Affinity in Distributed Systems

Thesis defense

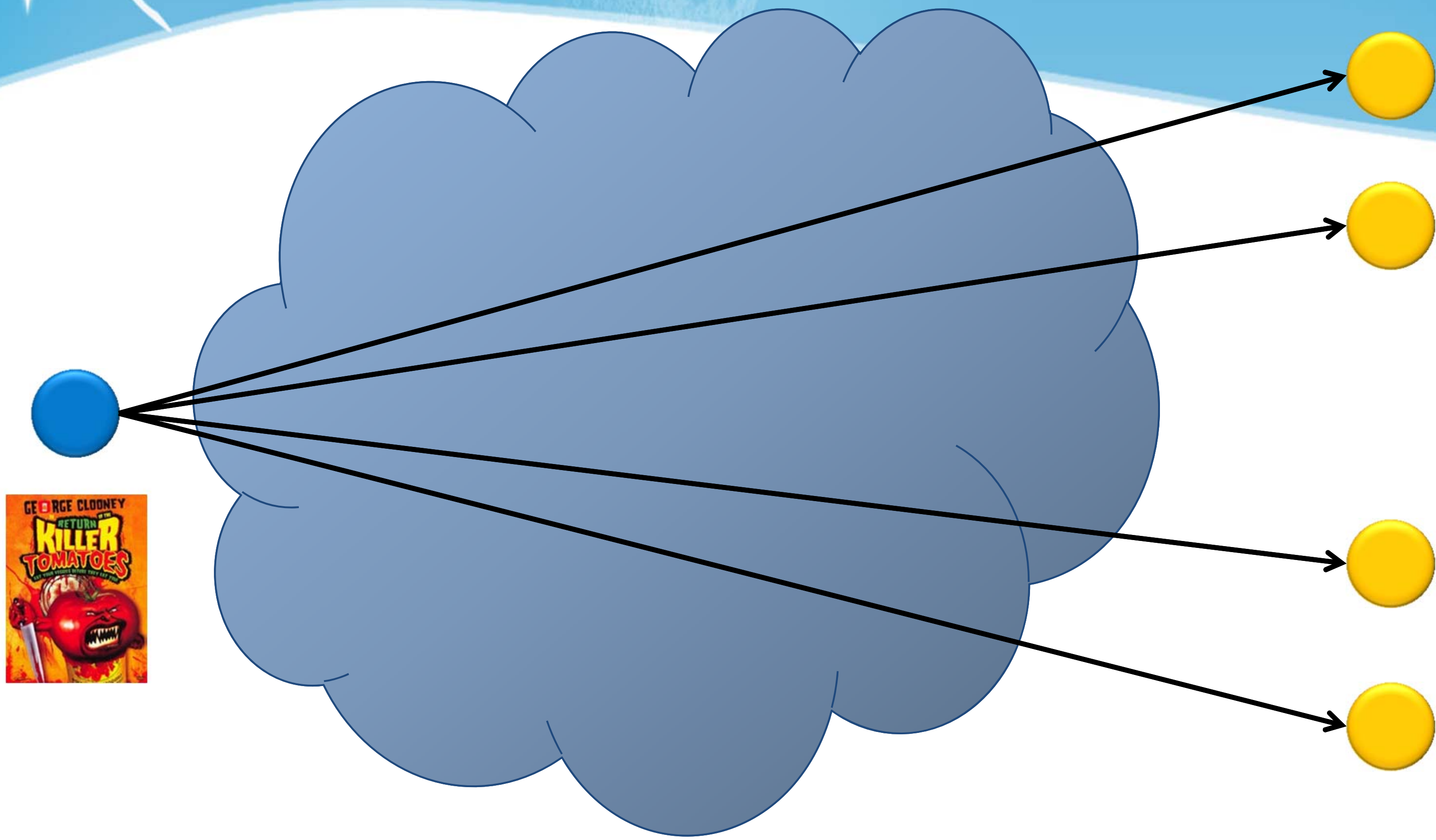
Ymir Vigfusson

Joint work with:
Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, Gregory Chockler, Qi Huang, Jure Leskovec, Deepak Nataraj and Yoav Tock.

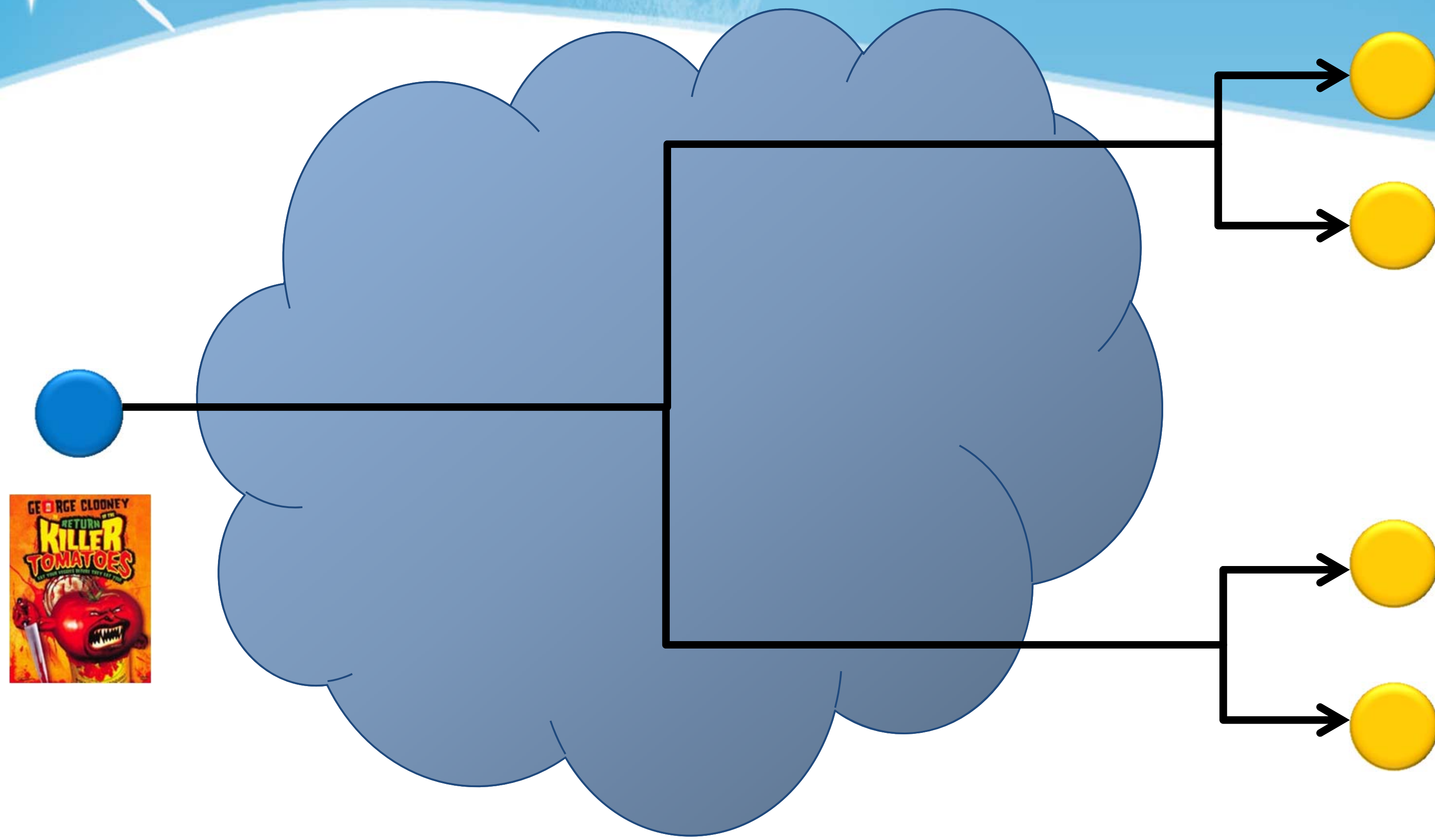
Group communication

- Most network traffic is *unicast* communication (one-to-one).
- But a lot of content is identical:
  - Audio streams, video broadcasts, system updates, *etc.*
- To minimize redundancy, would be nice to *multicast* communication (one-to-many).

