

# “High-throughput Distributed Services”

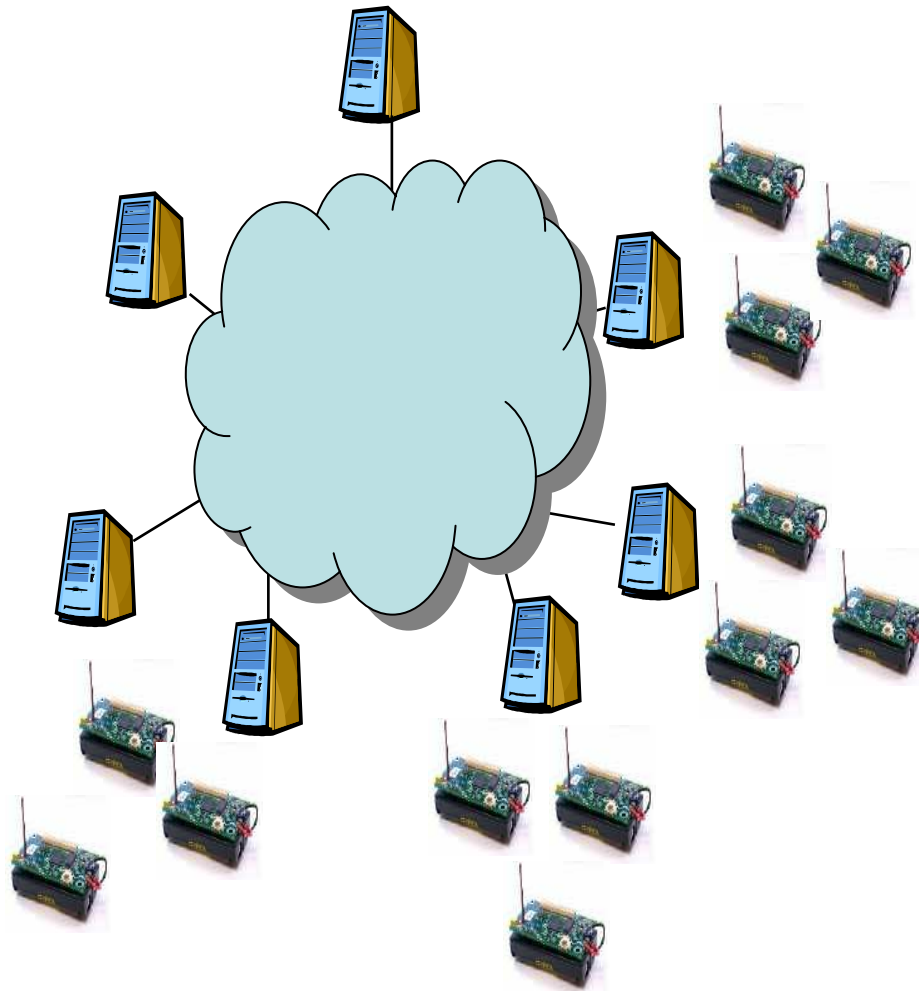
Dejan Kostić

Networked Systems Lab

EPFL

NeXtworking '07

# Towards the Global “Sensor Internet”



- Price of wireless sensor technologies diminishing rapidly
  - large numbers of WSNs deployed in the near future
- WSNs will be interconnected
  - global “Sensor Internet”
    - Global Sensor Network [MDM '07], Hourglass, IrisNet, etc.

