Constructing Low Latency Overlay Multicast Trees

Su-Wei, Tan Gill Waters John Crawford
{swt3,a.g.waters,j.s.crawford}@kent.ac.uk
Computing Laboratory
University of Kent, UK

Outline

- Introduction
- The Problem
- Proposal
- Simulation Results

Introduction

- Efficient delivery trees for delay-sensitive point-to-multipoint applications
  - Live webcasting, Audio/video conferencing
  - Potentially have large number of participants

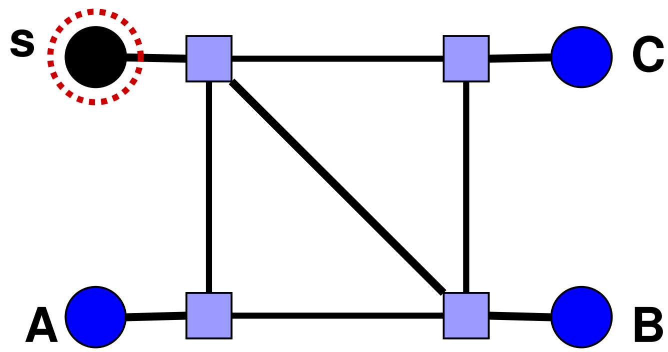
- Multicasting: provides efficient data delivery mechanism
  - Spare global network layer infrastructure support

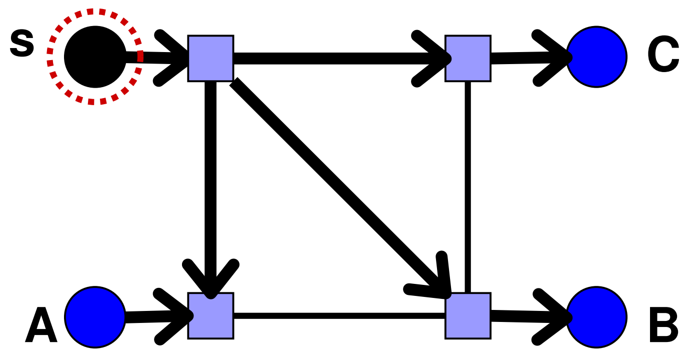
→ Application Layer Multicast (ALM)

Application Layer Multicast (ALM)

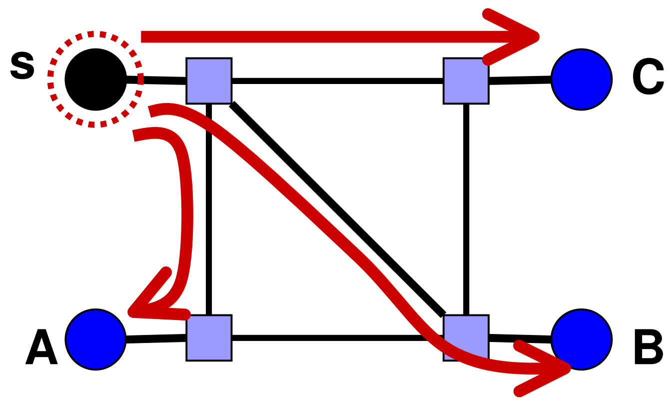
- Multicast functionalities such as packet replication are implemented directly at end systems
- End systems are organised into a logical topology – overlay
  - Overlay edges are unicast tunnels between the end systems

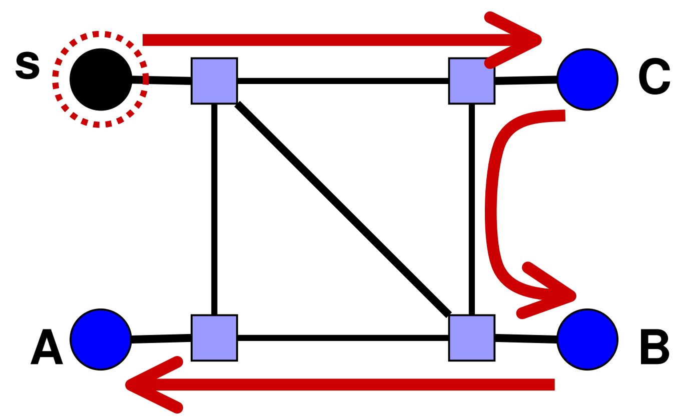
ALM Examples





IP Multicast





ALM Solutions

ALM Challenges

- Quality of data delivery: stress & stretch
- Limited topology information
- Capacity constraint & heterogeneity
  - Degree constrained overlay
- Robustness

Problem Statement

- Overlay network: *G*(*V,E*) where *V* is the set of end systems (including source), and *E* = *V* x *V* is the set of edges. Each node is degree-bounded due to its available capacity.