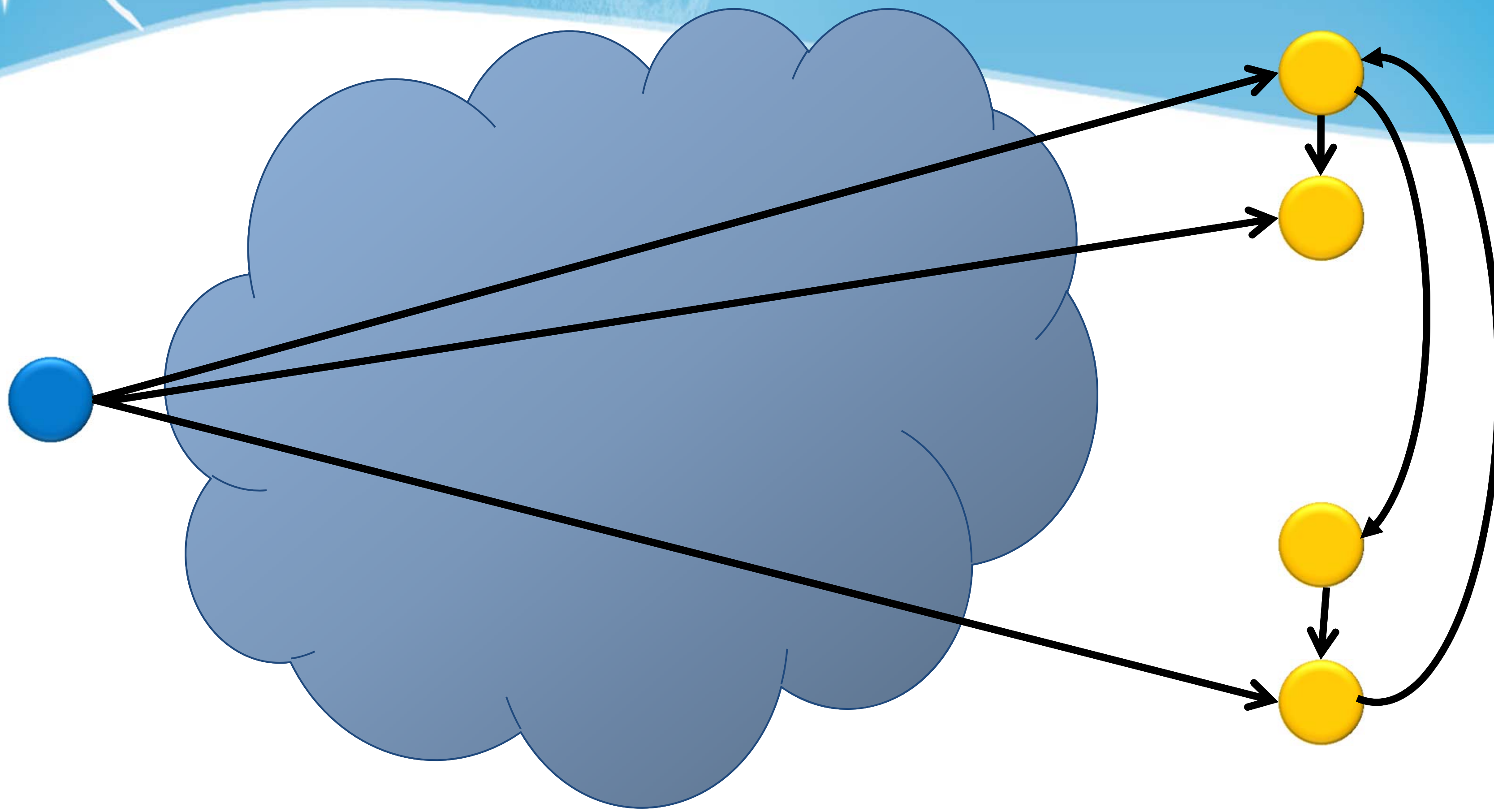
Multicast by Unicast



IP Multicast



Gossip



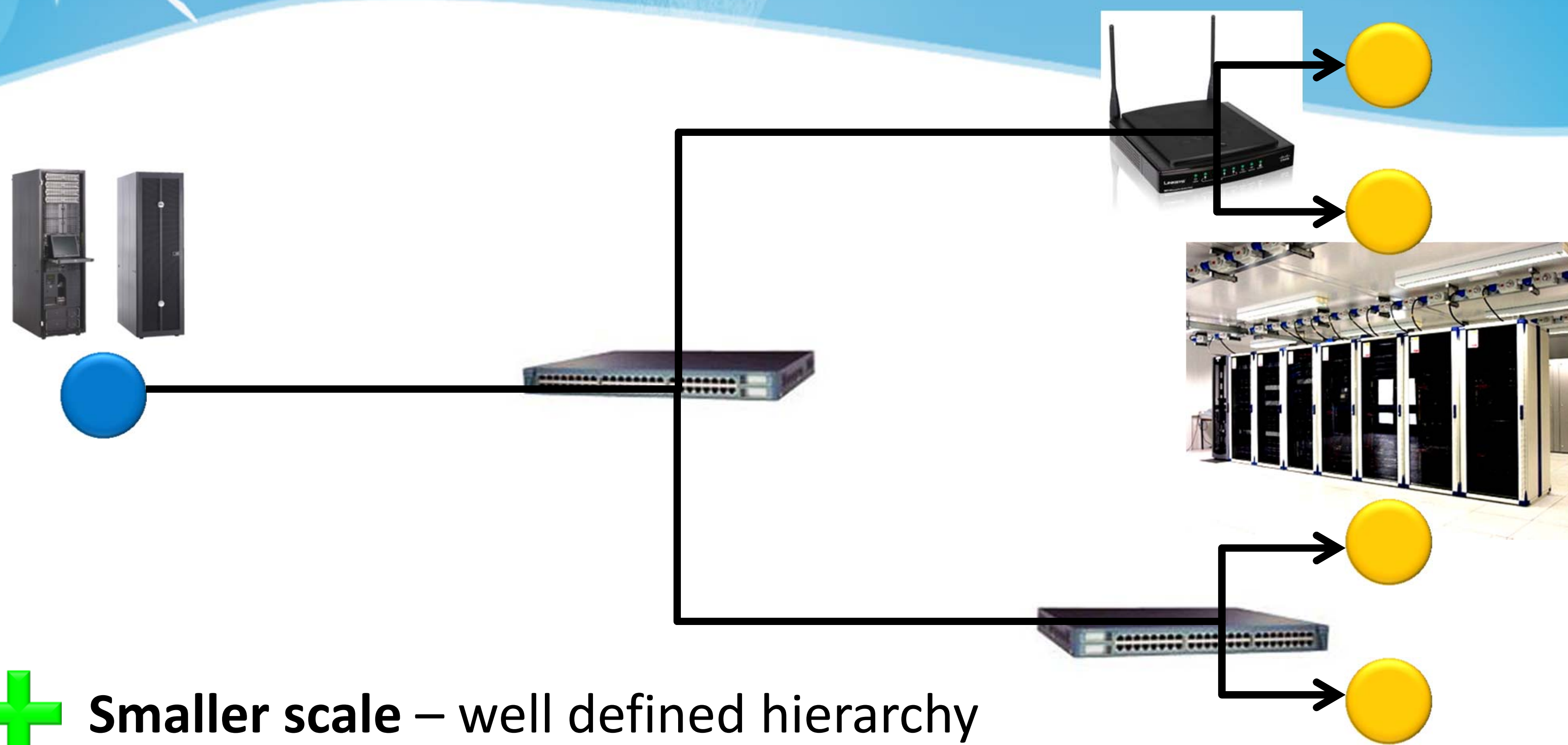
Group communication

Mechanism	Deliv. Speed	Redundancy	Scalable in # users?
Point-to-point unicast	Slow	High	No
IP Multicast (IPMC)	Fast	None	Yes
Gossip	Slow	Low	Yes

