Comparison of k-ary n-cube and de Bruijn Overlays in QoS-constrained Multicast Applications

Richard West, Gary Wong and Gerald Fry

Boston University

Computer Science

Introduction

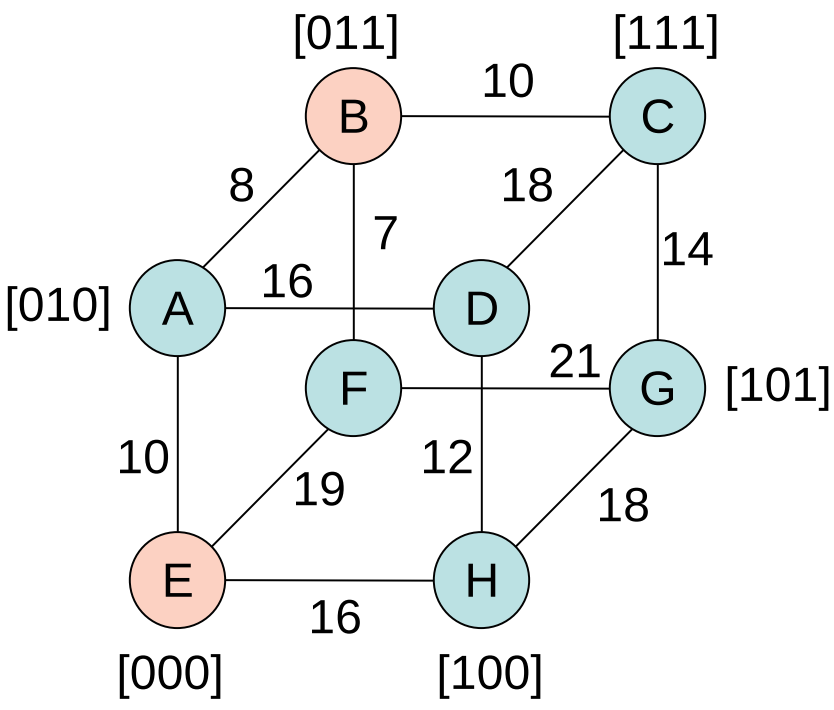
Computer Science

- Our goal is a multicast system which can:
  - Guarantee timely delivery of data
  - Scale to many thousands of end hosts
- We consider an overlay infrastructure built using a regular graph topology, to:
  - Reduce the end-to-end hop count
  - Allow simple and flexible routing
  - Minimise link stress on the underlying physical network
- Two regular graphs: k-ary n-cubes and de Bruijn graphs

k-ary n-cubes

Computer Science

- M=k^n nodes
- Node ID: n base-k digits
- Neighbors have n-1 common digits in their IDs
  - ith digit in each ID differs by +/- 1 mod k
- Graph diameter: $n\lfloor k/2 \rfloor$