- Minimum maximum-latency degree-bounded spanning tree:
  - Find a spanning tree, *T* of *G* rooted at the source, *s* satisfying the degree bound at each node, such that the maximum delay from *s* to the receivers is minimised.

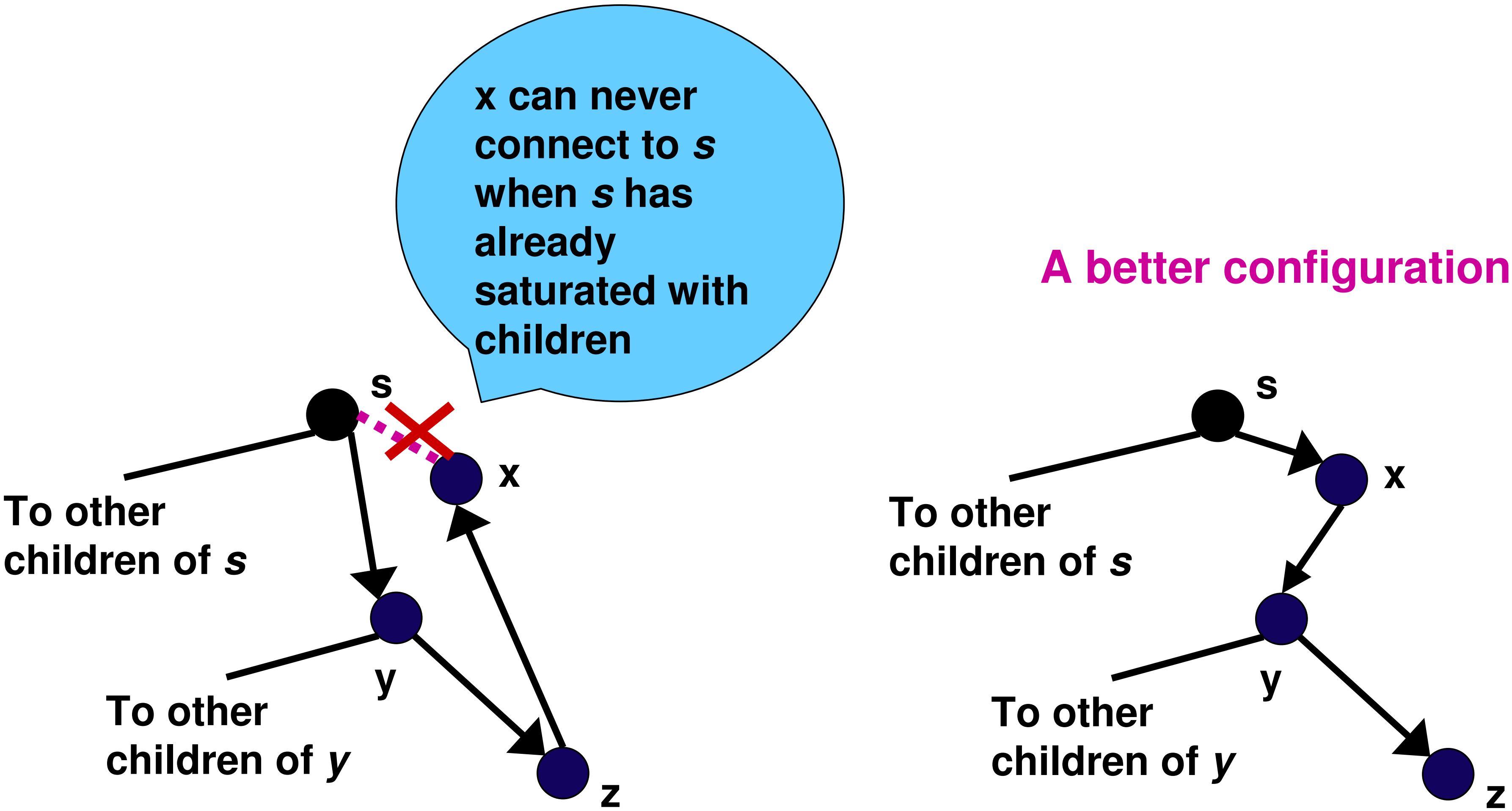
- NP-hard!

