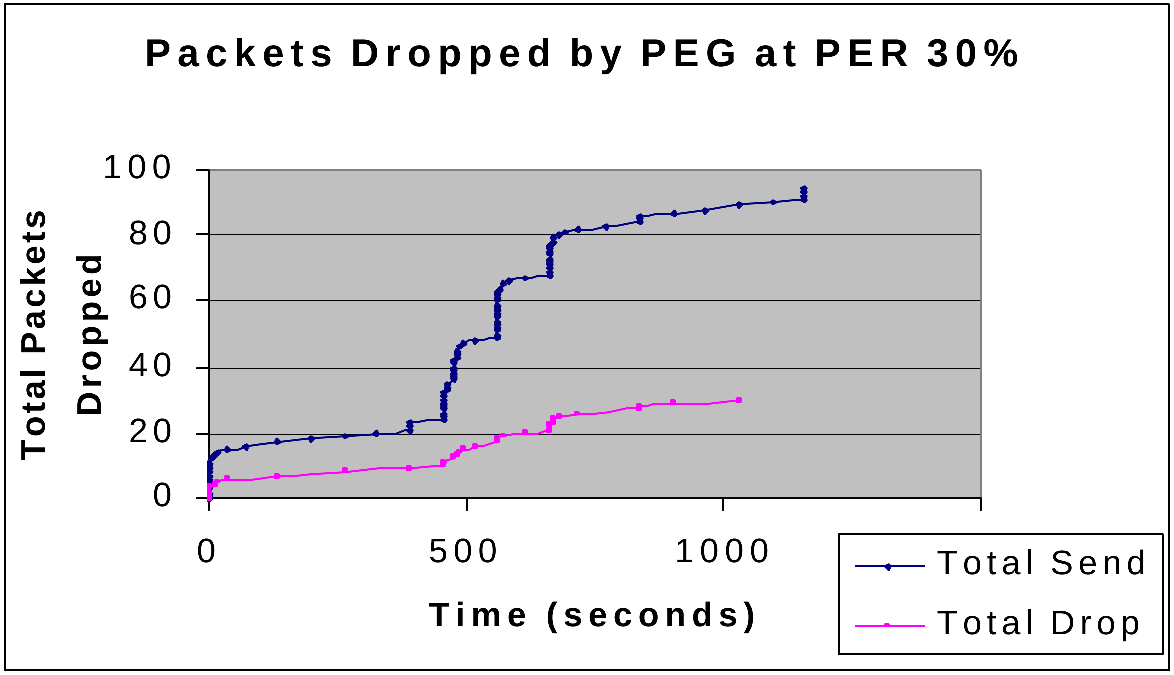
```
typedef struct
    {
            unsigned int  src_ip;
            unsigned int  dest_ip;
            int             src_port;
            int             dest_port;
            unsigned int   seq_num;
            unsigned int   ack_num;
            unsigned int   rcv_win;
            int              urgent_pointer;
            int              data_len;
            int              urg;
            int              ack;
            int              push;
            int              rst;
            int              syn;
                        int              fin;
    } TcpInfo;
```

