Gnutella Network Robustness

ENSC427 OPNET Project

Group Members:

Ken Wu
Hao Su

P2P networks

- P2P – 2 or more peers
- Share resources directly with one another
- Each peer is both server and client simultaneously
- File sharing, real time telephony traffic
- “True” P2P vs non-True P2P
  - Napster use client-server structure for file searching
  - Gnutella is a true P2P, all task use P2P structure

Gnutella Network

- Developed by Justin Franel and Tom Pepper of Nullsoft in early 2000
- Facilitated by directory servers to inform peers of network addresses of other peers
- Each peer is called a SERVENT
- Servent send request to all other connected servents, the request is forwarded, until it reaches maximum hops

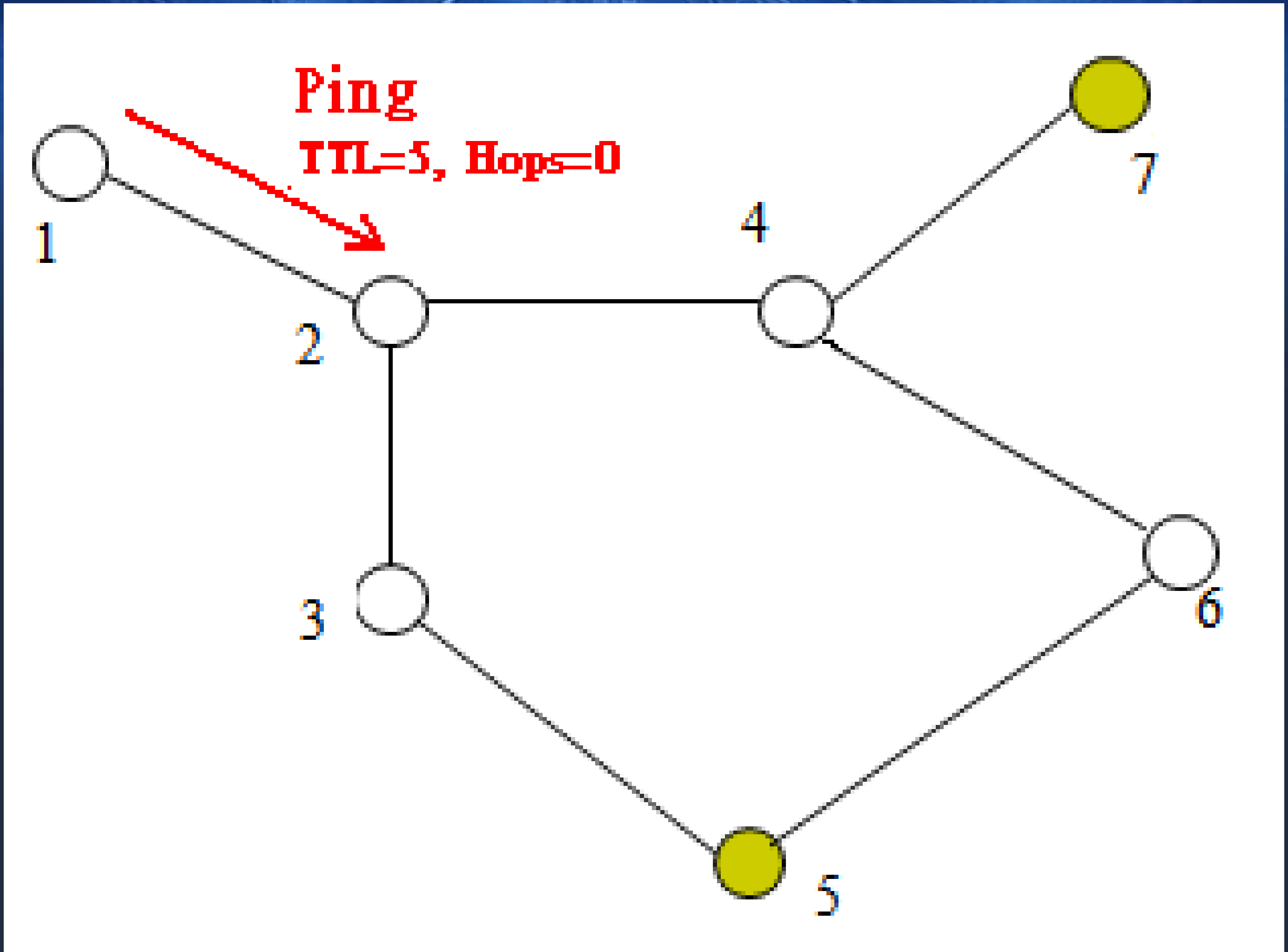
Gnutella descriptors

- 5 descriptors to communicate data:
- Ping – actively discover hosts on the network
- Pong – response to a Ping, includes address of the connected Servent and data sharing information
- Query – search mechanism to find files
- QueryHit – response to a Query, contains information about the file found
- Push - Allows firewalled Servents to share files

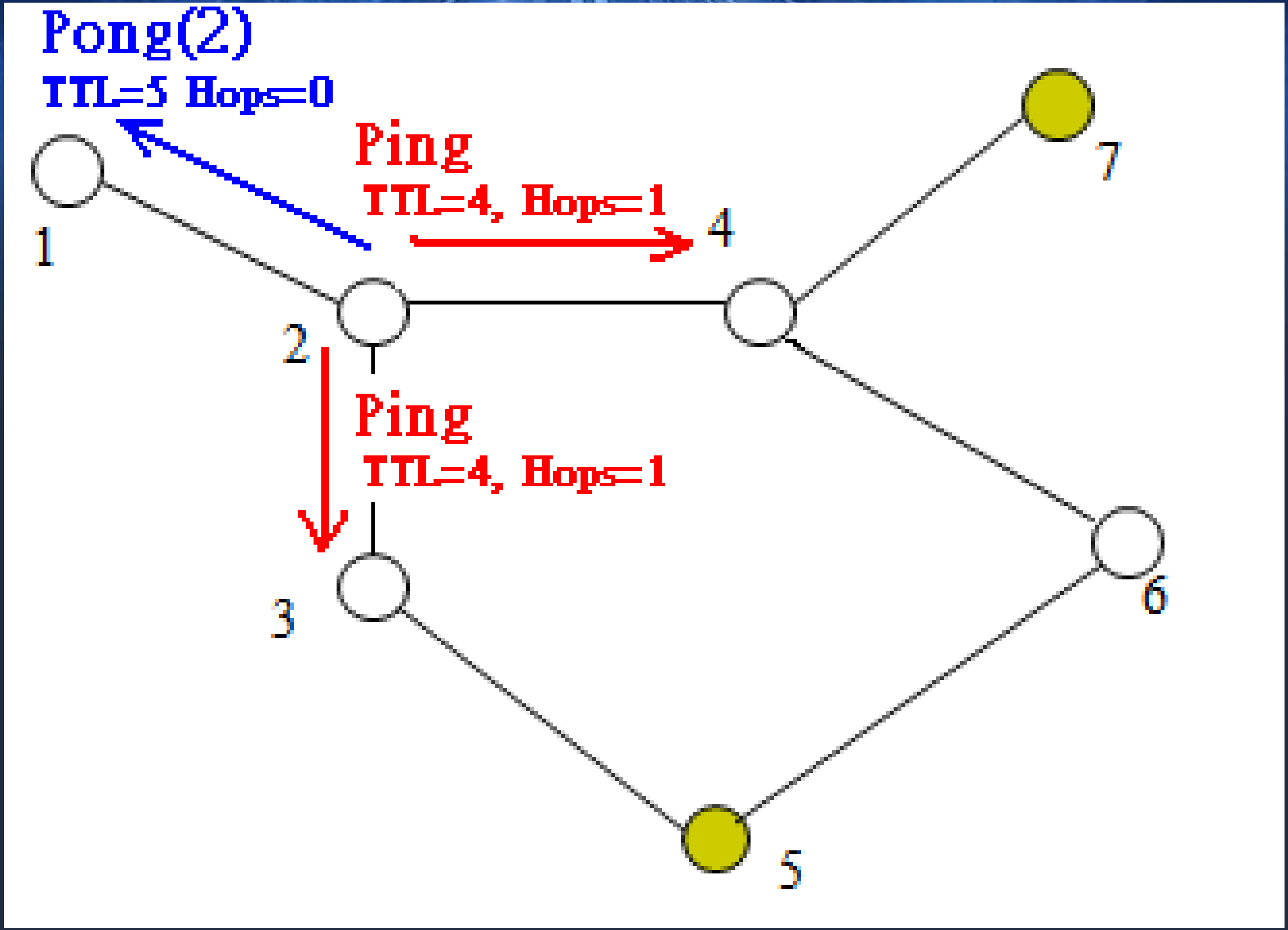
Gnutella Network

- Developed by Justin Franel and Tom Pepper of Nullsoft in early 2000
- Facilitated by directory servers to inform peers of network addresses of other peers
- Each peer is called a SERVENT
- Servent send request to all other connected servents, the request is forwarded, until it reaches maximum hops

