Adaptive Routing of QoS-Constrained Media Streams over Scalable Overlay Topologies

Gerald Fry and Richard West
Boston University
Boston, MA 02215
{gfry,richwest}@cs.bu.edu

Computer Science

Introduction

- Internet growth has stimulated development of real-time distributed applications
  - e.g., streaming media delivery, interactive distance learning, webcasting (e.g., SHOUTcast)
- Peer-to-peer (P2P) systems now popular
  - Efficiently locate & retrieve data (e.g., mp3s)
  - e.g., Gnutella, Freenet, Kazaa, Chord, CAN, Pastry
- To date, limited work on scalable delivery & processing of QoS-constrained data streams

Objectives

Computer Science

- Scalable overlay networks
  - Devise a logical network that can support many thousands of hosts
  - Minimize the average (logical) hop count between nodes
- Efficient delivery of data streams
  - Route arbitrary messages (eg., video data packets) along the overlay topology
  - Reduce routing latency by considering physical proximity
- How can logical positions of hosts be adapted to reduce lateness with respect to deadlines?

Contributions

Computer Science

- Focus on scalable delivery of real-time media streams
  - Analysis of *k-ary n-cube* graphs as structures for overlay topologies
  - Comparison of overlay routing algorithms
  - Dynamic host relocation in logical space based on QoS constraints
- Applications: live video broadcasts, resource intensive sensor streams, data intensive scientific applications

Introduction (4)

- Overview of this talk
  - Definition and properties of k-ary n-cube graphs
  - Optimization through M-region analysis
  - Overlay routing policies
  - Adaptive node relocation based on per-subscriber QoS constraints
  - Concluding remarks and future work

Definition of k-ary n-cube Graphs

- A k-ary n-cube graph is defined by two parameters:
  - n = # dimensions
  - k = radix (or base) in each dimension
- Each node is associated with an identifier consisting of n base-k digits
- Two nodes are connected by a single edge iff:
  - their identifiers have n-1 identical digits, and
  - the ith digits in both identifiers differ by exactly 1 (modulo k)

Properties of k-ary n-cube Graphs

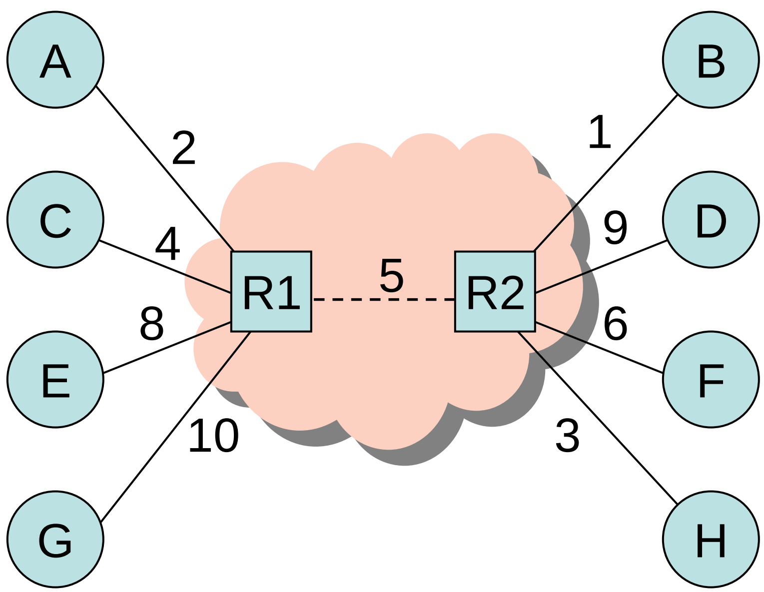
Computer Science

- $M = k^n$ nodes in the graph
- If k = 2, degree of each node is n
- If k > 2, degree of each node is 2n
- Worst-case hop count between nodes:
  - $n\lfloor k/2 \rfloor$
- Average case path length:
  - $A(k,n) = n \lfloor (k^2/4) \rfloor 1/k$
- Optimal dimensionality:
  - n = ln M
  - Minimizes A(k,n) for given k and n