Motivation: The Greedy Problem

- Minimise the delay from the tree root
  - Every node tries to get as close as possible to the root

- (delay optimisation) + (Lack of complete topology information) + (degree constraint)
  - Greedy problem

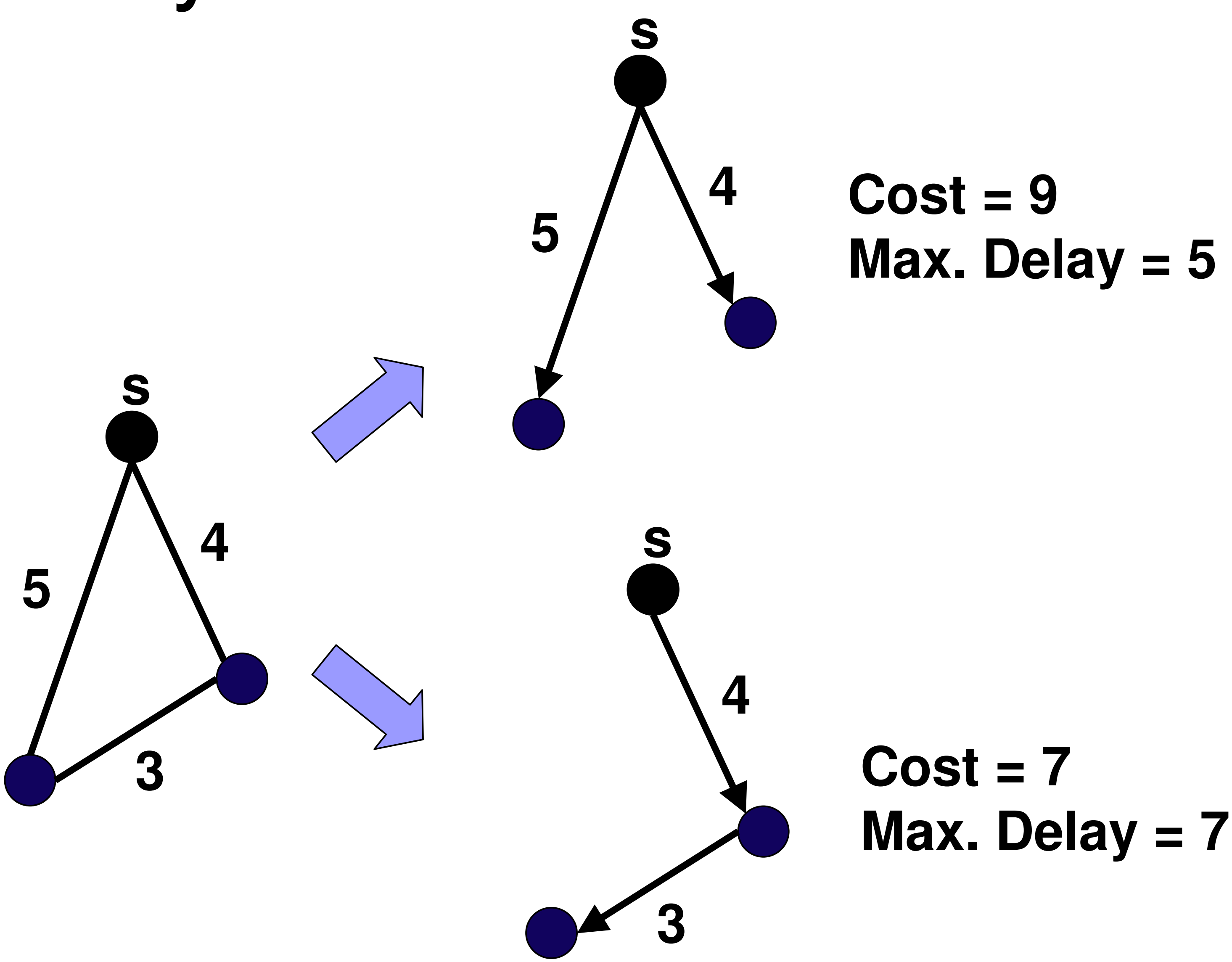
Greedy Problem: Example



Greedy Problem: Possible Solution

- Organise nodes (approximately) based on their relative position in the underlying network
  - Can be viewed as a minimum spanning tree problem where edge delay = cost
- BUT low cost tree $\neq$ low delay tree

Delay-cost trade-offs



Proposal

- Create a overlay mesh which consists of:
  - Low cost backbone
  - Additional links to reduce the delay

- Questions
  - How to create the overlay?
  - How to get the degree-bounded delivery tree?

