

# Exchange Mechanisms and Cooperative Distributed System Design

**PhD Dissertation Defense**

**Kostas G. Anagnostakis**

***CIS Department, University of Pennsylvania***

**Committee:** Michael Greenwald (advisor)

Matt Blaze

Joan Feigenbaum (external)

Honghui Lu

Andre Scedrov

Jonathan M. Smith

# Cooperative Distributed Systems

- Definition:

- distributed systems where autonomous participants pool together their local resources towards a global service

- Examples:

- peer-to-peer systems (file sharing, telephony, backup)
- some types of sensor nets

- Importance:

- *Could* make existing services better, cheaper
- *Could* enable new services

- The gamble:

- autonomy implies lack of trust

# Motivation: Enforcing Cooperation

- The importance of cooperation
  - Usually not guaranteed by design
  - Result of non-cooperation could be system collapse
- Incentive mechanisms to enforce cooperation
  - Hard problem because of local autonomy
  - An economic problem in computational setting
  - A “systems” problem: need to be efficient

# Other incentive mechanisms

## ● **Monetary** mechanisms:

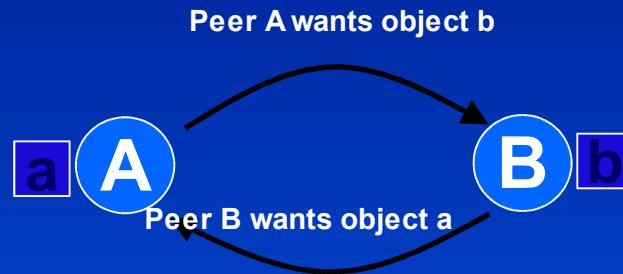
- Actions coupled with economic transactions (payments)
- Requires single point of trust
- Vulnerable to pseudo-spoofing under weak identities
- Highly flexible but potentially very complex

## ● **Reputation** mechanisms:

- Maintain shared history of user behavior
- Vulnerable to pseudo-spoofing under weak identities
- Shown to perform poorly unless there are *some* trusted nodes

# Exchange Mechanisms

- **Definition:** mechanisms where actions are linked to verifiable reciprocal actions between participants in the system
- **Example:** simple file exchange



- **Benefits:** simple, decentralized, memory-less, based directly on evidence of cooperation
- **Potential problem:** “*double coincidence of simultaneous wants*” limits applicability

# Thesis statement

*Although exchange/barter mechanisms are not usually applicable in the real world, they are widely applicable in the setting of cooperative distributed systems.*

# Methodology & Roadmap

- Generalized exchange mechanism

- Demonstrate applicability in specific applications

- . File sharing

- . P2P VoIP-PSTN peering

- Examine exchange in other applications

- . BitTorrent file sharing, P2P backup, query forwarding, hotspot sharing

- Decision procedure determining applicability of exchange

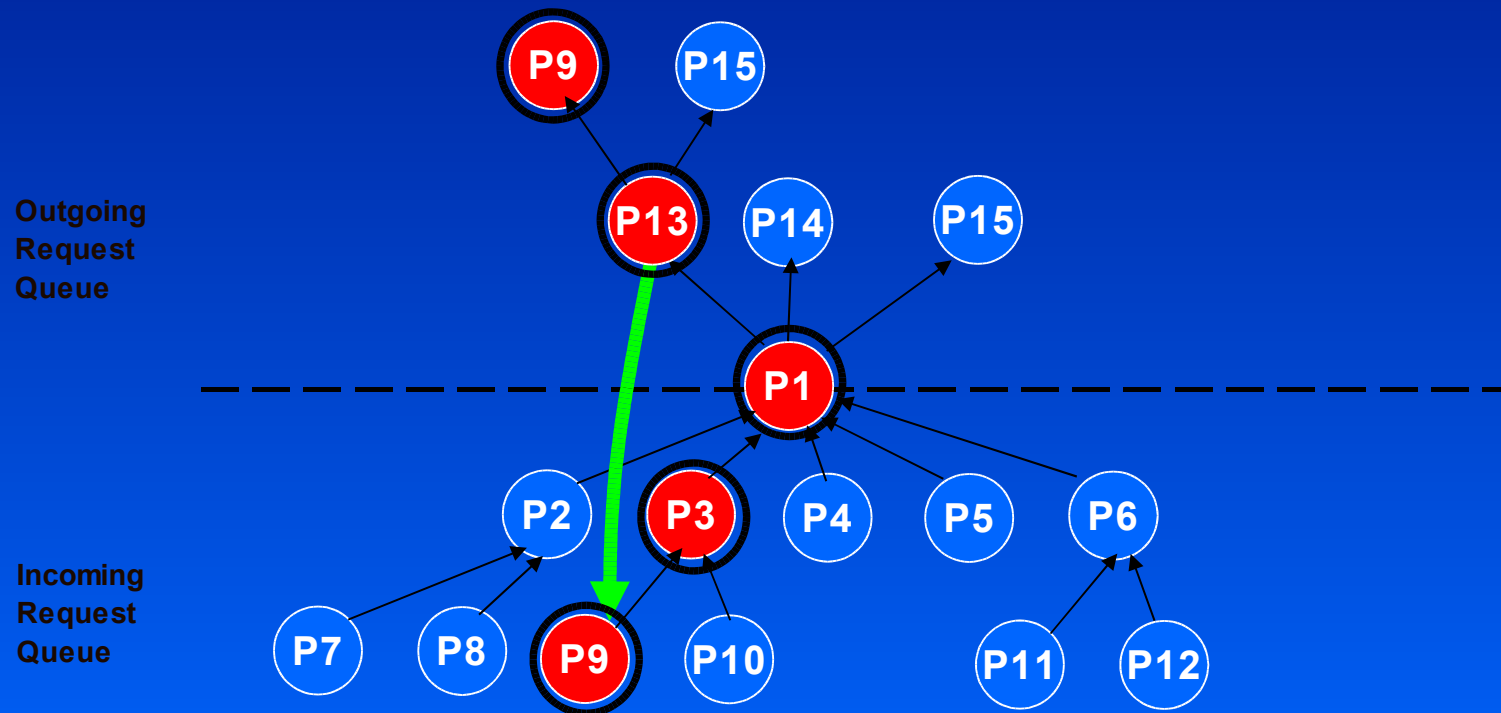
# File sharing: an exchange mechanism



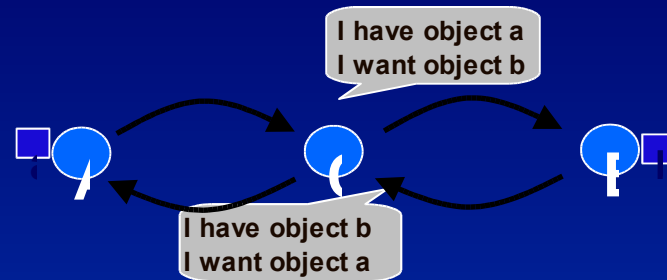
- **Approach:** give upload *priority* to peers that can provide a reciprocal download in return
  - Two-way, or N-way **exchange rings**
  - As long as there is at least one exchange ring for each peer, freeloaders will have to wait in low-priority queues
  - New users still get service, populating their disk and slowly improving their ability to participate in exchanges

# File sharing: Locating exchange rings

- Maintain Incoming and Outgoing Request Queues: **IRQ/ORQ**
- Incrementally extend IRQ and ORQ until some node is on both
- Limit each branch to **K** nodes – complexity  **$O(K^{[N/2]})$**



# Robustness I



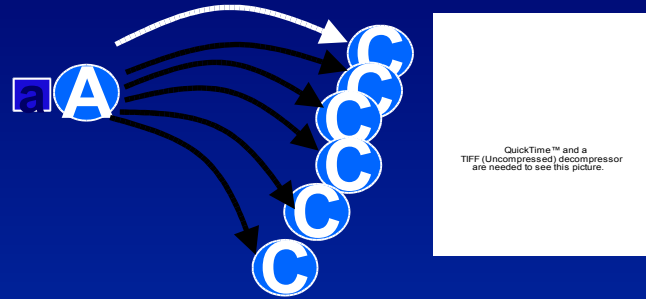
## ● Threats:

- **One-sided defection:** cheater leaves ring after acquiring object
- **False content:** cheater joins ring, sends bogus content
- **Middleman:** cheater joins ring, peddles blocks between A,B.

## ● Solution:

- *Prefer long exchanges, encrypt transfers, embed **peer-id** in transfer*
- *Use mediator to validate content and assist in final exchange of keys*

# Robustness II



## ● Threats:

- **Pseudospoofing:** if exchange makes downloads  $M$  times faster for cooperating peers, cheater can multiply his bandwidth share by signing up with  $M'$  identities

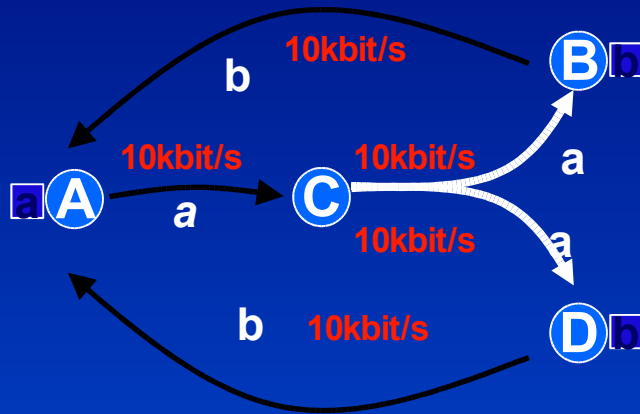
## ● Solution:

- *Freeriding is prevented -- Each identity is equally likely to be chosen on the exchange graph*

# Mixed bandwidth object exchange

- Mixed bandwidth-object exchange

- More effective for **C** than being a middleman
- Lowers “entry cost” for new users, so we can disable non-exchange service



