



UNIVERSITY OF
STIRLING



Modelling the EpiChord P2P Overlay in an XCAST enabled Network

Mario Kolberg

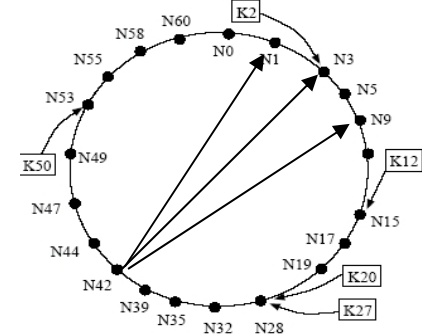
University of Stirling

Part of this work has been presented at IEEE ICC 2007, Glasgow, UK.



- **Peer-to-Peer (P2P)**

- Overlay – build on top of the IP network
- **Nodes** in the overlay are connected by **virtual** or **logical** links corresponding to a path (possibly through many physical links) in the underlying network.
- Concentrated on one-hop structured P2P overlays
- use a DHT for data indexing and discovery
- (near) single hop from source node to destination node
- Full routing table, maintenance traffic
- EpiChord, D1HT, OneHop



- **DHTs are the indexing mechanism for P2P systems**

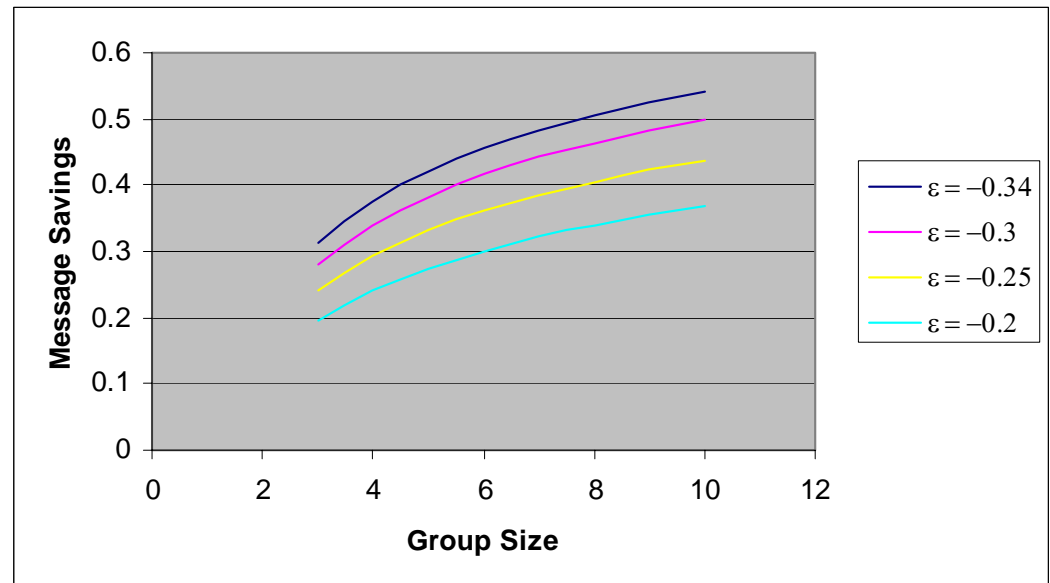
- DHT - Node IDs and Data Keys
- O(1)-hop overlays have better latency characteristic than multi-hop overlays, but require more maintenance traffic
- How to obtain best performance in a large-scale wide area context for DHT operations is an important question.