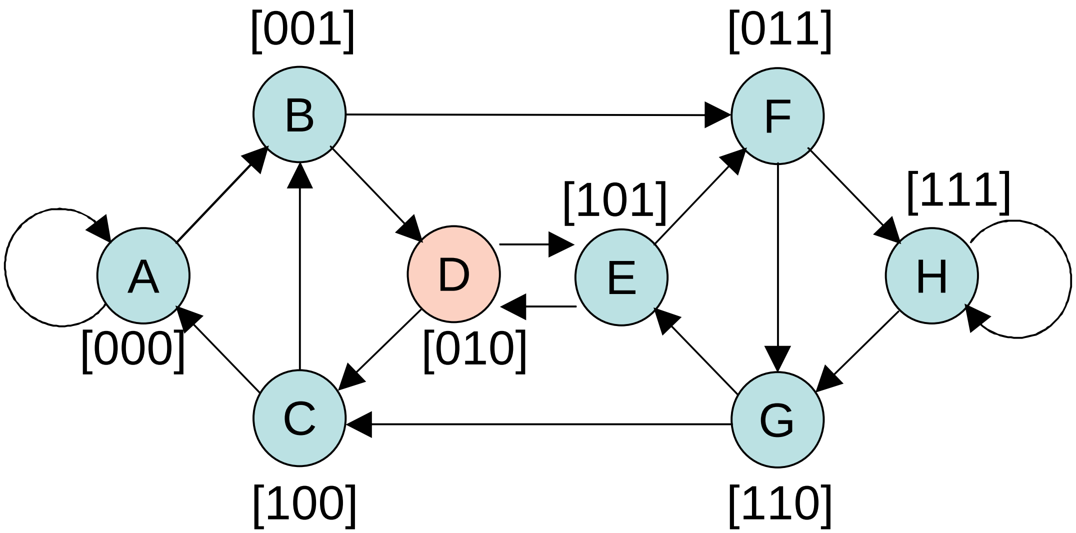
Node F ID = 001, Node D ID = 110

Nodes B and E are failed nodes

de Bruijn Graphs

Computer Science

- M=k^n nodes
- Node ID: n base-k digits
- Neighbors: directed edge from A to B iff last n-1 digits of A match 1st n-1 digits of B
- Graph diameter: n



Route Availability

- How many routes exist between a given source/destination pair?
- *k*-ary *n*-cubes: $(\lfloor k/2 \rfloor n)! / (\lfloor k/2 \rfloor !)^n$
- de Bruijn graphs:
  - Only a single path with minimal hop count exists
  - If we allow the source to route via an alternative peer (for redundancy), then in general there exist k-1 non-overlapping "backup" paths, of length n+1

Fault Resilience

- What if a node along path from source (S) to destination (D) fails?
- Suppose node H hops from D fails:
  - k-ary n-cubes: (H-1)(H-1)! alternative shortest paths
  - de Bruijn graphs: no backup paths as short as original

Table of Various Properties

				Hop Count		Local Routes			Global Routes	
		Nodes	Degree	Med	Max	Min	Med	Max	Med	Max
k	n	k-ary n-cubes								
2	20	1M	20	10	20	1	10	20	3.6M	2×10^{18}
3	13	1.6M	26	9	13	1	9	13	363K	6.2G
	k	de Bruijn graphs								
	2	1M	2	19	20	1	1	(2)	1	(2)
	3	1.6M	3	13	13	1	1	(3)	1	(3)
	4	1M	4	10	10	1	1	(4)	1	(4)
	5	2M	5	9	9	1	1	(5)	1	(5)
	20	3.2M	20	5	5	1	1	(20)	1	(20)
	26	12M	26	5	5	1	1	(26)	1	(26)