A. Creating the overlay

- Start with a randomly connected mesh
- Every node (except root) performs periodically refinement
  - Add/delete links within the degree bound
  - Prioritise links in the backbone over links in the delivery tree
  - Each operation involves only nodes engaged in the process ➔ No global coordination is needed

B. Getting the delivery tree

- Path-vector routing protocol
  - Using *s* as the sole destination
- Reverse path forwarding to get the tree rooted at *s*

Solution Properties

- Decentralised
  - Use partial topology information
  - Scalable
- Adaptation
  - Adapt to changes in membership & underlying network conditions
- Feasible
  - Degree-constrained is enforced throughout the multicast session

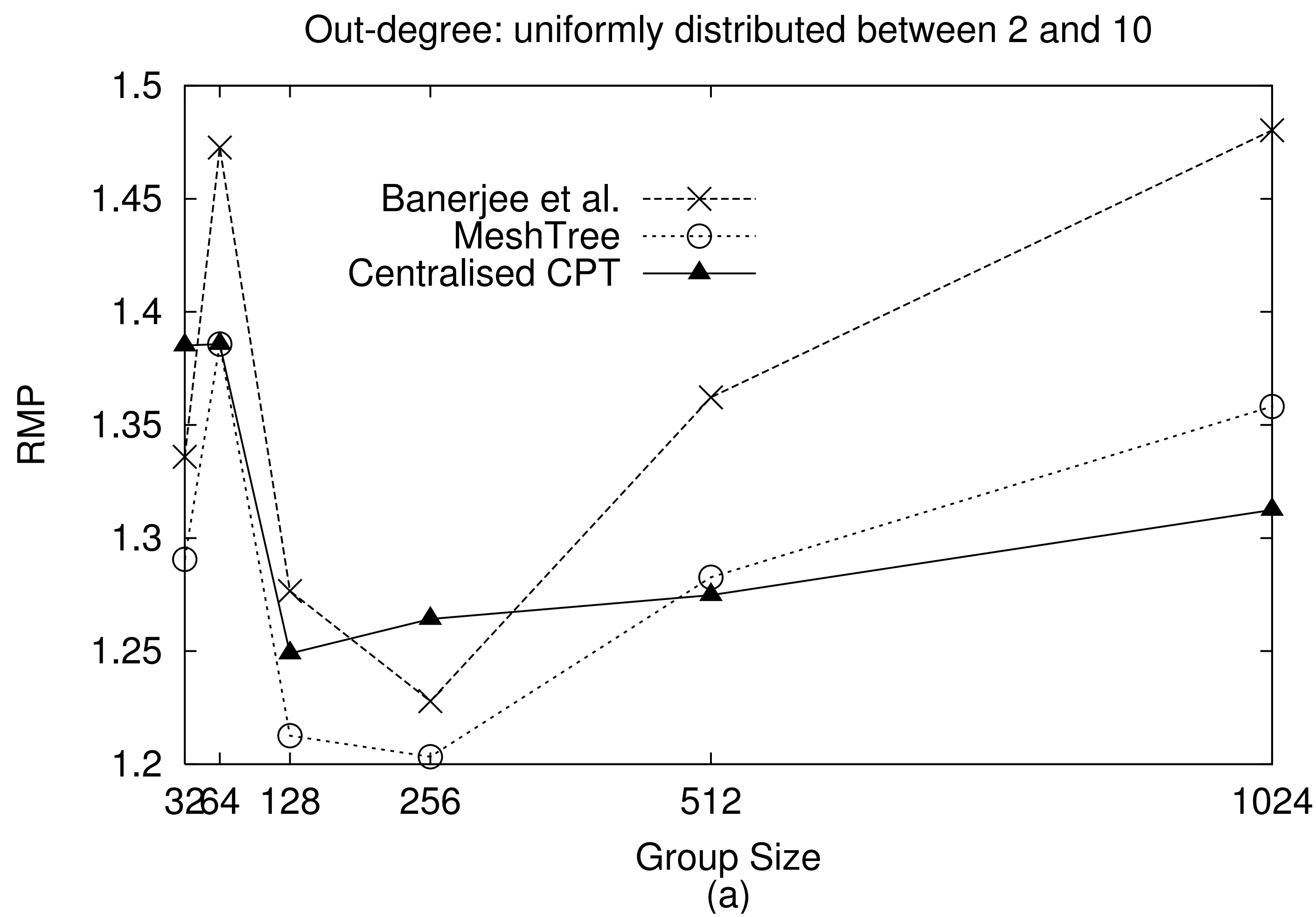
Simulation Experiments

- 10100-node transit-stub topology generated by GT-ITM topology generator
- Overlay members: 32 - 1024
- Max. out-degree: uniformly distributed between 2 to 10