Talk Outline

- Dr. Multicast (**MCMD**)
  - Group scalability in IP Multicast.
- Gossip Objects (**GO**) platform
  - Group scalability in gossip.
- Affinity
  - GO+MCMD optimizations based on group overlaps
  - Explore the properties of overlaps in data sets
- Conclusion

IP Multicast in Data Centers

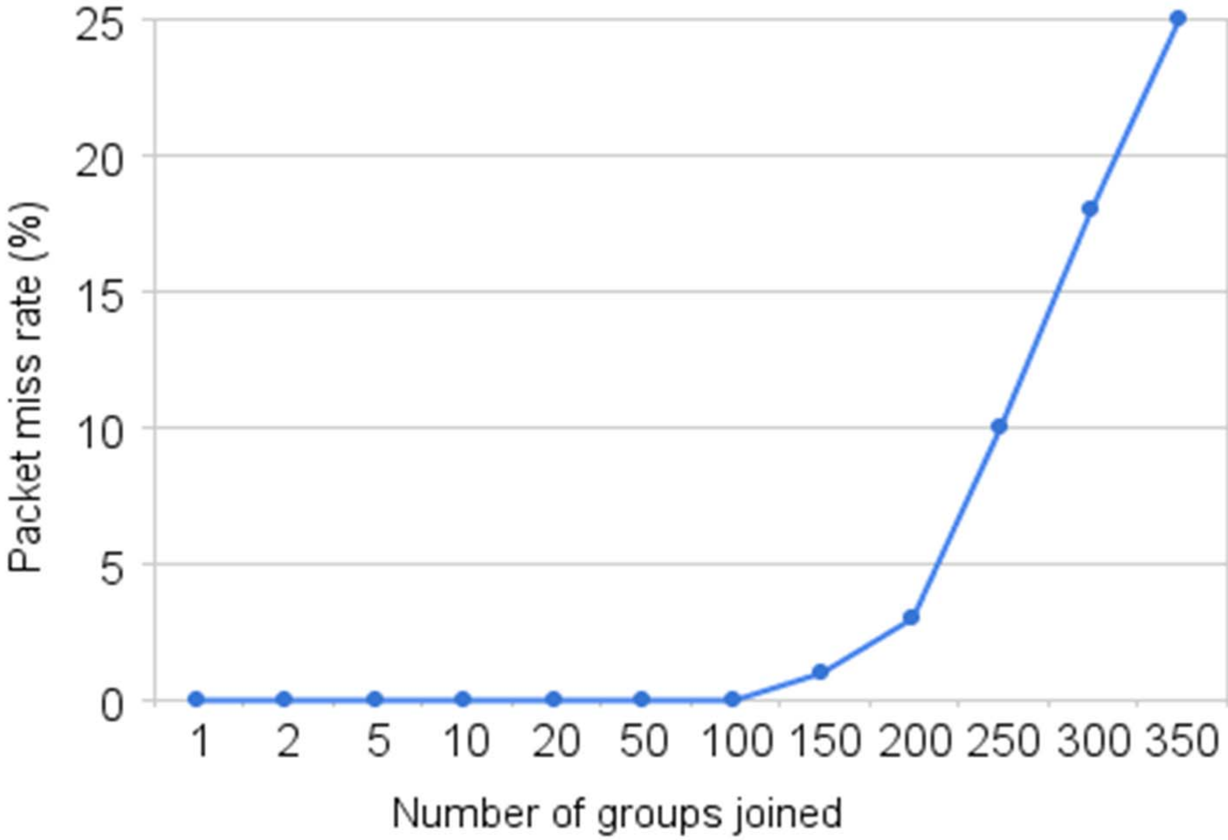


- **Smaller scale** – well defined hierarchy
- Single **administrative** domain
- **Firewalled** – can ignore malicious behavior

IP Multicast in Data Centers

- Useful, but rarely used.
- Various problems:
  - Security
  - Stability
  - Scalability

IP Multicast in Data Centers



IP Multicast in Data Centers

- Useful, but rarely used.
- Various problems:
  - Security
  - Stability
  - Scalability
- **Bottom line:** Administrators have no *control* over IPMC.
  - Thus they choose to disable it.