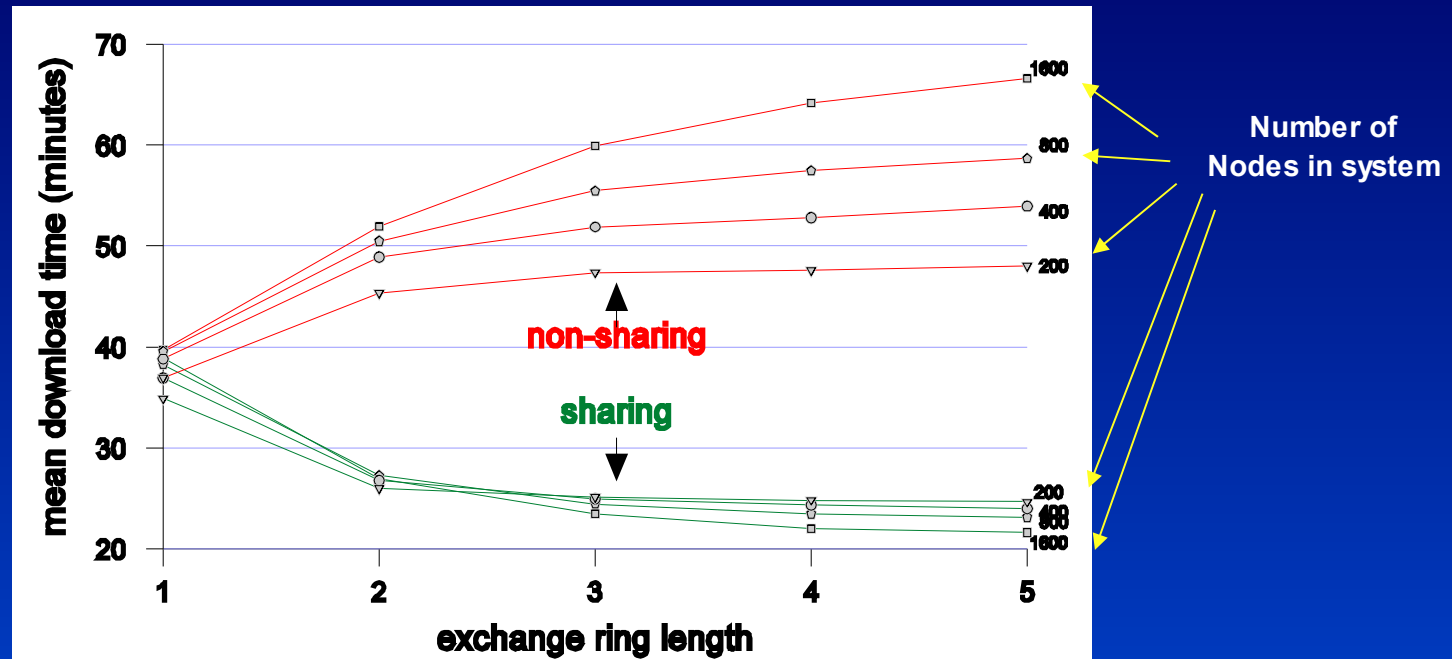
- Conditions:

- **A** is upload bandwidth-limited
- **C** wants **a**, does not have any files, but can act as *amplifier* for **A**
- Works for any set of rings with **a**-edges out of **A**

- Result:

- **A** gets to download object **b** twice as fast
- **B** and **D** both get object **a**
- **C** gets object **a** without serving any local object (but has to spend twice as much bandwidth)

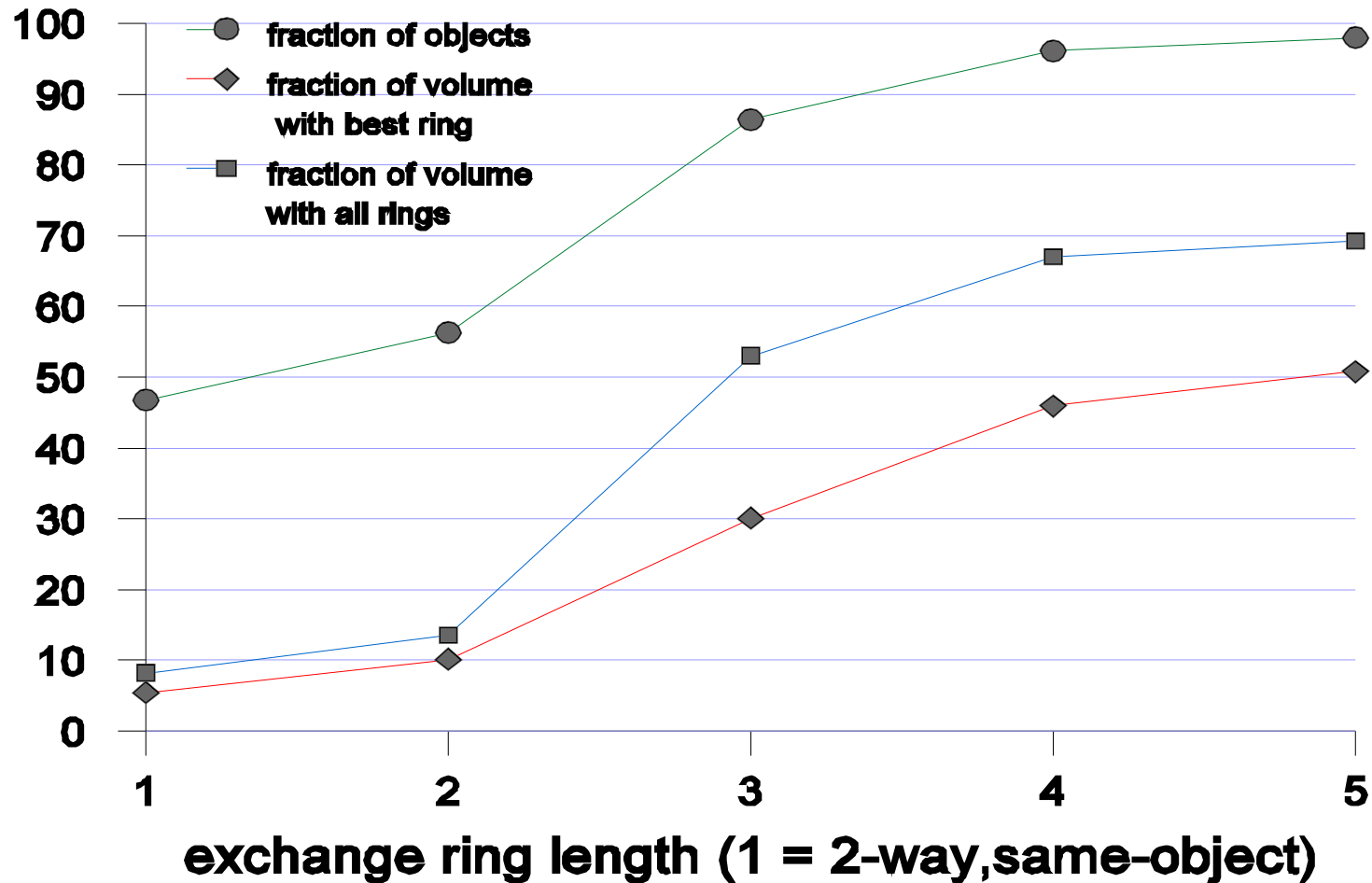
# Effectiveness of exchange: simulation



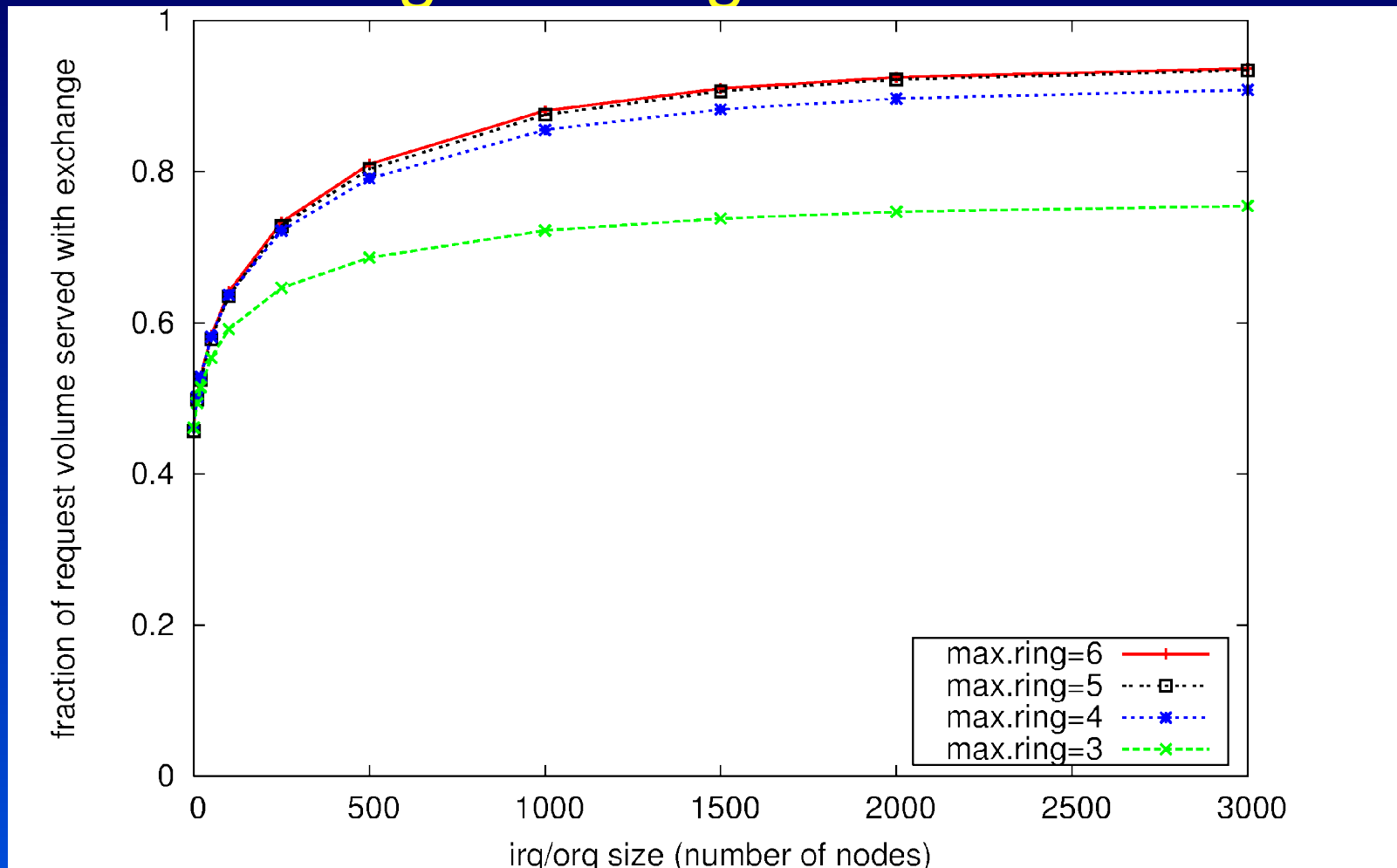
- Strong incentive to share (2x-3x faster downloads)
- Substantial improvement for  $N > 2$
- Diminishing returns for  $N > 4$
- Incentive improves with system scale

# Effectiveness of exchange: real-world data

- Dataset: 30,000 nodes of *eMule* P2P file sharing system
- Computed exchange rings for all requests in dataset



# Cost of locating exchanges: real-world data



- Can achieve 95% effectiveness with IRQ/ORQ sizes of < 3000 nodes.
- For 8-byte node id's this is 24KB << median object size