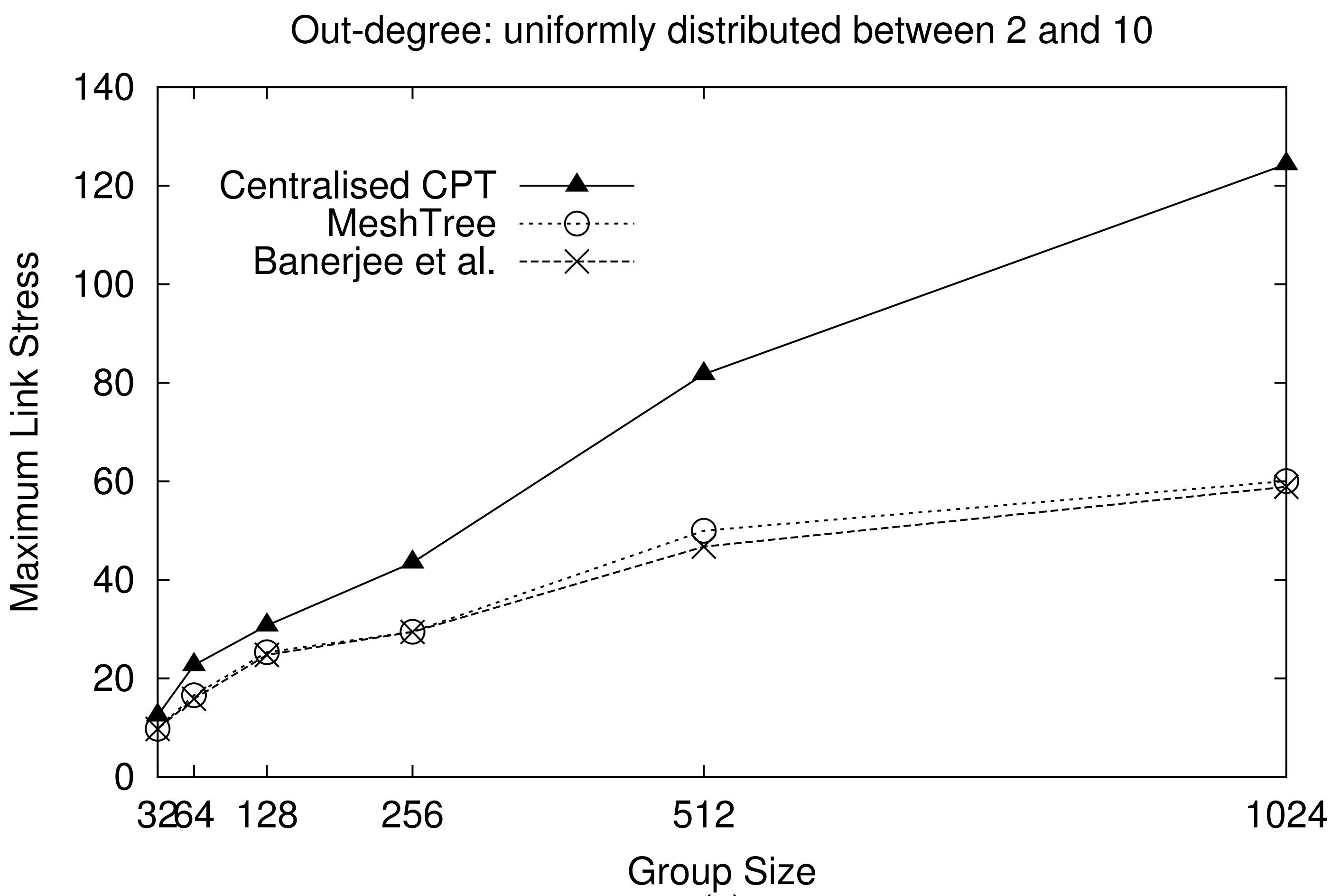
Comparisons

- Compact Tree algorithm (JSAC’02)
  - Centralised greedy heuristic
- Banerjee et al. scheme (Infocom’03)
  - Flexible tree-reconfiguration for nodes within 2-level of each another
  - Our previous study shows that it performs better than several other ALM protocols, e.g. switch-trees, HostCast, NICE, TBCP.