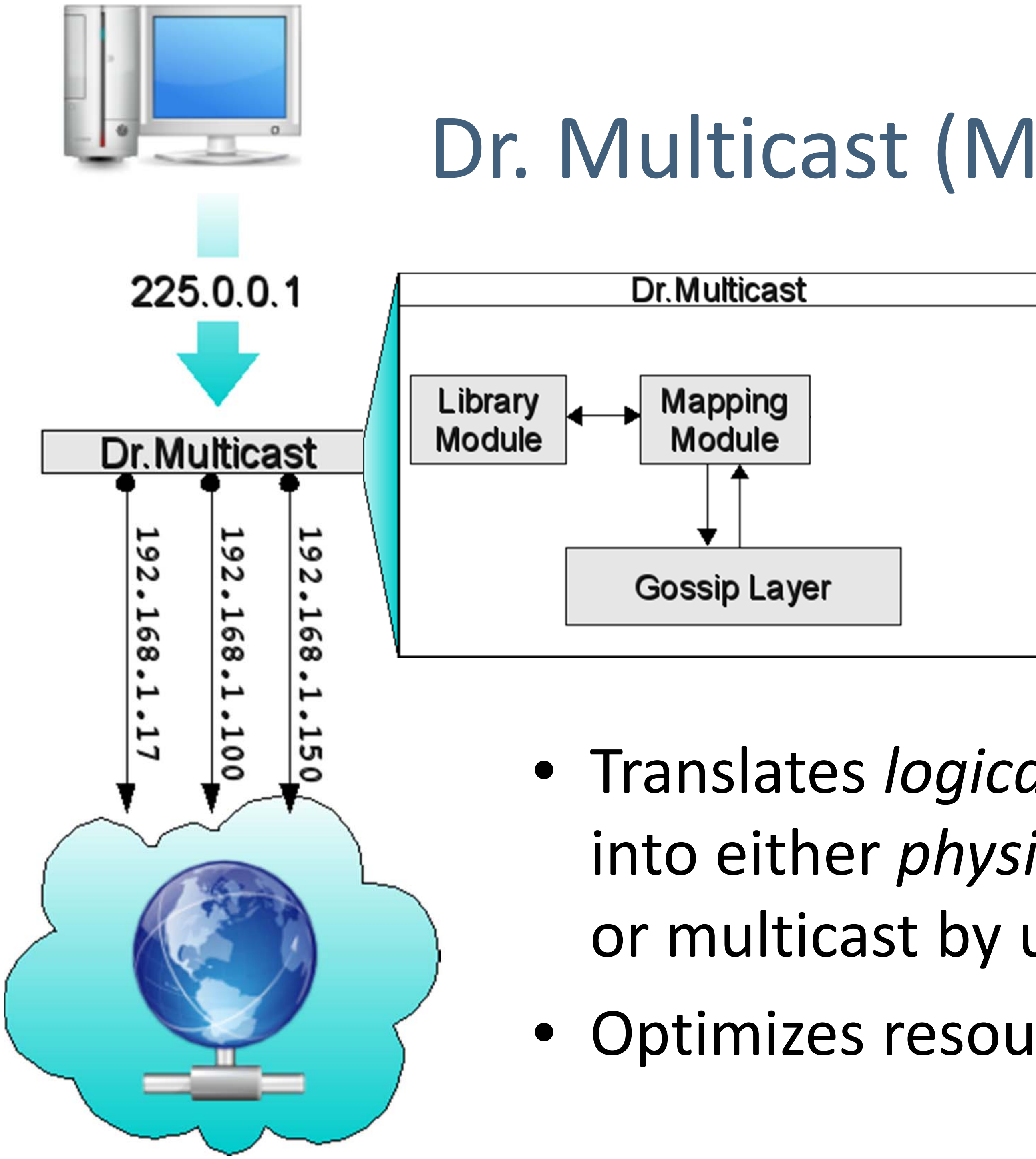
Wishlist

- ***Policy:*** Enable control of IPMC.
- ***Transparency:*** Should be backward compatible with hardware and software.
- ***Scalability:*** Needs to scale in number of groups.
- ***Robustness:*** Solution should not bring in new problems.

Acceptable Use Policy

- Assume a higher-level network management tool compiles policy into primitives.
- Explicitly allow a process (user) to use IPMC groups.
  - ***allow-join(process ID, logical group ID)***
  - ***allow-send(process ID, logical group ID)***
- Point-to-point unicast always permitted.
- Additional restraints.
  - ***max-groups(process ID, limit)***
  - ***force-udp(process ID, logical group ID)***

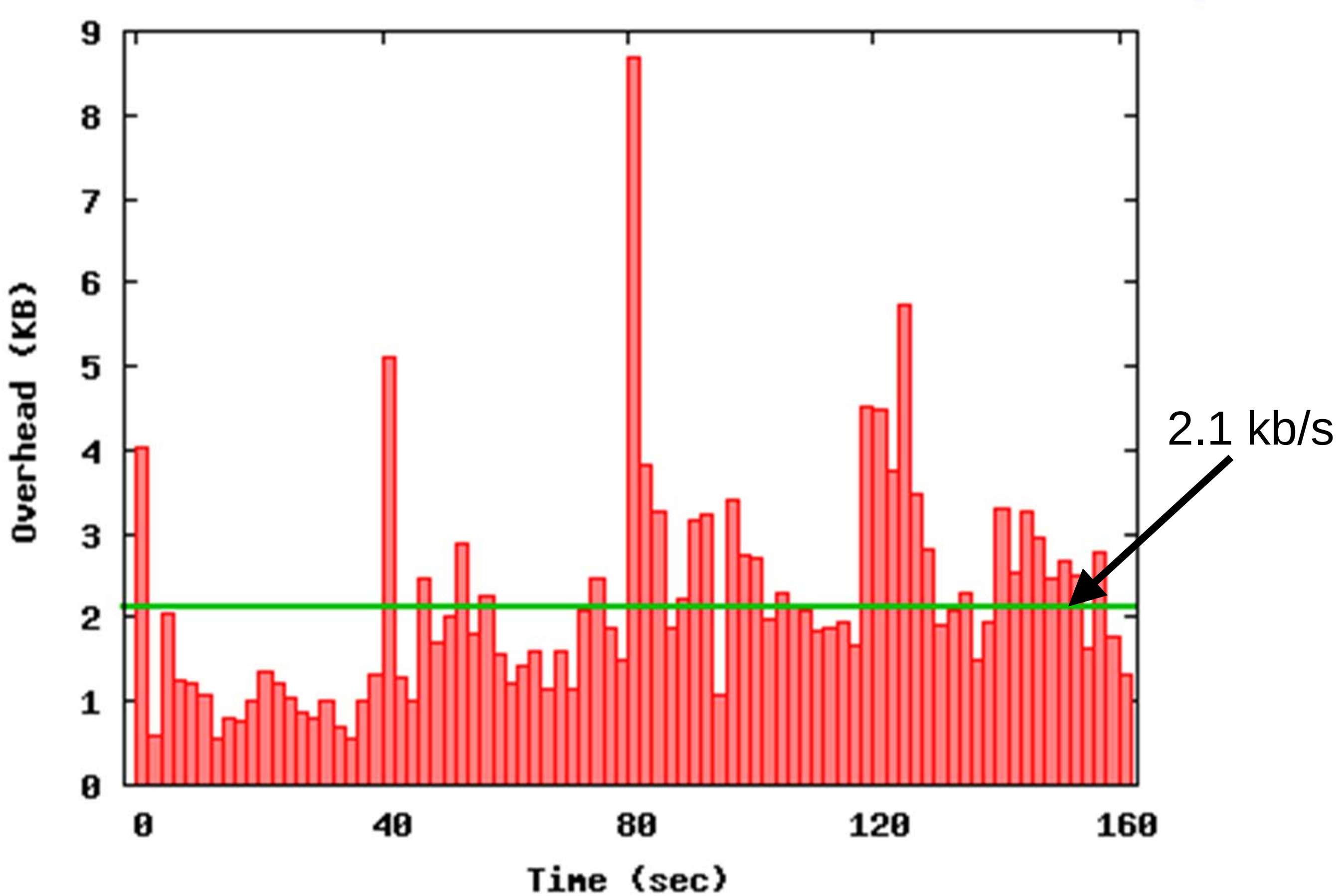
Dr. Multicast (MCMD)



- Translates *logical* IPMC groups into either *physical* IPMC groups or multicast by unicast.
- Optimizes resource use.