# Application #1

- Aggregation of sensor measurements
  - Use scalable, decentralized, highly-available data store, such as a DHT
- Global Sensor Network measurements are small, < 100 bytes
- Have to insert **billions of small identifier-value pairs** into a DHT

# Towards the Next-gen Internet

- Last-mile link capacities reaching 10s of Mbps
- Broadband users demand easy access to entertainment
- **Leverage the increased end host bandwidth to provide:**
  - Better service, and
  - New functionality

# Application #2: Video-on-Demand

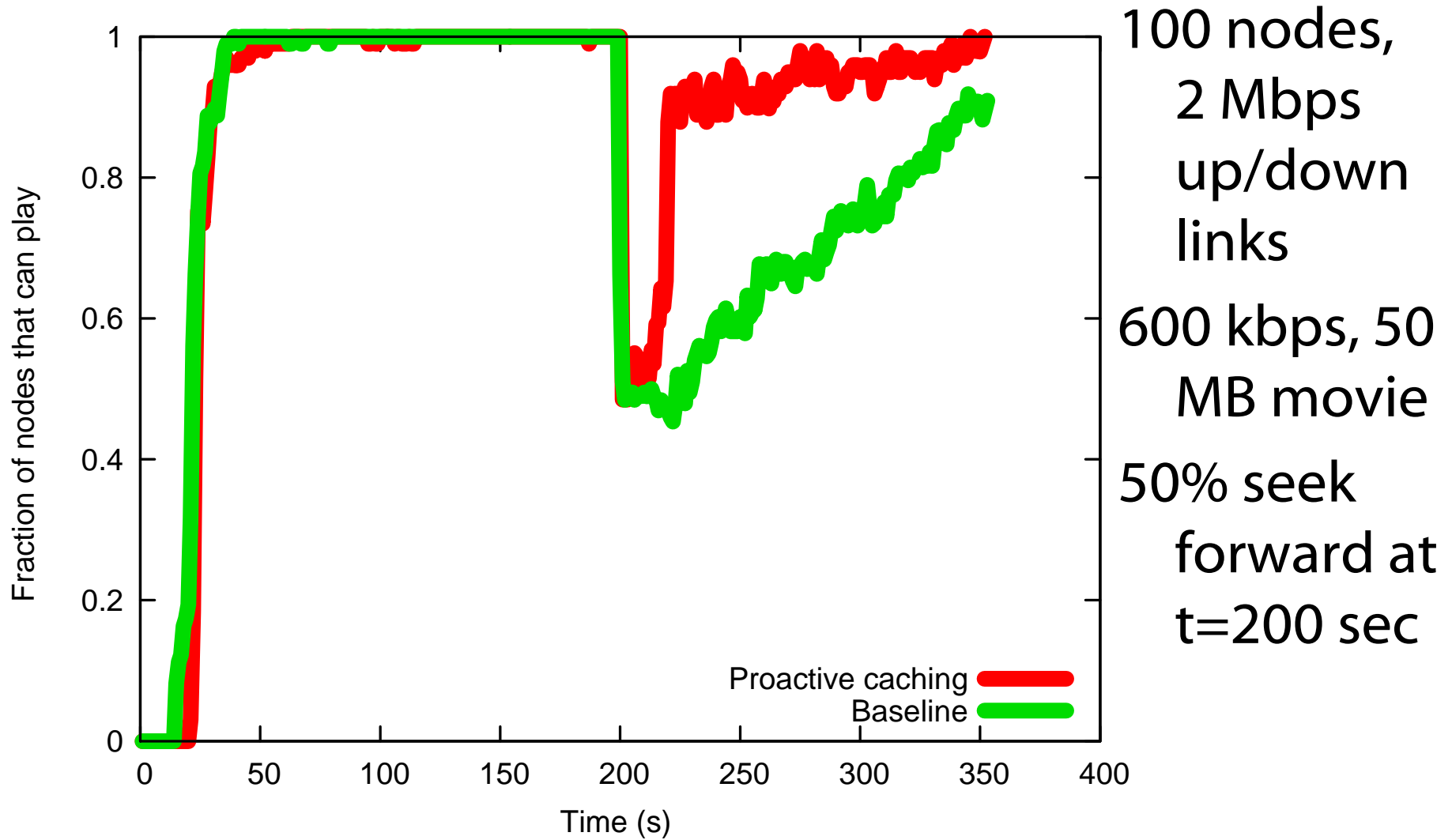
- Customer demand
  - Real-time TV streaming is popular
  - Downloads of TV shows soon after the first airing
- Commercial VoD push
  - YouTube, Apple, AOL, Yahoo, Zattoo, Joost
  - Media companies realizing the benefits, developing new service models
- Existing large-scale Video-on-Demand systems and proposals either:
  - Do not have VCR/DVD features, or
  - Can easily overwhelm the content source