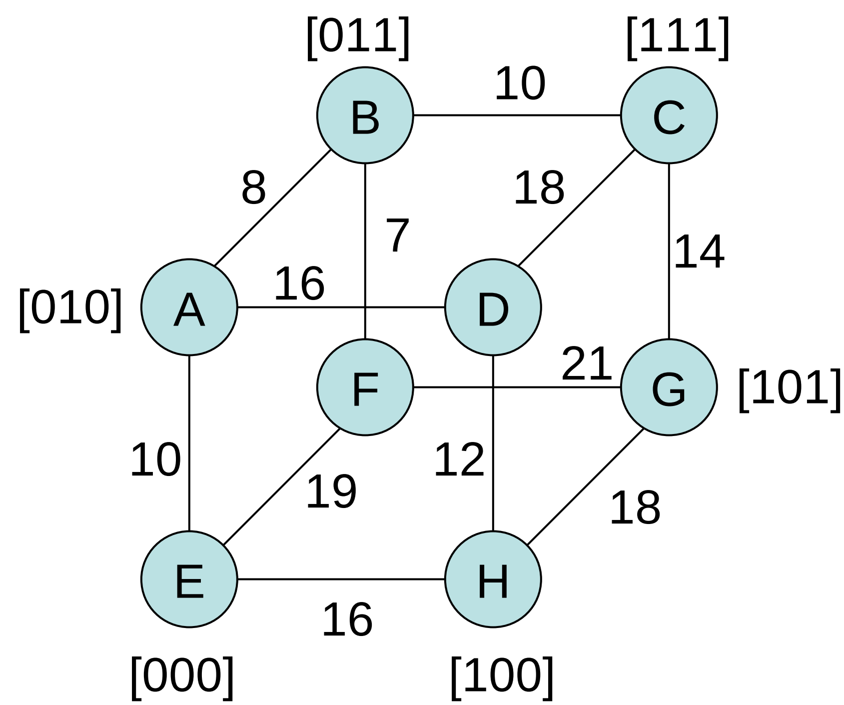
Overlay Routing Example

- Overlay is modeled as an undirected k-ary n-cube graph
- An edge in the overlay corresponds to a uni-cast path in the physical network

Physical view



Logical view



Average Hop Count

- H(k,n): sum of the distances from any one node to every other node in a k-ary n-cube graph

- Proof by induction on dimensionality, n
  - Base case: $H(k,1) = \lfloor (k^2/4) \rfloor$
  - $H(k,n) = H(k,n-1)k + k^{n-1} \lfloor (k^2/4) \rfloor$
  - Thus, $H(k,n) = k^n (n \lfloor (k^2/4) \rfloor 1/k)$

- Avg. hop count between pairs of nodes
  - Given by $A(k,n) = H(k,n) / k^n = n \lfloor (k^2/4) \rfloor 1/k$

Worst-case Hop Count

Computer Science

- Each k-ary n-cube node is represented by a string of n digits in base k
- Given two node identifiers:
  - $A = a_1,a_2,\ldots,a_n$; $B = b_1,b_2,\ldots,b_n$
  - Distance between corresponding nodes is given by the sum of each $a_i - b_i$ (modulo k)
  - Maximum distance in one dimension = $\lfloor k/2 \rfloor$
- Thus, the maximum path length for n dimensions = $n\lfloor k/2 \rfloor$

Logical versus Physical Hosts

Computer Science

- Mapping between physical and logical hosts is not necessarily one-to-one
  - M logical hosts
  - m physical hosts
- For routing, we must have m <= M
  - Destination identifier would be ambiguous otherwise
- If m < M, some logical nodes are unassigned

M-region Analysis

- Hosts joining / leaving system change value of m
  - Initial system is bootstrapped with overlay that optimizes A(k,n)
- Let M-region be range of values for m for which A(k,n) is minimized