Multicast Tree Construction

Computer Science

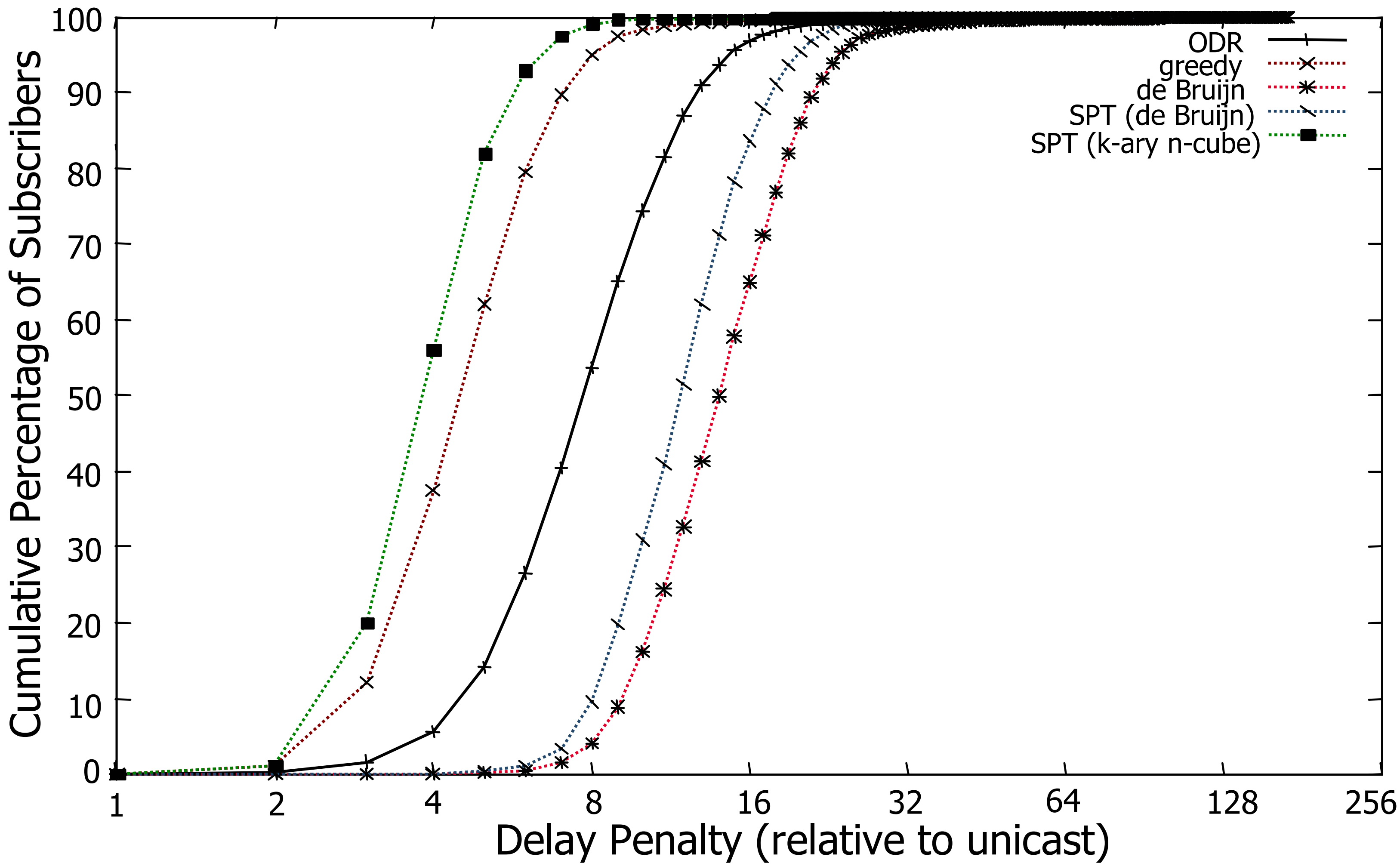
- Consider different methods for multicast tree construction using regular overlay topologies, that affect:
  - Relative delay penalty: ratio of end-to-end delay across overlay to equivalent unicast latency at (physical) network level
  - Link stress: ratio of total msg transmissions to number of physical links involved
  - Normalized lateness:
    - 0 if end-to-end overlay delay (d) within subscriber deadlines (D)
    - (d – D) / D otherwise
  - Success ratio: Fraction of all subscribers satisfying their deadlines (D)

Experimental Evaluation

- GT-ITM used to simulate physical network w/ 5050 routers
- Compare performance of each overlay using various routing strategies:
  - k-ary n-cubes:
    - ODR – route in a specific order of dimensions
    - Random – route in random dimensions as long as distance to destination is reduced at each hop
    - Greedy – choose next hop with lowest latency
  - de Bruijn – shift-based routing
    - e.g. 000 $\rightarrow$ 010 : 000 $\rightarrow$ 001 $\rightarrow$ 010