# Accommodating Random Seeks/FF

- Use **proactive caching** in a P2P VoD system to leverage the extra bandwidth
- Quickly establish (and maintain) multiple copies of content **in-overlay**
  - enable advanced features without requiring a well provisioned server

[With Ant Rowstron, MSR Cambridge]

# Benefits of Proactive Caching in P2P VoD



# Challenges

- Controlling content block replication under dynamic conditions (e.g., [TotalRecall, NSDI '04])
- Maximizing the use of spare bandwidth without hurting playback
  - Multi-overlay bandwidth provisioning  
[2<sup>nd</sup> Arcadia meeting, Dec '06, Lisbon]
- Rapid discovery of content blocks when peers perform seeks
  - High throughput “block resolver”
    - Can use a DHT

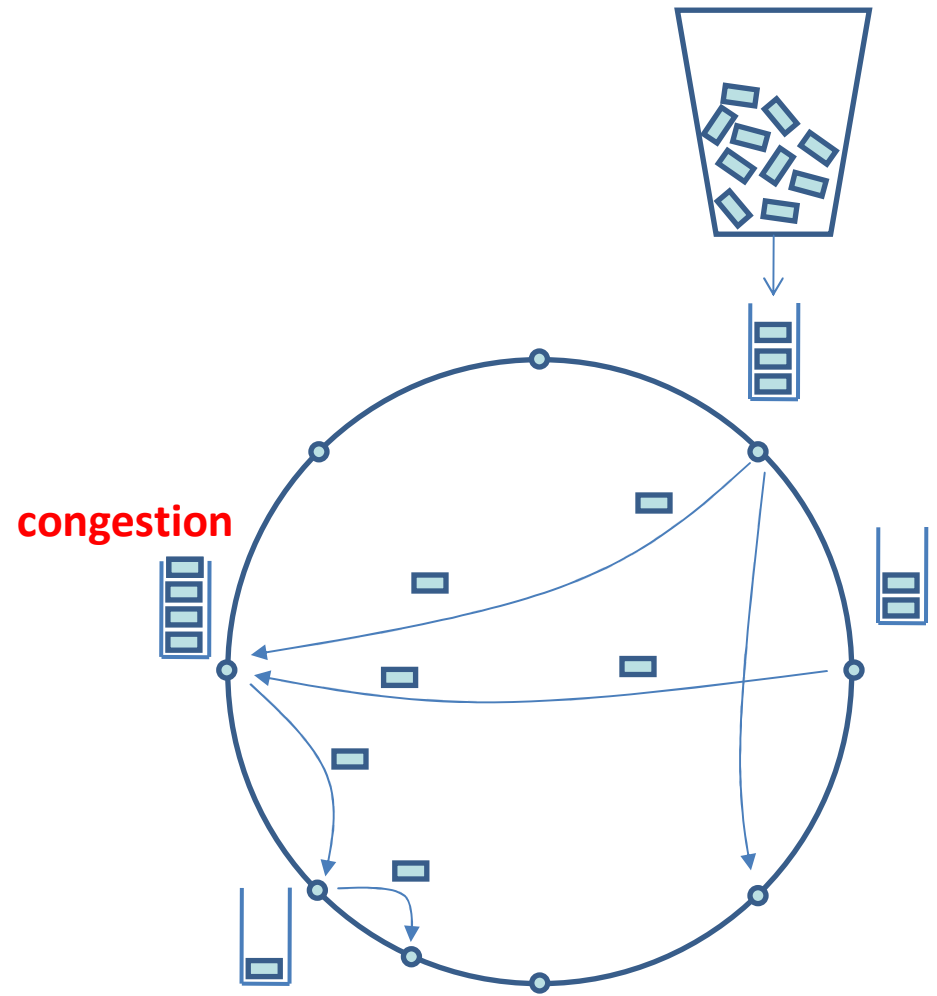
# High-throughput DHTs

- Build a DHT that can efficiently handle **large numbers** of (small) packets
  - Assumption: the bottleneck is routing and not the task performed at the destination.