Maximum session delay penalty: RMP

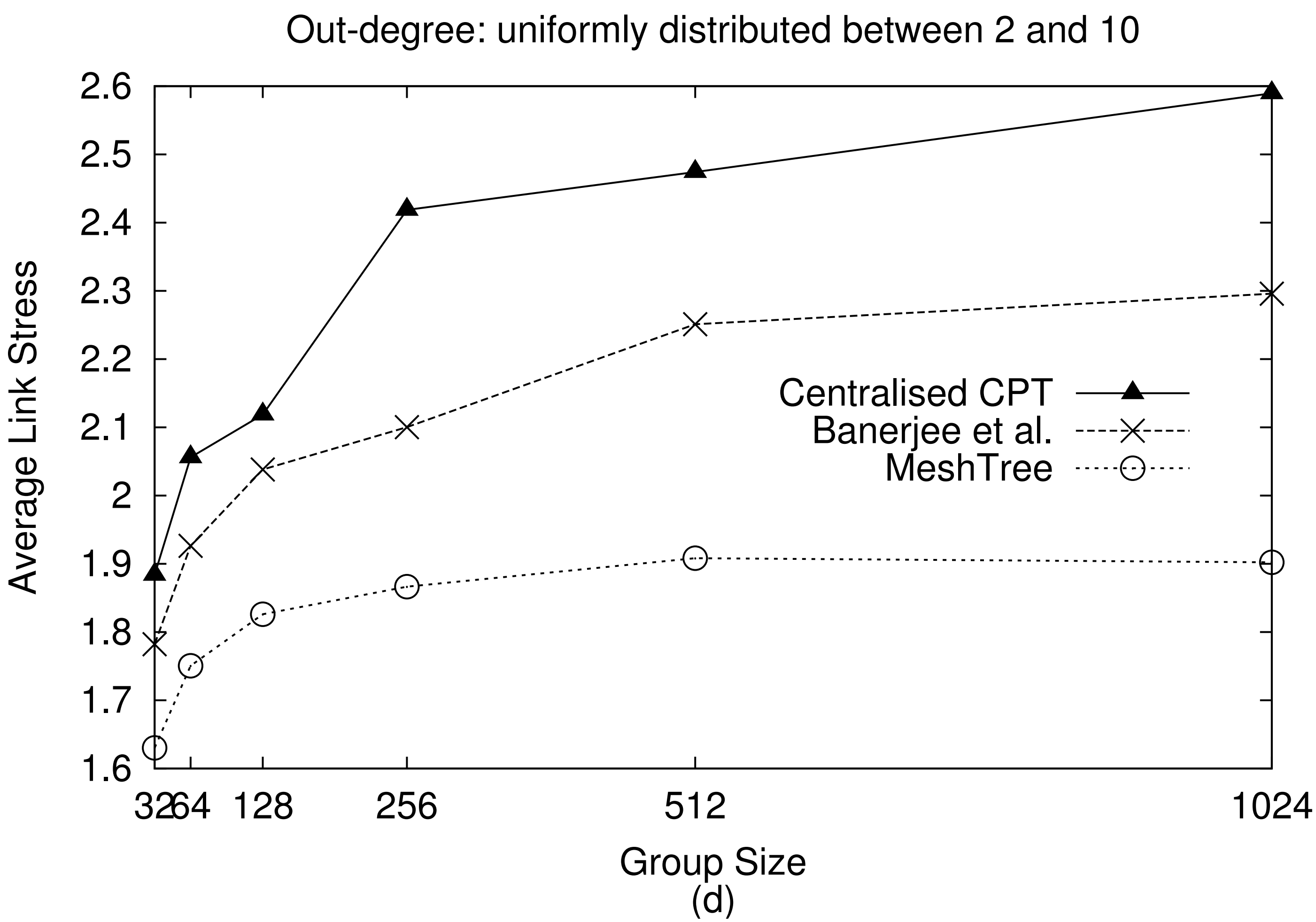


Maximum Link Stress