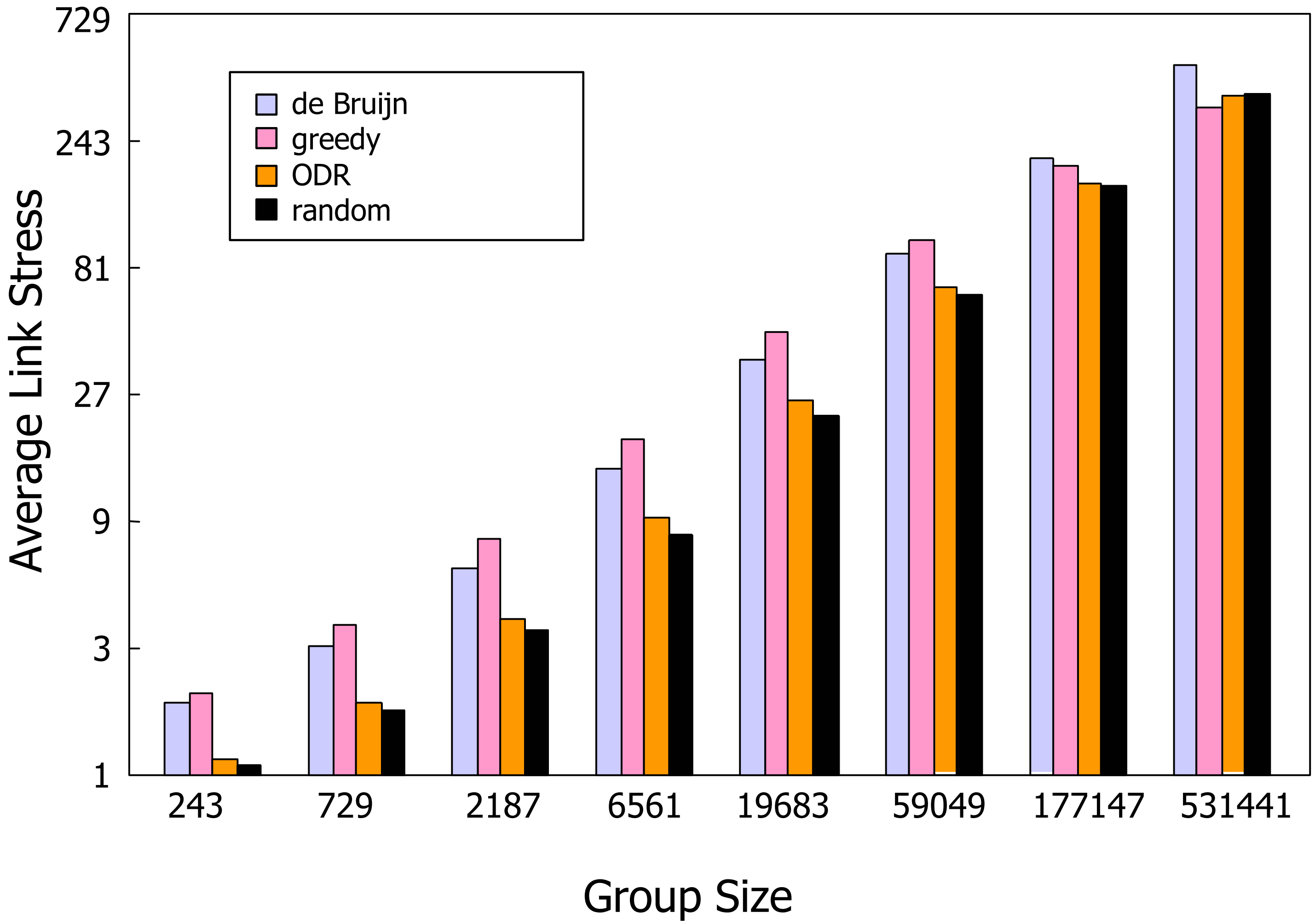
Relative Delay Penalty

Computer Science



- k=2 n=16, SPT = Dijkstra's shortest path routing across overlay

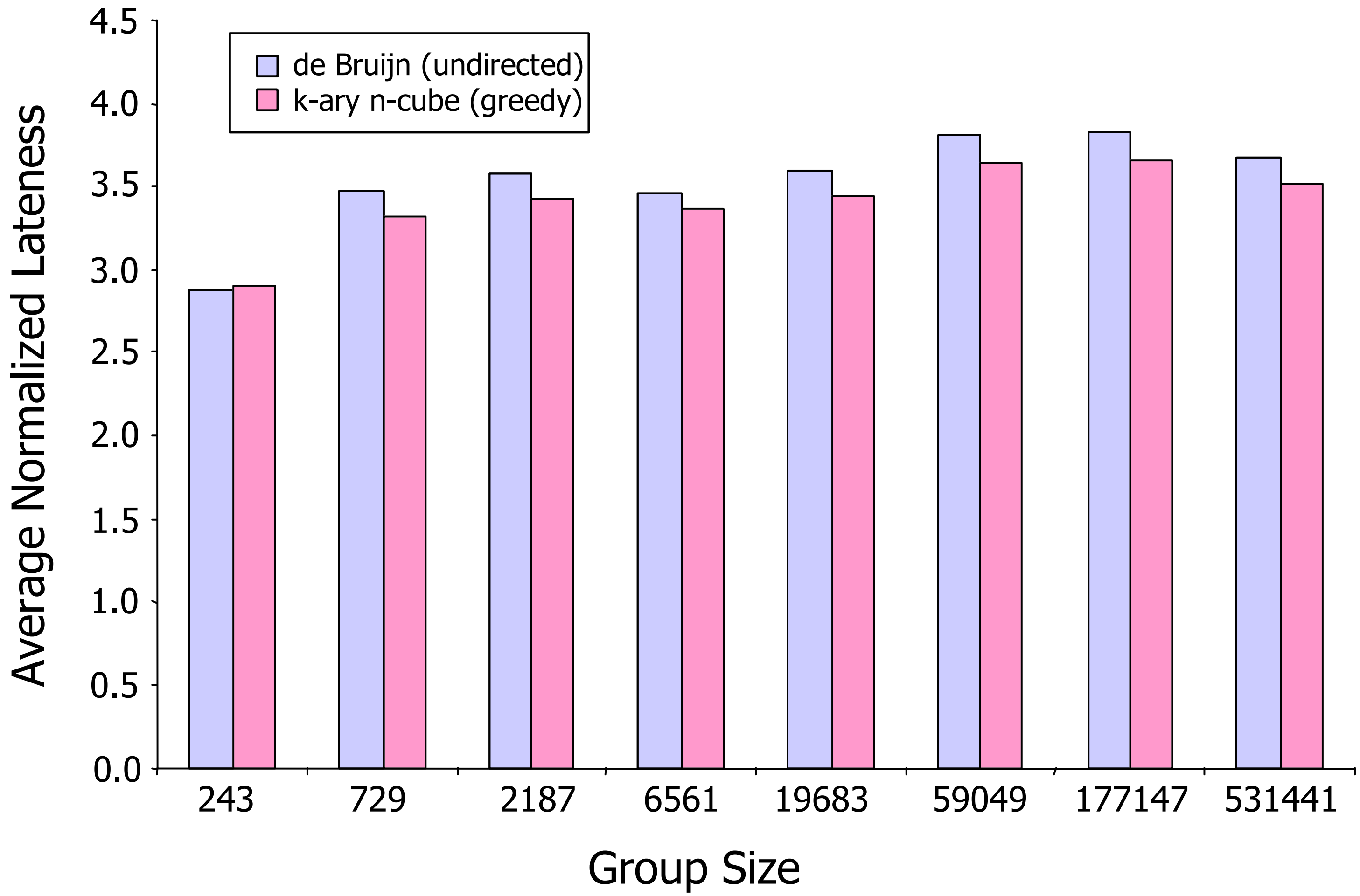
Link Stress



- 3-ary 13-cube versus de Bruijn graph with k=10 and n=6

Lateness

Computer Science

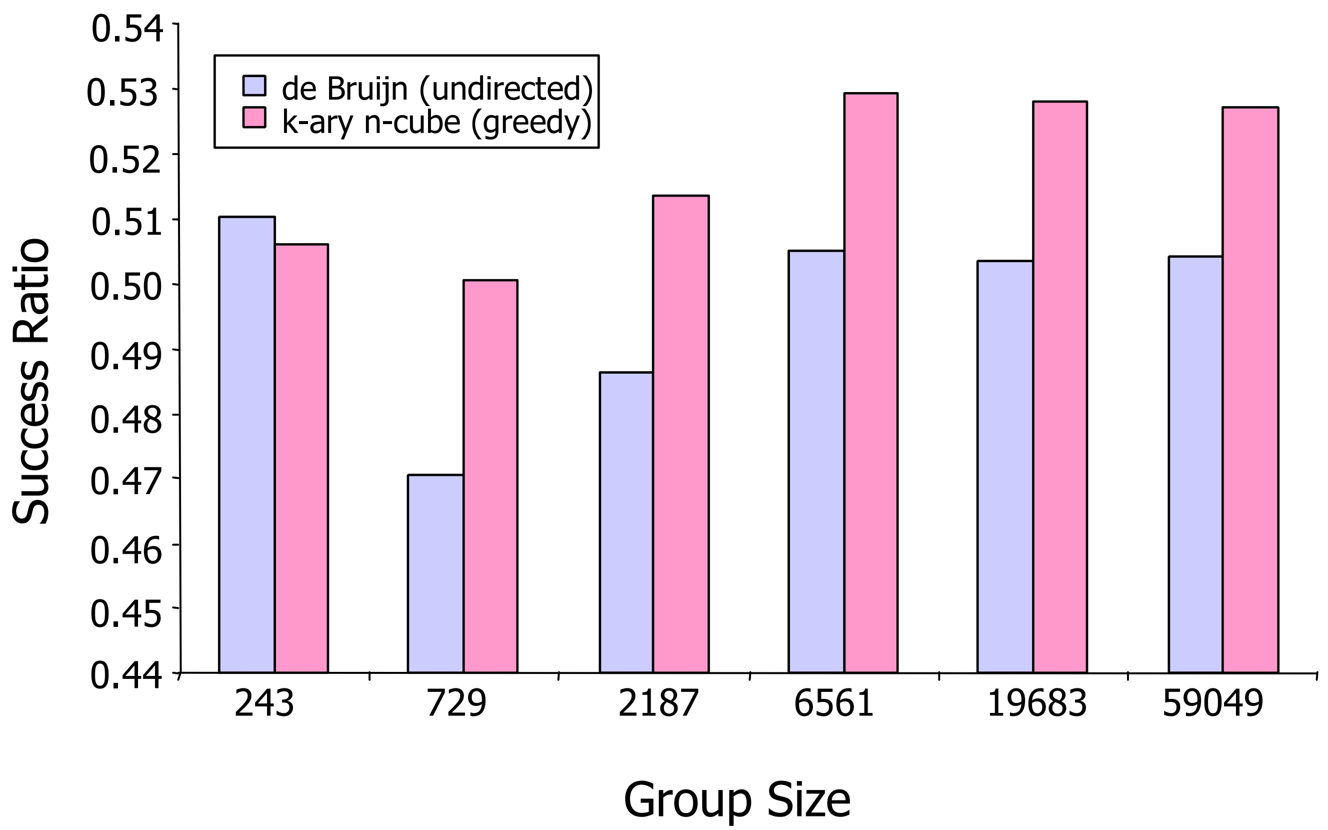