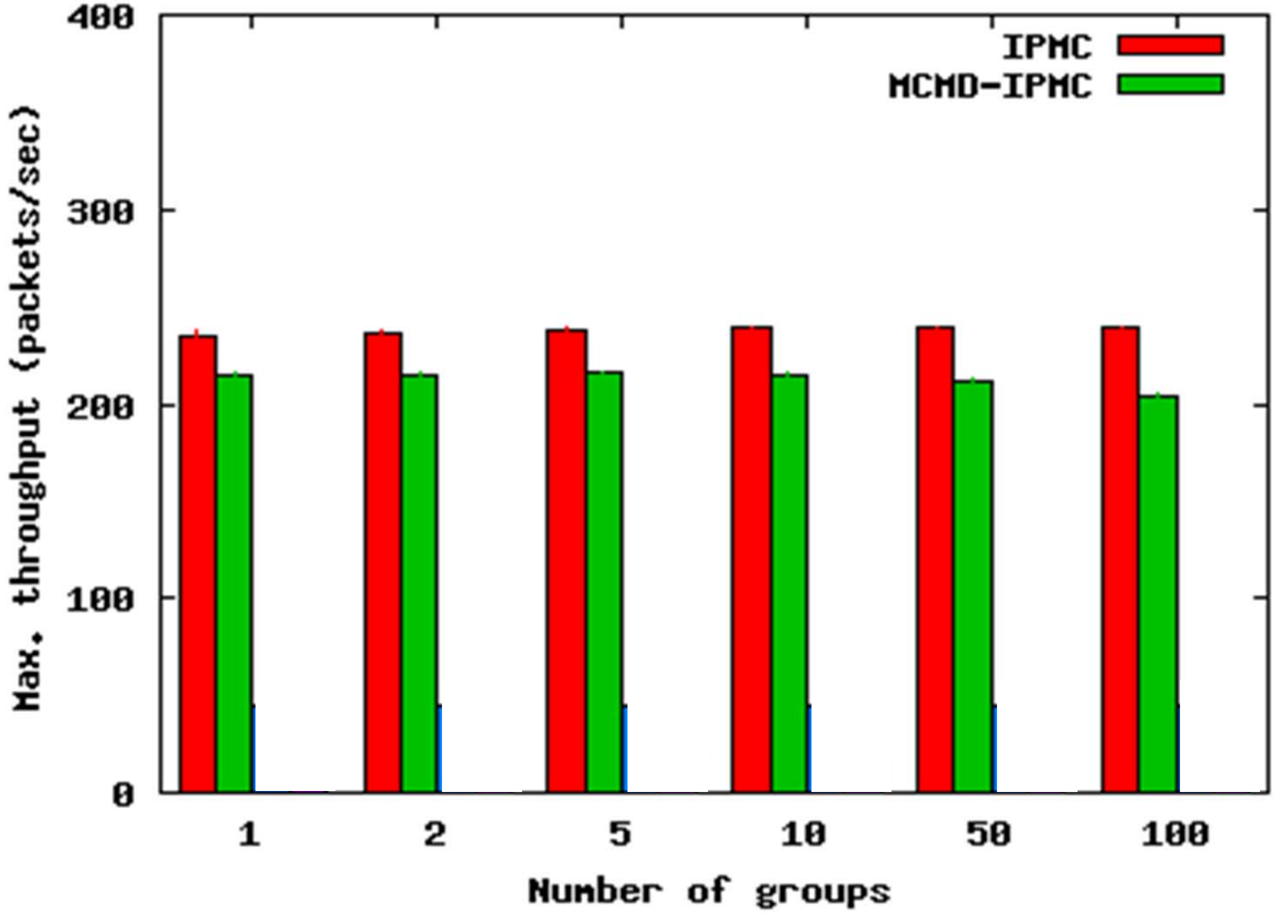
Network Overhead

- Gossip Layer uses constant background bandwidth on average

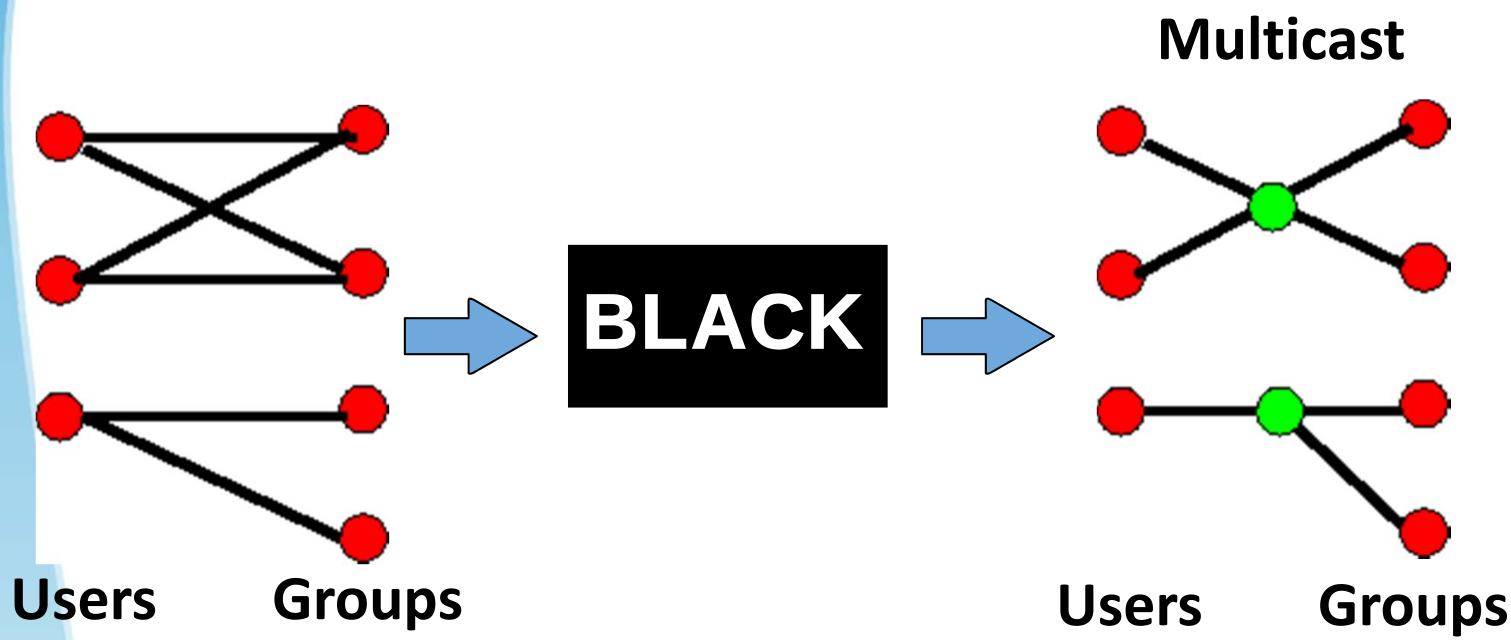


Application Overhead

- Insignificant overhead when mapping logical IPMC group to physical IPMC group.



Optimization questions



Optimization Questions

- Assign IPMC and unicast addresses s.t.
  - Min. receiver filtering
  - Min. network traffic
  - Min. # IPMC addresses
  - ... yet have all messages delivered to interested parties

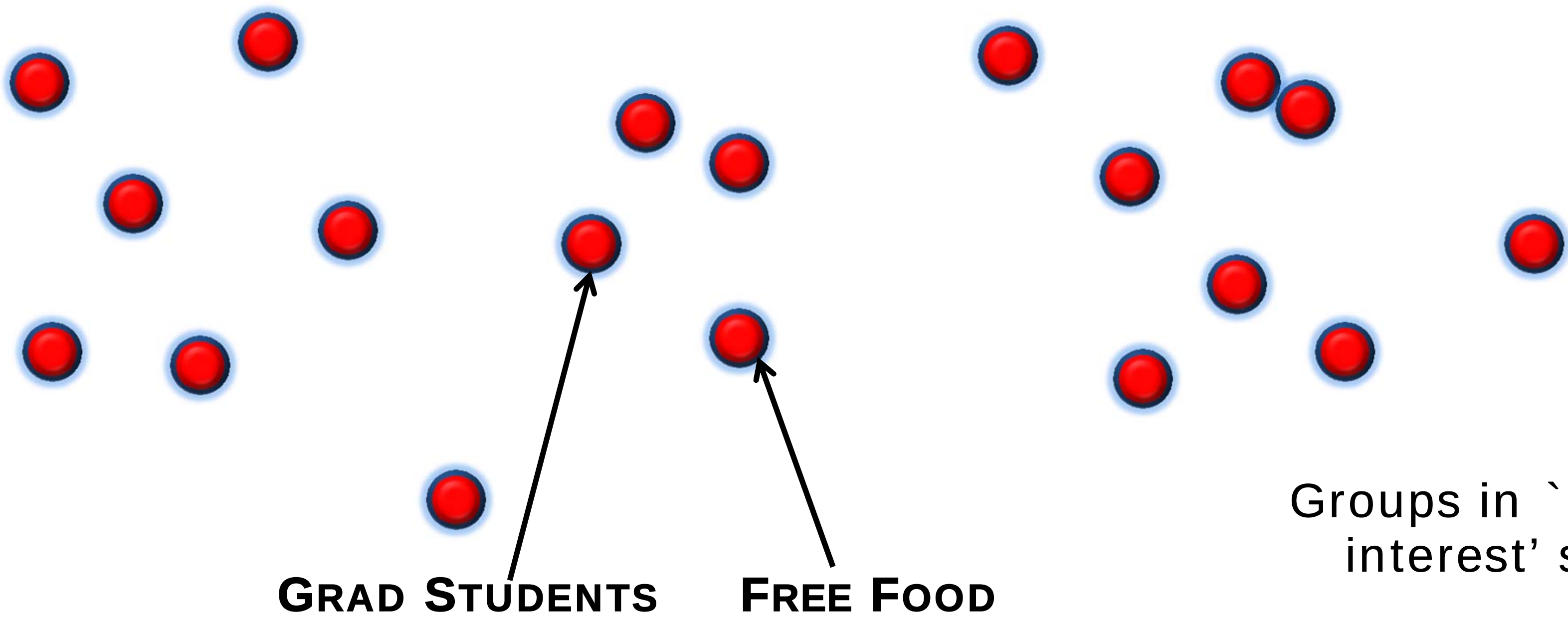
Optimization Questions

- Assign IPMC and unicast addresses s.t.
  - $\leq \alpha$ % receiver filtering (hard)
  - (1) Min. network traffic
  - $\leq M$ # IPMC addresses (hard)