Cache Functions

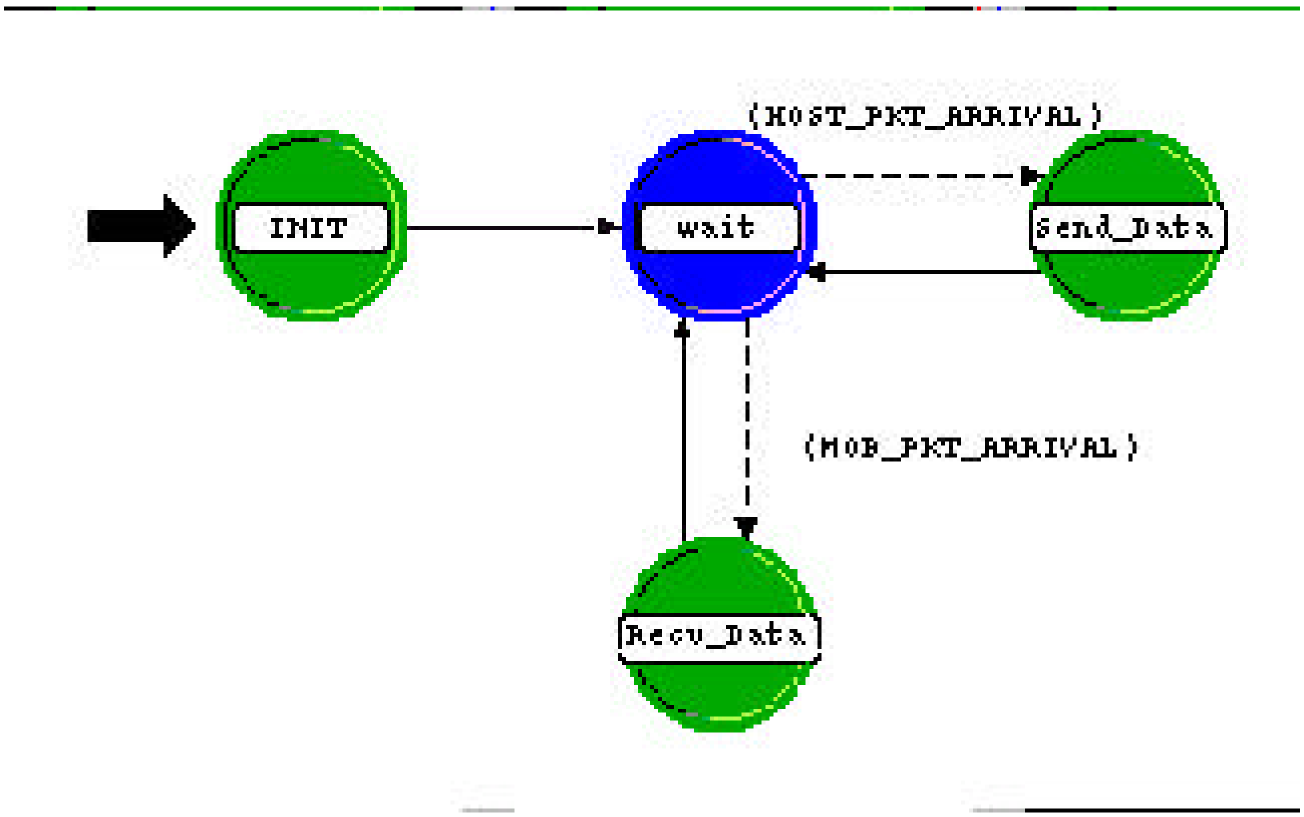
Function	Description
snCacheInit	Initialize the cache records
snCachedPkt	Copies a packet into the cache
snCacheRetrieve	Retrieves a packet from the cache
snCacheDestroy	Deletes a packet from the cache

