Using Gossip to Build Network Overlays.

Ken Birman

Cornell University. CS5410 Fall 2008.

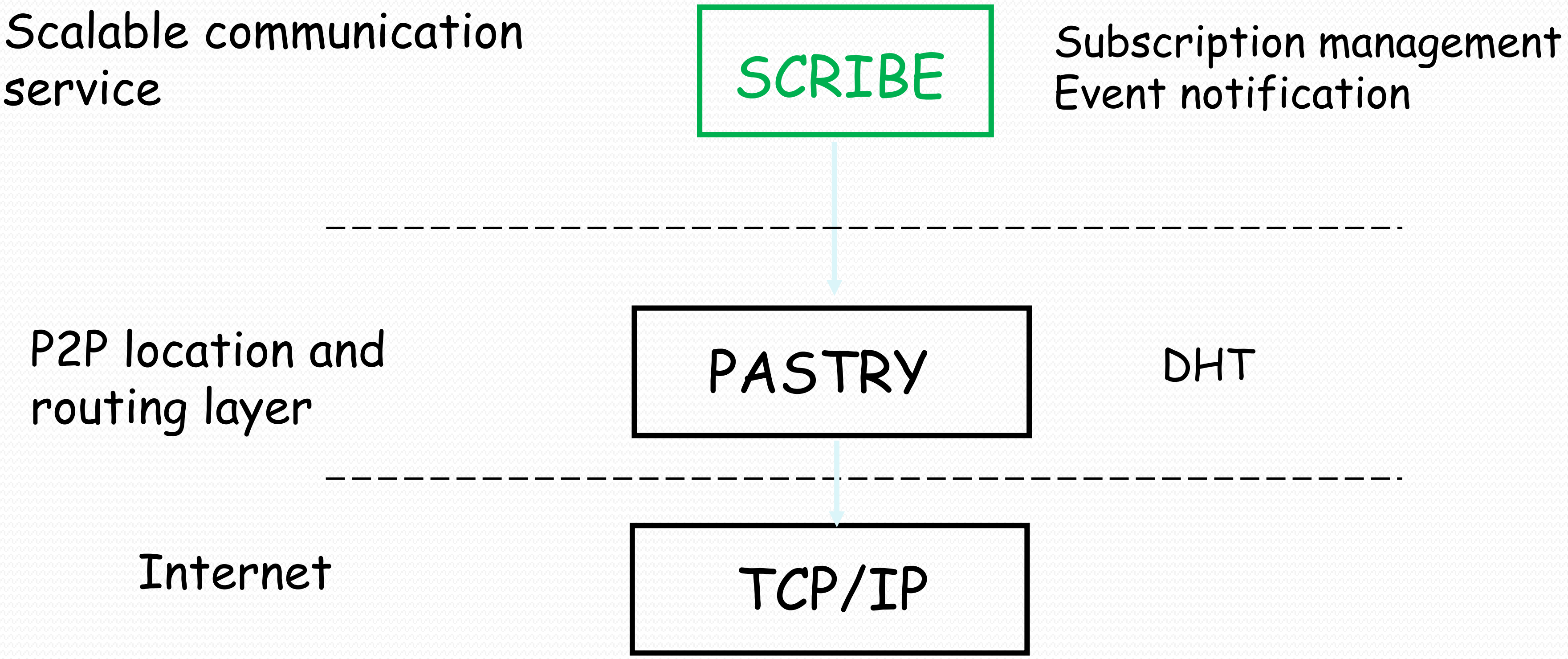
Gossip and Network Overlays

- A topic that has received a lot of recent attention
- Today we'll look at three representative approaches
  - Scribe, a topic-based pub-sub system that runs on the Pastry DHT (slides by Anne-Marie Kermarrec)
  - Sienna, a content-subscription overlay system (slides by Antonio Carzaniga)
  - T-Man, a general purpose system for building complex network overlays (slides by Ozalp Babaoglu)

Scribe

- Research done by the Pastry team, at MSR lab in Cambridge England
- Basic idea is simple
  - Topic-based publish/subscribe
  - Use topic as a key into a DHT
    - Subscriber registers with the “key owner”
    - Publisher routes messages through the DHT owner
  - Optimization to share load
    - If a subscriber is asked to forward a subscription, it doesn’t do so and instead makes note of the subscription. Later, it will forward copies to its children