# Methodology & Roadmap

- Generalized exchange mechanism

- Demonstrate applicability in specific applications

- . File sharing

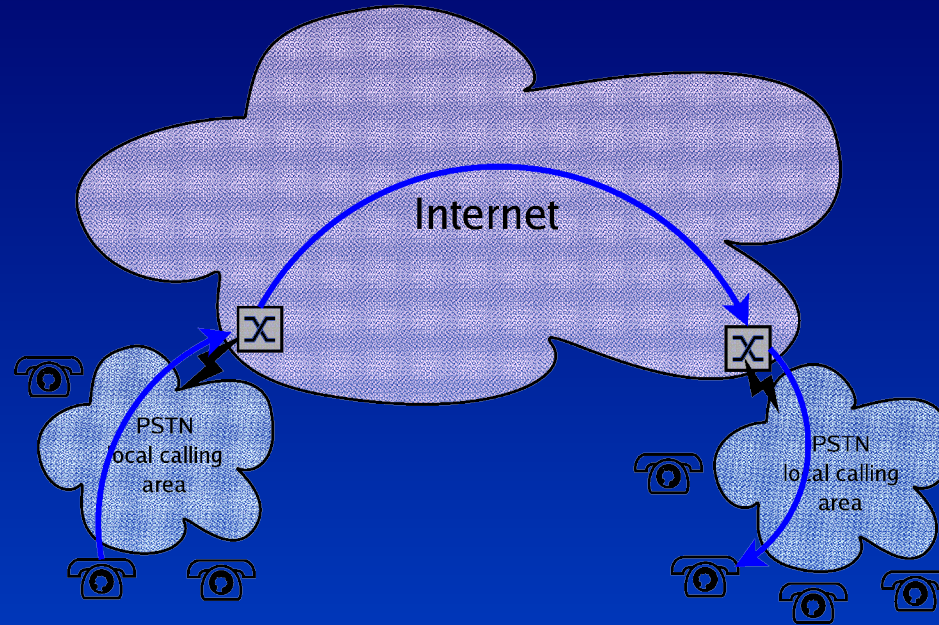
- . P2P VoIP-PSTN peering

- Examine exchange in other applications

- . BitTorrent file sharing, P2P backup, query forwarding, hotspot sharing

- Decision procedure determining applicability of exchange

# Cooperative VoIP-PSTN peering



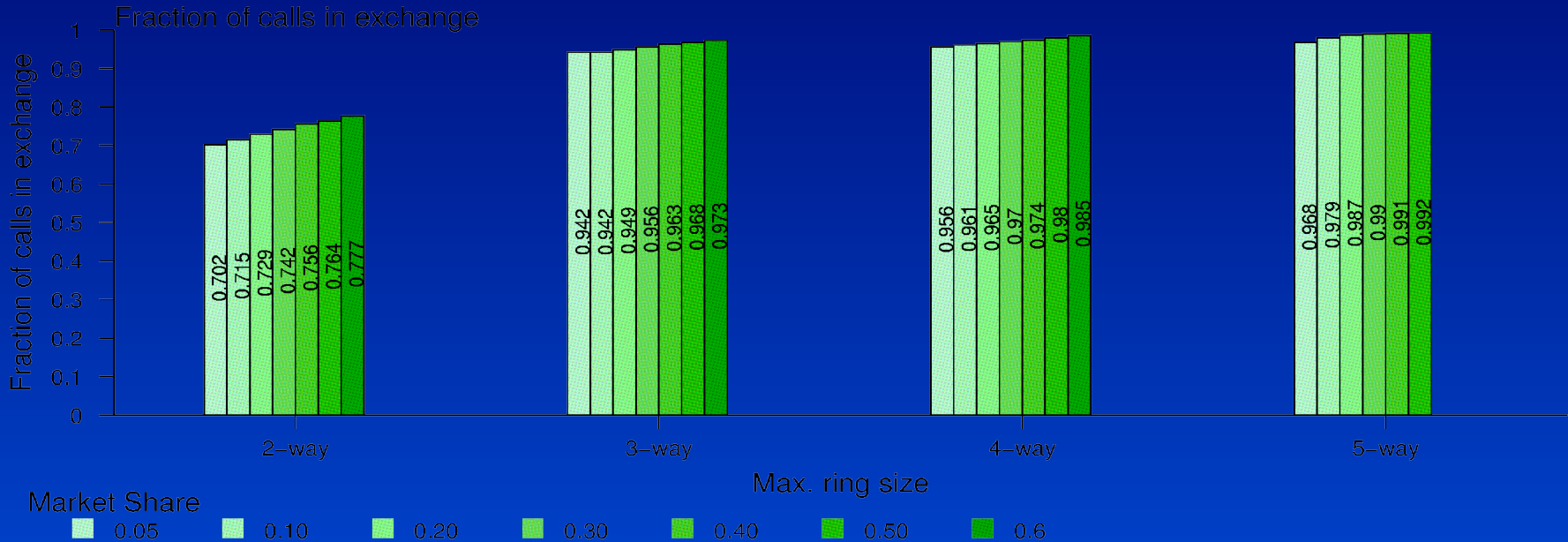
- Free PSTN local calling areas, gateways peering over IP
  - Similar to Skype, but decentralized/cooperative
  - Need incentive for LCA gateways to carry calls
  - Goal is ultra-cheap service, thus cost of enforcing cooperation should be very low

# VoIP-PSTN peering: exchange mechanism

- For each incoming call, try to find matching outgoing call to form exchange
  - Look for two-way, N-way exchanges
  - If no matching outgoing call, delay call establishment
  - If delay reaches timeout, serve call anyway
  - If possible, replace edge when call completes
- Implications
  - Cooperating LCAs that are willing to serve incoming calls,  
can establish their outgoing calls with lower delay
  - Non-cooperating LCAs will always hit the timeout

# VoIP-PSTN peering: simulation experiments

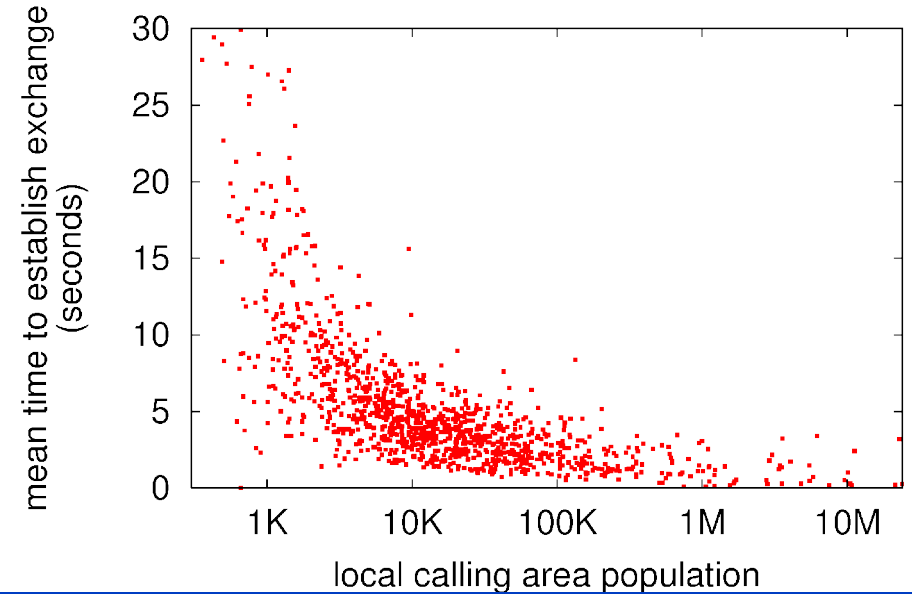
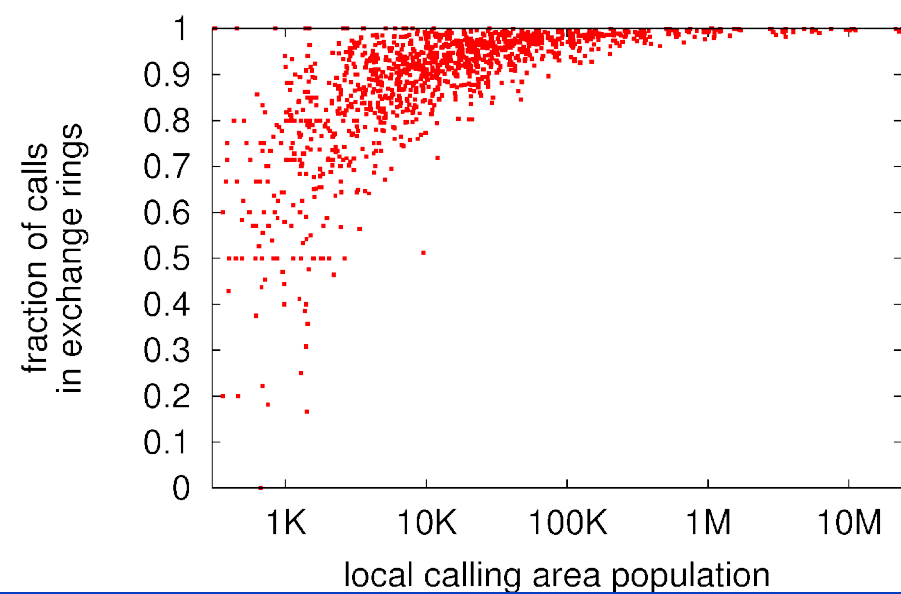
- Simulated 3 hours of calls using model of US long-distance net



- Generally very high fraction of calls in exchange
- N-way makes a big difference → 99.2% coverage for N=5
- Smaller market share → lower exchange probability

*But how good is 99.2% ?*

# VoIP-PSTN peering: simulation experiments



- Works well for large calling areas
- Unfairly penalizes small areas
- Reason is high rate of calls entering the system
- 99.2% was misleading

# Methodology & Roadmap

- Generalized exchange mechanism

- Demonstrate applicability in specific applications

- . File sharing

- . P2P VoIP-PSTN peering

- Examine exchange in other applications

- . BitTorrent file sharing, P2P backup, query forwarding, hotspot sharing

- Decision procedure determining applicability of exchange

## ● Approach:

*provide upload bandwidth to users who  
provide upload bandwidth in return*

- A tit-for-tat two-way same-object exchange

## ● Efficiency:

- Works well for large popular files.