- **How to make P2P Overlays more efficient? → Multicast**



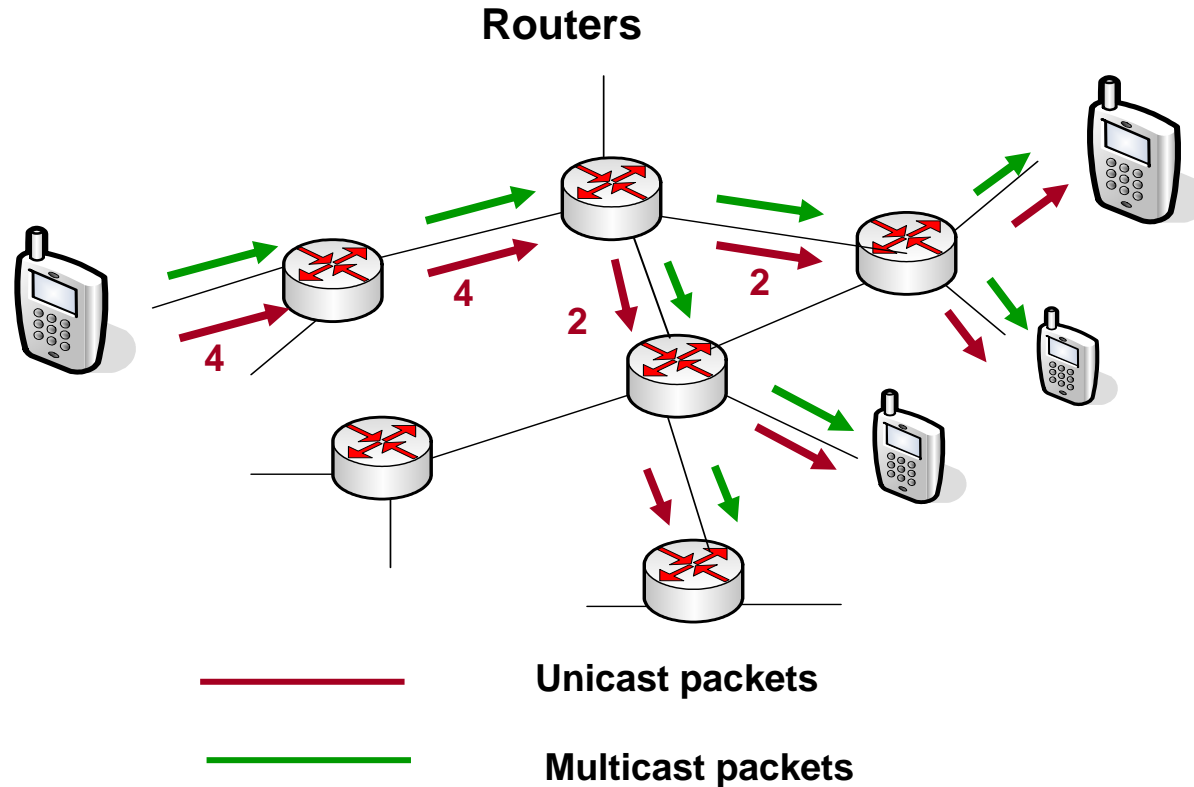
- **Why multicast?**

- Chuang-Sirbu multicast scaling law states message savings are related to group size: $1 - m^{-\epsilon}$, $-0.34 < \epsilon < -0.2$
- 5-way: 28% to 42%, 10-way: 37% to 54%
- Host group multicast vs. multidestination multicast
 - Overhead, group size, group numbers, life time of a group





