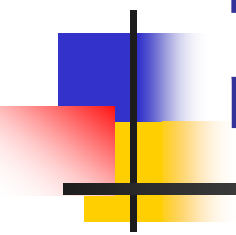
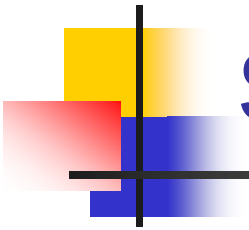


Evaluation of TCP congestion control mechanisms using OPNET simulator

ENSC 835: COMMUNICATION NETWORKS

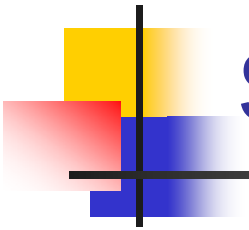
FINAL PROJECT DEMONSTRATION
Spring 2008

Laxmi Subedi
URL: http://www.sfu.ca/~lsa38/project.html
Email: lsa38@cs.sfu.ca



Simulation: Simple client and server

- Client and Server connected with 1.5 Mbps line
- Packet discarder, between server to client link, impose packet loss
- Application is defined
- Profile is created
- File of size 3 MB is transferred from server to client using ftp application
- Run Simulation for 2 minutes (actual simulation time 5 minutes) and collect statistics



Simple Client Server: Scenarios

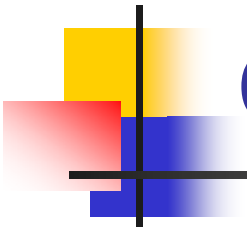
- No packet loss
- One packet loss within 0.5 seconds
- Two packets loss within 0.5 seconds
- Five packets loss within 0.5 seconds
- All packets loss within 0.5 seconds

Simulation: Client and Server with disconnection node

- consist of two subnets with client and server on each
- client and server connected to routers by 100 Mbps link
- routers are connected to Internet cloud by 45 Mbps link
- 150MB is transferred using ftp application
- link between client and router is disconnected via failure recovery node
- Scenarios:
  - 0.05s, 0.1s and 0.2s
  - 10s
- simulated for 10 minutes (actual simulation time 14.5 minutes)

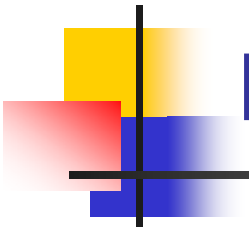
Simulation: Multiple clients and multiple servers network

- consists of multiple subnets with multiple clients and servers
- client and server are connected to subnet router by 100 Mbps link
- router is connected to backbone internet by 45 Mbps link
- database and ftp application are used in server
- profile using server application are run on client
- congested network is simulated
- simulation time 5 minutes (Actual simulation time 1hrs 32 minutes)



Conclusion

- Simple client server model:
  - Reno, SACK, and New-Reno recovers congestion window similarly in case of all and no packets loss
  - SACK performs best for multiple packets loss
  - Reno is the worst among three
- Client and server connected via IP cloud:
  - all algorithms are incapable to differentiate link disconnection.
  - recovery process varies with disconnection interval
  - SACK is better for small disconnection interval
- Multiple client and server model:
  - Conflicting behavior for heavily congested network
  - SACK performs better in lightly congested network



Future Work

- model wireless node with disconnection behavior for detailed analysis
- analyze variation in drop time for more further performance analysis
- investigate to identify the cause of conflicting behavior of Reno, SACK, and NewReno for heavily congested network
- implement new algorithms and compare their performance with these basic algorithms