Architecture



Scalable communication service

SCRIBE

Subscription management
Event notification

P2P location and routing layer

PASTRY

DHT

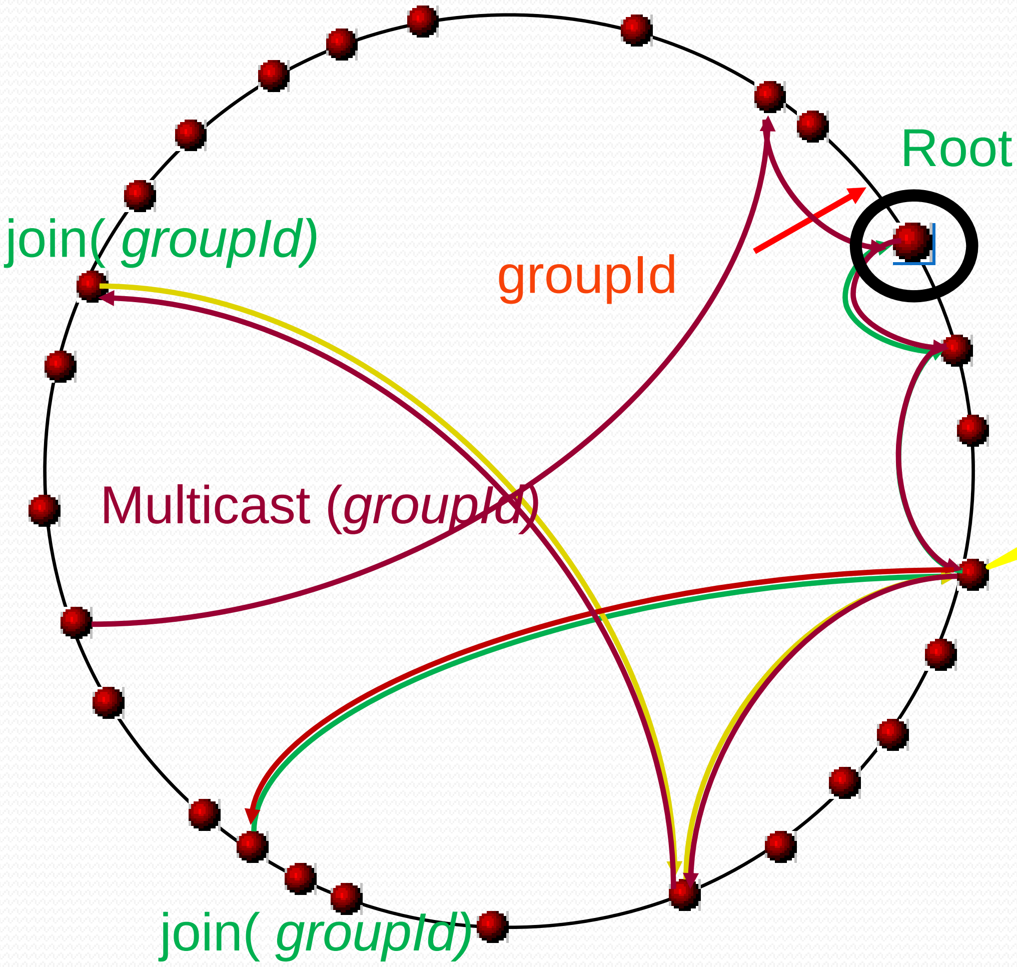
Internet

TCP/IP

Design

- Construction of a multicast tree based on the Pastry network
  - Reverse path forwarding
  - Tree used to disseminate events
- Use of Pastry route to create and join groups