- Prefers sender load over receiver load.

- Control knobs part of administrative policy.

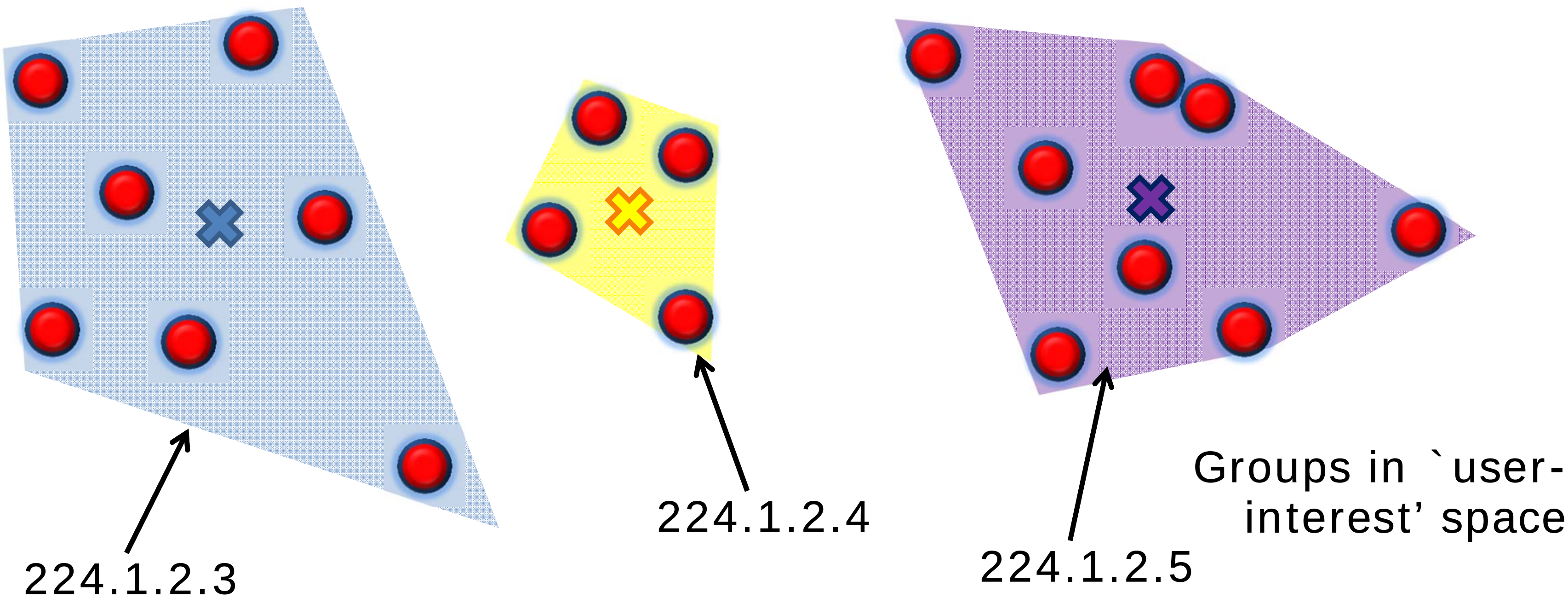
MCMD Heuristic



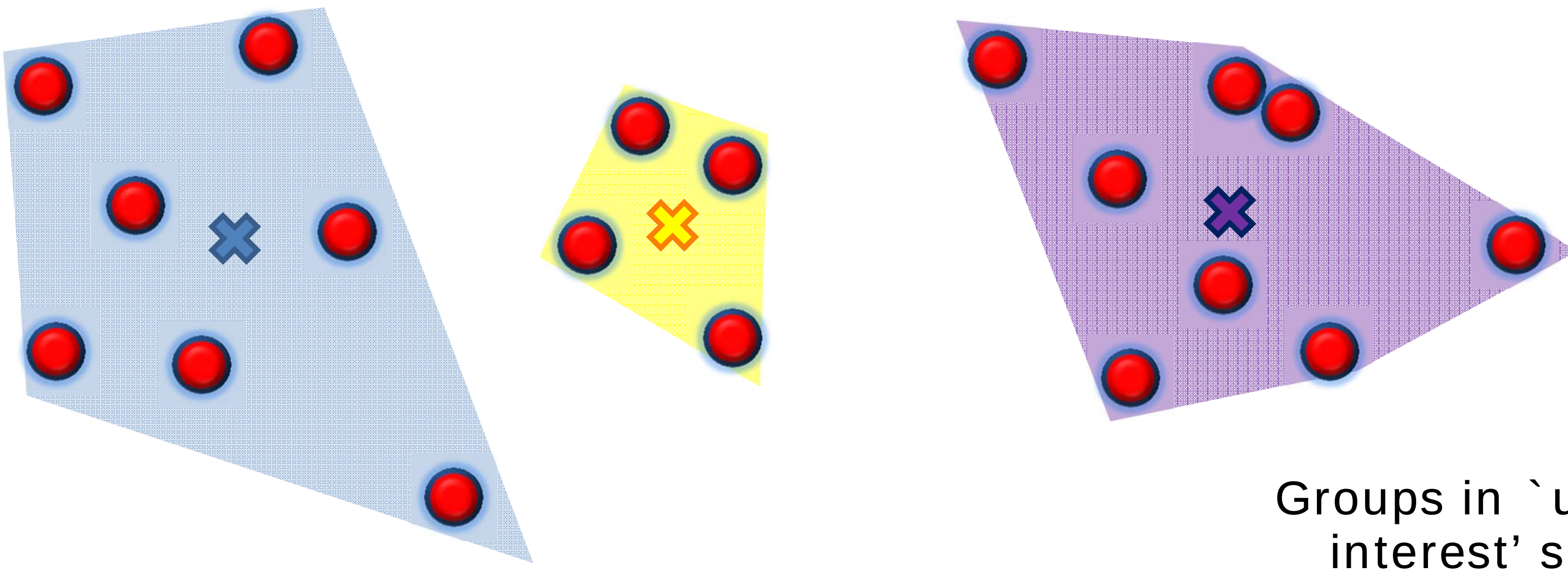
Groups in `user-interest' space

$$(0,1,1,1,1,0,1,0,0,0,1,1)$$

MCMD Heuristic



MCMD Heuristic

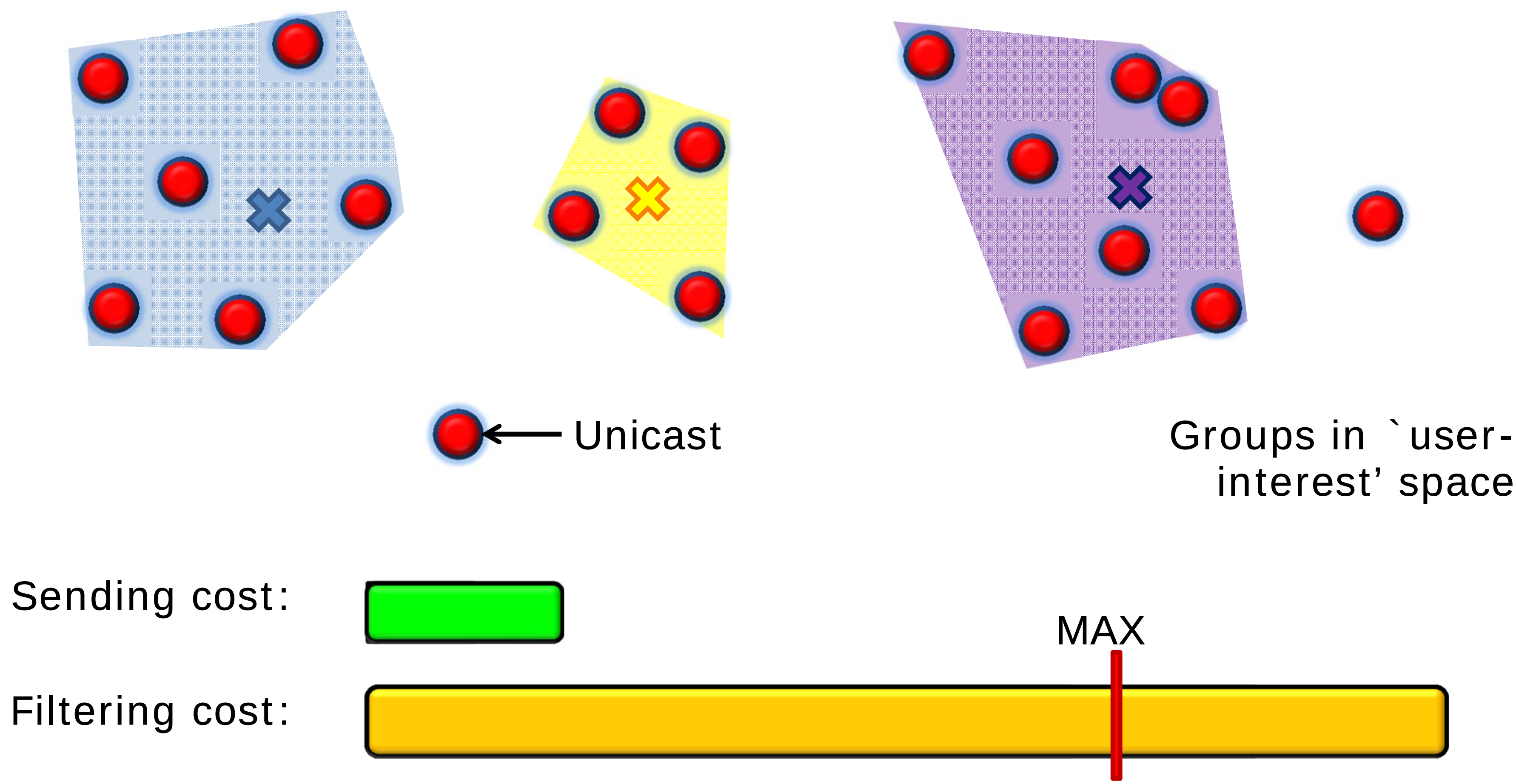


Groups in `user-interest' space

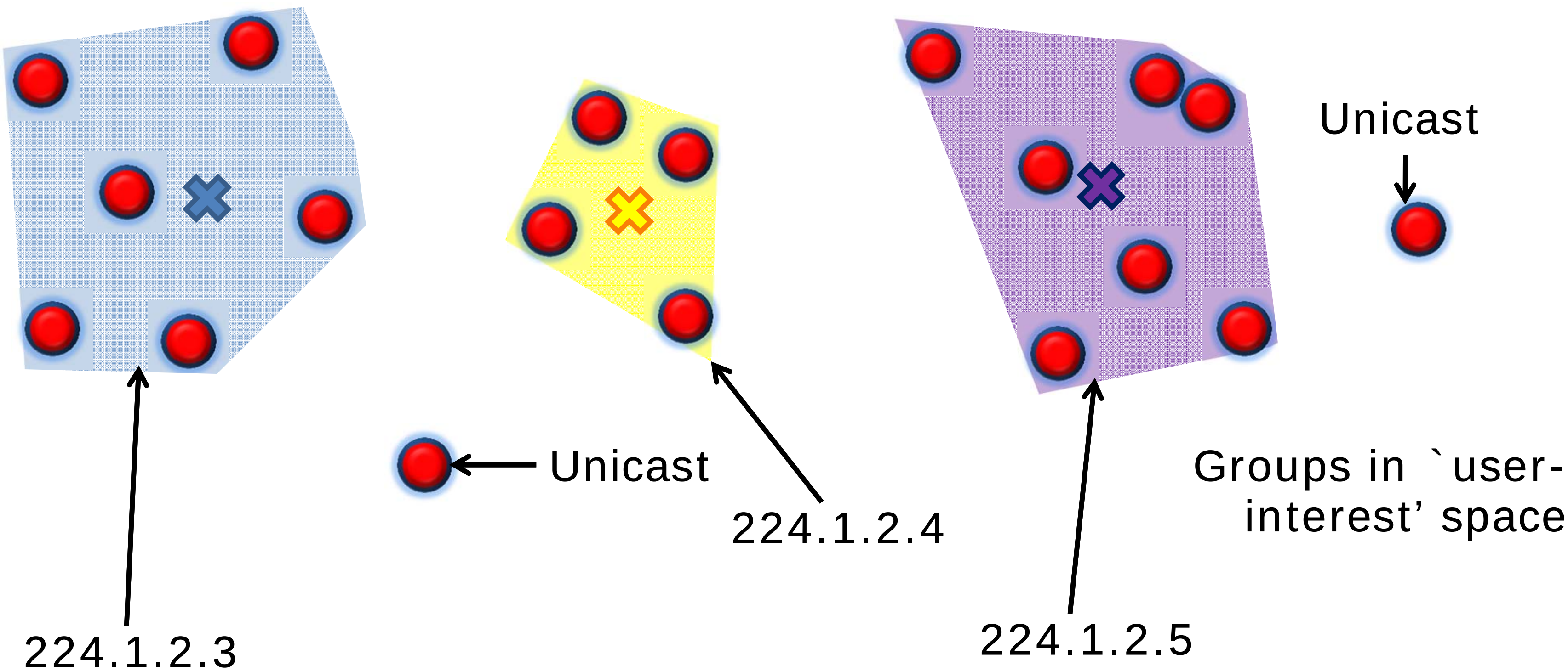
Sending cost:

Filtering cost:

MCMD Heuristic



MCMD Heuristic



Dr. Multicast

- ***Policy:*** Permits data center operators to selectively enable and control IPMC.
- ***Transparency:*** Standard IPMC interface to user, standard IGMP interface to network.
- ***Scalability:*** Uses IPMC when possible, otherwise point-to-point unicast.
- ***Robustness:*** Distributed, fault-tolerant service.