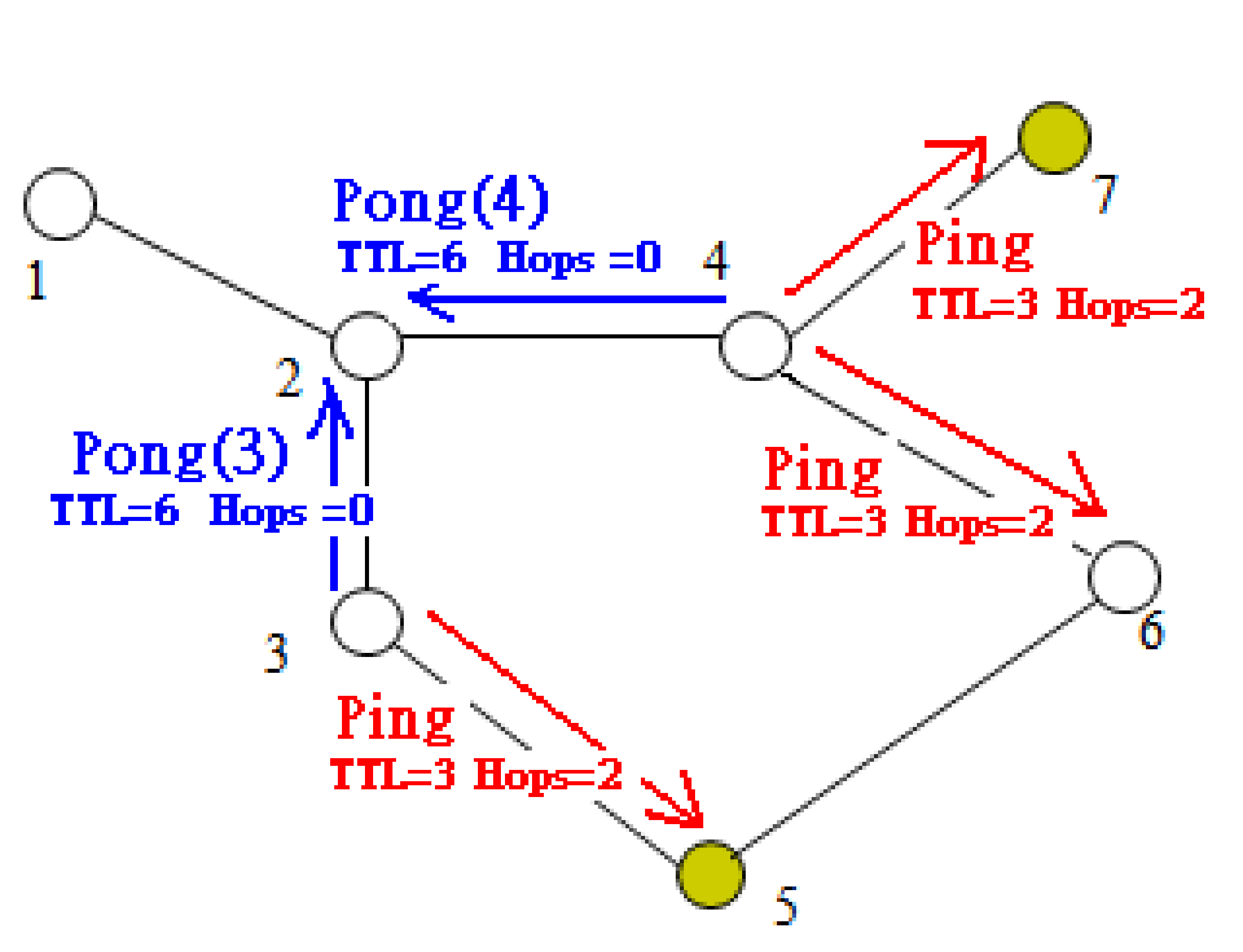
Ping and Pong



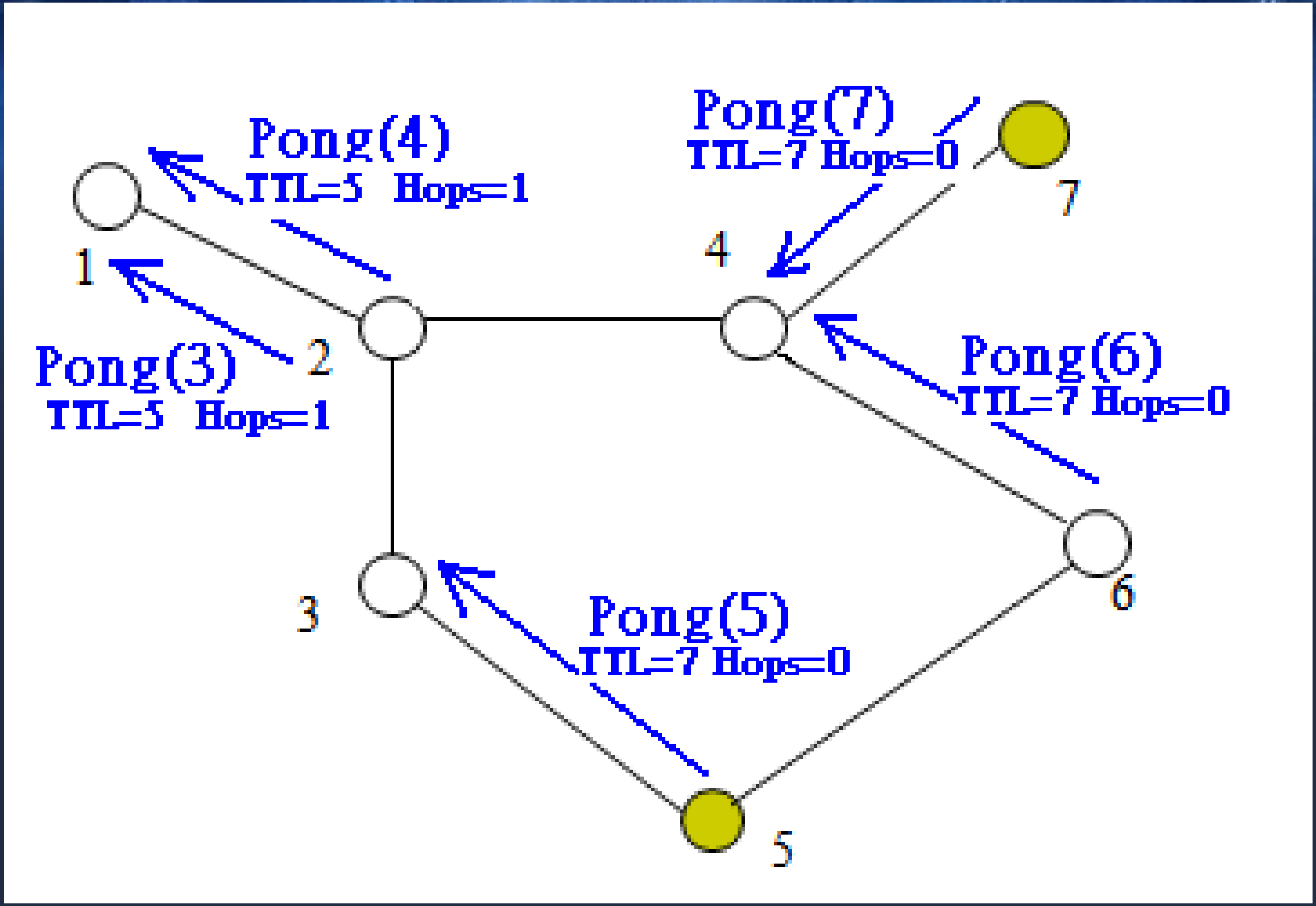
Ping and Pong



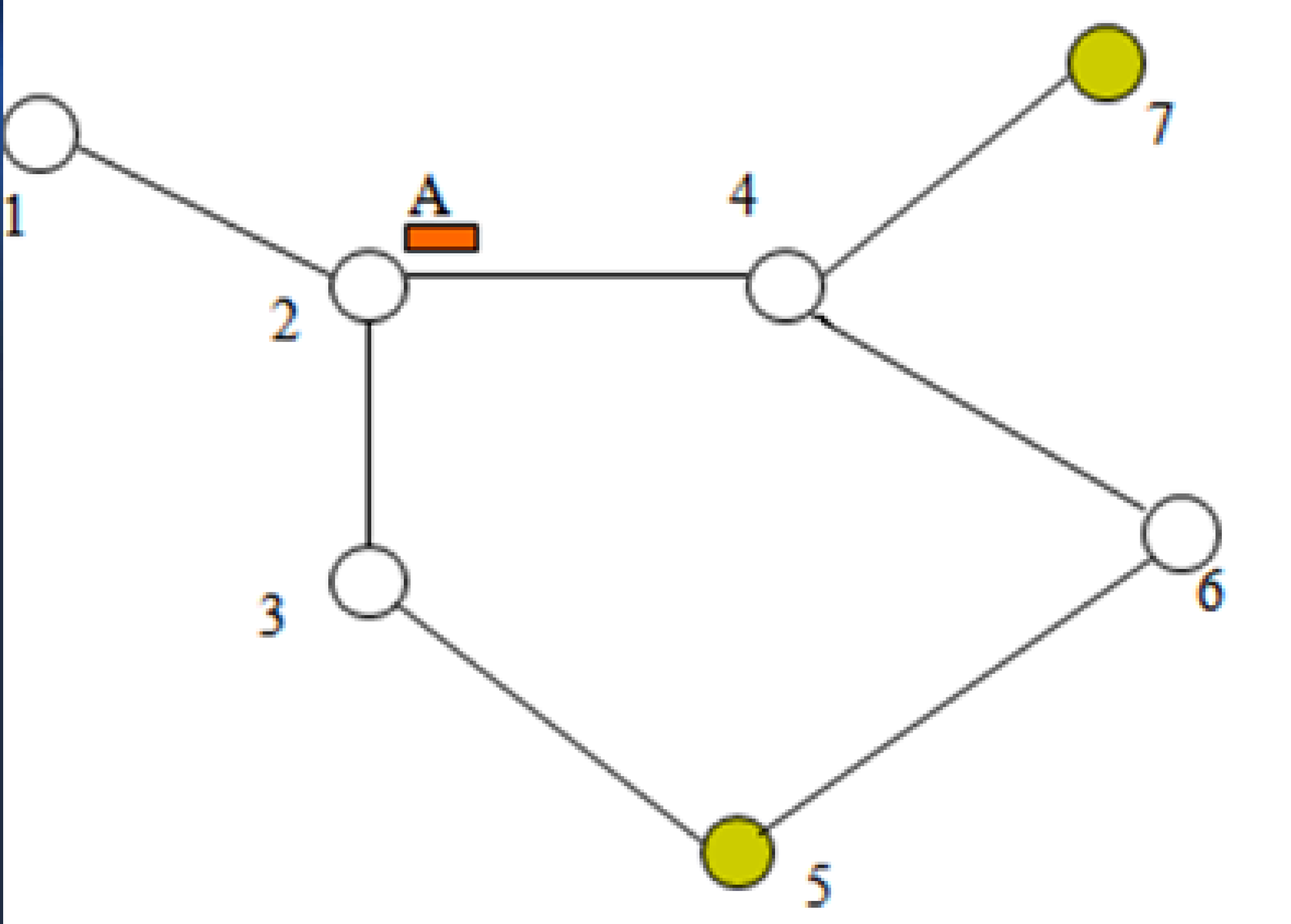
Ping and Pong



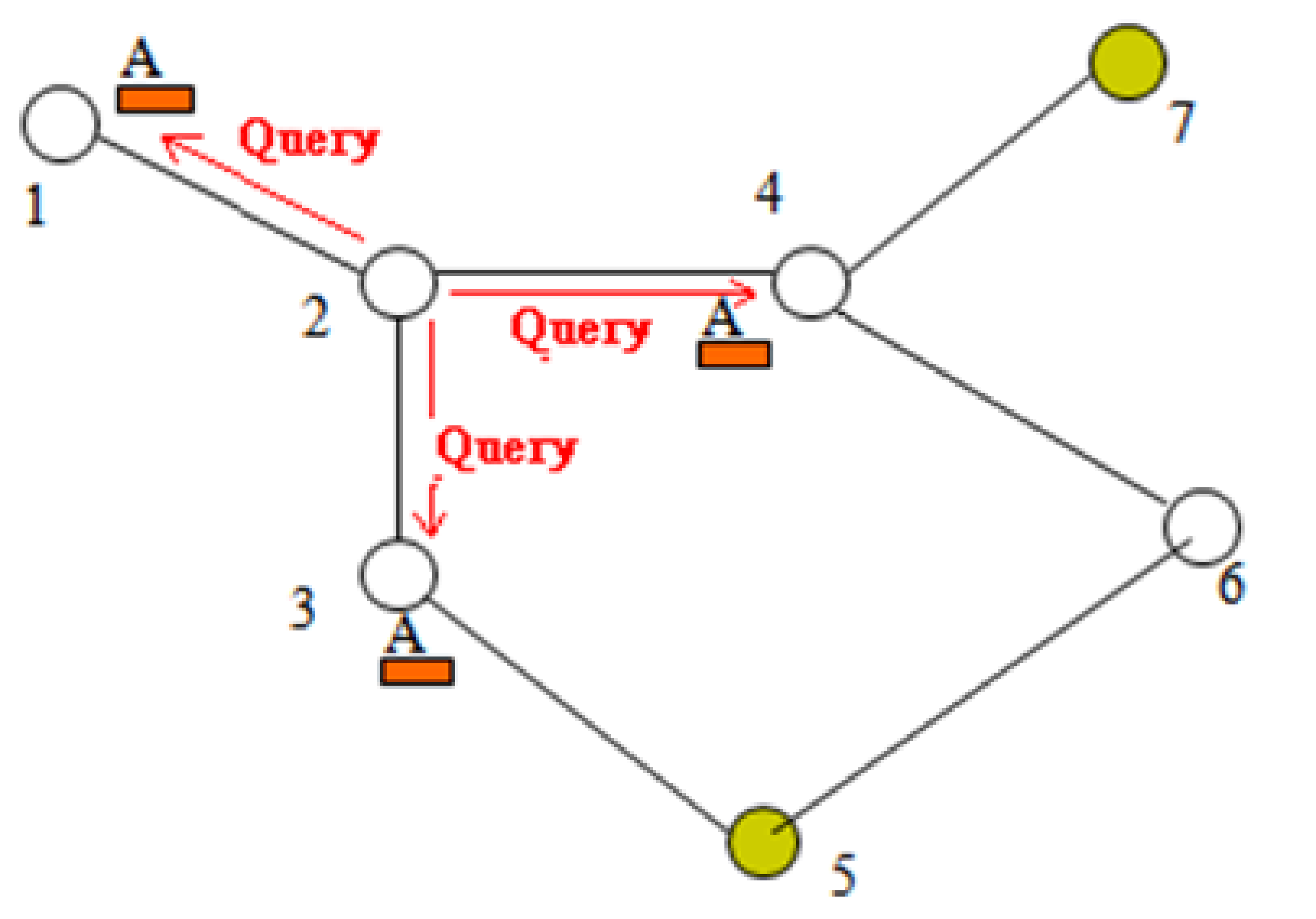
Ping and Pong



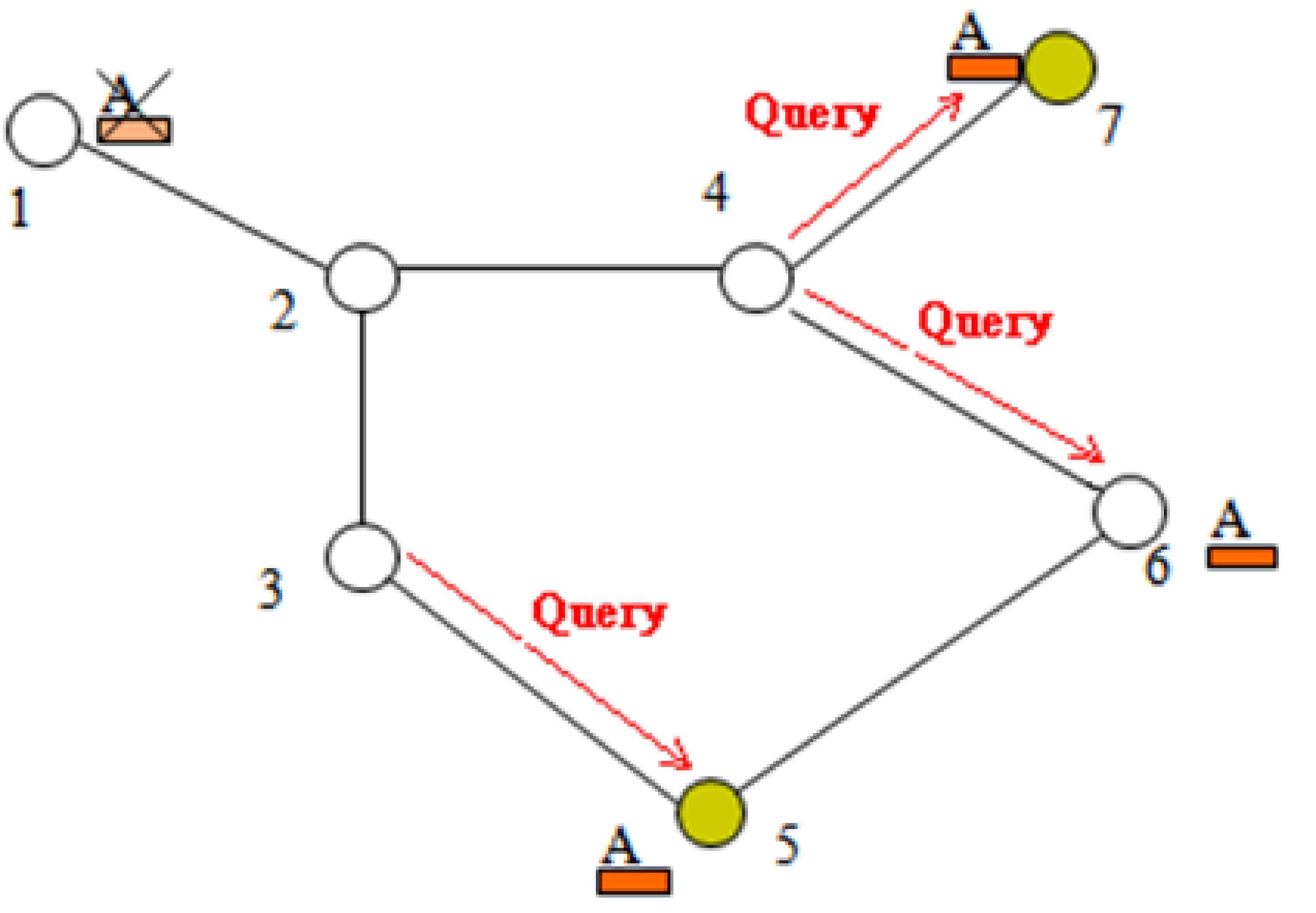
Query and QueryHit



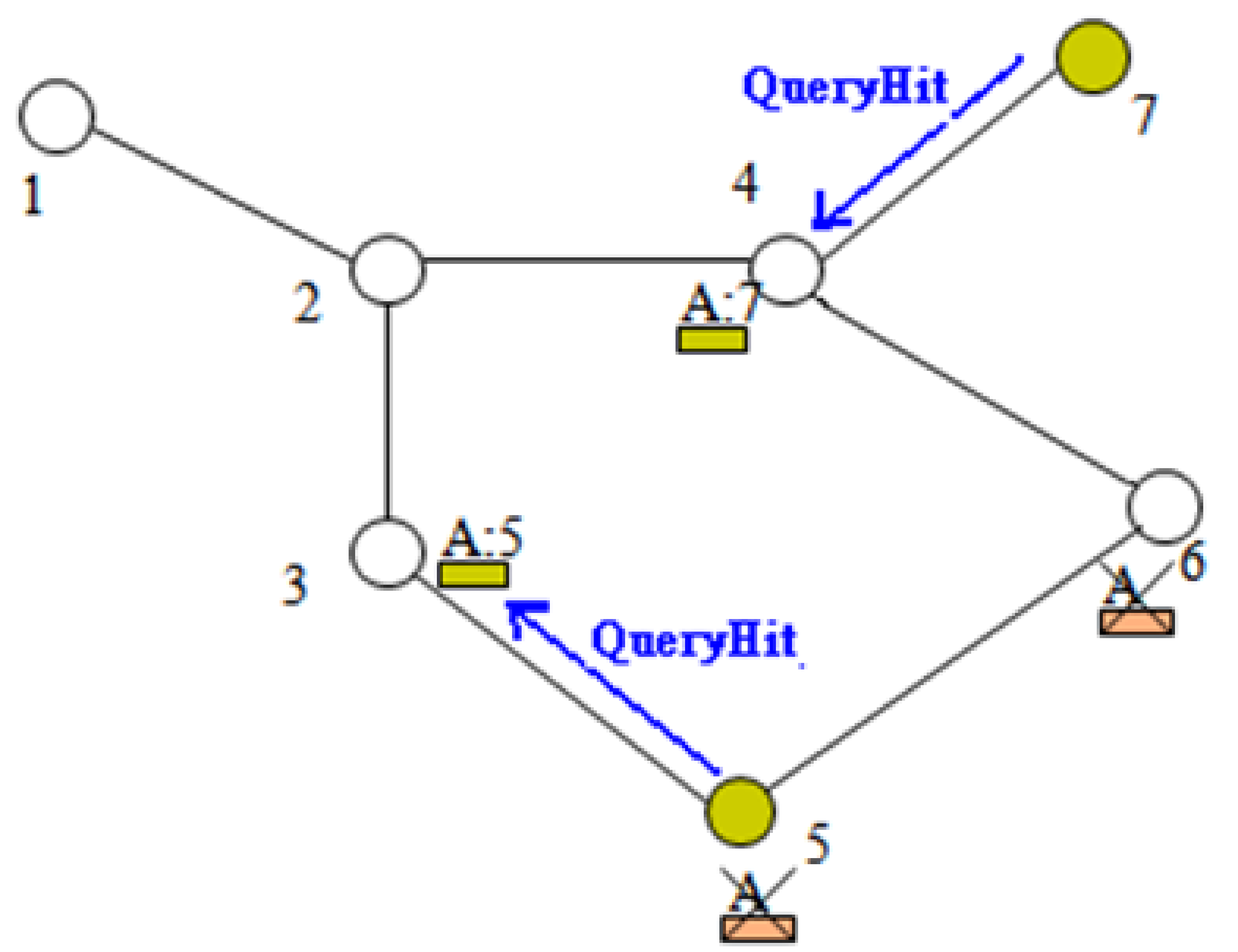
Query and QueryHit



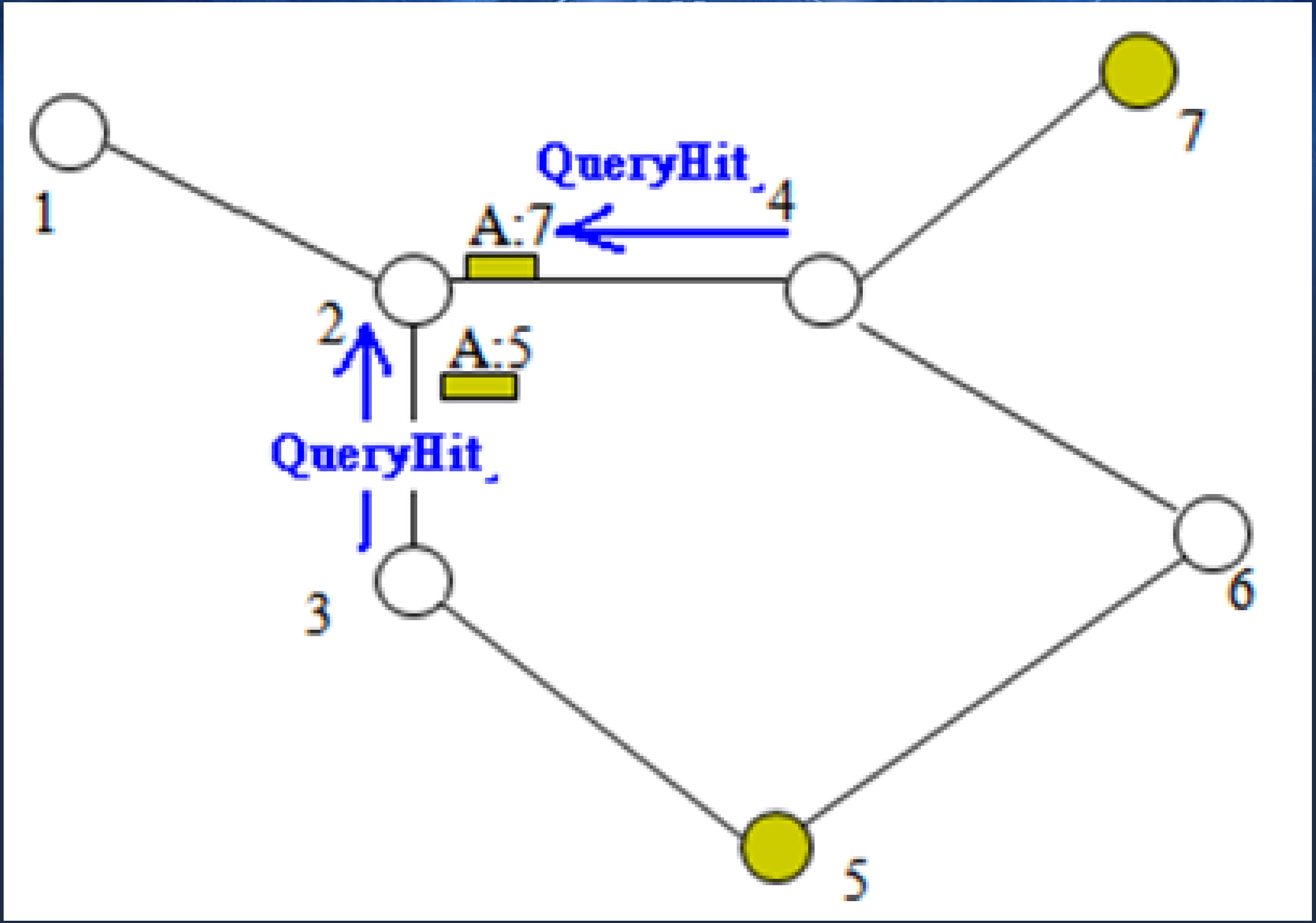
Query and QueryHit



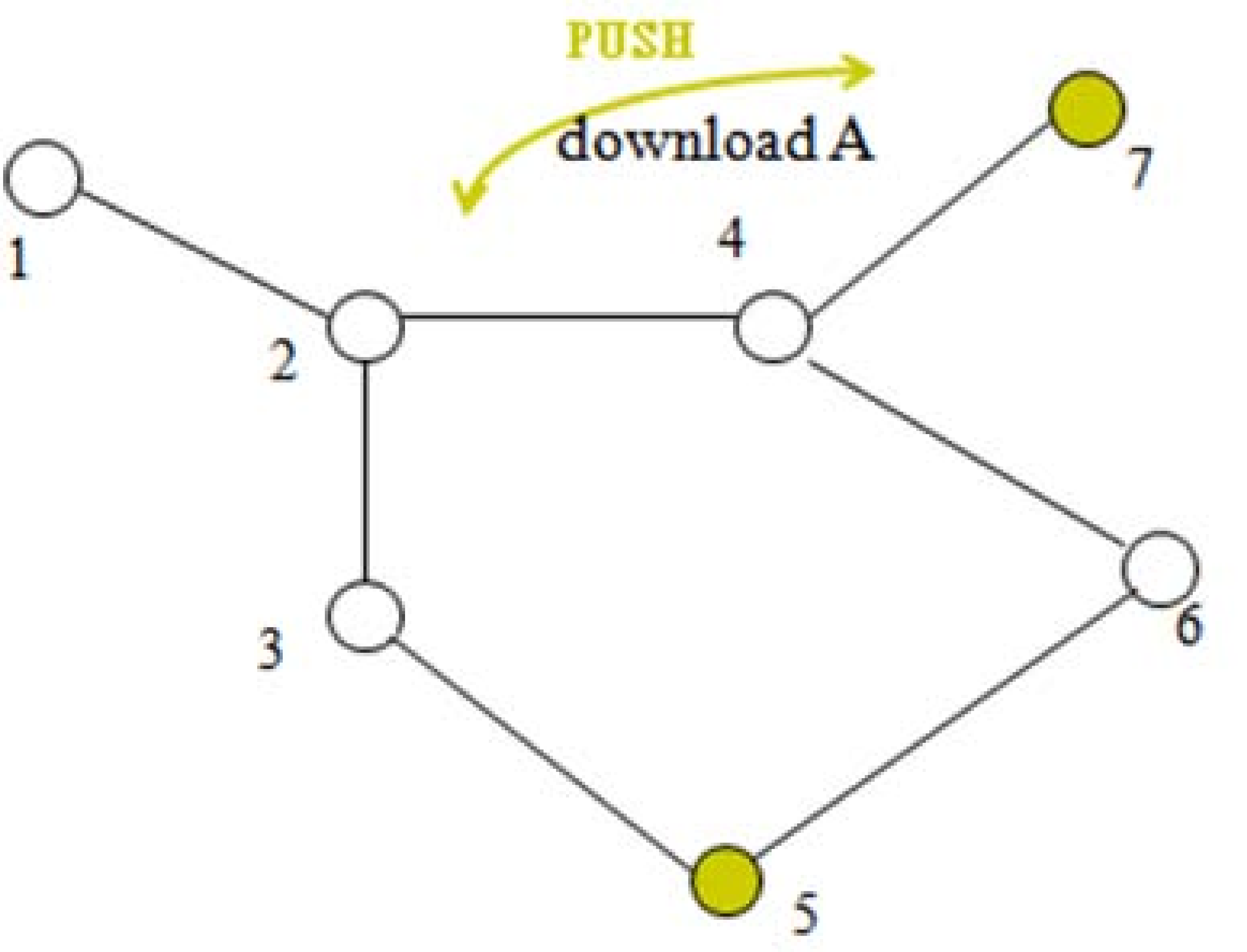
Query and QueryHit



Query and QueryHit



Query and QueryHit



Gnutella optimization problems