Success Ratio

Computer Science

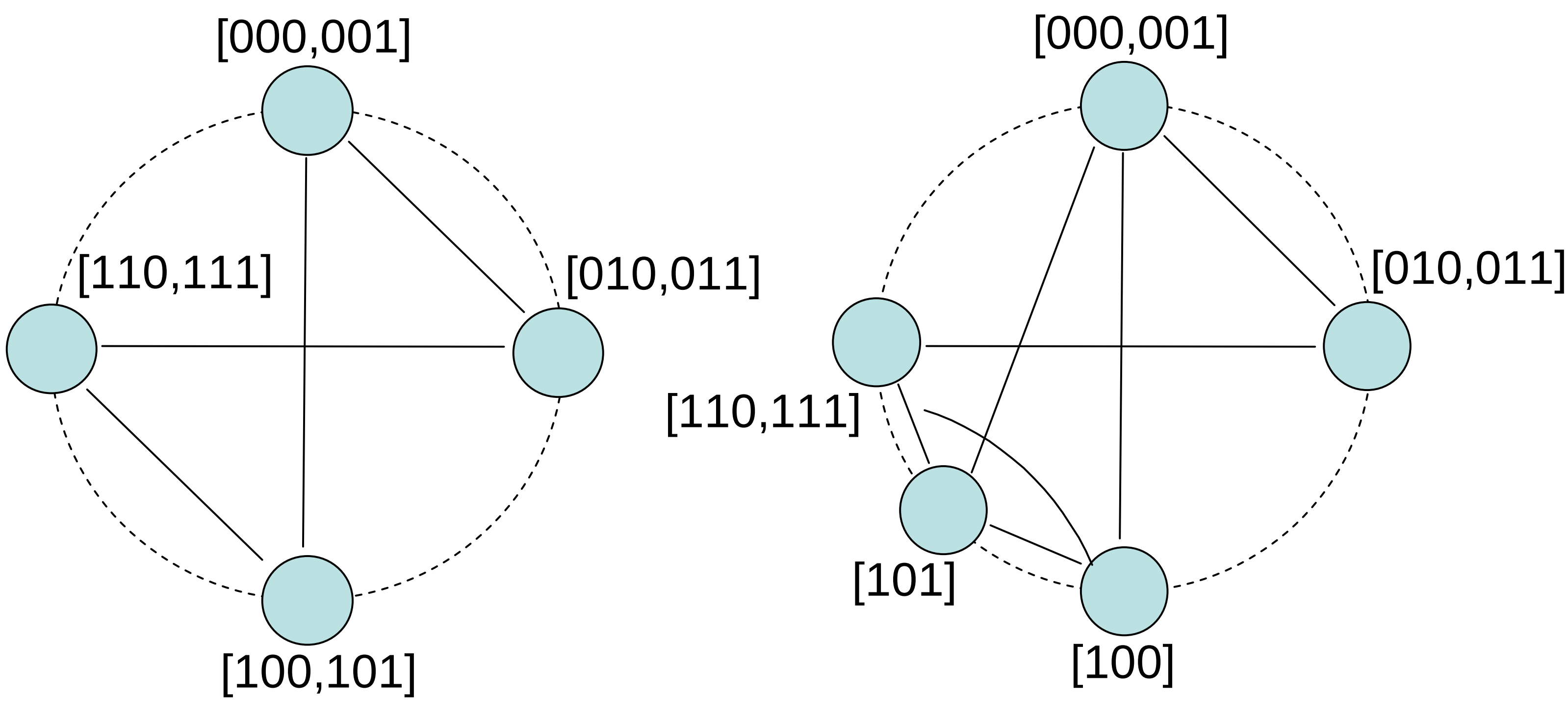
subscriber deadline = random[min physical link delay,
max link delay * diameter of k-ary n-cube]

NOTE: success ratio is a relative metric --
Can be improved by increasing subscriber deadlines



Dynamic Characteristics

- e.g., supporting hosts joining system for k-ary n-cubes
- ID space is set to $M=k^n$ with physical hosts randomly assigned logical IDs in this space
- Each host responsible for 1 or more logical IDs depending on ID originally chosen randomly



Conclusions and Future Work

Computer Science

- Compare k-ary n-cubes and de Bruijn graphs for routing data between source and many destinations w/ per-subscriber service constraints
- May be less effective than building end-system multicast trees from the "ground up" (w/o considering overlay topology) BUT much simpler
- Regular topologies could be candidates for large-scale streaming applications
- Future work: An Internet-wide system for processing & delivery of data w/ per subscriber QoS