Multi-Destination Routing

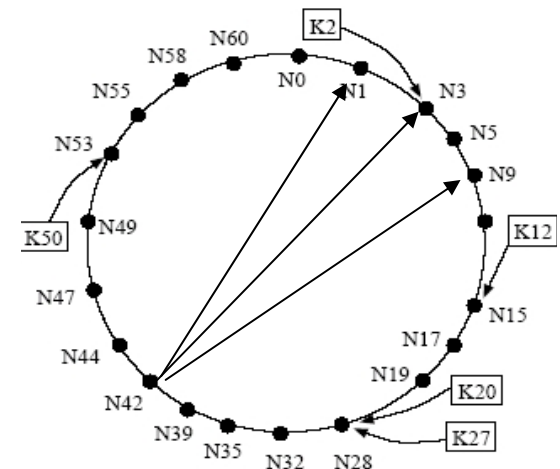


XCAST = Experimental Multi-Destination Routing Protocol

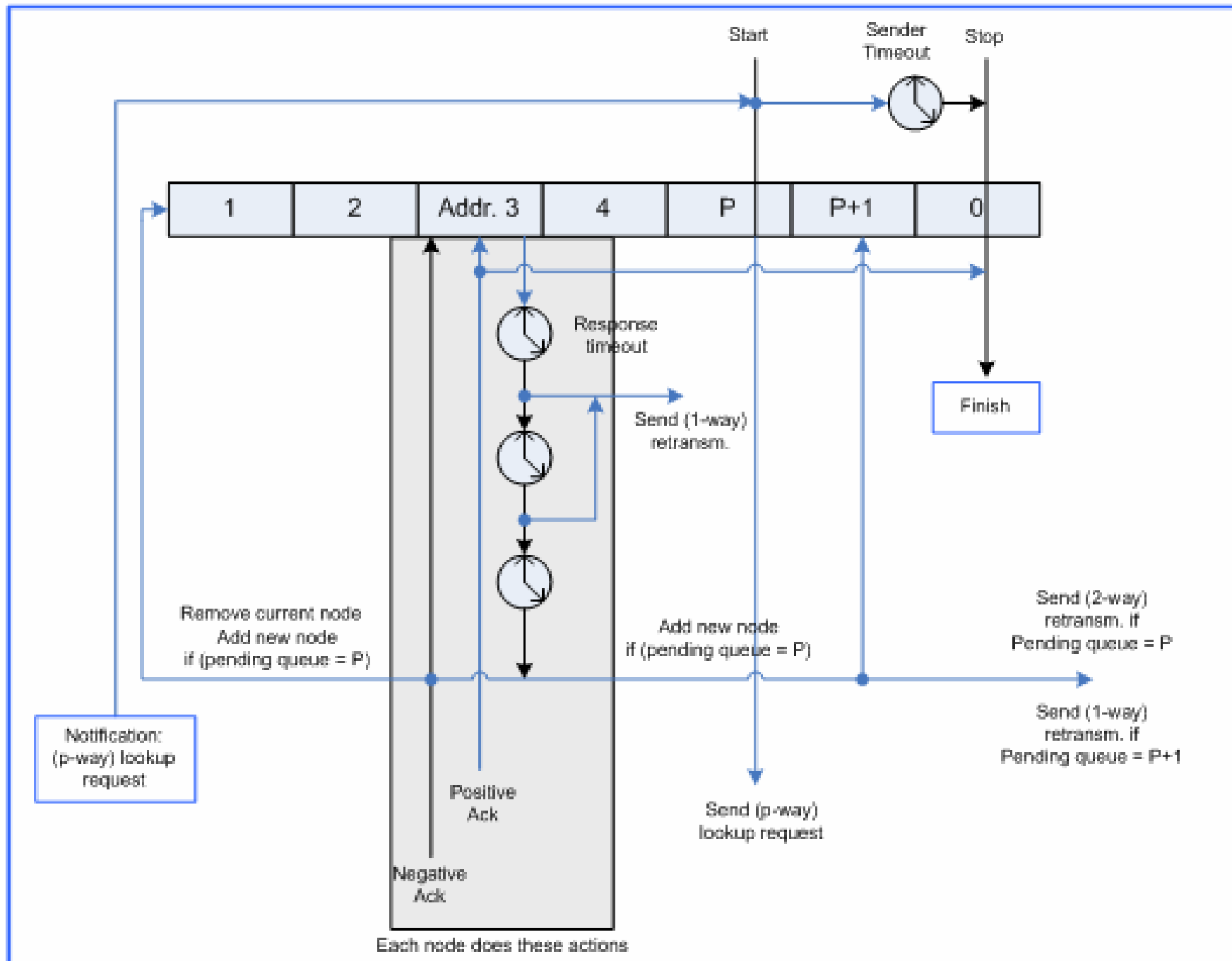


Experimentation

- To determine whether multi-destination routing is applicable to Overlay systems, we used simulation and modelling:
 - EpiChord (simulation).
 - Markov Model(s)
- Simulations were carried out using a 10,450 node network in the SSFNet simulation environment. Overlay sizes varied from 1k to 9k nodes.
- DHT lookups and routing table maintenance use parallel unicast requests
- Failed responses are used iteratively to update routing table and narrow the search
- Opportunistic maintenance of routing table