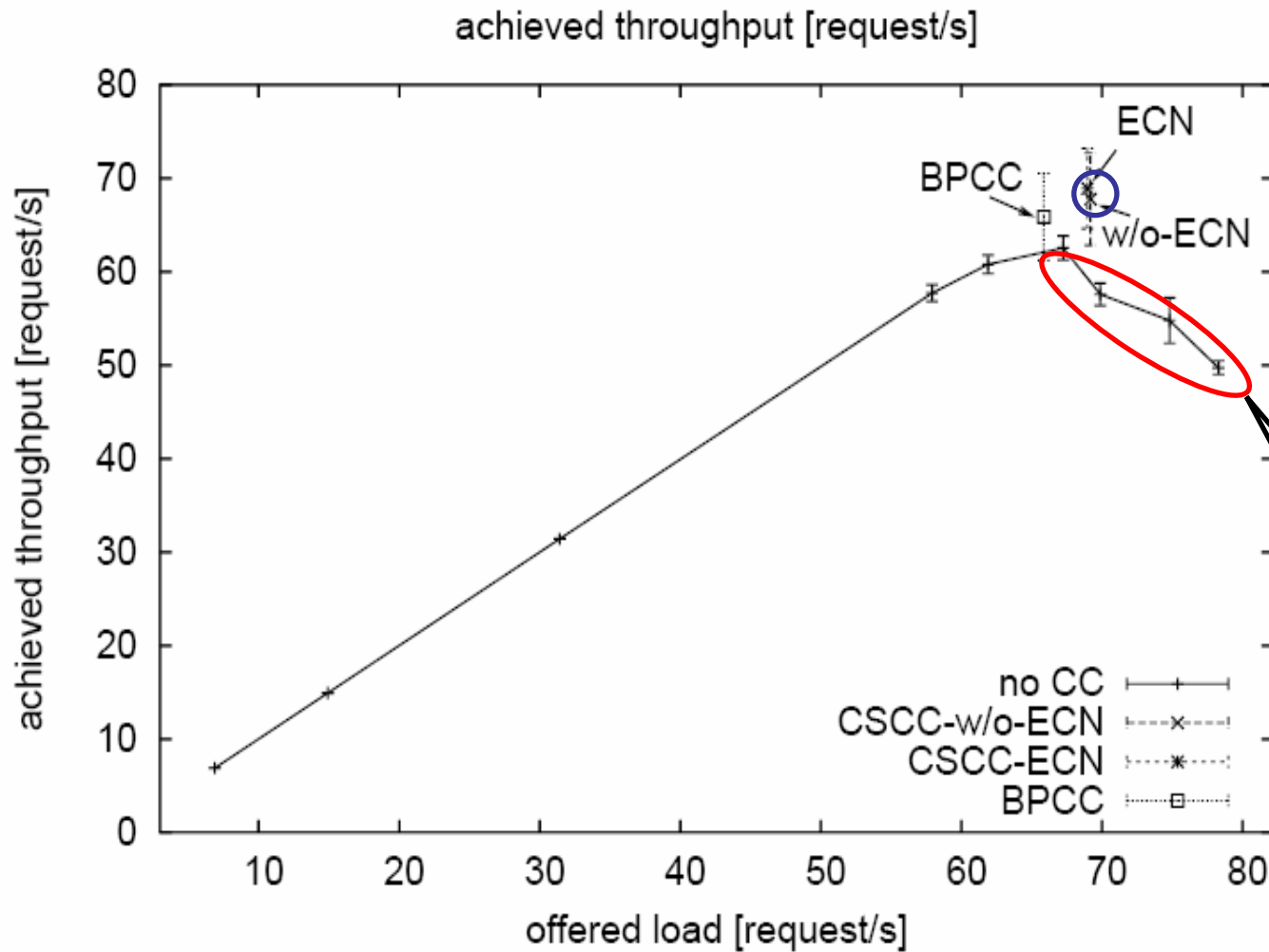
[With Karl Aberer and Jean-Yves Le Boudec]