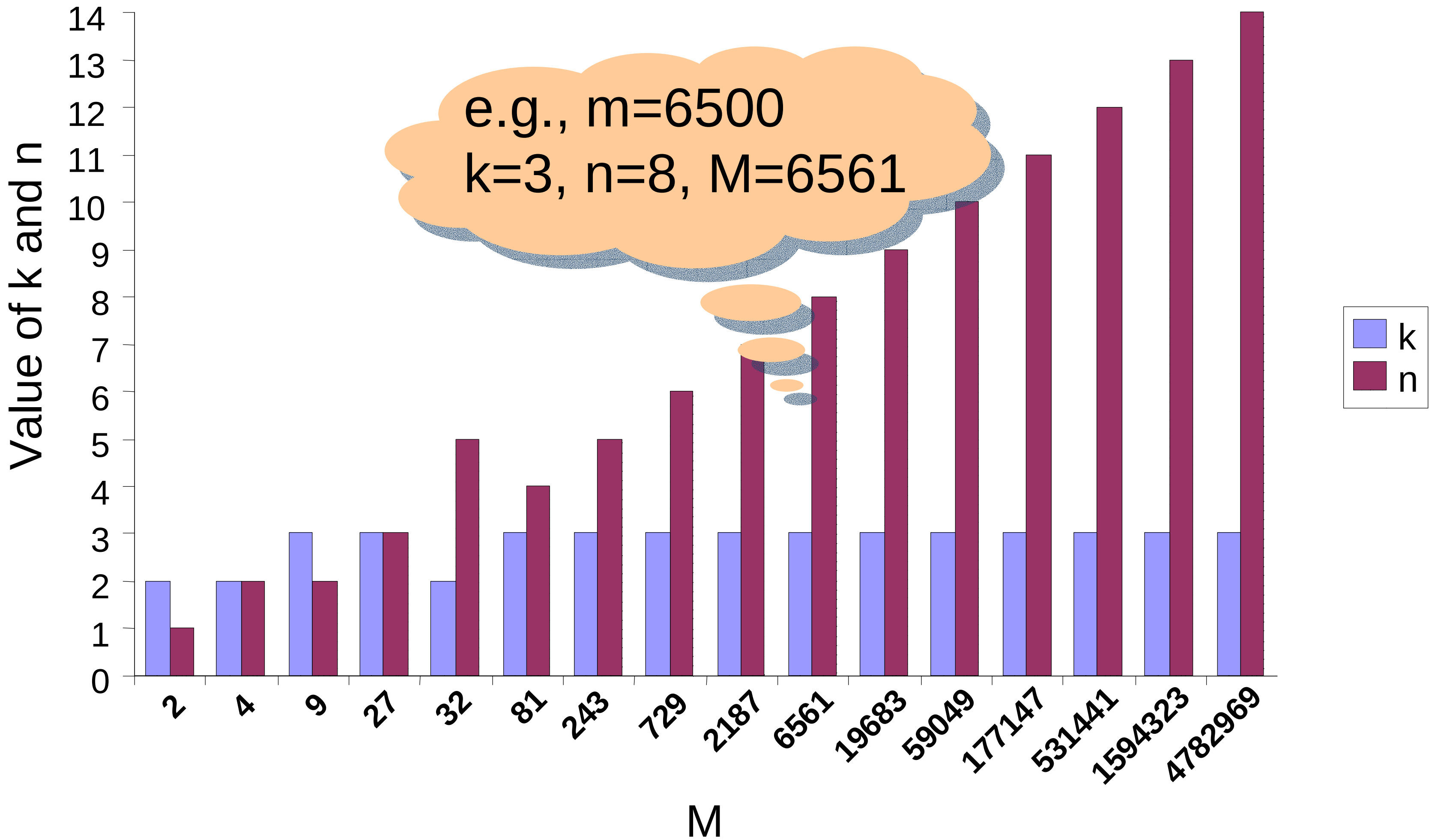
Calculating M-regions

```
Calculate_M-Region(int m) {
    i = 1;    k = j = 2;
    while (M[i,j] < m) i++; // Start with a hypercube
    n = i;
    maxM = M[i,j];
    minA = A[i,j];
    incj = 1;
    while (i > 0) {
        j += incj; i--;
        if ((A[i,j] <= minA) && (M[i,j] > maxM)) {
            incj = 1;
            maxM = M[i,j];
            minA = A[i,j];
            n = i; k = j;
        }
        else incj = 0;
      }
    return k, n;
  }
```