SCRIBE: Tree Management



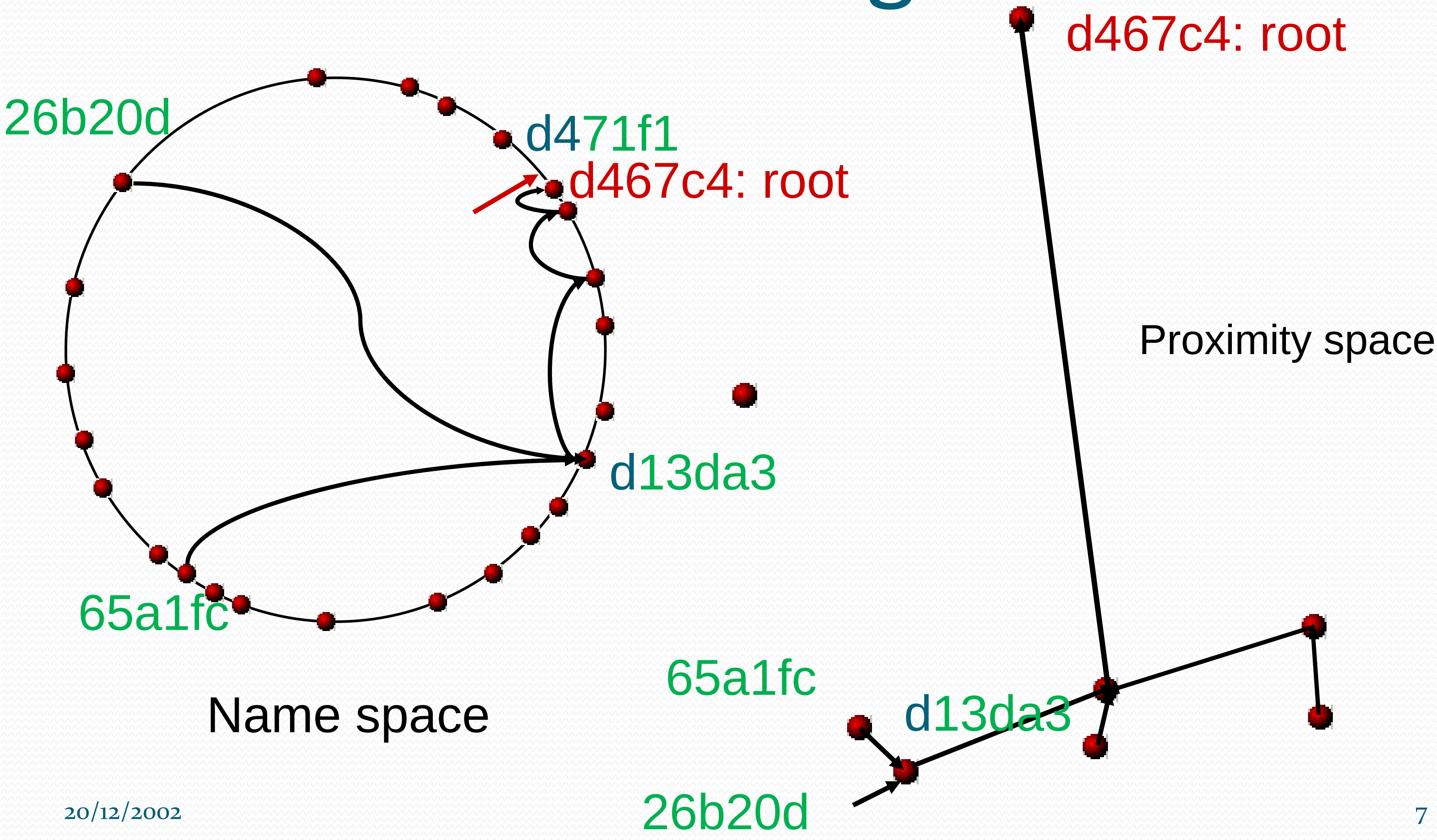
Forwards two copies

- Create: route to groupId
- roupId
  Pastry
  es from members
  to the root.
- Multicast: from the root down to the leaves

  Low link stress

  Low delay

SCRIBE: Tree Management



Concerns?

- Pastry tries to exploit locality but could these links send a message from Ithaca… to Kenya… to Japan…
- What if a relay node fails? Subscribers it serves will be cut off
  - They refresh subscriptions, but unclear how often this has to happen to ensure that the quality will be good
  - (Treat subscriptions as “leases” so that they evaporate if not refreshed… no need to unsubscribe…)