Analysis shows only 10% coverage for general file sharing

## ● Threats:

- One-sided defection after first round
- Aggressively connect to object replicas
- Pseudo-spoofing

**→ Mechanism has flaws, but most are fixable  
see our ICDCS'04 file sharing design**

# Other applications: WiFi hotspot sharing [Efstathiou&Polyzos, 2005]

## ● Approach:

*provide hotspot access to nodes that provide hotspot access in return*

- Non-simultaneous: receipt for access, receipt is edge in exchange graph
- N-way exchanges

## ● Threats:

- Cheater can lurk for newbies that don't have any receipts to form exchange rings
  - . *Plausible threat, especially in early deployment stages*
  - . *Unlikely to be a big problem in the long run*

→ ***Mechanism seems reasonable***

# Other applications: P2P backup

[Cox & Noble,

SOSP'03]

## ● Approach:

*provide backup storage for nodes that  
provide backup storage in return*

- Use placeholders for non-simultaneous exchange
- Probabilistically drop storage for nodes that don't honor reciprocal storage/placeholders

## ● Threats:

- Aggressively backup on many nodes to beat probabilistic drop
  - . *Shown to be impractical for current disk-bandwidth trends*

**→ Mechanism seems to work well.**

***Only instance of fully-decentralized exchange***

## ● Approach:

*forward queries for peers that forward queries in return*

- Adapt own bandwidth based on measured query response rate for each peer
- Essentially a two-way tit-for-tat exchange

## ● Threats:

- Fake responses
- Aggressive connection strategy
- Pseudo-spoofing

**→ Mechanism has major flaws**

***Lack of verification + tit-for-tat weaknesses***

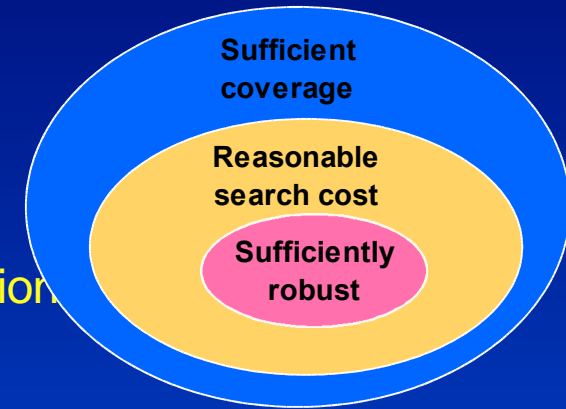
# Methodology & Roadmap

- Generalized exchange mechanism
  - Demonstrate applicability in specific applications
    - . File sharing
    - . P2P VoIP-PSTN peering
  - Examine exchange in other applications
    - . BitTorrent file sharing, P2P backup, query forwarding, hotspot sharing
- Decision procedure determining applicability of exchange

# A Decision Procedure

## ● Determine **coverage**

- Type of symmetric relationship → exchange graph  $G$
- Utility  $U$ , differentiation function  $D$
- Coverage  $C(G)$  sufficient s.t.  $Uc - Unc > threshold$
- Fraction of edges is a reasonable measure, but not always adequate
- Static analysis sometimes ok, but may need simulation



## ● Determine **search cost**

- Choice between IRQ/ORQ algorithm, centralized matchmaking, full-mesh exchange, ...
- No general solution, may need to get creative

## ● Determine **robustness**

- Tit-for tat: level of replication, feasibility of aggressive TFT, strong/weak id's
- One-shot non-simultaneous: possible when users bound to return
- Simultaneous: purest, safest form of exchange, but at the expense of mediator

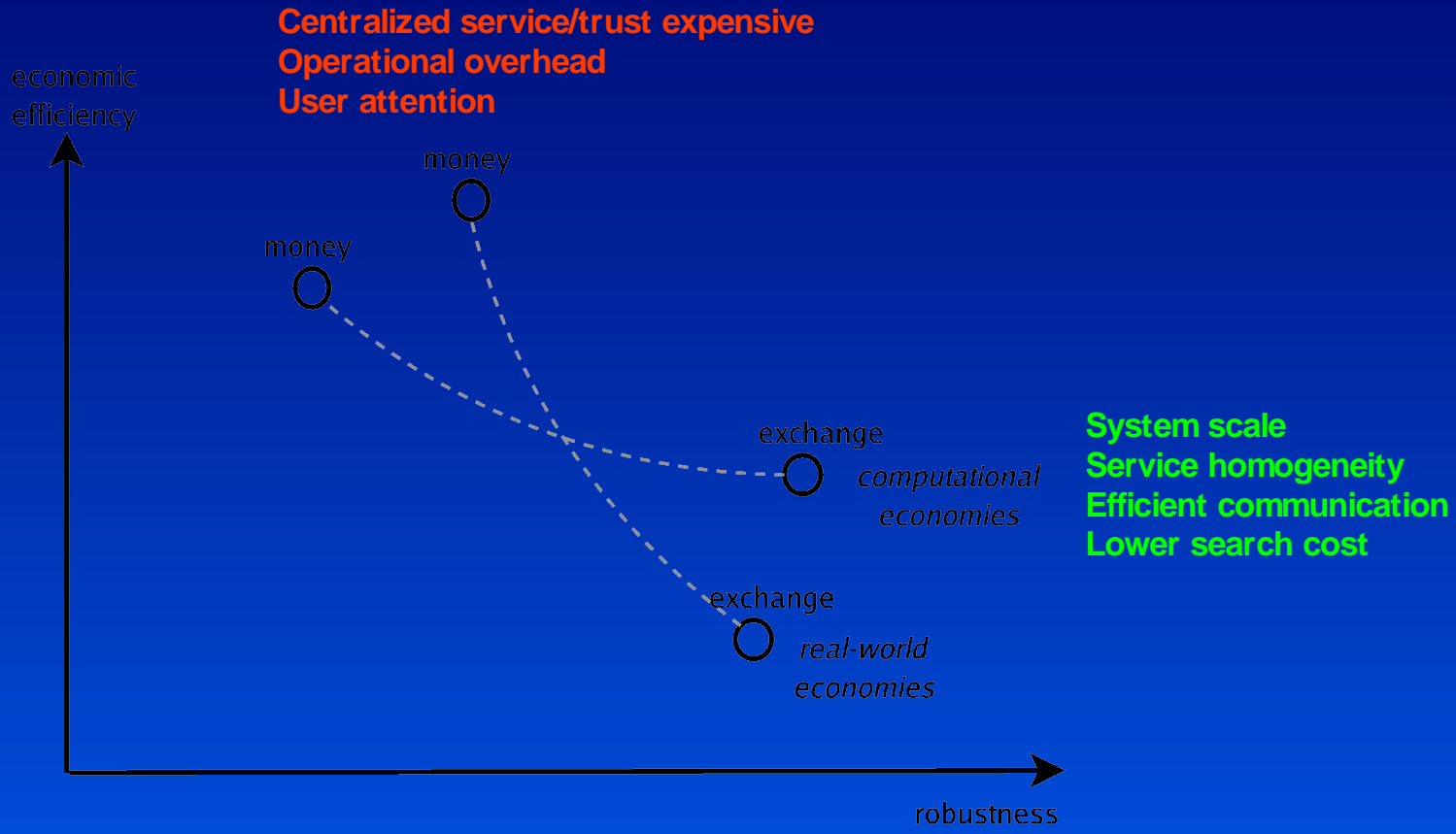
# Summary of Results

- ***We have shown that exchange mechanisms are widely applicable***
  - + Much more widely applicable than previously thought
  - Not universal
- ***We have outlined a decision procedure that helps determine the applicability of exchange***
  - + Consistent with observations made in applications
  - Informal, as many dependencies

# Additional Contributions

- First to propose and analyze N-way exchanges
- First exchange mechanism for generalized file sharing
- First incentive mechanism for P2P VoIP-PSTN peering
- Identified problems with previous approaches, incl. BitTorrent
- Publicly available experimental artifacts: simulators & datasets

# The big picture: money vs. exchange



Although exchange mechanisms are not usually applicable in the real world, they are widely applicable in the setting of cooperative distributed systems.

# Open Questions and Future Work

- More applications to accurately characterize design space
- Multi-commodity exchanges
- Hybrid mechanisms
- Framework unifying exchange, reputation, monetary mechanisms
- Collusion as the evil twin of cooperation

Backup  
slides

# Previous work: cooperation theory

- Repeated Prisoners' Dilemma (RPD) model
  - Users repeatedly choose between cooperating and defecting
  - Assumptions: symmetry, verifiability, repeated interaction
- Theory suggests that **tit-for-tat** is the “ideal” strategy
  - **Optimism**: cooperate if peer cooperated in previous round
  - **Retaliation**: defect if peer defected in previous round
  - **Forgiveness**: ignore history, just look at previous round
- But mapping to cooperative distributed systems not obvious
  - Context of theory is international relations/conflict resolution
    - Cost of trusted mediation >> cost of verifying cooperation
  - RPD assumptions often do not hold in cooperative distributed systems

# Previous work: exchange in the real-world

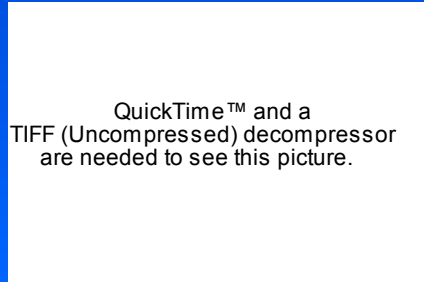
- Numerous examples of exchange (barter) economies throughout history

- Considered primitive, inefficient: “double coincidence of wants” constraint, and limited scalability
- Inefficiency believed to have led to introduction of money as a *medium of exchange*



- Exchange systems not completely extinct. Many examples under certain (exceptional) conditions

- when centralized institutions fail:  
Argentina crisis, former Soviet Union, WWII camps
- in the presence of more efficient search:  
“Exchange and mart” magazine



# Previous work: exchange in the real-world

- Numerous examples of exchange (barter) economies throughout history
  - Considered primitive, inefficient: “double coincidence of wants” constraint, and limited scalability
  - Inefficiency believed to have led to introduction of money as a ***medium of exchange***
- Exchange systems not completely extinct. Many examples under certain (exceptional) conditions
  - when centralized institutions fail:  
Argentina crisis, former Soviet Union, WWII camps
  - in the presence of more efficient search:  
“Exchange and mart” magazine

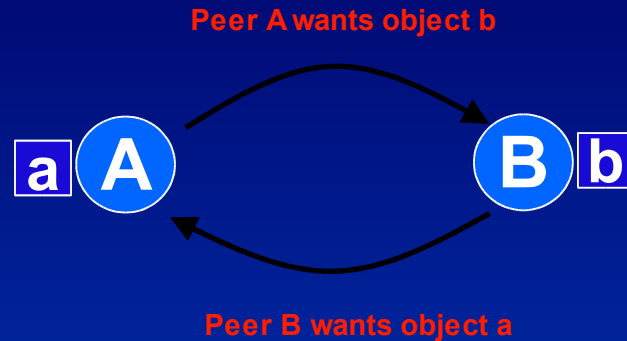
# Previous work: exchange in distributed systems

- Exchange in BitTorrent content distribution
  - Tit-for-tat upload/download
- Promising:
  - Fully decentralized
  - Intuition says it works
  - System has become very popular: 35% of Internet traffic
- Unproven:
  - % of traffic doesn't mean that the incentive works!
  - Focus on distribution of new, popular content.
    - . Even if it works, can it be extended to general file sharing?
    - . Other cooperative distributed systems ?
    - . Or is it just some degenerate example?