Try to find the largest M such that: m <= M & A(k,n) is minimized

M-regions

Computer Science

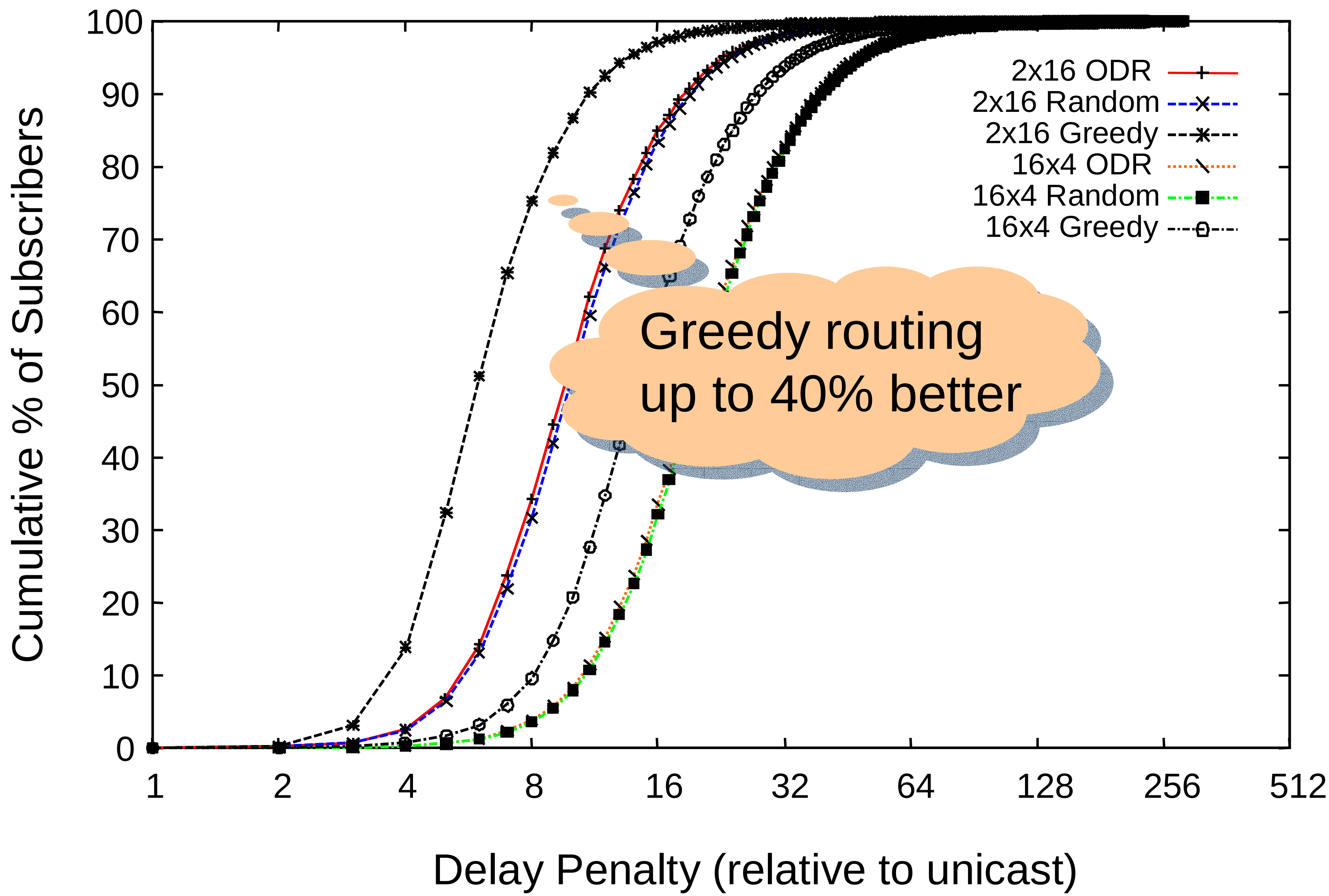


Overlay Routing

- Three routing policies are investigated
  - Ordered Dimensional Routing (ODR)
  - Random Ordering of Dimensions (Random)
  - Proximity-based Greedy Routing (Greedy)
    - Forward message to neighbor along logical edge with lowest cost that reduces hop-distance to destination
- Experimental analysis done via simulation
  - 5050 routers in physical topology (transit-stub)
  - 65536 hosts