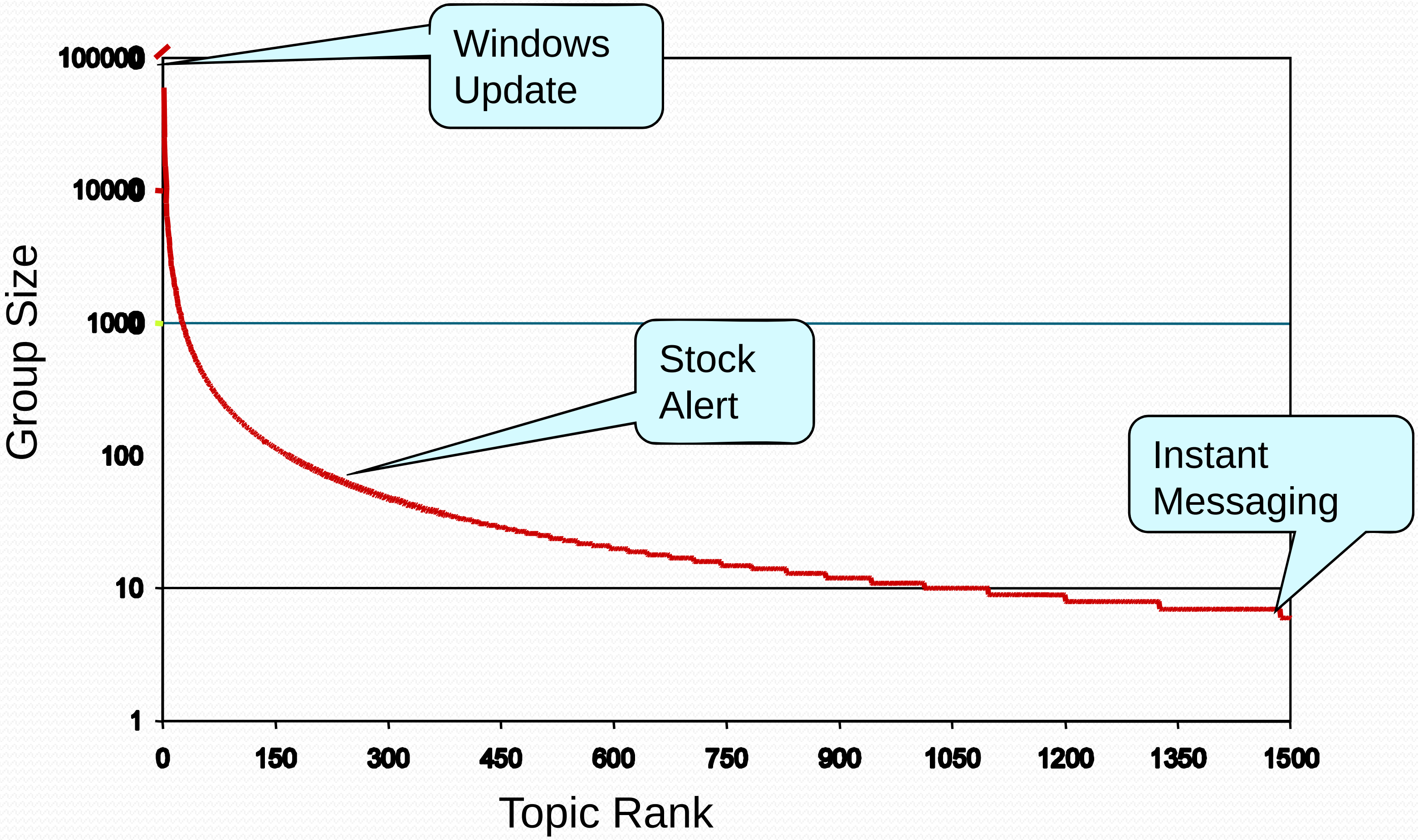
SCRIBE: Failure Management

- Reactive fault tolerance
- Tolerate root and nodes failure
- Tree repair: local impact
  - Fault detection: heartbeat messages
  - Local repair

Scribe: performance

- 1500 groups, 100,000 nodes, 1msg/group
- Low delay penalty
- Good partitioning and load balancing
  - Number of groups hosted per node : 2.4 (mean) 2 (median)
- Reasonable link stress:
  - Mean msg/link : 2.4 (0.7 for IP)
  - Maximum link stress: 4*IP