# Congestion Control for DHTs

- Assume recursive routing:  
Peers relay packets en route to destination
- Congestion: a peer receives more packets than it can handle → drops packets
  - low link bandwidth
  - slow CPU
- Risk: **congestion collapse**
  - all “relay effort” is destroyed
- Goal: Control packet insertions to meet routing capacity in the DHT  
→ congestion control



# Experimental Results



128 node  
ModelNet  
experiment,  
600 kbps  
up/down  
links

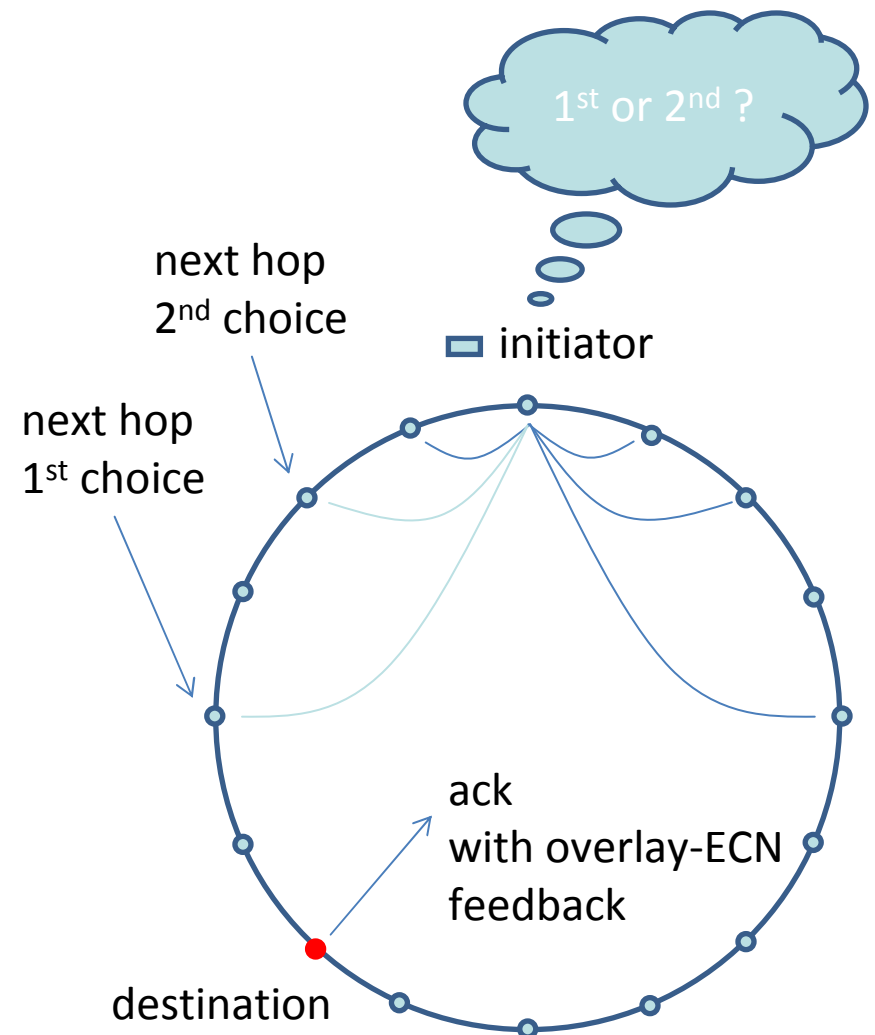
congestion  
collapse

# Step Further: Maximize throughput

- **Congestion-aware** routing:
  - Design a routing algorithm for DHTs that adapts to available capacities in the overlay network
    - Increase throughput relative to strictly greedy routing