16D Hypercube versus 16-ary 4-cube

Computer Science



Adaptive Node Assignment

- Initially, hosts are assigned random node IDs
- Publisher hosts announce availability of channels
  - Super-nodes make info available to peers
- Hosts subscribing to published channels specify QoS constraints (e.g., latency bounds)
- Subscribers may be relocated in logical space
  - to improve QoS
  - by considering "physical proximities" of publishers & subscribers

Adaptive Node Assignment (2)

Computer Science

```
Subscribe (Subscriber S, Publisher P, Depth d) {
  if (d == D) return;

  find a neighbor i of P such that
    i.cost(P) is maximal for all neighbors

  if (S.cost(P) < i.cost(P))
    swap logical positions of i and S;
  else
    Subscribe (S, i, d+1);
}
```

- Swap S with node i up to D logical hops from P

Simulation Results

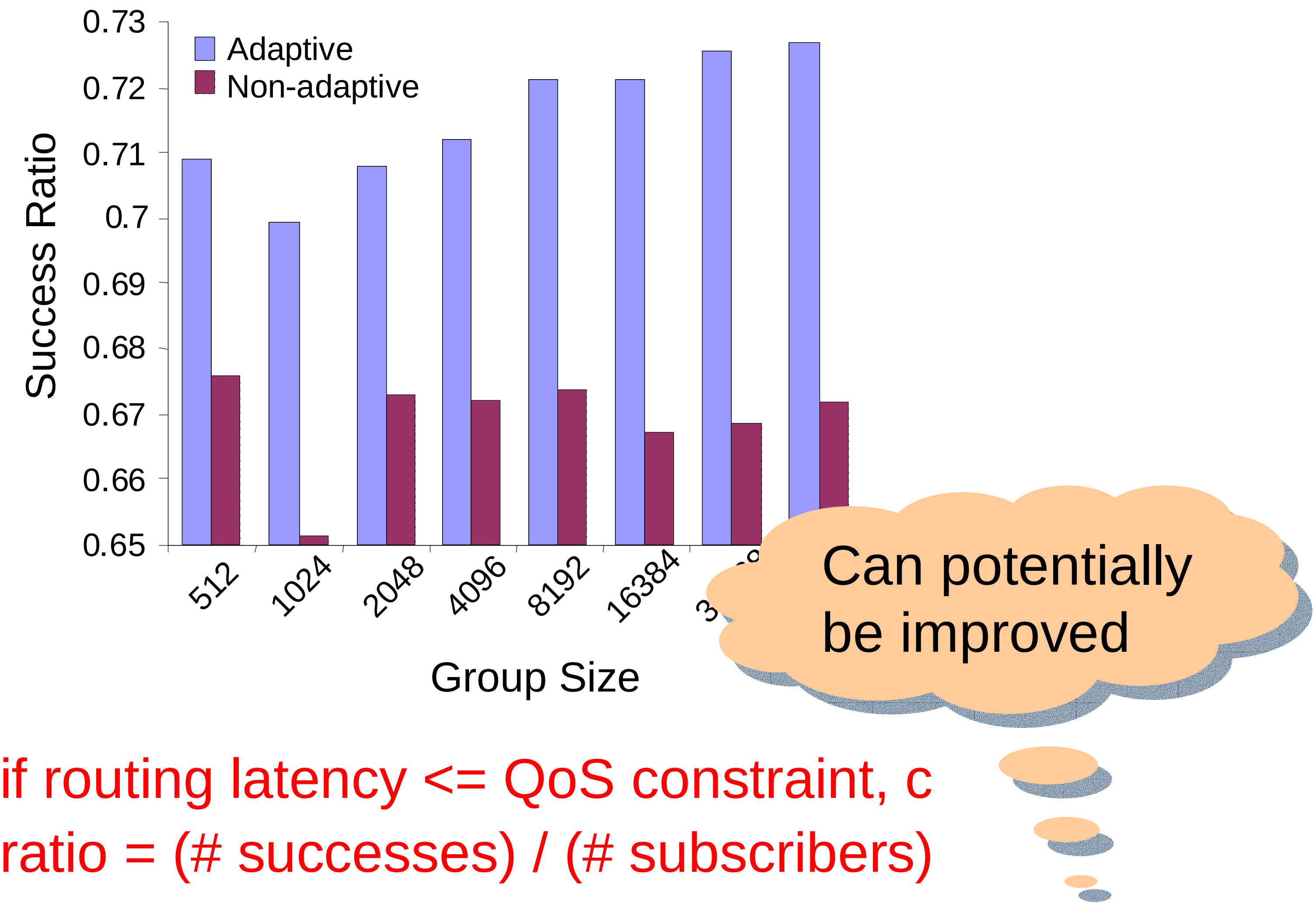
Computer Science

- Randomly generated physical topology with 5050 routers
- M=65536 and topology is a 16D hypercube
- Randomly chosen publisher plus some number of subscribers with QoS (latency) constraints

- Adaptive algorithm used with D=1
- Greedy routing performed with & without adaptive node assignment

Success Ratio vs Group Size

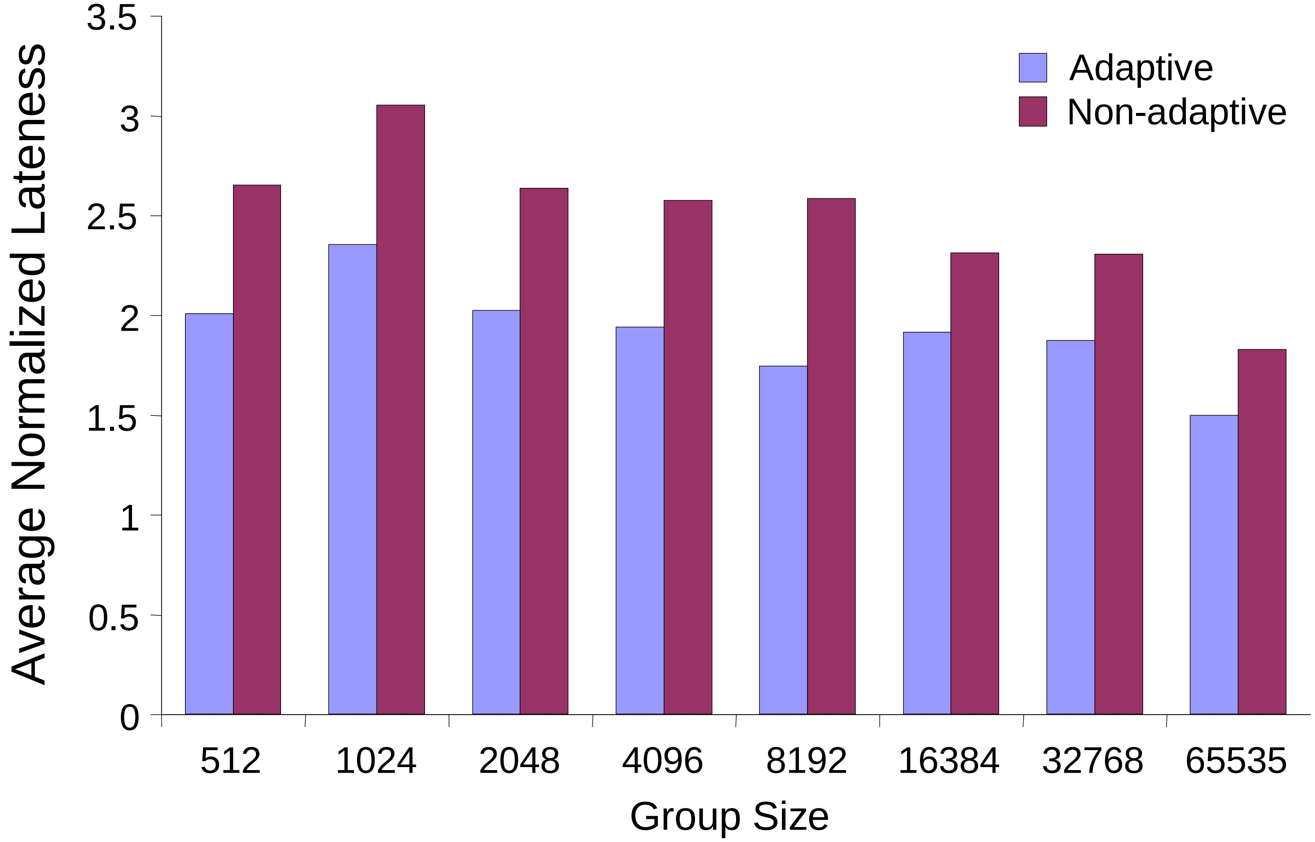
Computer Science



- Success if routing latency <= QoS constraint, c
- Success ratio = (# successes) / (# subscribers)
- Adaptive node assignment shows up to 5% improvement