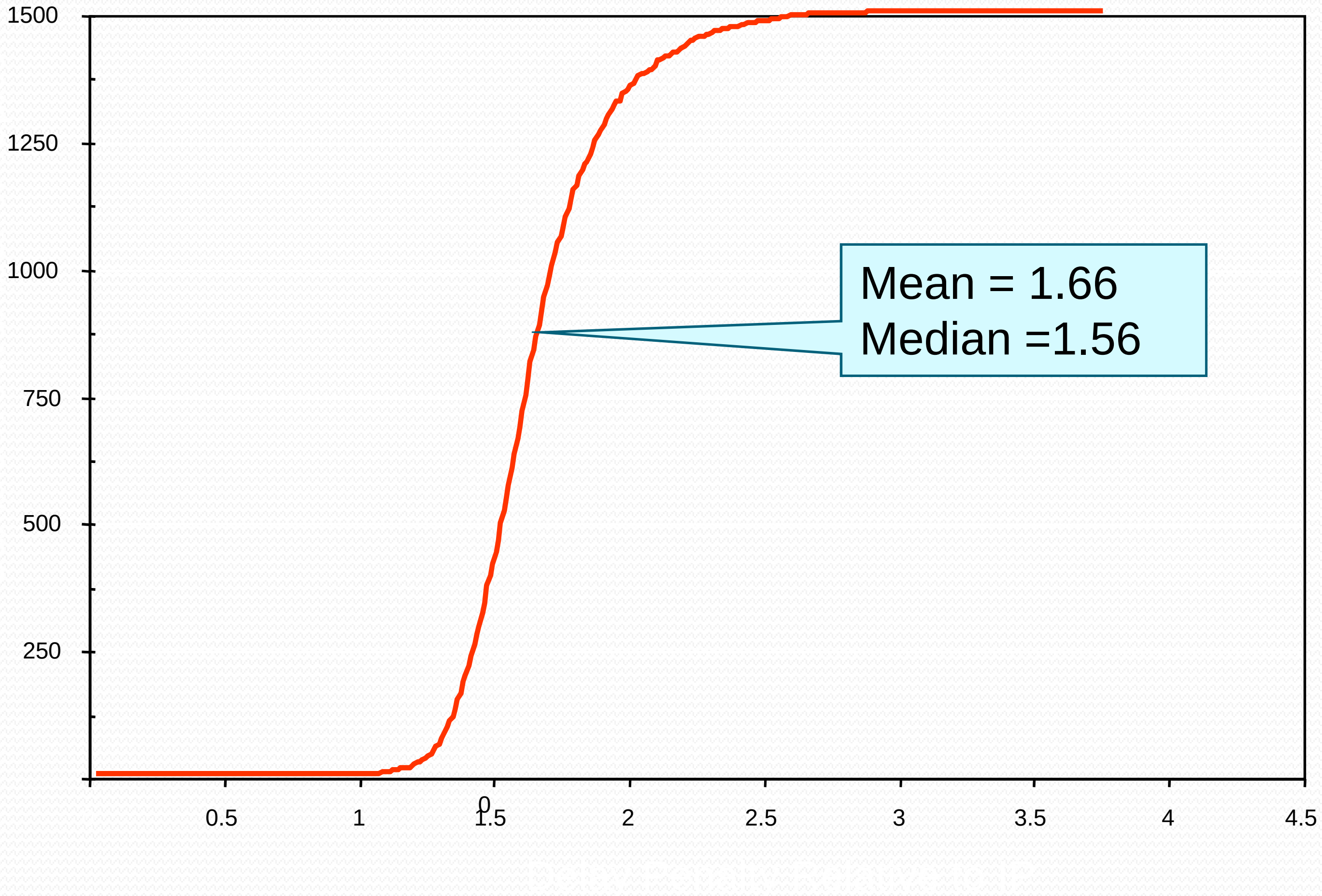
Topic distribution



Concern about this data set

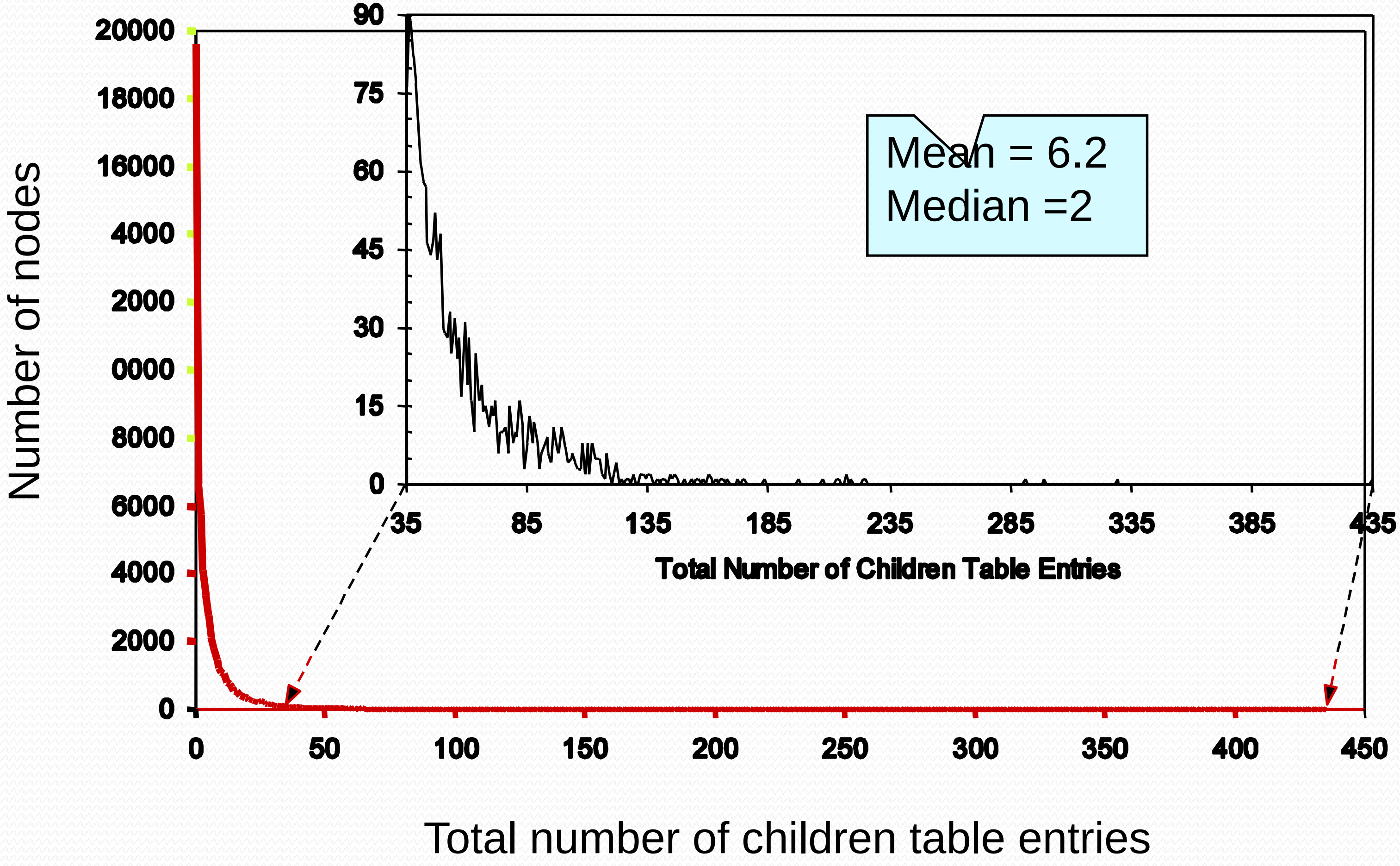
- Synthetic, may not be terribly realistic
  - In fact we know that subscription patterns are usually power-law distributions, so that's reasonable
  - But unlikely that the explanation corresponds to a clean Zipf-like distribution of this nature (indeed, totally implausible)
  - Unfortunately, this sort of issue is common when evaluating very big systems using simulations
  - Alternative is to deploy and evaluate them in use… but only feasible if you own Google-scale resources!