# Thesis questions

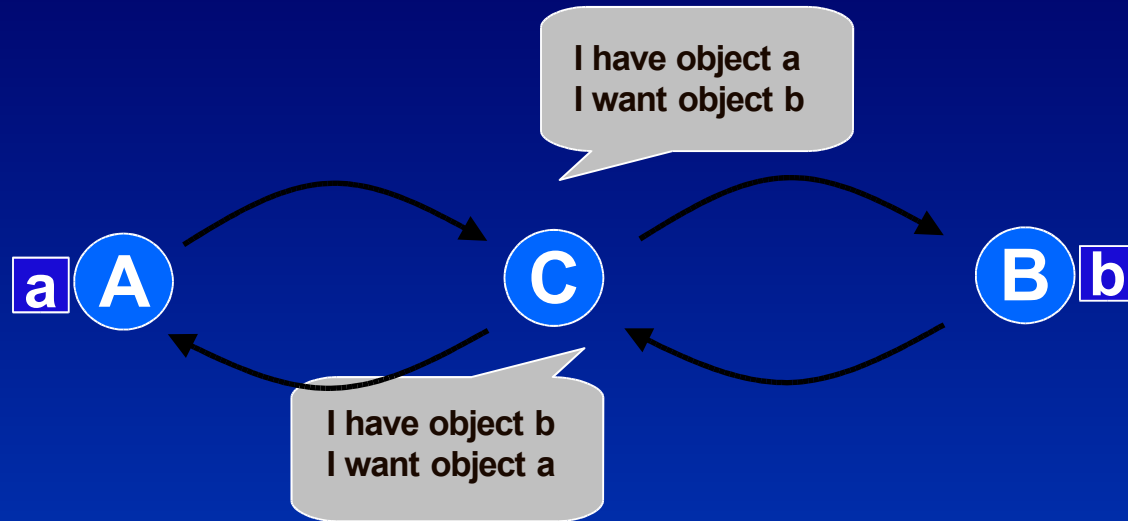
- ***Are exchange mechanisms suitable for providing incentives in cooperative distributed systems?***
  - *Strong enough incentives for cooperation?*
  - *Robust enough to selfish adversaries?*
  - *Practical enough to implement?*
- ***Is there a decision procedure that can help determine whether exchange applies to a given problem?***
  - *What system attributes determine whether exchange is applicable?*
  - *How do these attributes affect the design of the mechanism?*

# Robustness I



- **One-sided defection:** cheater leaves ring after acquiring object, without honoring exchange
  - Block-by-block transfers (baby steps, *aka tit-for-tat*) not good enough: cheater may be happy even with one-block exchanges
  - *Enforce repeated interaction: prioritize long exchanges, encrypt transfers, and only swap keys when done*
- **False content:** cheater joins ring, sends bogus content
  - *Requires content validation service and a mediator to validate transaction and assist in final exchange of keys*

# Robustness II

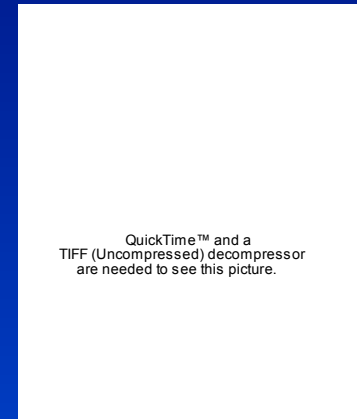
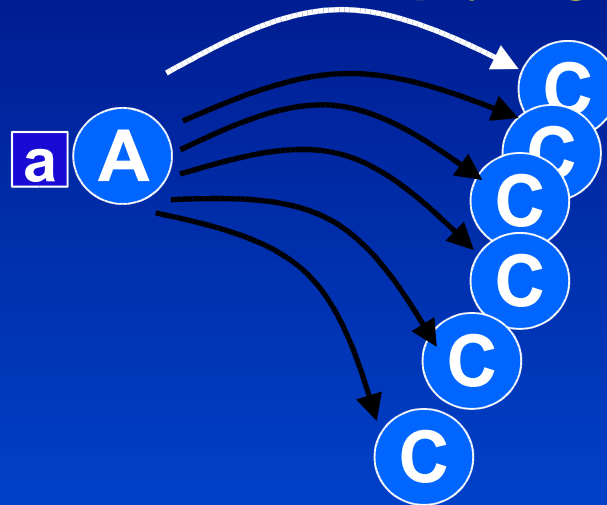


- **Middleman #1:** cheater joins ring, peddles blocks between A,B
  - *Embed **peer-id** in encrypted blocks, use third-party to complete exchange of keys (third party sends key to **peer-id**)*
- **Middleman #2:** cheater obtains seed blocks using non-exchange transfers, uses seed blocks for one-block exchanges
  - *Prioritizing longer exchanges helps, but doesn't entirely solve problem*

# Robustness III

## ● Pseudospoofing

- Exchange makes downloads  $M$  times faster for cooperating peers  
*what if cheaters could simply sign up with  $M'$  identities?*

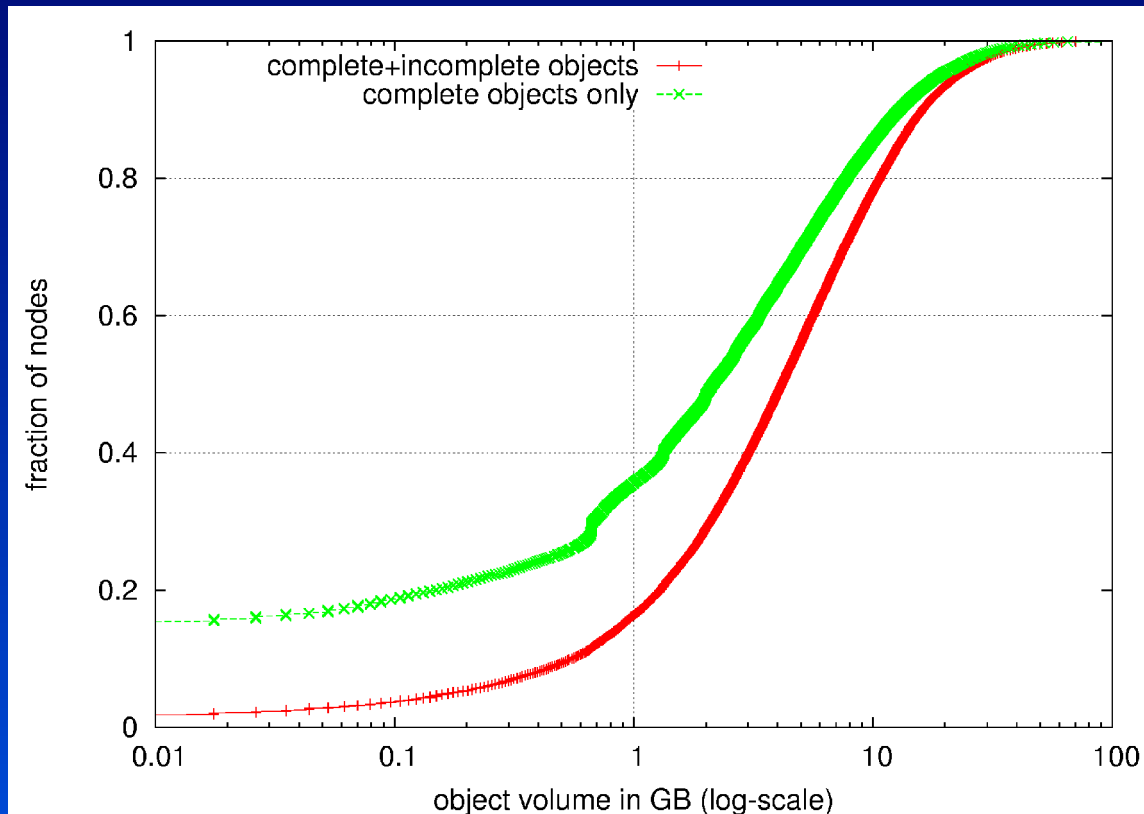


- *May need to completely prohibit non-exchange transfers,  
but how do we prevent lock-out of new users?*

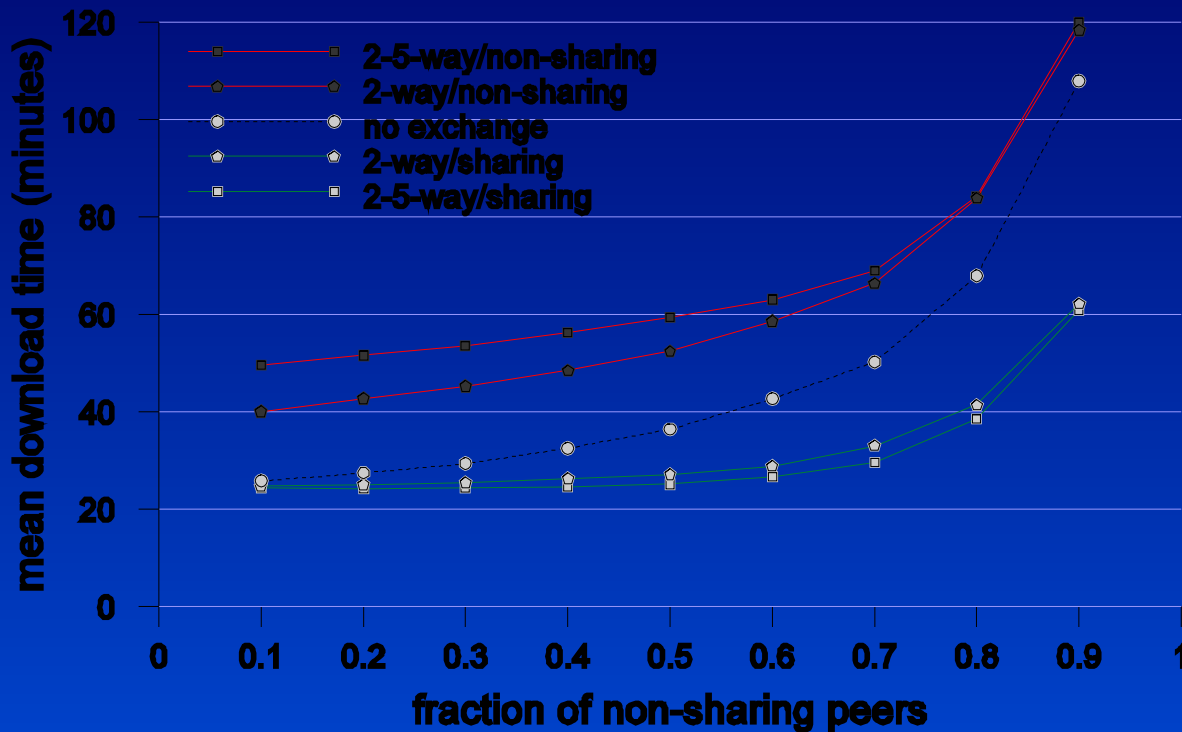
# Some preliminary observations...