- Measures are currently taken to optimize Gnutella search speed and minimize traffic conjestion:
  - Reply to Ping and Query descriptors do not go back the way it came from
  - Duplicate descriptors in the same Servent will be discarded
  - TTL is implemented so that descriptor packets will not propagate network forever.

    Less time to live: travels fewer hops and more difficult to reach destination

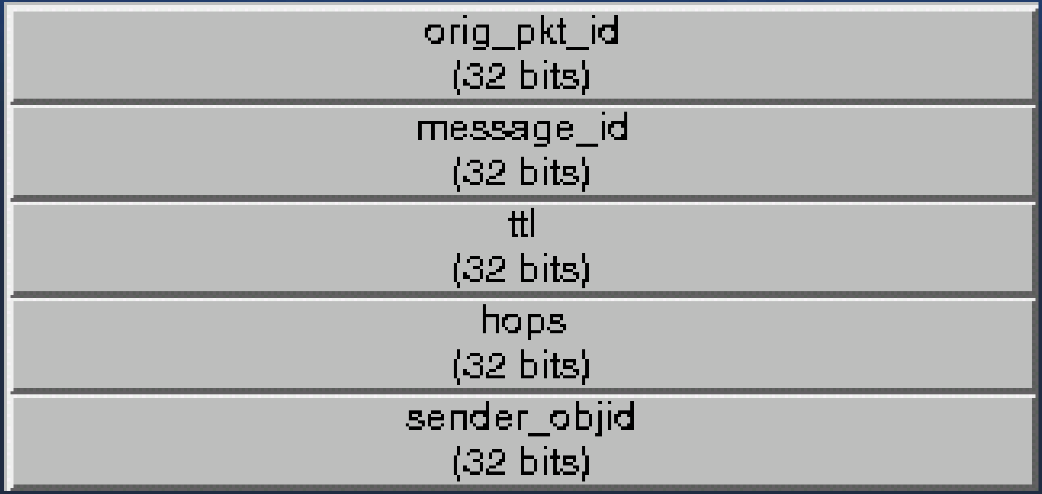
    More time to live: travels further but more likely to be duplicated and congest traffic

Project Scope