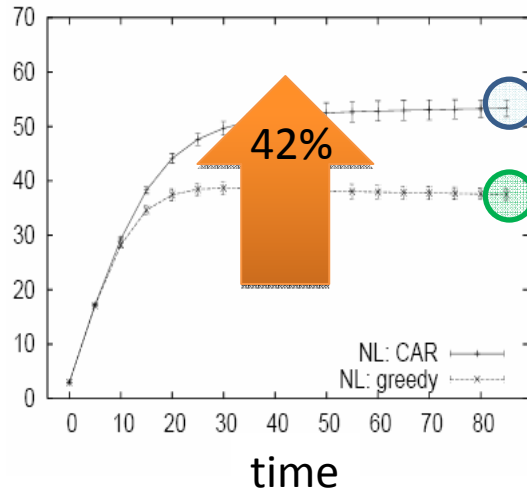
# Congestion-Aware Routing

- Loosen greedy routing
- Consider the  $k$  closest neighbors to the searched destination for the next hop
- Successful insert returns congestion feedback
- Adjust forwarding choices accordingly

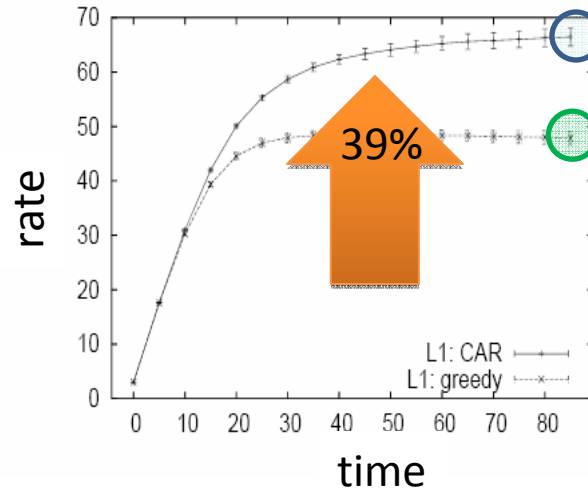


# Initial Results

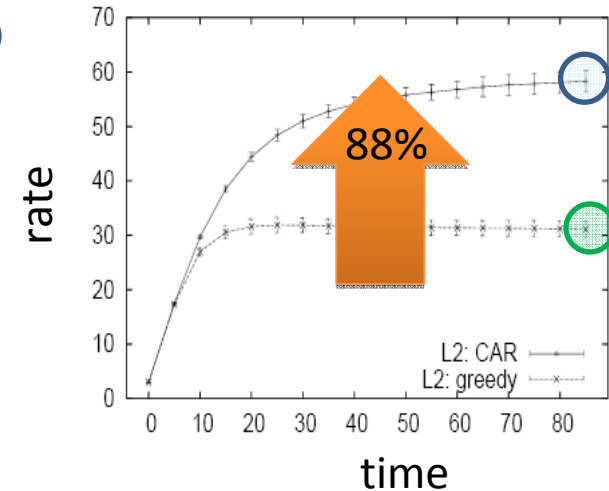
- Analysis with 256 peers, 14 routing entries ( $2 * \log_2(n-1)$ )
- Peers start with rate = 0 and use AIMD to control query rate



no locality:  
rate increase: 42%  
hop inc: 11%



L1: close on the ring  
→ physically close  
rate increase: 39%  
hop increase: 11%



L2: physically close  
long-range neighbors  
rate increase: 88%  
hop increase: 6%

# Conclusion

- New large-scale, **high-throughput** distributed services are important for the future Internet
  - Data aggregation in the Global Sensor Internet
  - (High Definition) Video-on-Demand with advanced features
- Require experimentation on a new testbed
  - Substantial bandwidth, CPU power, storage
    - no PlanetLab-style bw caps, CPU overload, process aborts