- Exchange applicable: two out of two so far
  - Applicability not universal, depends on system parameters
  - N-way makes a big difference compared to two-way
  - Search cost is reasonably low
  - Security is crucial
  - Fraction of requests/calls in exchange is a good first-order measure but could be misleading

# File sharing: Dataset characteristics

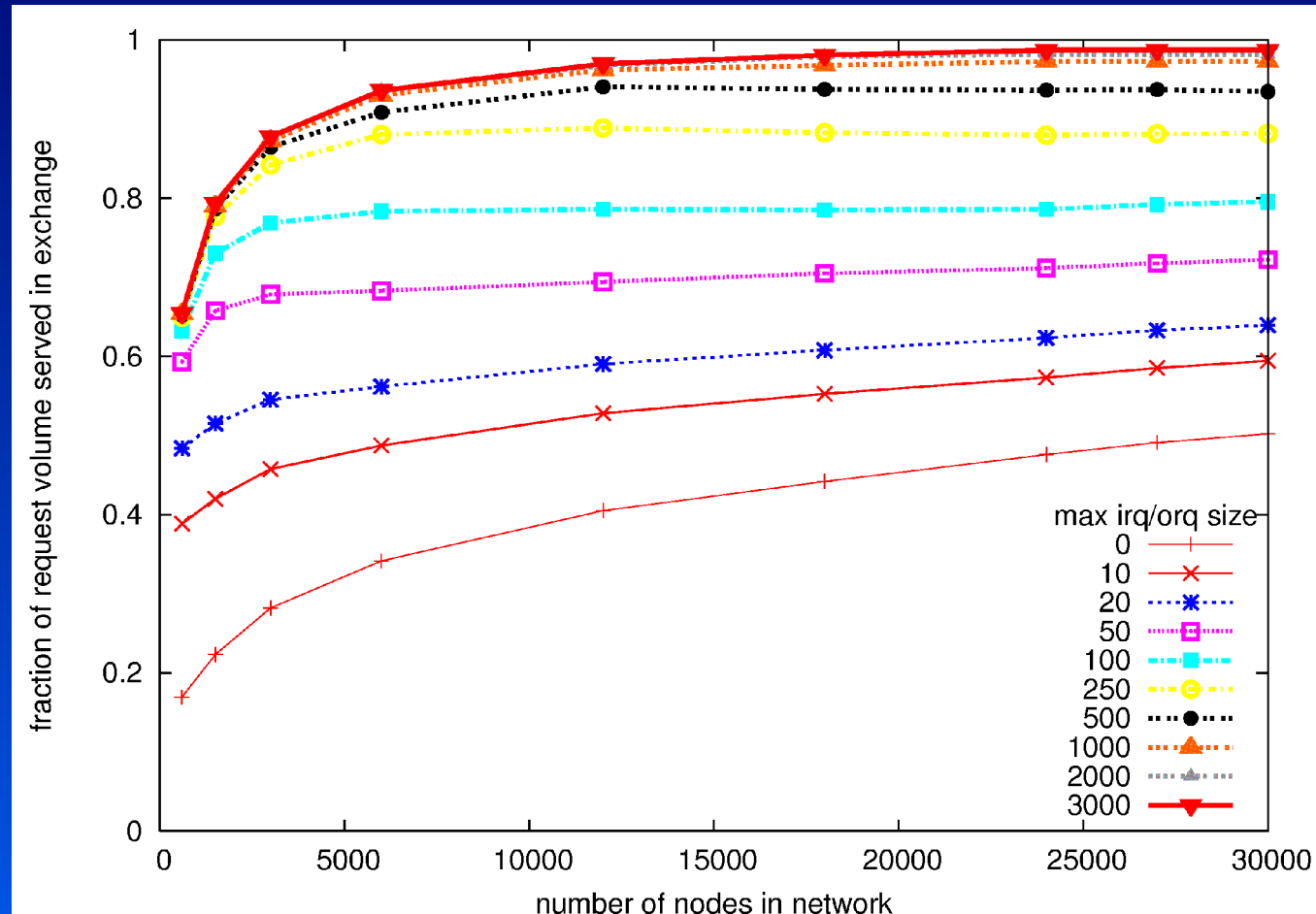


# Incentive to deploy exchange: simulation

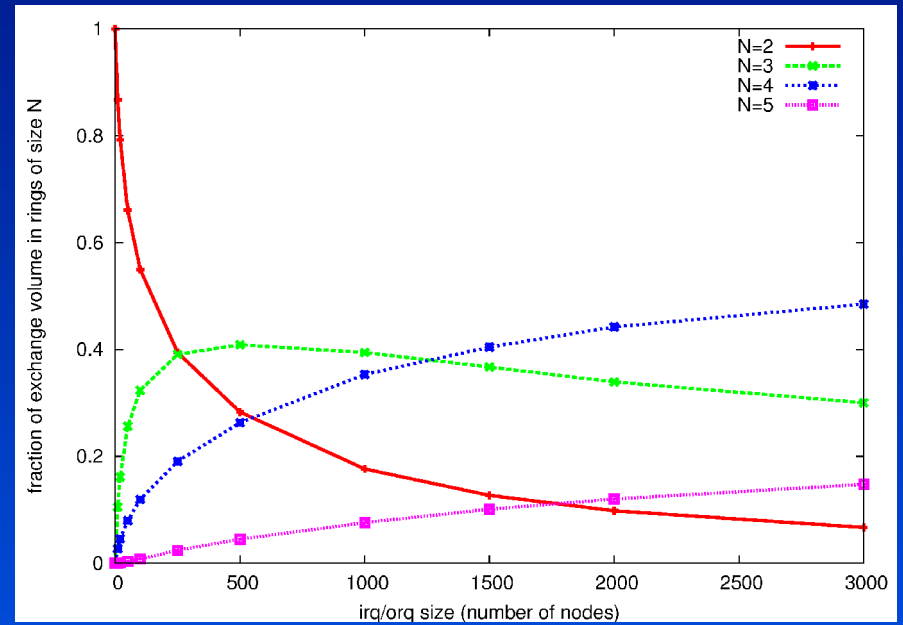
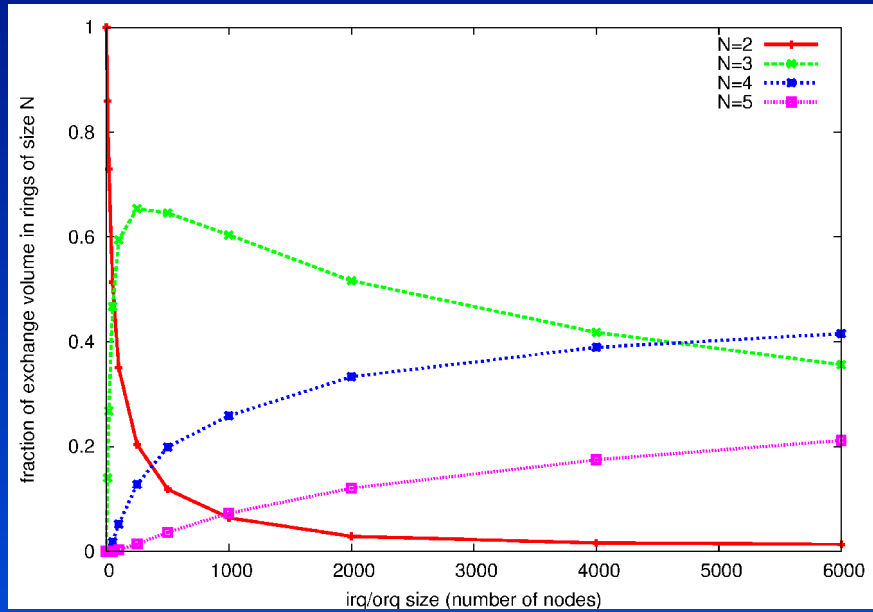


- Under current conditions (70% non-sharing), and even worse, sharing peers have strong incentive to deploy exchange
- Freeloaders are generally worse off compared to sharers regardless of sharing/non-sharing population mix

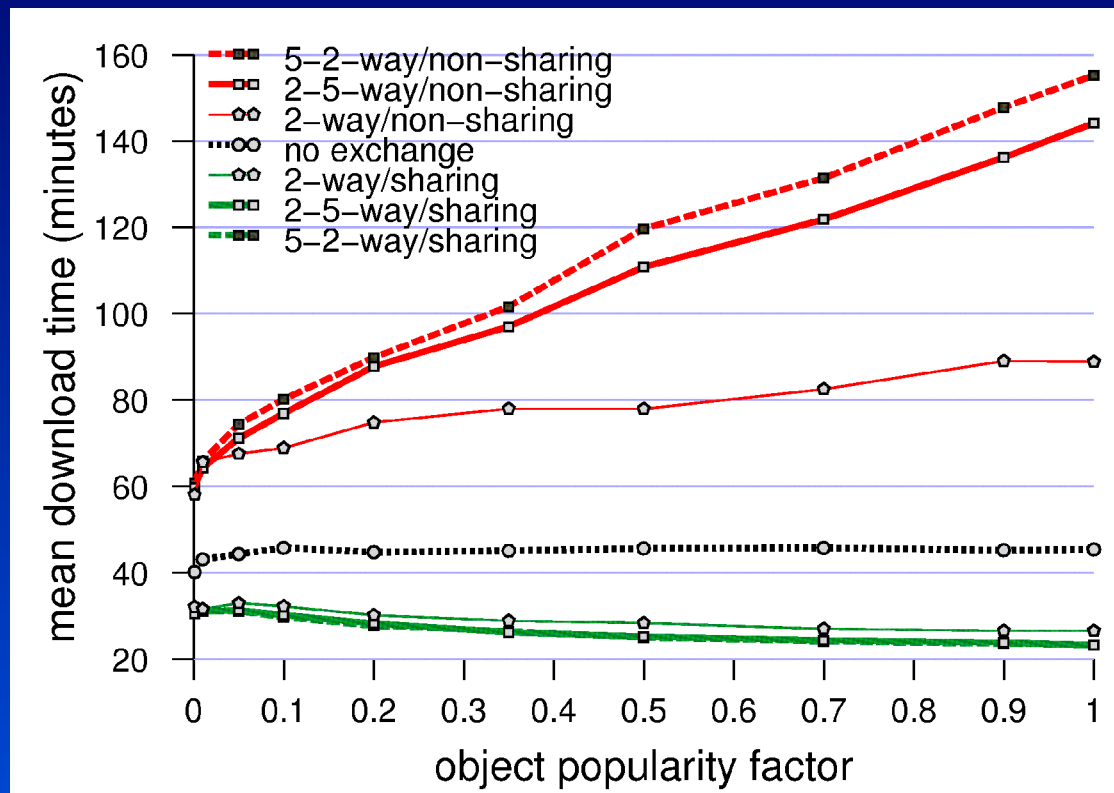
# File sharing: Experiment scale and effect on observations