- Implement a simplified model of Gnutella network using OPNET modler
- Only Ping and Pong descriptors will be implemented.
- Different topologies with varying number of Servents will be implemented
- Selected Servents within the network will be purposely disconnected to test robustness
- TTL will be varied to maximize robustness and decrease traffic congestion

Packet Format

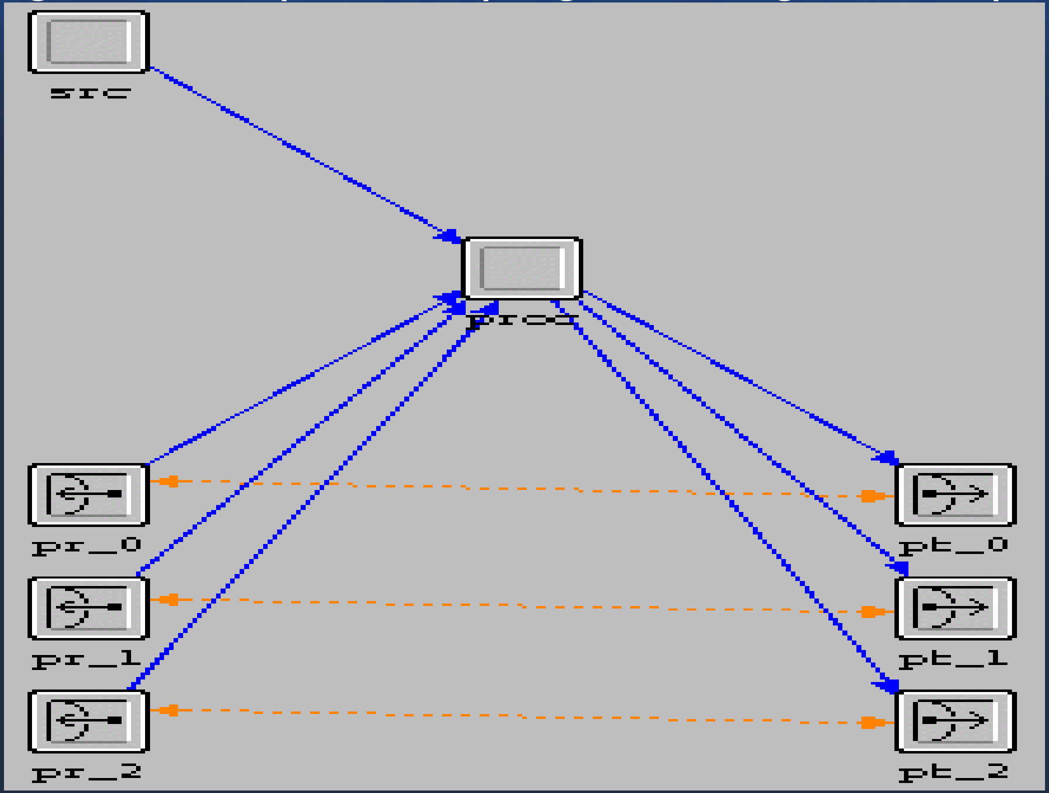
Consisted of original packet id, message id, time to live, hops and sender id



orig_pkt_id (32 bits)
message_id (32 bits)
ttl (32 bits)
hops (32 bits)
sender_objid (32 bits)