Delay penalty



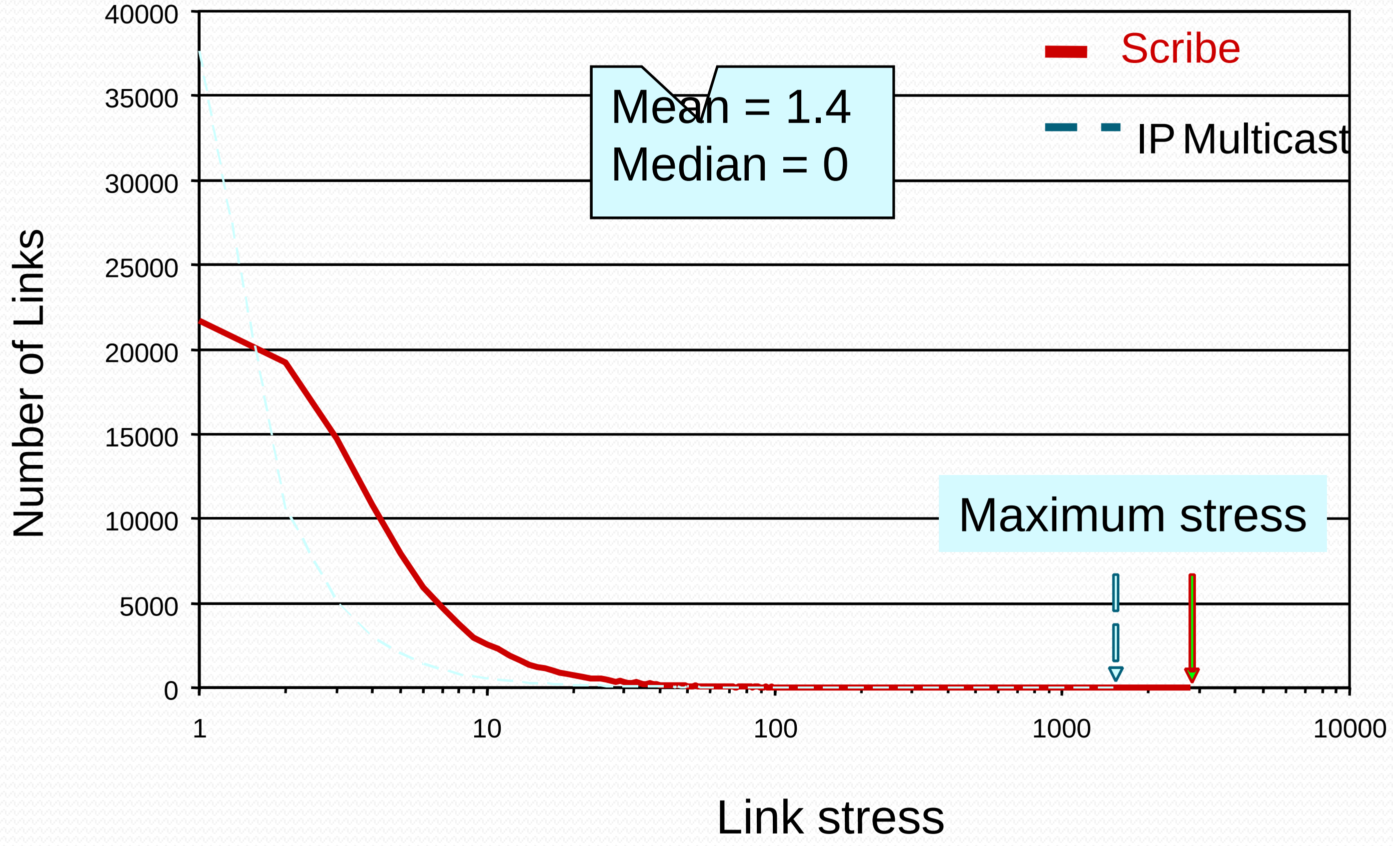
Mean = 1.66
Median =1.56

Node stress: 1500 topics



Mean = 6.2
Median =2

Link stress



Number of Links
40000
35000
30000
25000
20000
15000
10000
5000
0
1
10
100
1000
10000
Link stress
Scribe
IP Multicast
Mean = 1.4
Median = 0
Maximum stress

Anycast

- Supports highly dynamic groups
- Suitable for decentralized resource discovery (can add predicate during DFS)
- Results (100k nodes/.5M network):
    - Join: 4.1 msgs (empty group); avg 3.5 msgs (2,500 members)
    - 1,000 anycasts: 4.1 msg (empty group); avg 2.3 msgs (2,500 members)
    - Locality: For >90% of anycasts, <7% of member were closer than the receiver

Fireflies

Fireflies.ppt

T-Man

T-Man