Packet Error Generator

- Used to create packet loss
- Packet lost are uniformly distributed

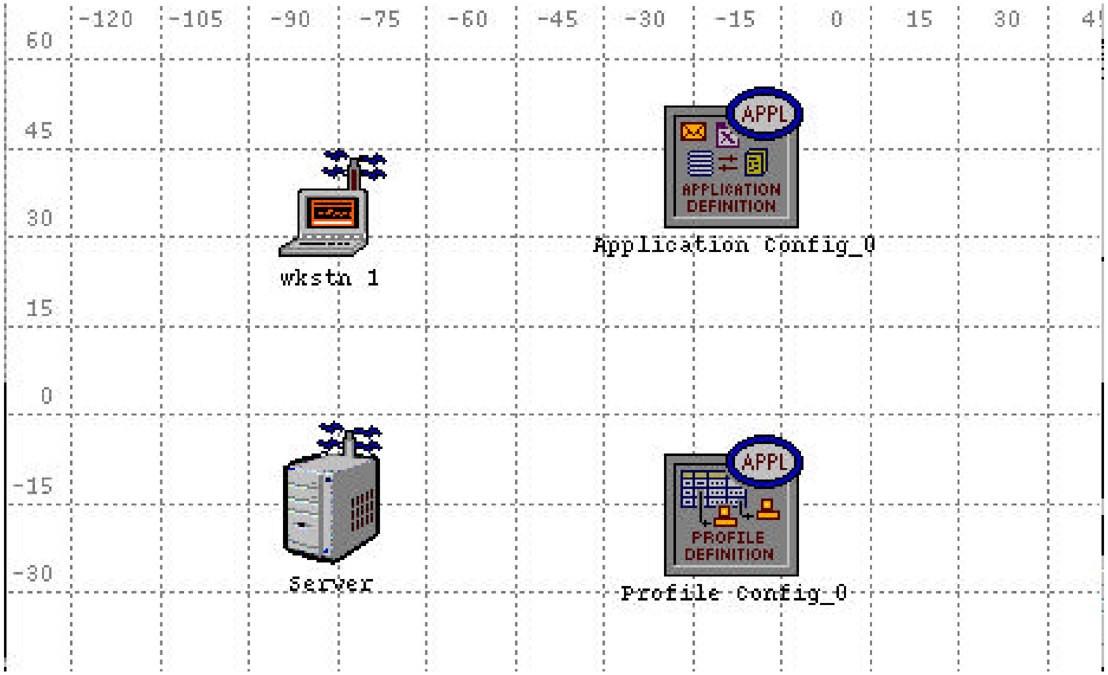


PEG State Transition Diagram



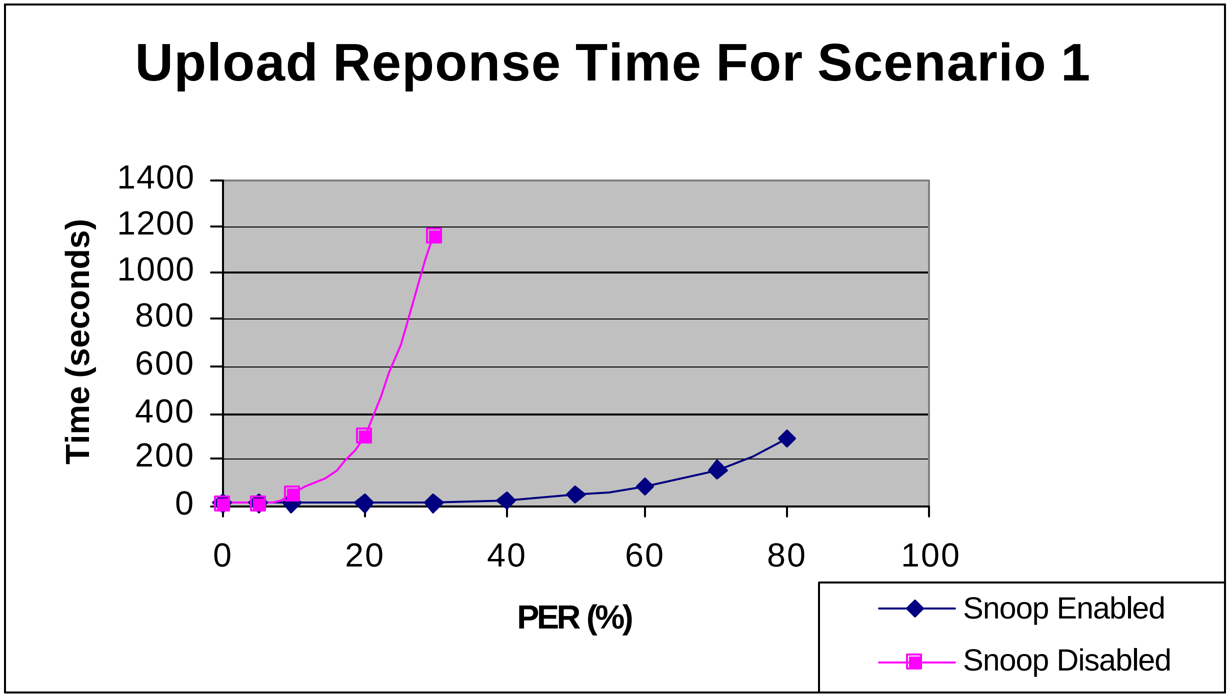
Scenario 1 – Single Mobile Upload

- To study how the Snoop Protocol improves the performance
- Upload 100, 000 byte file from Workstation 1