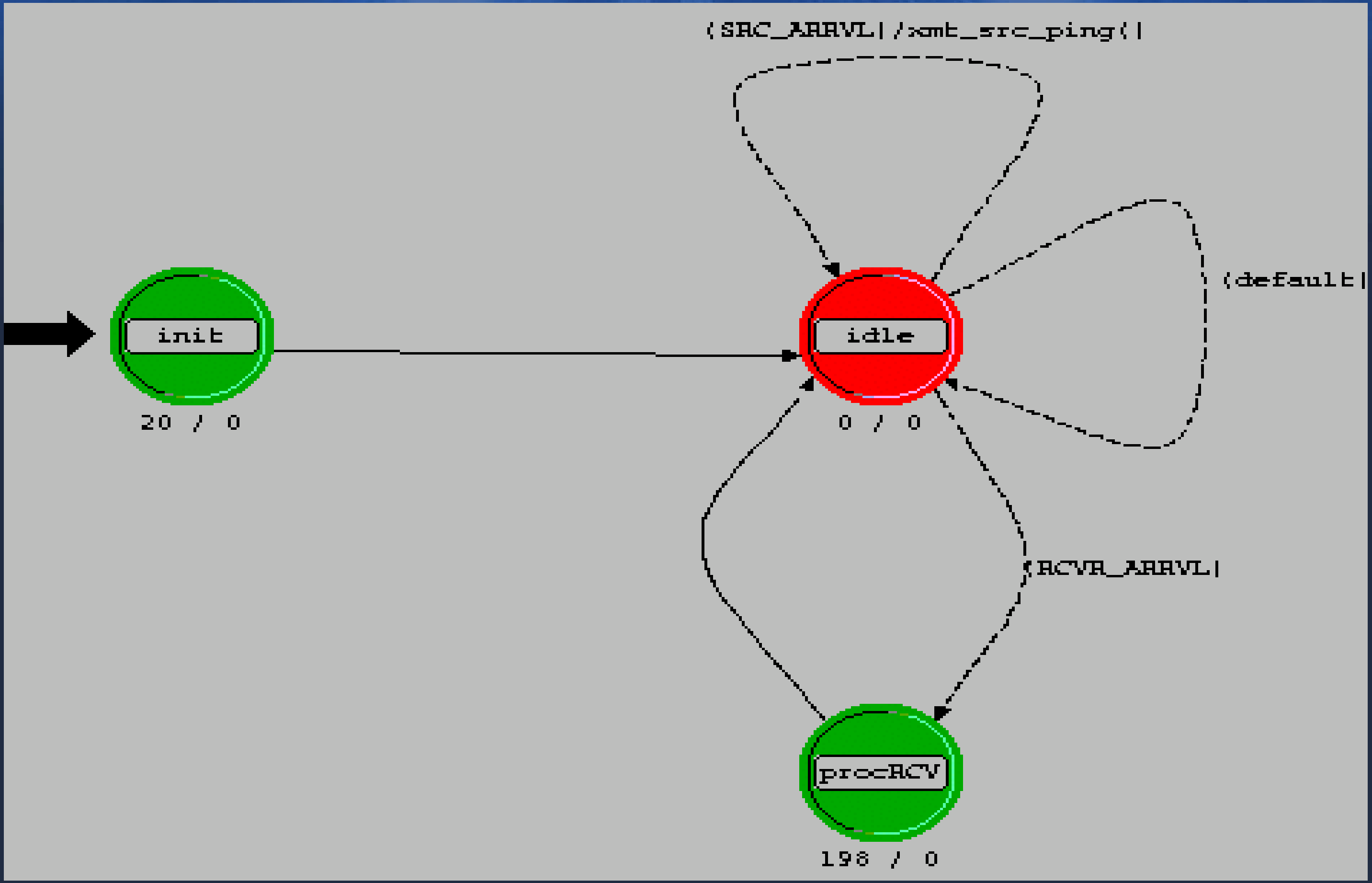
Node Model

- The node model is composted of a source a process model, 3 receivers and 3 transmitters
- Each node model represent one servent - peer/server
- The source generated periodic ping message to the process



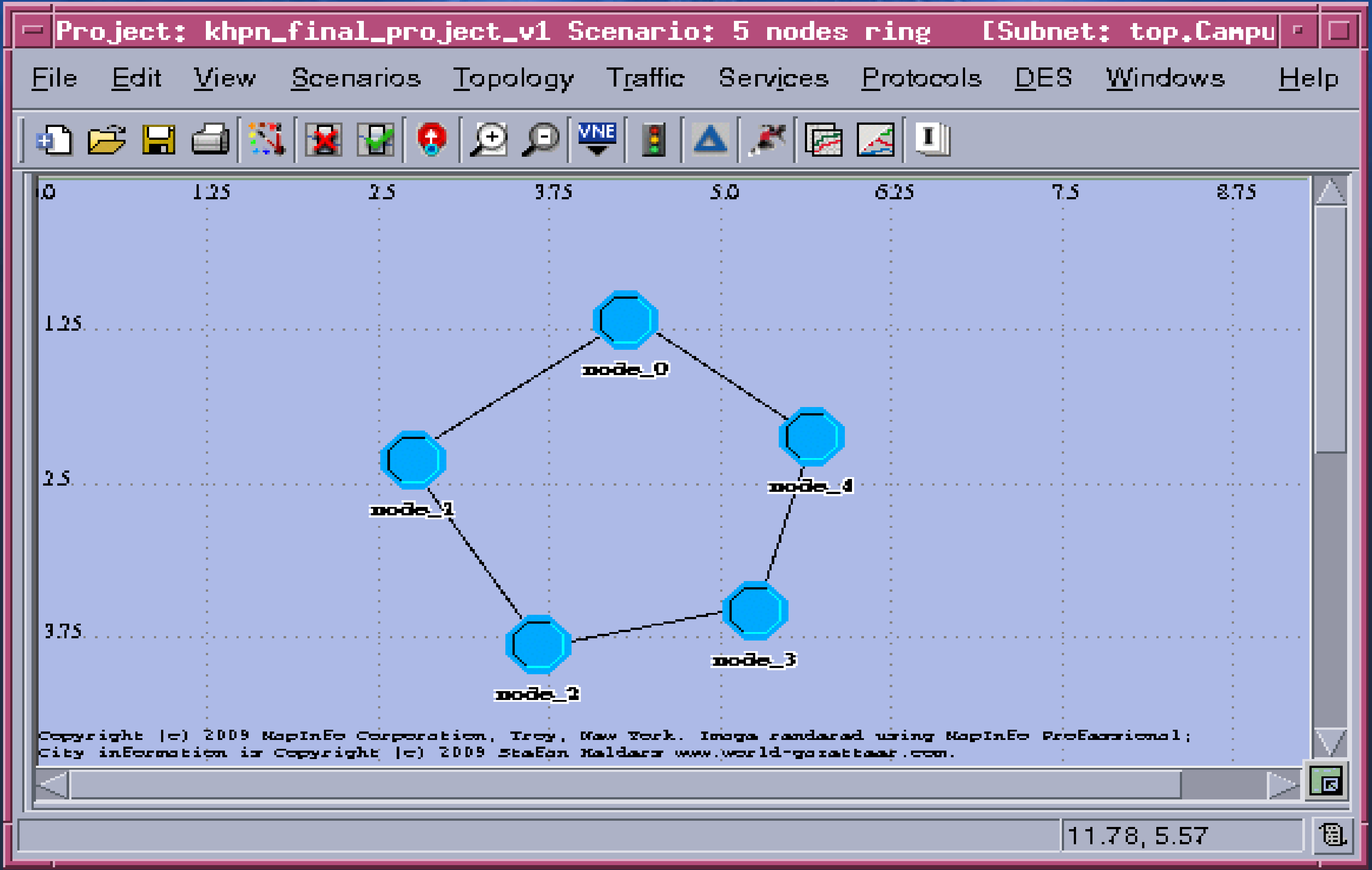
Process Model

- The meat of the process model is in procRCV state
  - When the receiver received the ping message
  - it reply with pong message then forward it to other node if ttl > 1