Lateness versus Group Size

Computer Science



- Normalized lateness = 0, if S.cost(P) <= c
- Normalized lateness = (S.cost(P)-c)/c, otherwise
- Adaptive method can yield >20% latency reduction

Adaptive Node ID Assignment

Computer Science

- Initial results look encouraging
- Improved performance likely if adaptation considers nodes at greater depth,D, from publishers
  - Expts only considered D=1
- Adaptive node assignment attempts to minimize maximum delay between publishers and subscribers

Link Stress

- Previously, aimed to reduce routing latencies

- Important to consider physical link stress:
  - Avg times a message is forwarded over a given link, to multicast info from publisher(s) to all subscribers

Link Stress Simulation Results

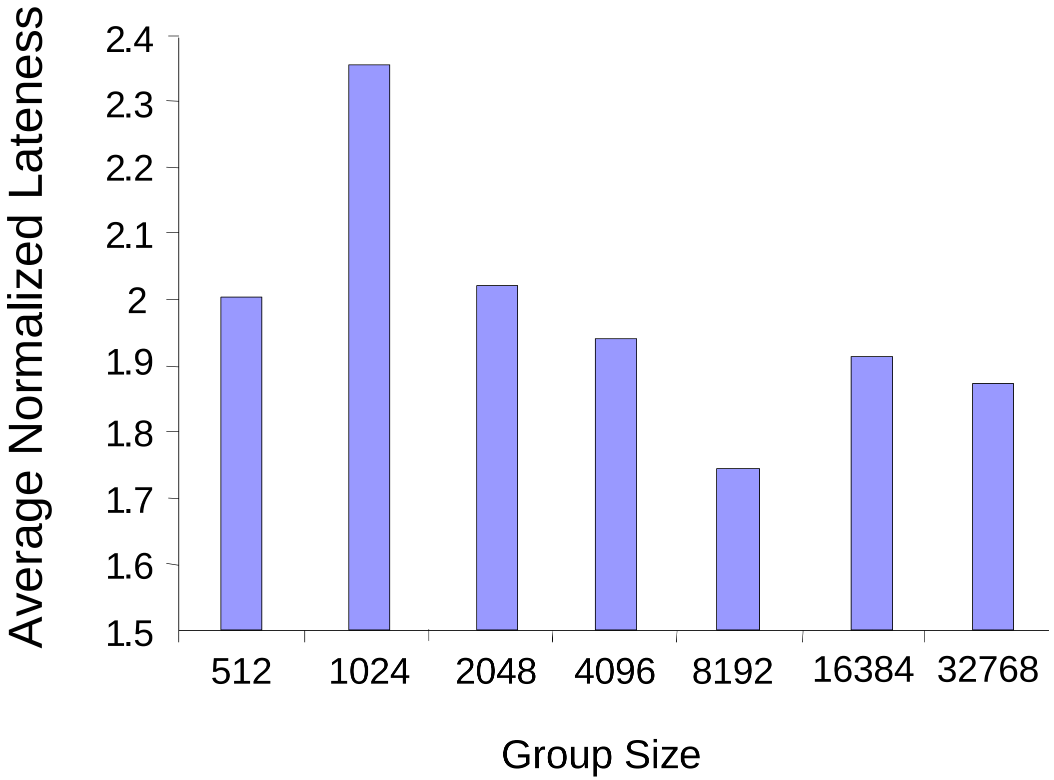
Computer Science

- 16D hypercube overlay on random physical network
- Randomly chosen publisher plus varying groups of subscribers
- Multicast trees computed from union of routing paths between publisher and each subscriber
  - Measure average physical link stress:

$$\frac{\text{(\# times message is forwarded over a link)}}{\text{(\# unique links required to route msg to all subscribers)}}$$