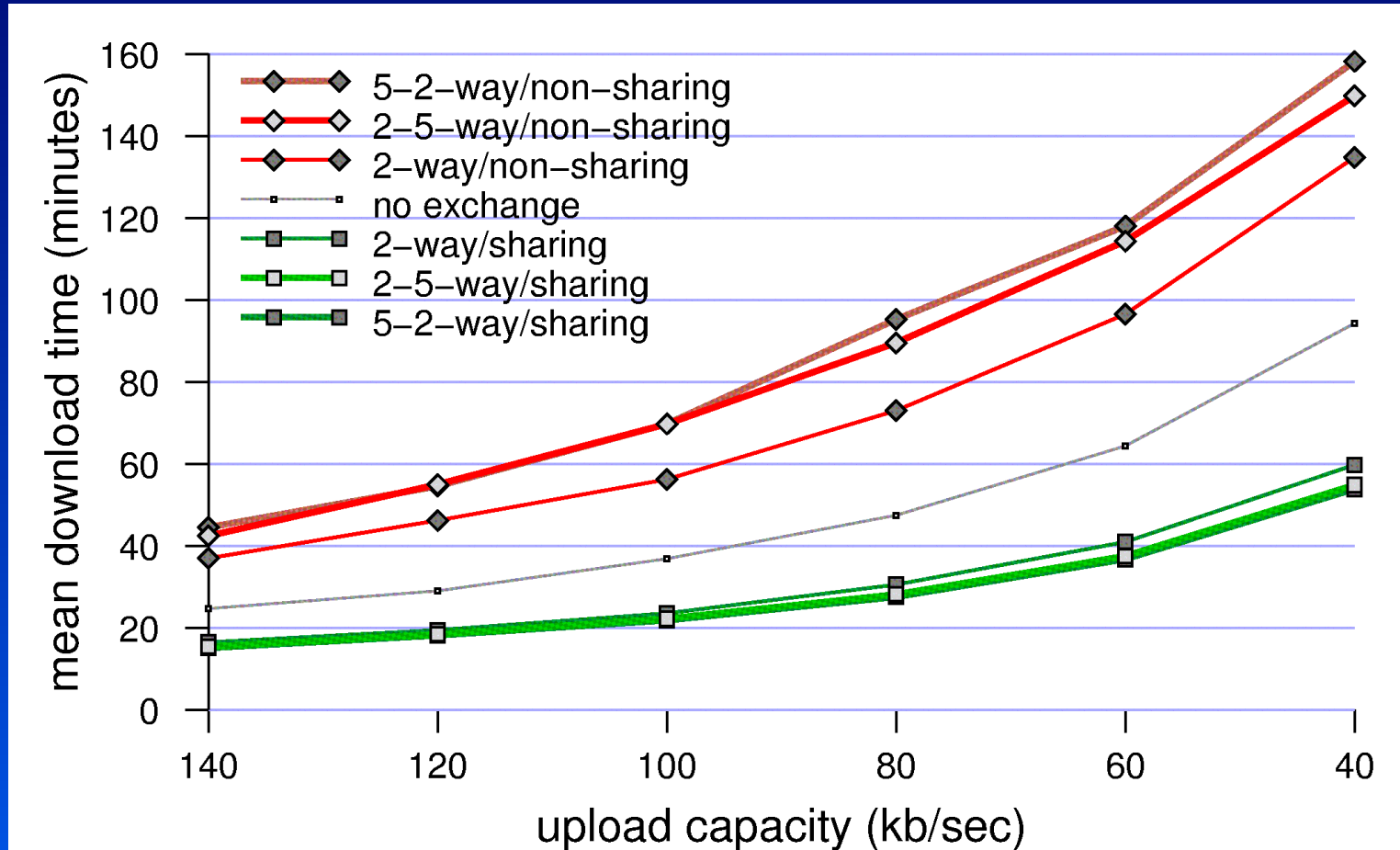
# File sharing: Breakdown of rings



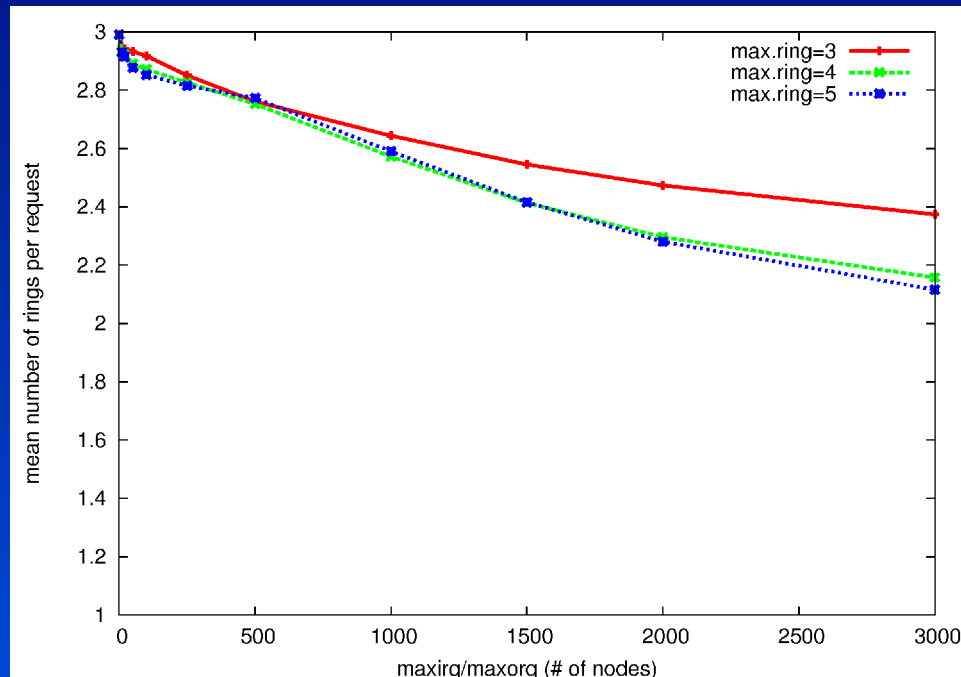
# File sharing: Impact of object popularity



# File sharing: Impact of resource scarcity



# File sharing: # of rings per request



# VoIP-PSTN: Example of local calling area

## Local calling area

Moorestown, NJ [[Prefix detail](#)]

LATA: 222 DELAWARE VALLEY NJ

V: 05228 H: 01432

ILEC: 9206 VERIZON NEW JERSEY, INC.

NPA 609:

304 410 744 760 781 923

NPA 856:

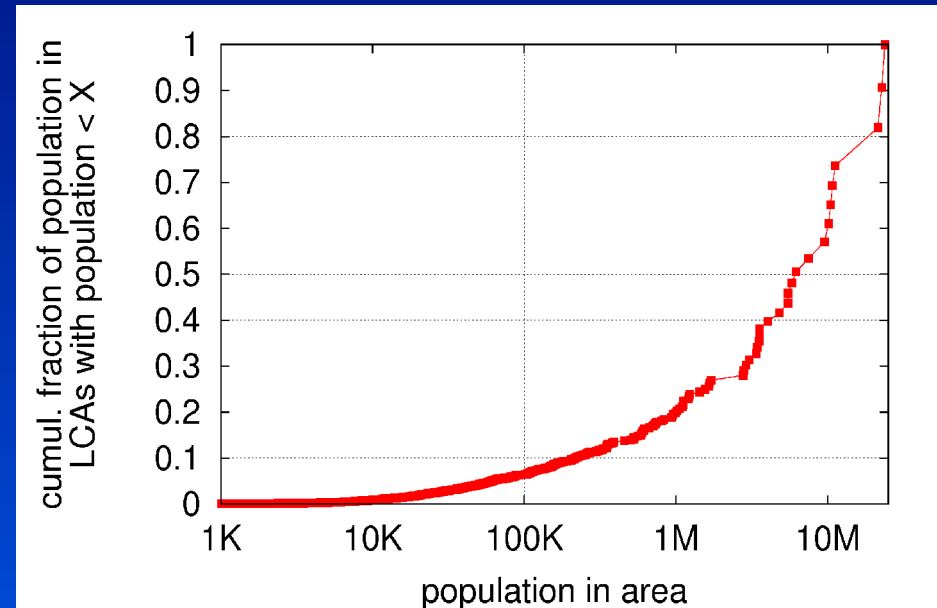
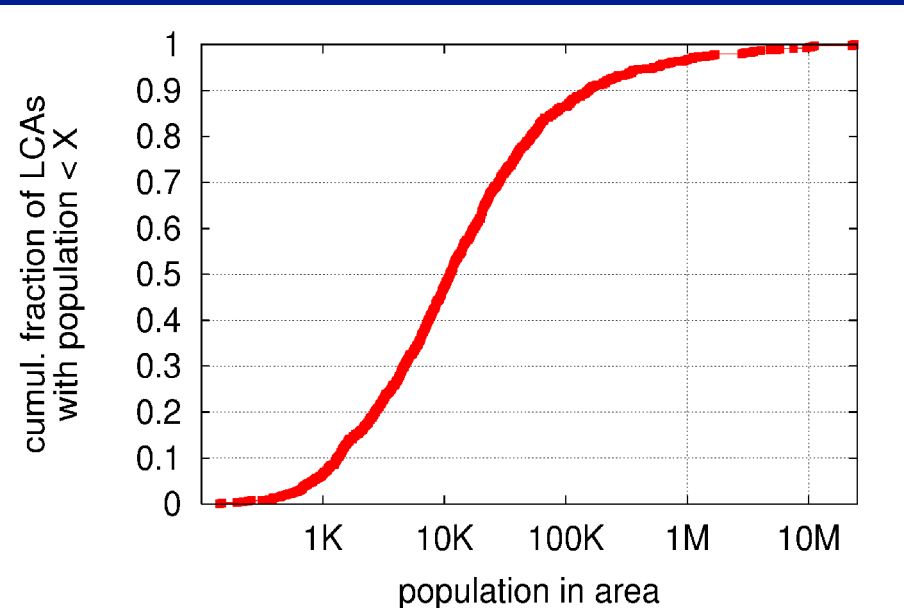
206 220 222 231 234 235 252 266 273 296 313 316 359 380 437 439 505 533 581 591 608 630 638 642 674 722 727 760 778 780  
787 793 802 813 840 866 912 914 917 924 995 996

See all [[Local Prefixes](#)]