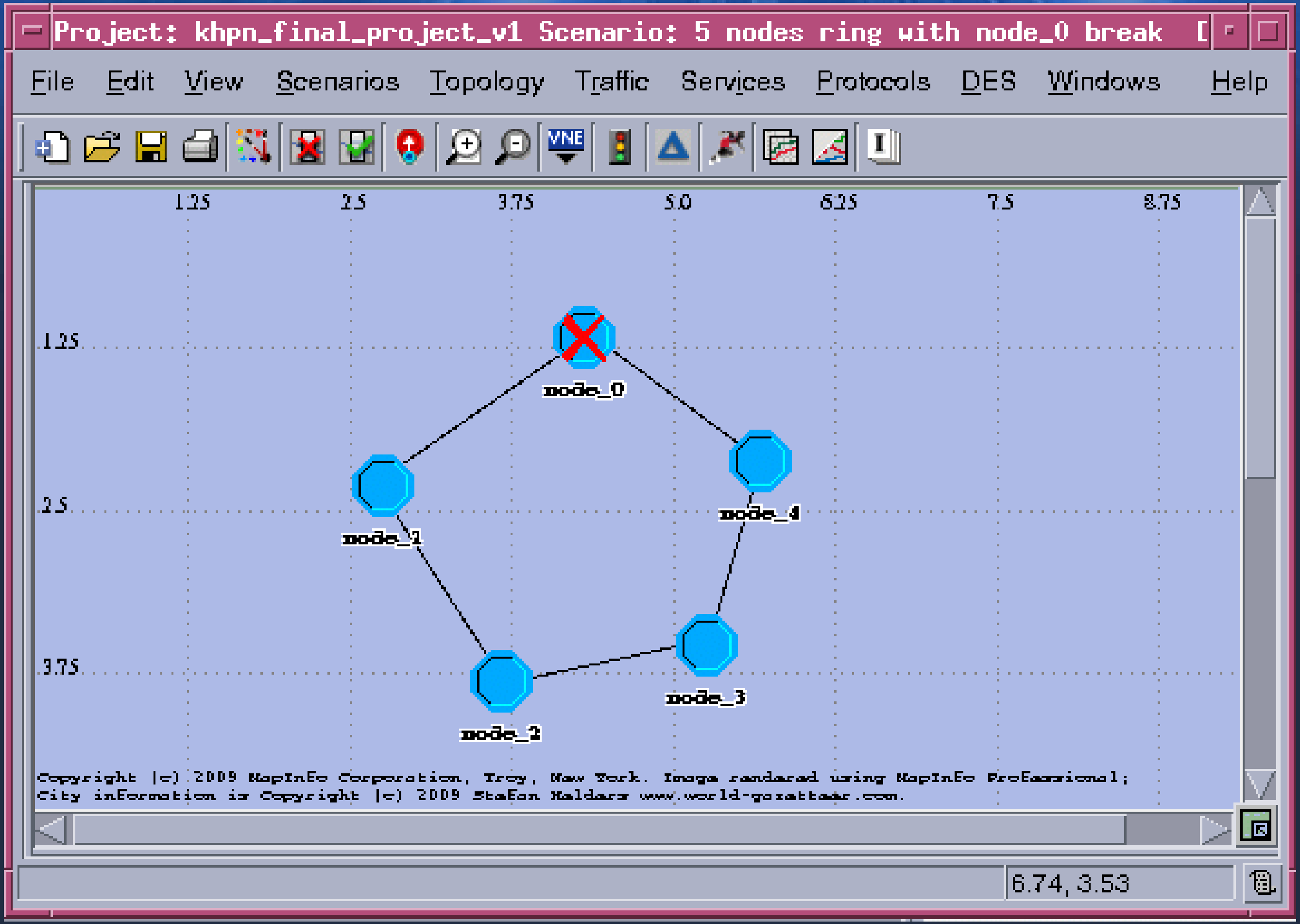
Scenario 1

- 5 peers /server model without failed node



Scenario 2

- 5 peers /server model fail node 0



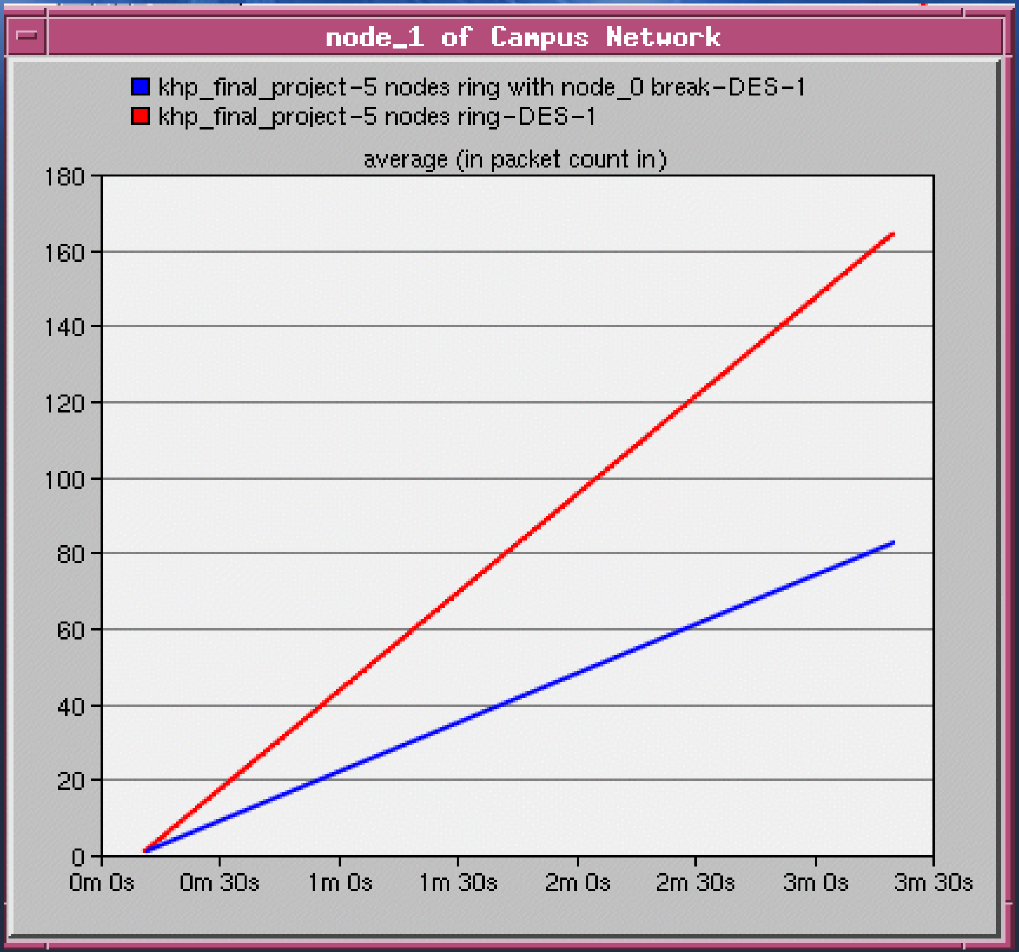
Project: khpn_final_project_v1 Scenario: 5 nodes ring with node_0 break [

Copyright (c) 2009 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2009 Stefan Helders www.world-gazetteer.com.

6.74, 3.53

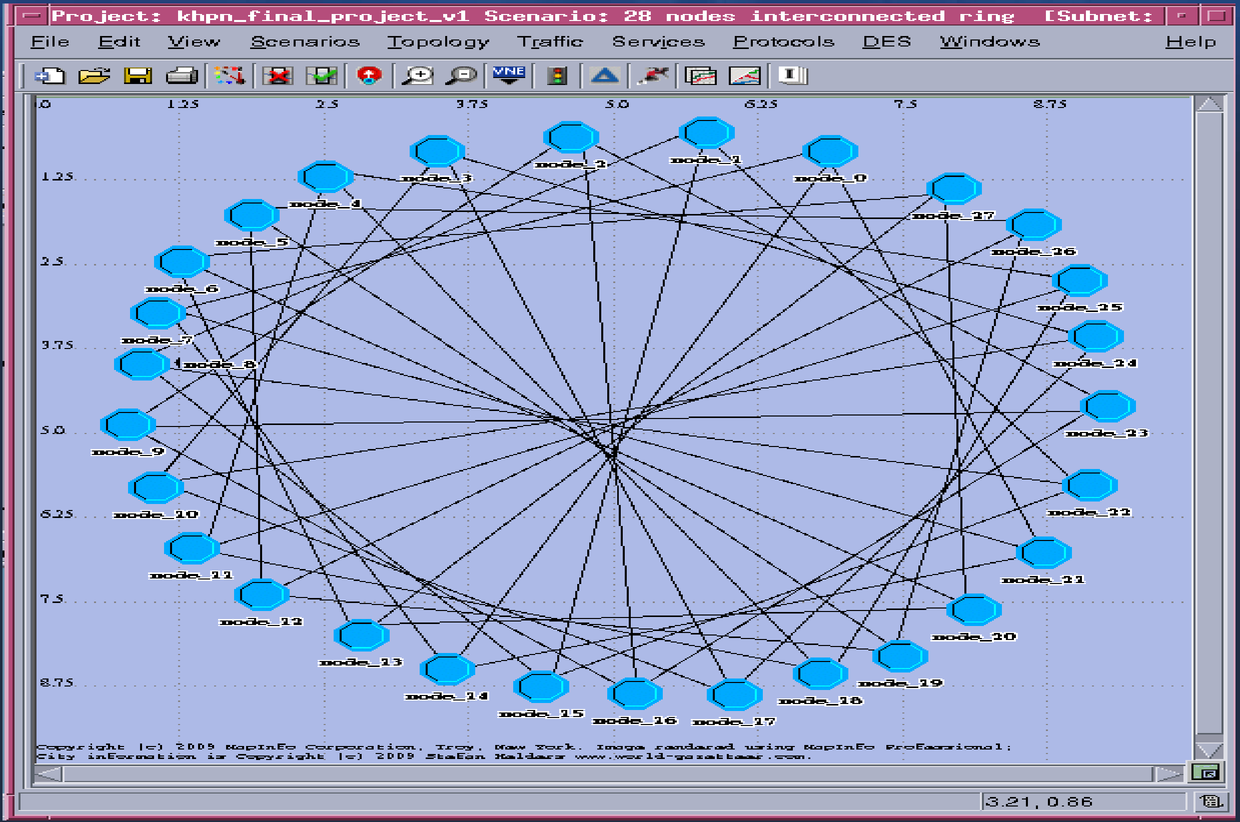
Test Result from node 1

- Packet count in - comparison between the two scenarios
- Red – no failed node
- Blue –failed node 0
- Red > Blue