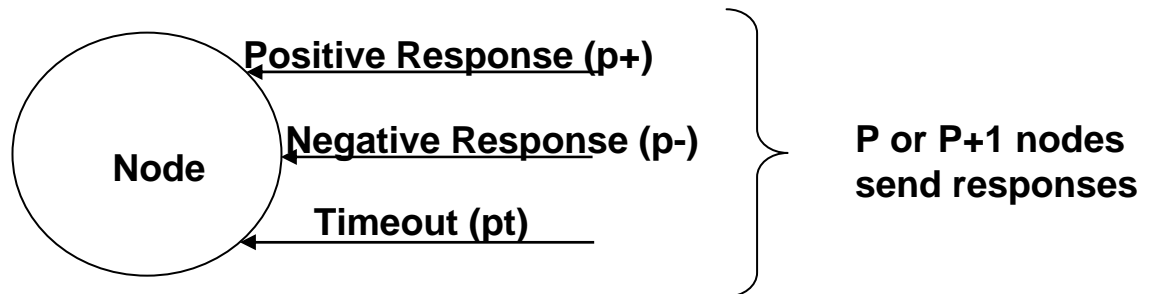
Sender pending queue





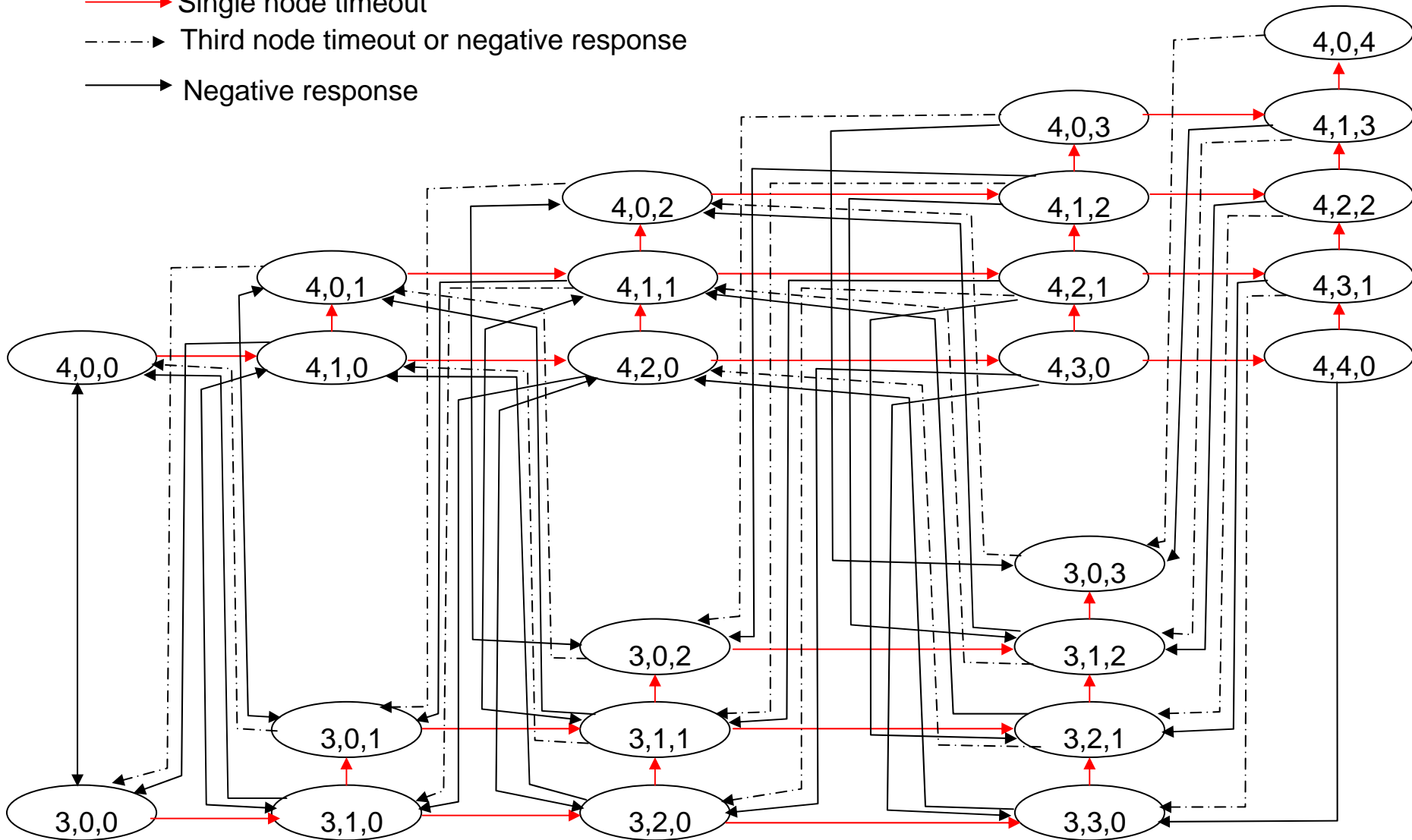
Analytical Model of XCAST enabled EpiChord