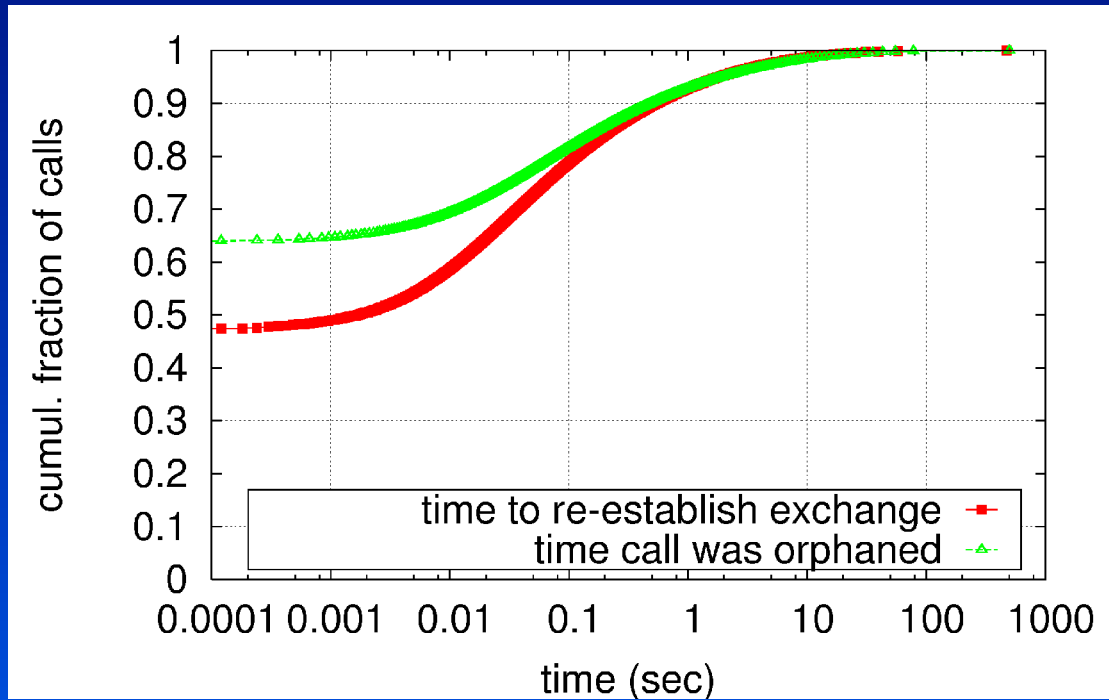
Local calling TO the following exchanges: [[Reverse direction](#)]

Rate centre	Plan type	Call type	Monthly limit (minutes)	Note	Effective date
<a href="#">Camden, NJ</a>					
<a href="#">Haddonfield, NJ</a>					
<a href="#">Marlton, NJ</a>					
<a href="#">Medford, NJ</a>					
<a href="#">Merchantville, NJ</a>					
<a href="#">Mount Holly, NJ</a>					
<a href="#">Riverside, NJ</a>					
<a href="#">Riverton, NJ</a>					

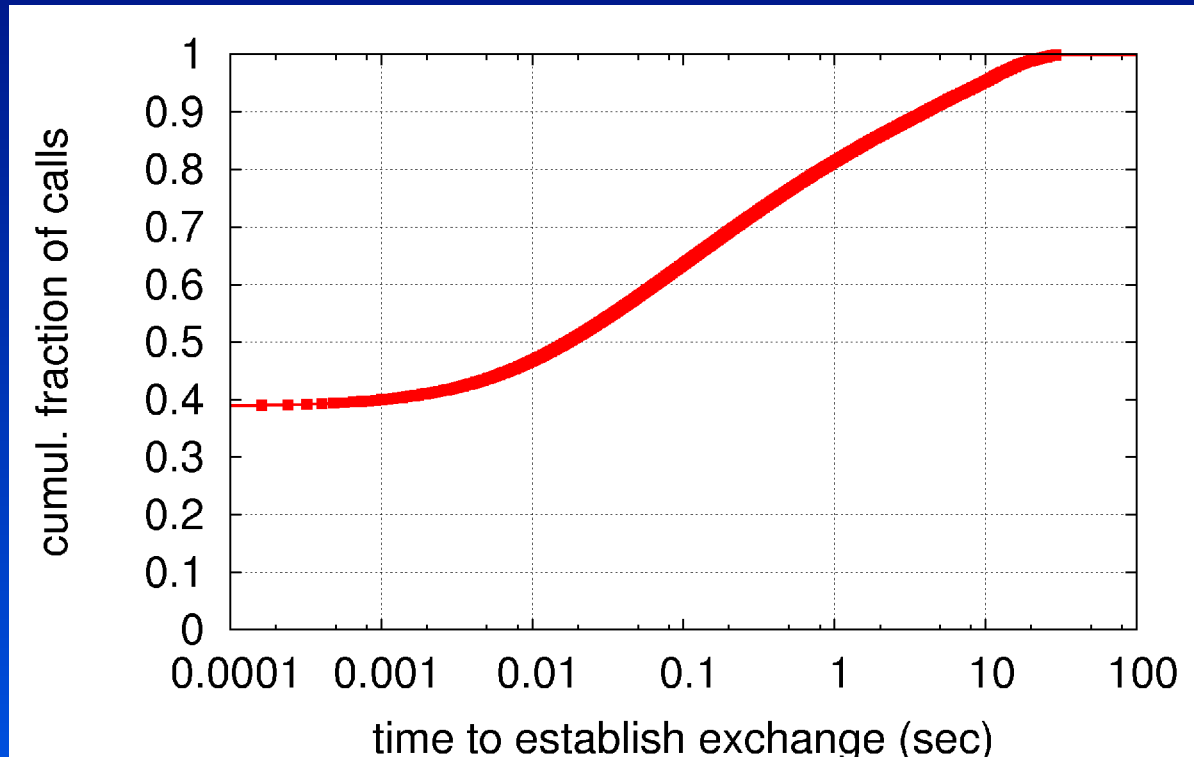
# VoIP-PSTN: LCA stats



# VoIP-PSTN: Orphan calls



# VoIP-PSTN: Time to establish exchange



# VoIP-PSTN peering: Locating rings

- VoIP-PSTN peering different from file sharing
  - Small # of peers: ~1000 LCAs in US PSTN system
  - Tighter timing requirements: latency matters
  - Higher rate of change in exchange graph: mean call time 3.3 min
- Approach:
  - NxN full-mesh periodic updates of pending call bitmap
  - Every node knows full graph, can compute rings locally
- Performance:
  - Conservative cost estimate: 1200 nodes, updates every 4 seconds result in 45KB/sec, equivalent to 5 calls
  - Cost reasonable, but could do much better

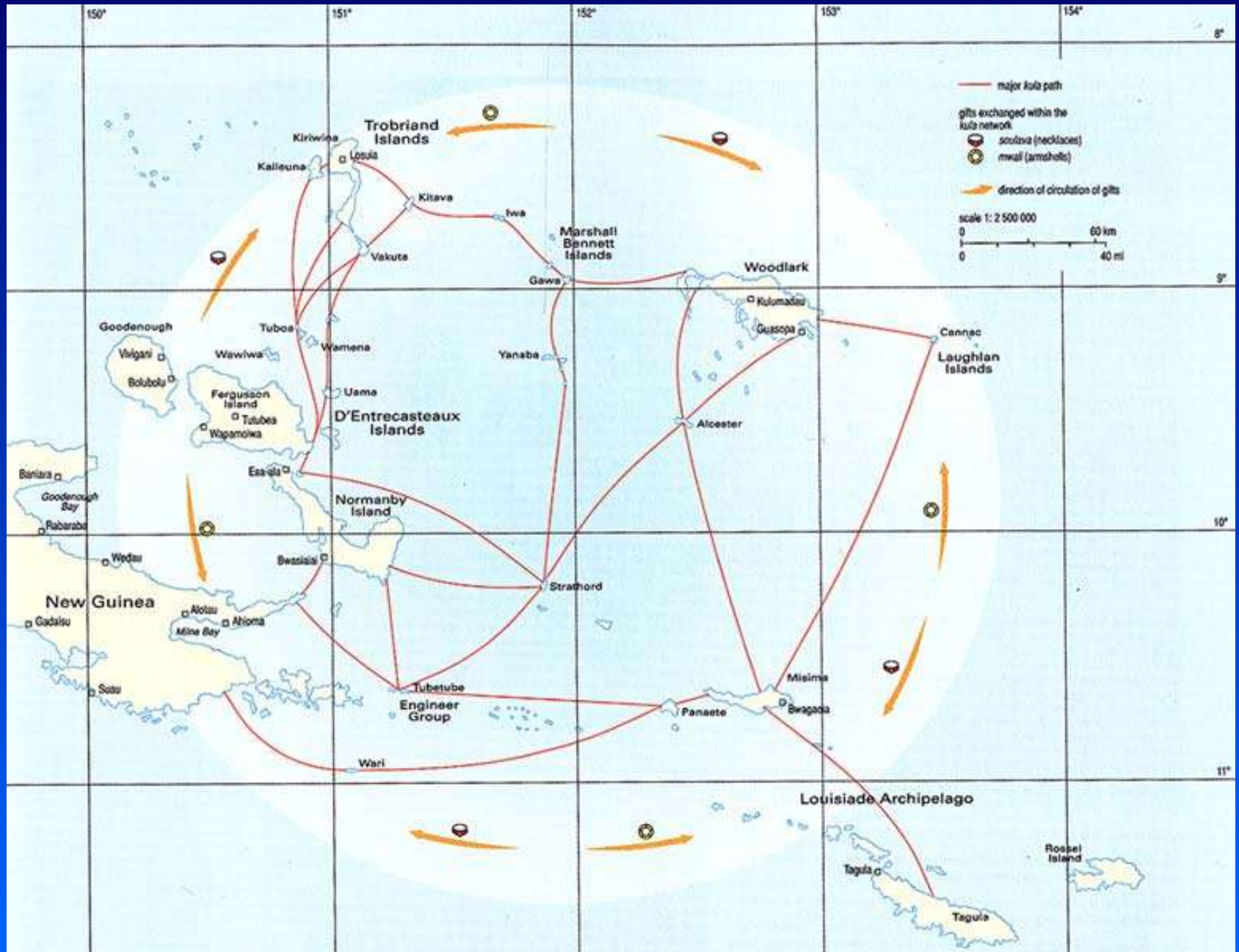
# VoIP-PSTN peering: Threats

- Assumptions:

- Strong identities
- Long-lived participation

- Premature drop

- Cannot distinguish
- Need reputattion



# Methodology & Roadmap

- Demonstrate applicability of exchange in specific applications
  - File sharing
  - P2P VoIP-PSTN peering
  - **Exchange is applicable**
- Examine feasibility+structure of exchange in other applications
  - BitTorrent file sharing, P2P backup, query forwarding, hotspot sharing
  - **Decision procedure determining applicability of exchange**
- Results + contributions
- Why exchange? Exchange vs. money in real-world vs. cooperative distributed systems
- Future work + concluding remarks

# Related work

## ● CS:

- Systems using mechanisms based on money and reputation
  - . *MojoNation, Fileteller, Karma, EigenTrust, and others*
- Several other systems using exchange
  - . *PFE (UW), Samsara (Michigan), SLIC (Stanford), BitTorrent, hotspot sharing*

## ● ECON:

- Real-world exchange systems
  - . *Primitive economies, but also modern examples: WWII camps, Argentina crisis, ...*
- Cooperation theory
  - . *Tit-for-tat exchange ideal strategy under certain (tight) conditions*

# Closing remarks

- Exchange seems widely applicable, but very easy to make mistakes. Careful design still necessary
- Security the most important and often ignored issue
- Decision procedure informal and fuzzy. Ideal would be an abstract model/flowchart, but hard to get at this point
- Caution when porting concepts and methods from other domains to computer systems