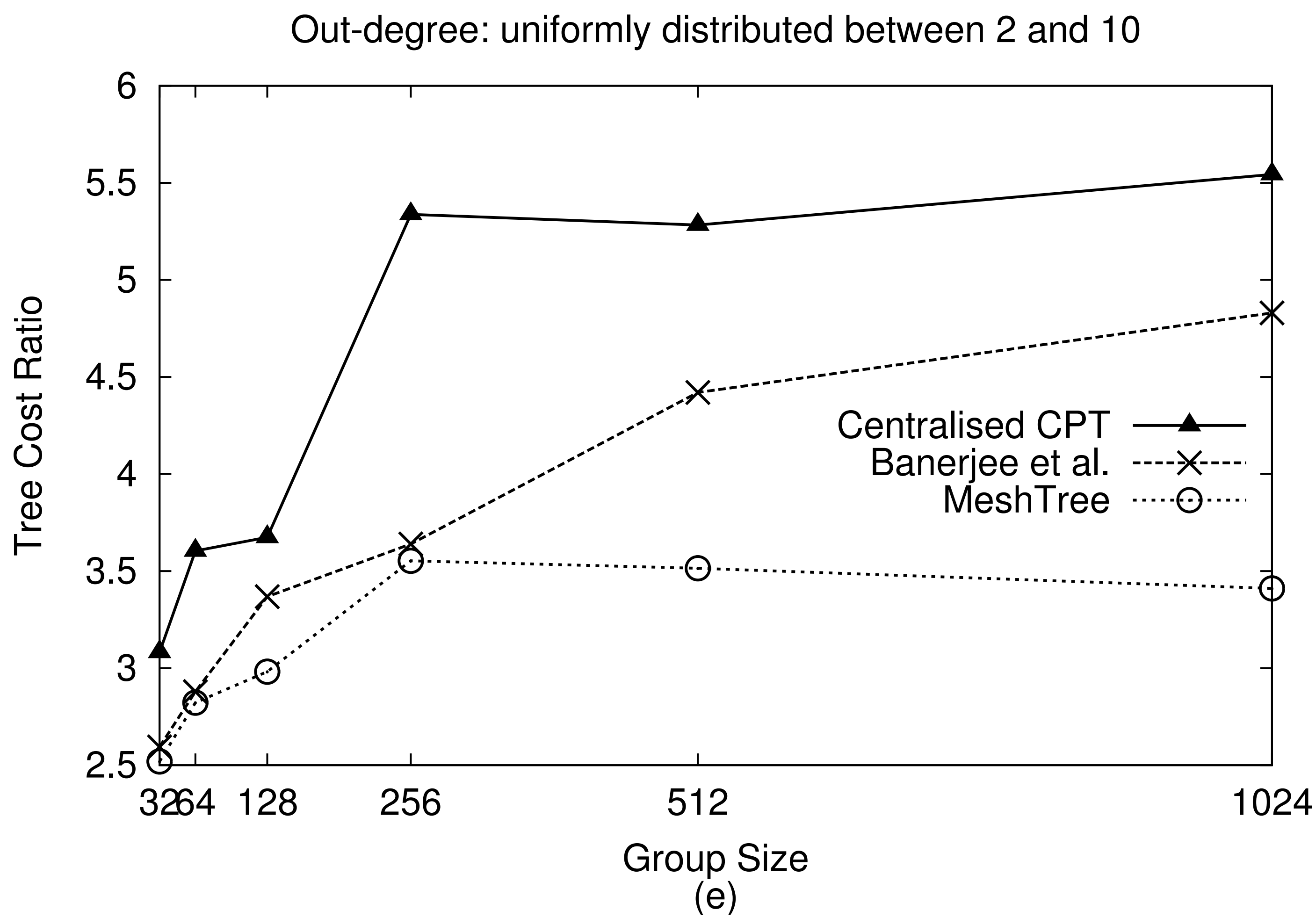


(c)

Average Link Stress



Tree Cost Ratio



Convergence