Scenario 3

- 28 node ring topology
- No failed nodes
- Each node connected to 3 other nodes

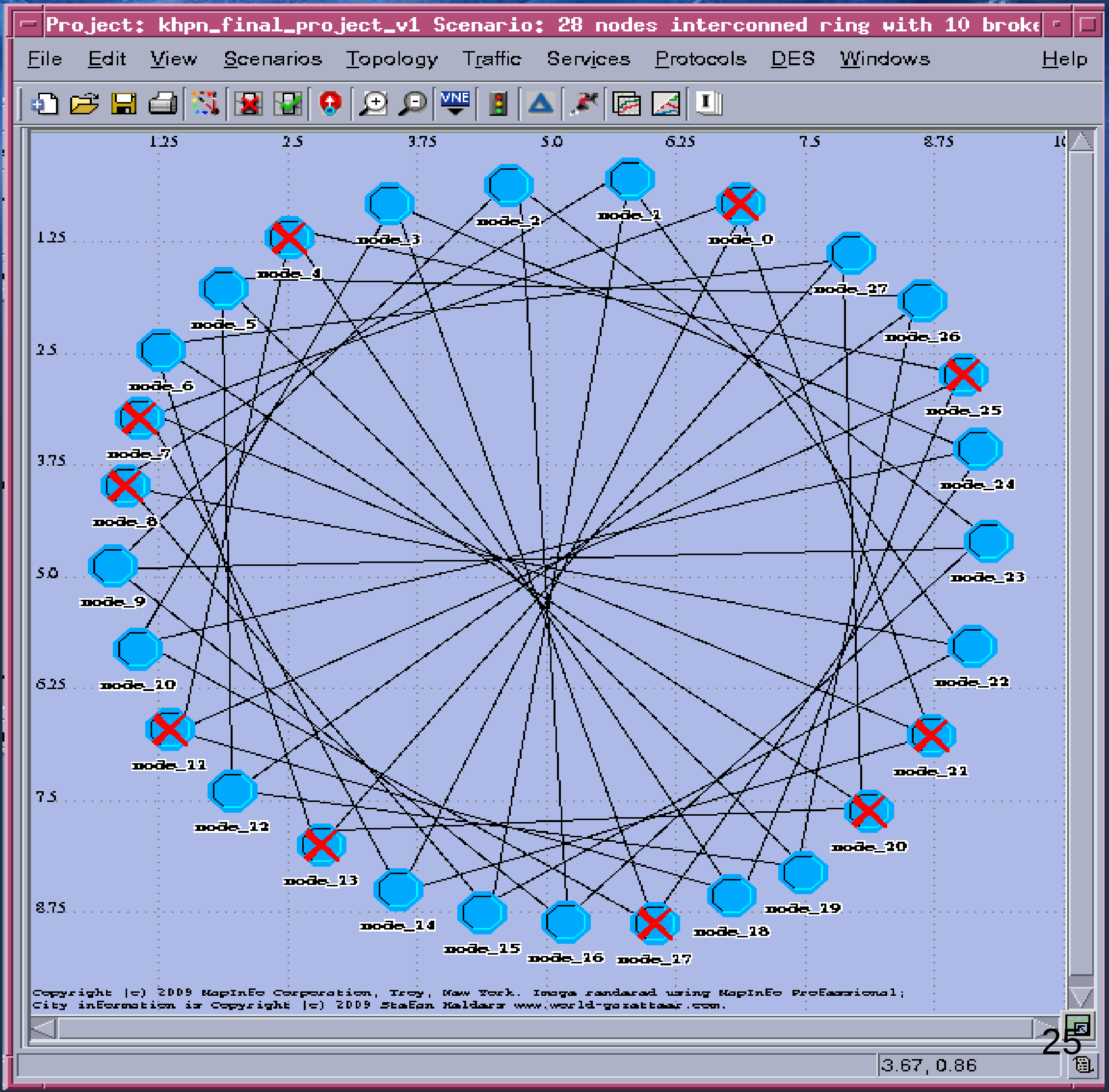


Project: khpn_final_project_v1 Scenario: 28 nodes interconnected ring [Subnet:

Copyright (c) 2005 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional; City information is Copyright (c) 2009 Stefan Helders www.world-gazetteer.com.

Scenario 4

- 28 node ring topology
- 10 failed nodes

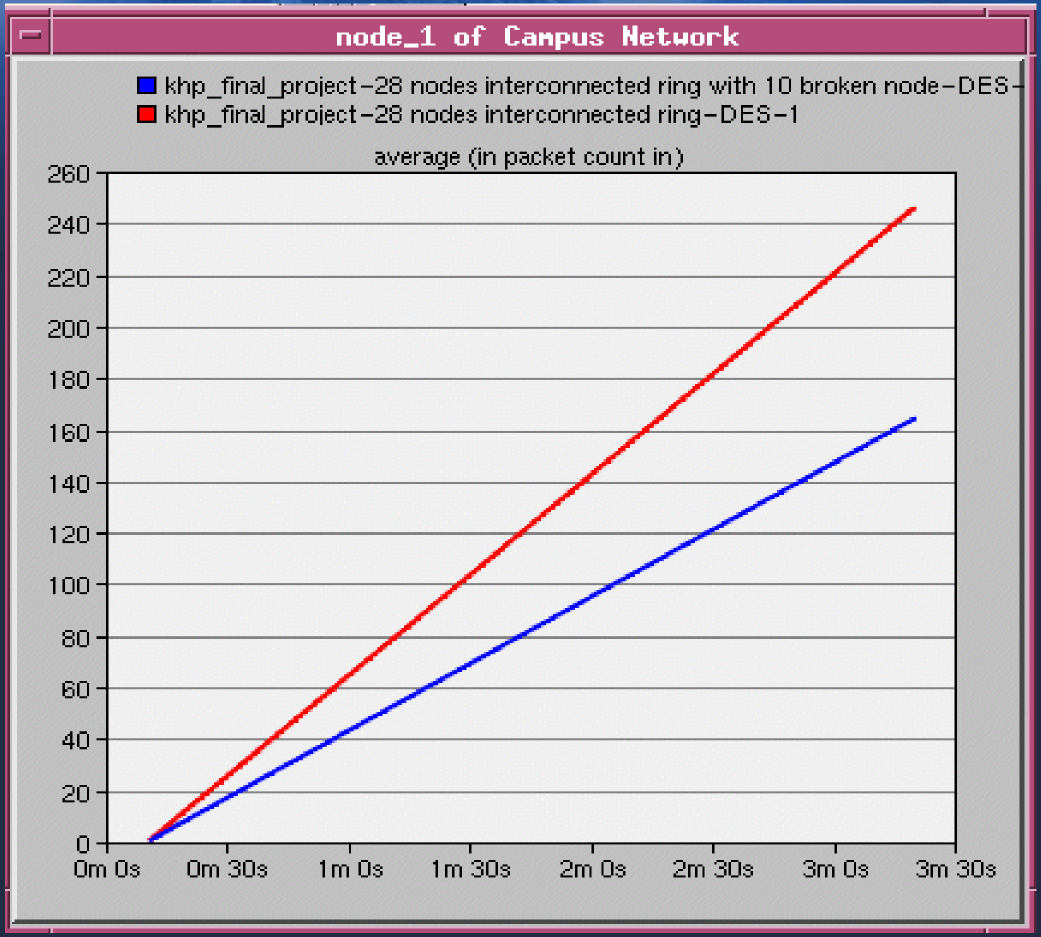


Project: khpn_final_project_v1 Scenario: 28 nodes interconned ring with 10 broke

Copyright (c) 2009 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional; City information is Copyright (c) 2009 Stefan Helders www.world-gazetteer.com.

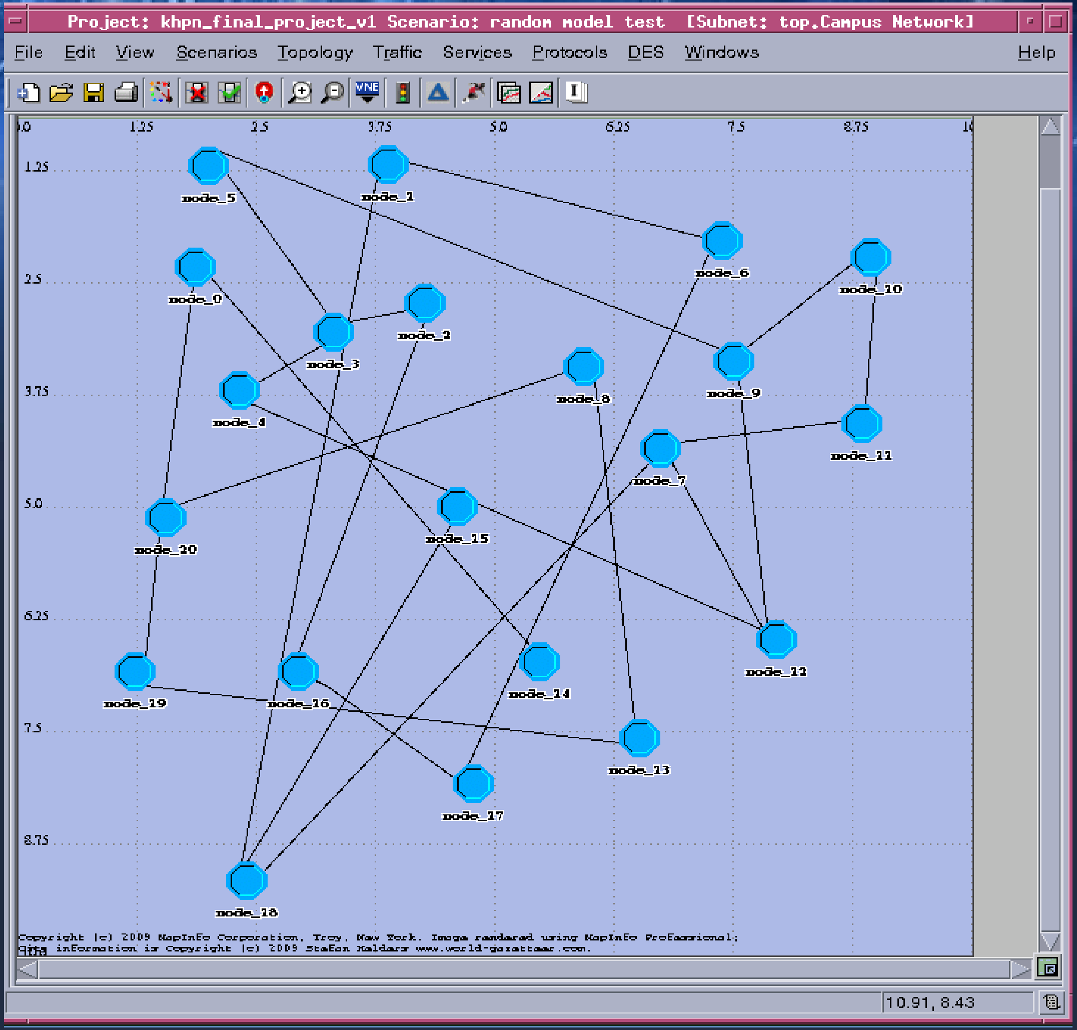
Test Result from node 1

- Packet count in - comparison between the two scenarios
- Red – no failed node
- Blue –failed node 0
- Red > Blue