Lateness versus Group Size

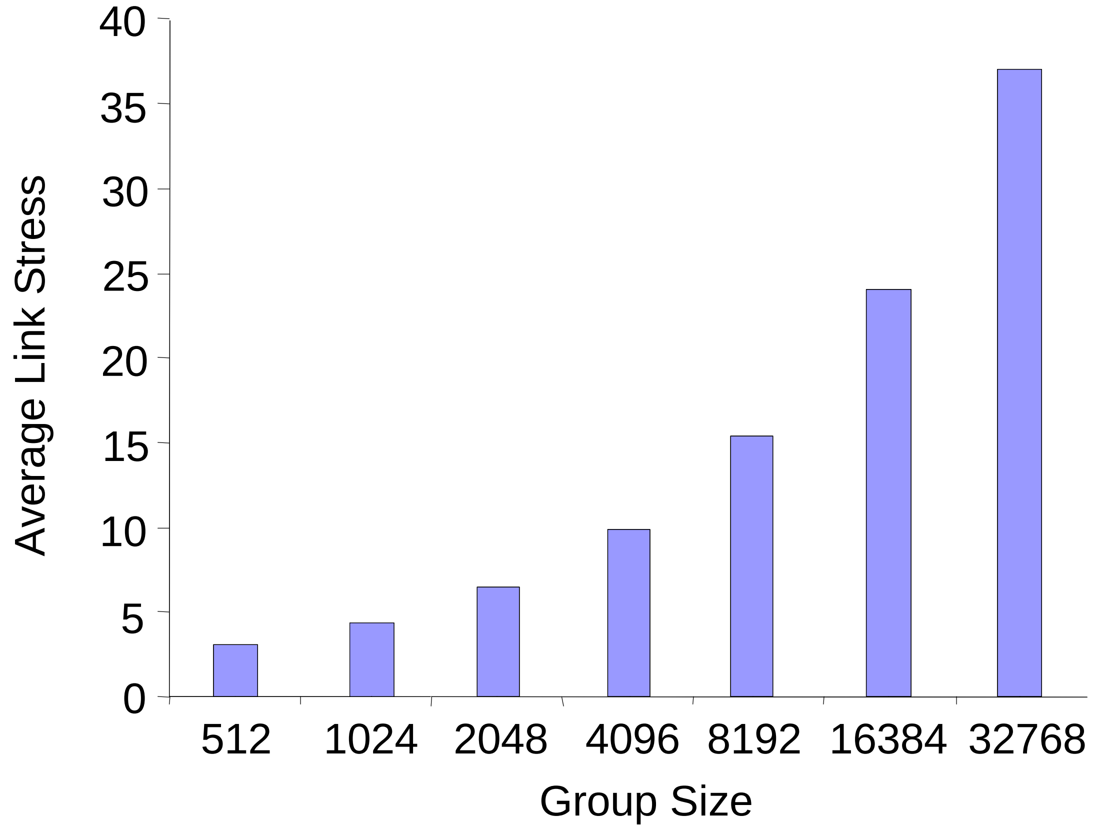
Computer Science



- Variations in lateness (for pairs of columns) due in part to random locations of subscribers relative to publisher

Link Stress versus Group Size

Computer Science



- Greedy routing performs worse as group size increases
- Appears to be due to greater intersection of physical links for multicast tree (i.e. fewer physical links)

Conclusions

Computer Science

- Analysis of k-ary n-cube graphs as overlay topologies
  - Minimal average hop count
  - M-region analysis determines optimal values for k and n.
- Greedy routing
  - Leverages physical proximity information
  - Significantly lower delay penalties than existing approaches based on P2P routing
- Adaptive node ID re-assignment for satisfying QoS constraints

Future and Ongoing Work

Computer Science

- Further investigation into alternative adaptive algorithms
- How does changing the overlay structure affect per-subscriber QoS constraints?
- Analysis of stability as hosts join and depart from the system
- Goal is to build an adaptive distributed system
  - QoS guarantees of NARADA
  - Scalability of systems such as Pastry/Scribe