Results

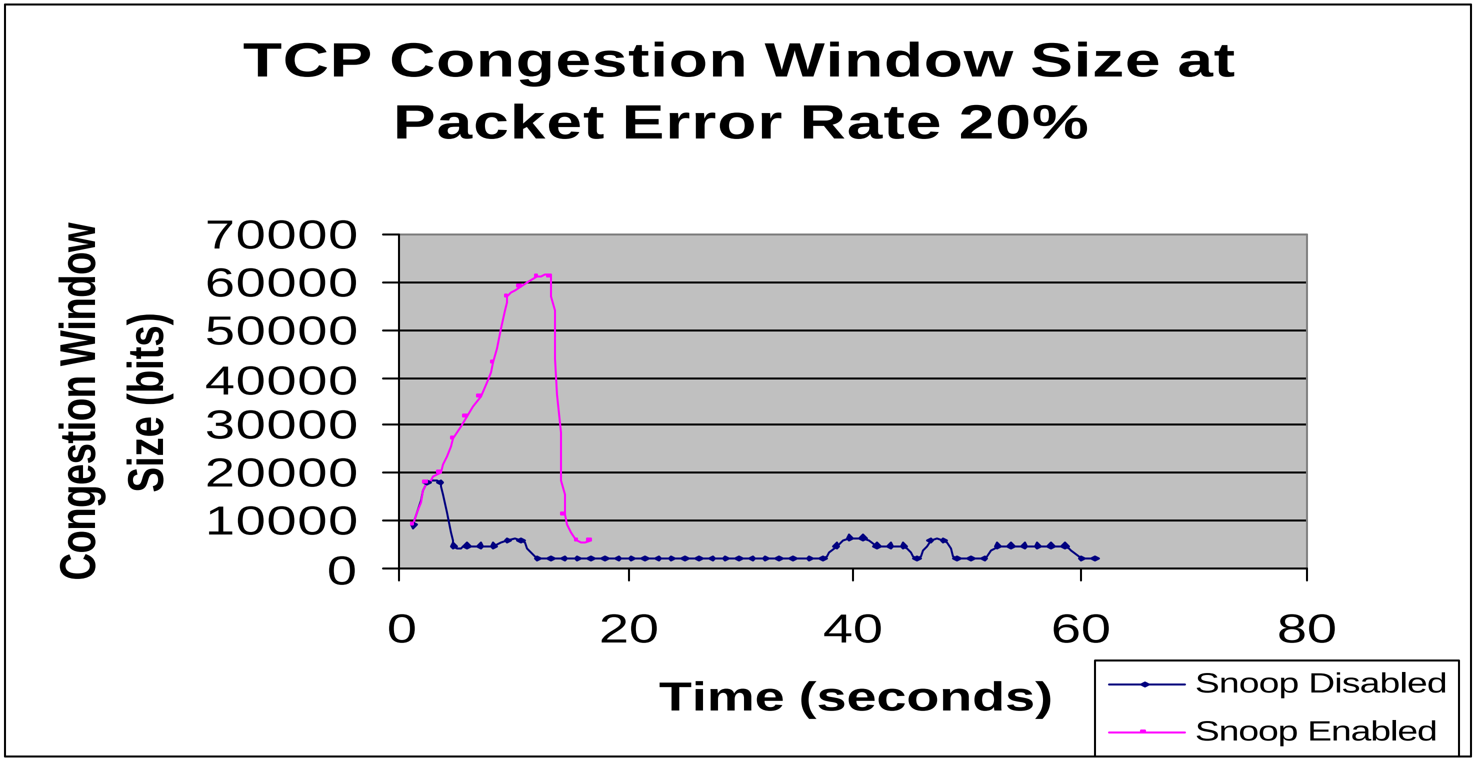
- Upload Response Time



- Improve 68 times at error rate of 30%

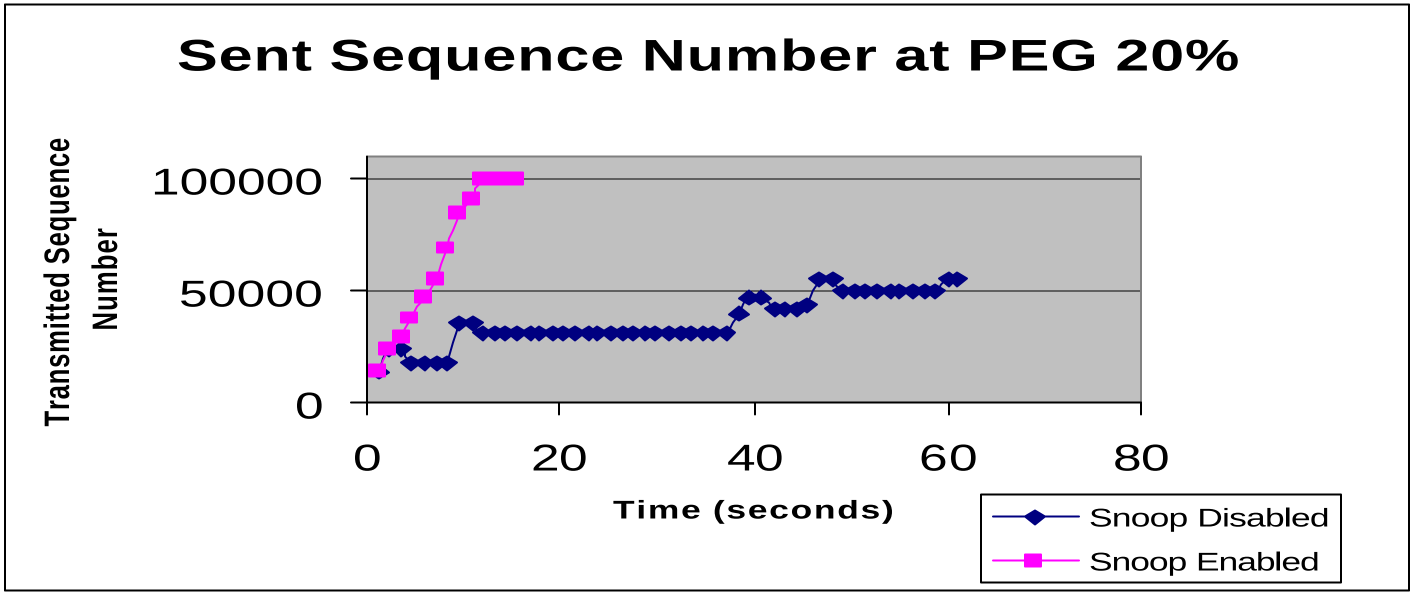
Congestion Window Size

- Congestion Window Size



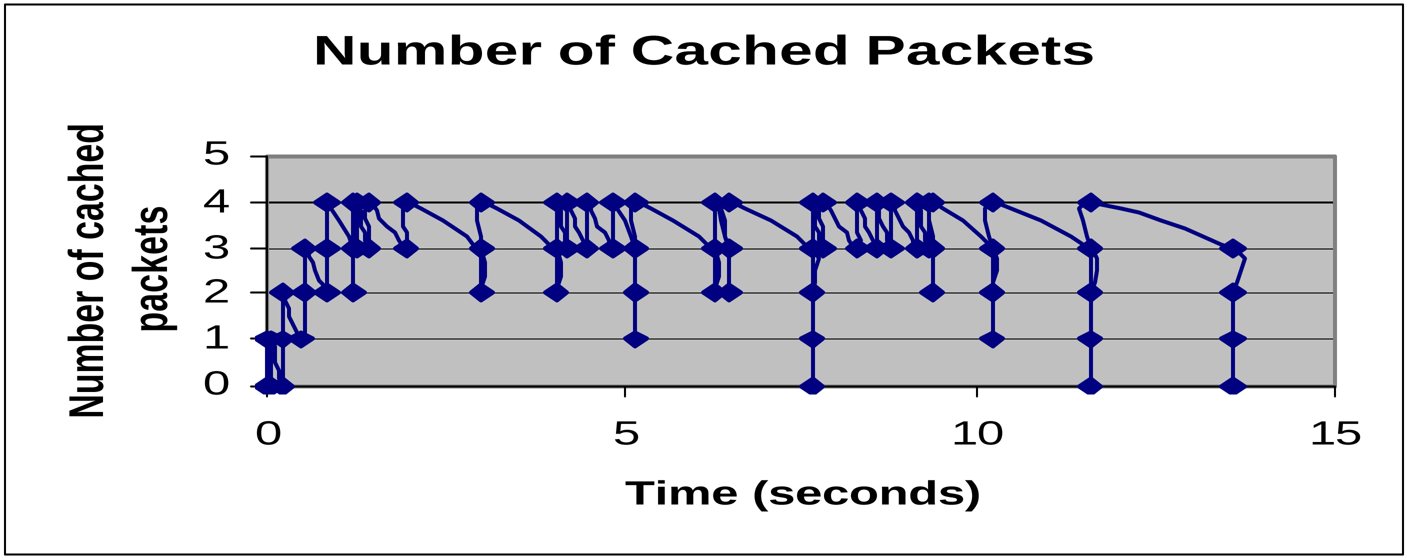
Sent Sequence Number

- Each data byte is represented by a sequence number



Packet Cache

- Study how are the packet cached



- Transmission Window: 4 packets
- Number of packets cached at the end is 0

Difficulty of Project

- Develop Snoop and PEG model from scratch. Need significant amount of time for developing and debugging the code
- Need to place the two models between the ARP and IP nodes. Need to study the source code of ARP and IP node and separate them

Future Work

- Vary the retransmission timer based on calculated round trip delay in the wireless link

References

- [1] IEEE 802.11 Workgroup
- http://grouper.ieee.org/groups/802/11/index.html
- [2] Performance Enchancing Proxy (PEP) Request for Comments (RFC)
- http://community.roxen.com/developers/idocs/drafts/draft-ietf-pilc-pep-04.html
- [3] Improving TCP/IP Performance over Wireless Networks
- http://www.cs.berkeley.edu/~ss/papers/mobicom95/html/mobicom-final.html
- [4] W.Richard Stevens, TCP/IP Illustrated Volume 1, Addison Wesley, Professional Computing Series, 1984
- [5] Andrew S. Tanenbaum, Computer Networks Third Edition, Prentice-Hall Press, 1996
- [6] Wireless LAN Model Description, Opnet Manual.