- Chuang Sirbu predict saving of $1 - m^{-\varepsilon}$, with $\varepsilon = -0.2$
- Does not take into account EpiChord retransmissions and timeouts
- A model will allow for more flexible and scalable analysis of the expected savings than simulation.
- Comparing results of the model with simulation
- The size of the pending queue changes depending on the type of response received
- Know Probabilities of receiving a certain response from simulations
- Hence pending queue size can be calculated, and so the average # of 2-way and 1-way retransmissions
- Pending queue has been modelled as a DTMC, transition matrix





- Single node timeout
- - - - -> Third node timeout or negative response
- Negative response





UNIVERSITY OF
STIRLING



DEPARTMENT OF COMPUTING SCIENCE AND MATHEMATICS

Assumptions

- Assumption 1:
 - probabilities do not change over time
 - The time the queue is in a certain state is ignored
- Assumption 2:
 - A transition occurs after one and only one response is received
 - Considers only a single node
- Assumption 3:
 - It is equally likely for a node to time out once, twice or three times
 - Probabilities of timing out is independent of the state



Results

P	Neg Resp. per lookup	Timeouts per lookup	Xcast (model)	Xcast (simul)	unicast (model)	unicast (simul)
3	1.44	1.3	0.77	0.75	0.87	0.9
4	1.98	1.54	1.06	1	1.02	1.05
5	2.54	1.77	1.35	1.22	1.18	1.2

P	Neg Resp. per lookup	Timeouts per lookup	Xcast (model)	Xcast (simul)	unicast (model)	unicast (simul)
3	6.1	3.16	3.19	3.05	2.2	2.22
4	7.27	3.67	3.81	3.45	2.52	2.6
5	8.49	4.23	4.49	3.92	2.88	3.03

- Use Pepa to model the system to get closer results...



UNIVERSITY OF
STIRLING



PEPA

- Two models
- Communicating model
 - Pending queue process
 - Processes for each process in the pending queue
- “Simple model” based on the states of the DTMC
- Expected results to be closer to simulation values
- Results show too many retransmissions



Communicating Model

- Node[4] <stop, success, thirddtimeout, negresp, start> PQ
- Nodes
 - Get started
Node = (start,infty).NodeS;
 - Positive responses, negative responses, timeouts
NodeS = (success,beta).Node
+ (timeout,gamma).NodeT
+ (negresp,delta).Node
+ (stop, infty).Node;
- Pending Queue Process
 - Starting nodes in the pending queue
 - “receiving responses” from nodes in PQ



- Pending Queue Process

PQ = (start,alpha).(start,alpha).(start,alpha).PQS;

PQS = (success,infty).DonePQ

+ (thirddtimeout,infty).PQplus1i

+ (negresp,infty).PQplus1i;

PQplus1i = (up, alpha).PQplus1;

PQplus1 = (start,alpha).(start,alpha).PQplus1S;

PQplus1S = (success,infty).DonePQPP1

+ (thirddtimeout,infty).PQS

+ (negresp,infty).PQS;

DonePQ = (stop, alpha).(stop, alpha). PQ;

DonePQPP1 = (stop, alpha).(stop, alpha).(stop, alpha). PQ;



UNIVERSITY OF
STIRLING



- Results:
- For 5-way: unicast: 1.65 (1.2 –simu, 1.18-DTMC)
xcast: 1.55 (1.22 - simu, 1.35-DTMC)
- Not very close!



- Developed “simple model”, following the DTMC

Node_300 = (positive,alpha).Node_300
+(negativeUP,beta).Node_400
+ (timeout, gamma).Node_310;

Node_301 = (positive,alpha).Node_300
+ (negativeUP,beta*(2/3)).Node_401
+ (negativeUP,beta*(1/3)).Node_400
+ (timeout, gamma*(2/3)).Node_311
+ (timeoutUP,gamma*(1/3)).Node_400;

...

- Results:
- For 3-way: unicast: 1.27 (0.9-simu, 0.87-DTMC)
xcast: 0.82 (0.75-simu, 0.77-DTMC)
- Also not very close!



UNIVERSITY OF
STIRLING



DEPARTMENT OF COMPUTING SCIENCE AND MATHEMATICS

- Potential issues
 - Timeouts
 - Rates vs. probabilities
 - ???