Scenario 5

- Random model
- Not entirely realistic

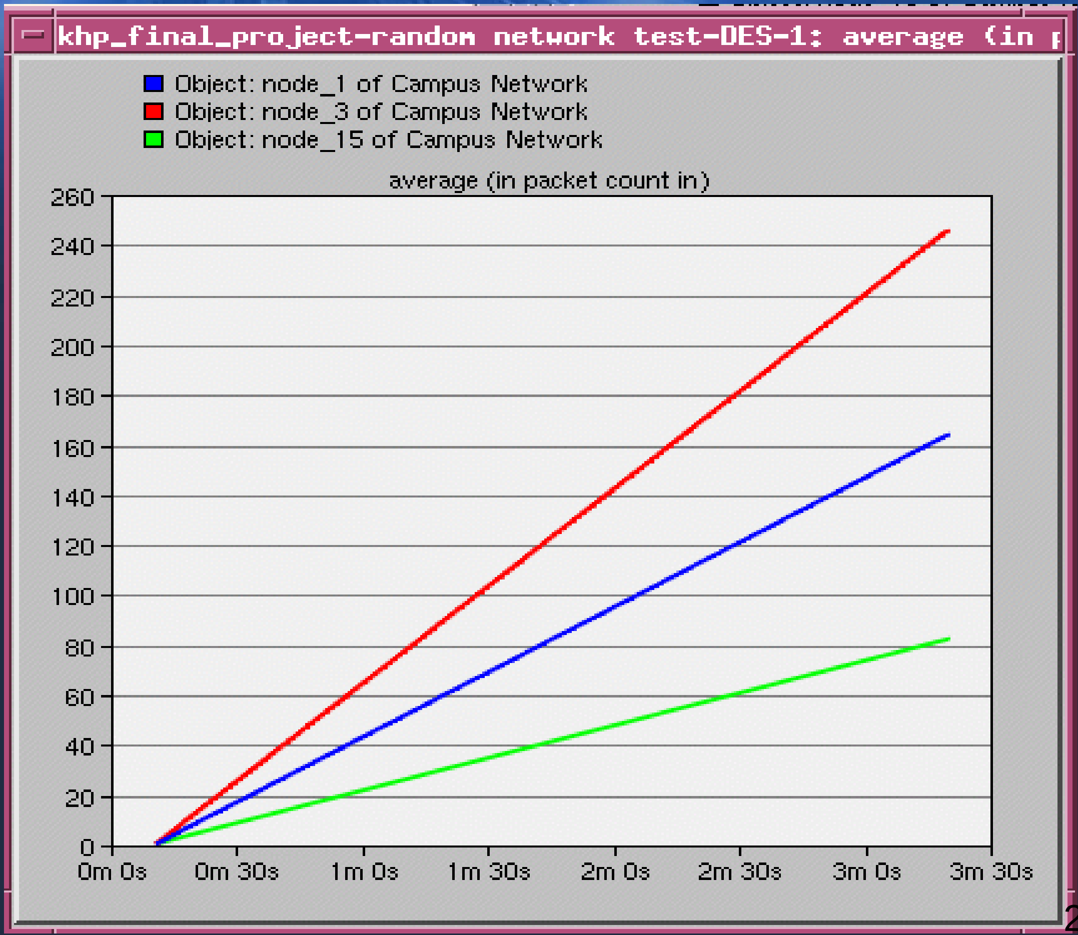


Project: khpn_final_project_v1 Scenario: random model test [Subnet: top.Campus Network]

Copyright (c) 2009 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2009 Stefan Helders www.world-gazetteer.com.

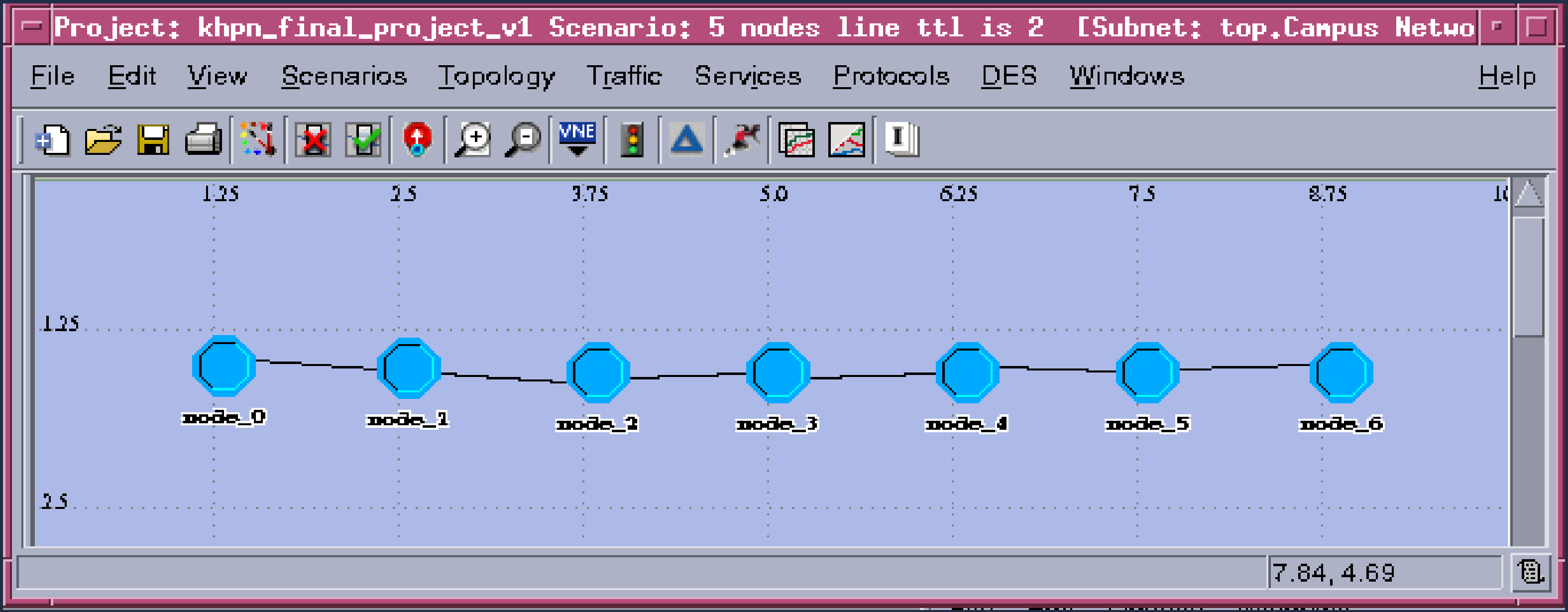
Test Result from node 1, 3 and 15

- Packet count in - comparison between the two scenarios
- Red – node 3
- Blue – node 1
- Green- Node5
- node3>node1>node5



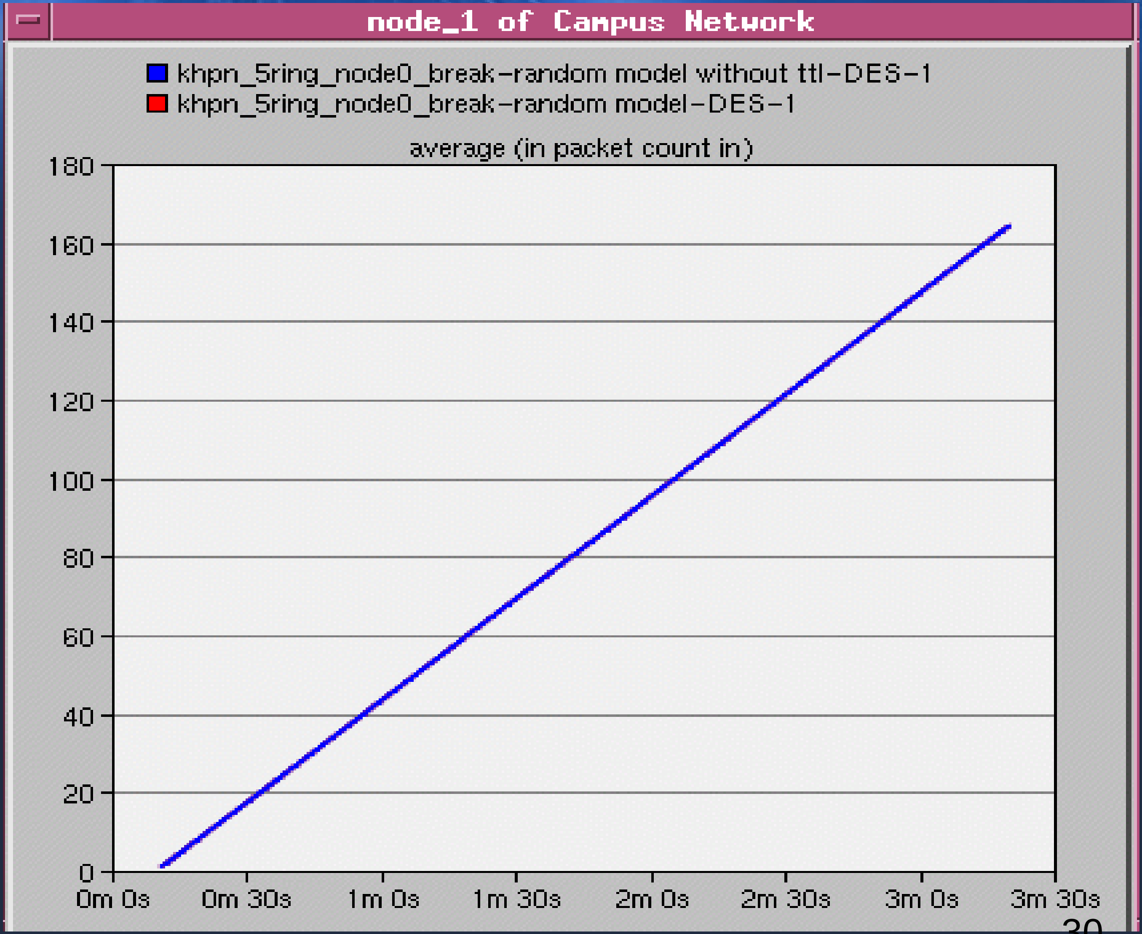
Scenario 6 and 7

- Line topology
- Test two TTL values, TTL=2 and TTL=5



Test Result from node 1

- Packet count in - comparison between the two scenarios
- Red – TTL=2
- Blue – TTL=5
- Same results



Conclusion

- Gnutella Very robust
  - Hard to completely disconnect a node unless all nodes around it are failed
- When IAT is very short, network topology has small effect on number of packets
  - Packets do not have time to “die out”
- Effects of TTL on packet received inconclusive
  - Need longer IAT to see an effect?

Future Work

- Create new type of node model that only pass on packet generated by other nodes but does not generate packets itself and use both type of node model in same network

- Change inter-arrival time to longer, so TTL effects can be seen

- Include Query, Queryhit and Push descriptors

- simulate Gnutella v0.6

- Bigger, more complex and realistic topologies

References

- [1] D. Andre, "Peer-to-Peer Networks as Content Distribution Networks," *ensc.sfu.ca*, report. Fall 2003. [Online]. Available: http://www.ensc.sfu.ca/~ljilja/ENSC835/Fall03/Projects/dufour/Report.pdf. [Accessed: Mar. 06, 2009].
- [2] “BitTorrent,” Mar. 10, 2008. [Online]. Available: http://wiki.limewire.org/index.php?title=User_Guide_Bittorent. [Accessed: Feb. 06, 2009]
- [3] E. Eman, “Scalability and Robustness of the Gnutella protocol,” *ensc.sfu.ca,* report. Spring 2006. [Online]. Available: http://www.sfu.ca/~eelghone. [Accessed: Feb. 06, 2009].
- [4] T. Kelvin, “Examination of Routing Algorithms in Distributed Hash Tables (DHTs) for Peer-to-Peer (P2P) Network,” *ensc.sfu.ca*, report. Spring 2008. [Online]. Available: http://www.sfu.ca/~kta18/ENSC835ProjectReport.pdf. [Accessed: Feb. 13, 2009].
- [5] T. Klingberg, “Gnutella 0.6,” *murdoch.edu.au*, report. June 2002. [Online]. Available: http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html. [Accessed: Feb. 13, 2009].
- [6] W. Stephanie, “How Kazaa Works,” *computer.howstuffworks.com*, document. [Online]. Available: http://computer.howstuffworks.com/kazaa3.htm. [Accessed: Feb 13, 2009].