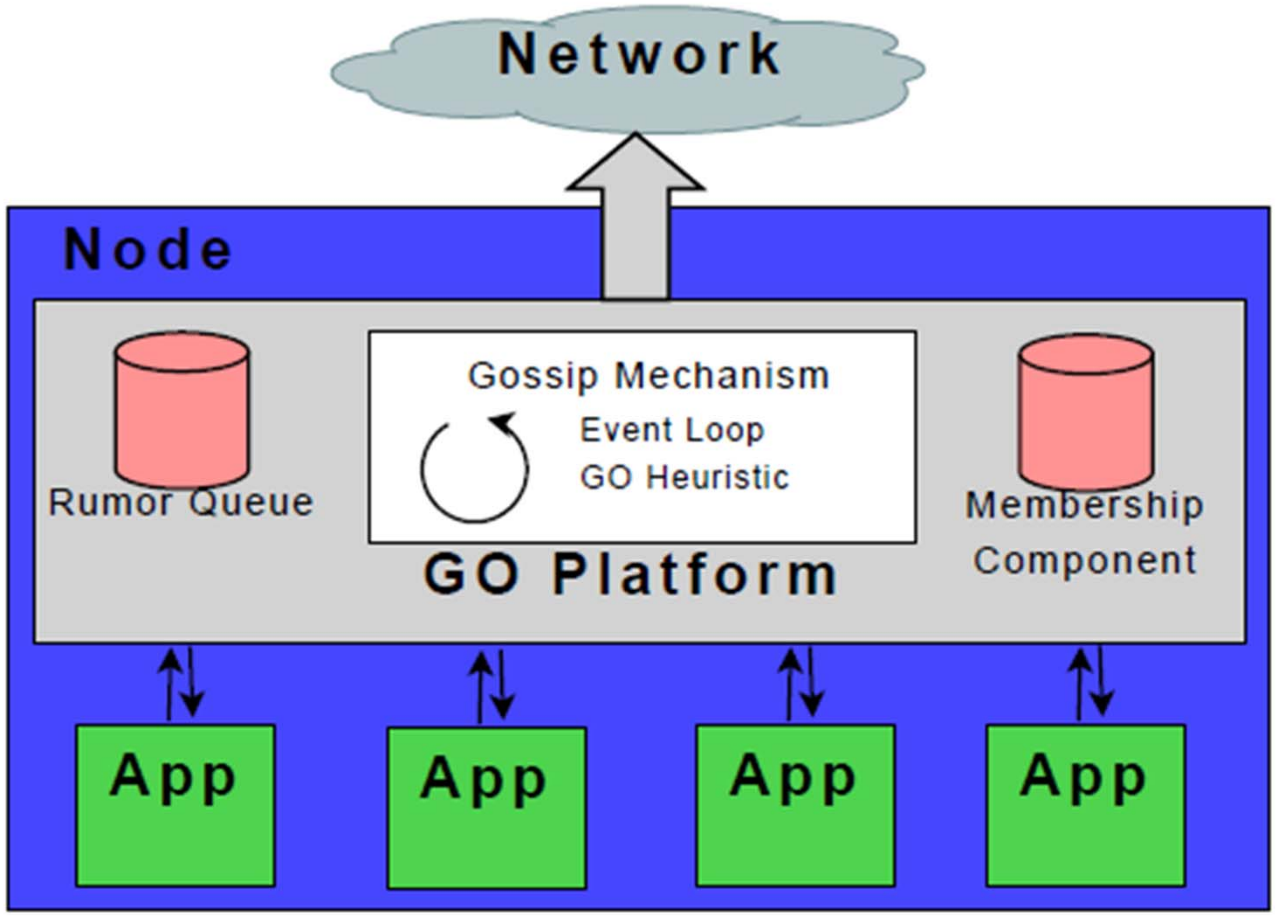
Talk Outline

- Dr. Multicast (**MCMD**)
  - Group scalability in IP Multicast.
- Gossip Objects (**GO**) platform
  - Group scalability in gossip.
- Affinity
  - GO+MCMD optimizations based on group overlaps
  - Explore the properties of overlaps in data sets
- Conclusion

Gossip

- **Def:** *Exchange information with a random node once per round.*
- Has appealing properties:
  - Bounded network traffic.
  - Scalable in group size.
  - Robust against failures.
  - Simple to code.
- When # of groups scales up, lose

GO Platform



Network
Node
Gossip Mechanism
Event Loop
GO Heuristic
Rumor Queue
Membership Component
GO Platform
App
App
App
App

Random gossip

- **Recipient selection:**
  - – Pick node *d* uniformly at random.
- **Content selection:**
  - – Pick a rumor *r* uniformly at random.

Observations

- Gossip rumors usually small:
  - Incremental updates.
  - Few bytes hash of actual information.
- Packet size below MTU irrelevant.
  - *Stack* rumors in a packet.
  - But which ones?
- Rumors can be delivered indirectly.
  - Uninterested node might forward

Random gossip w. stacking

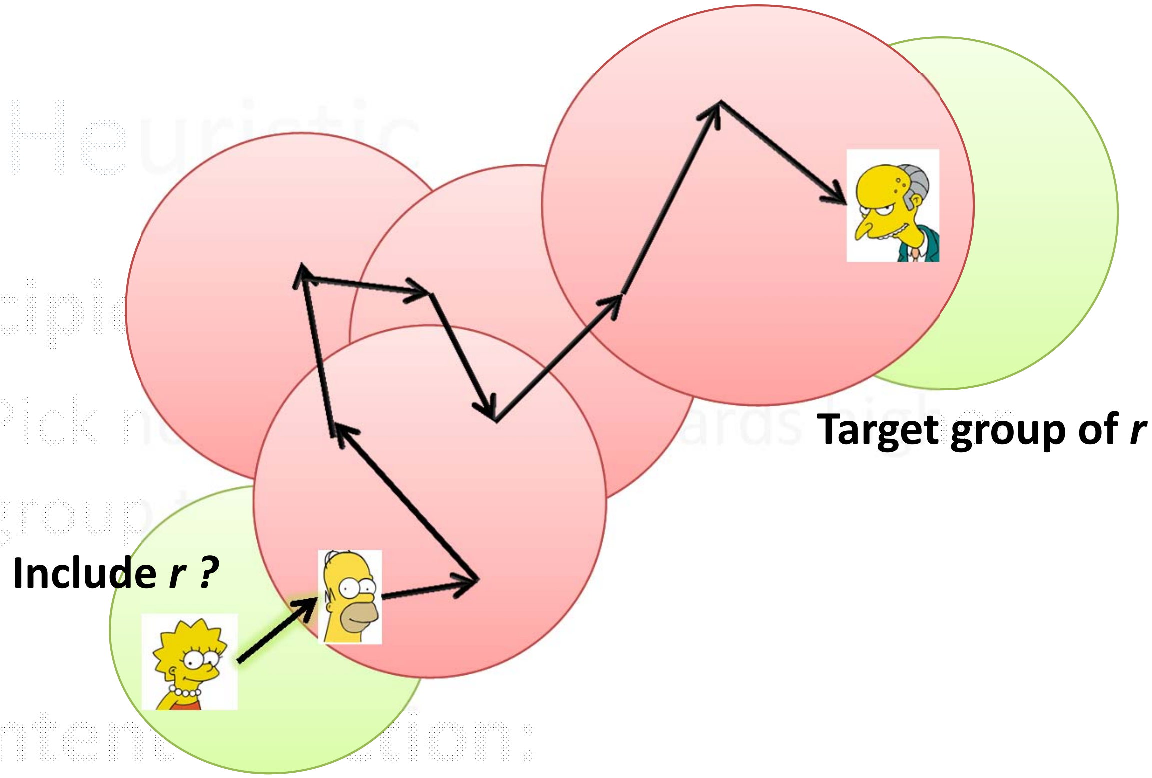
- **Recipient selection:**
  - Pick node *d* uniformly at random.

- **Content selection:**
  - Fill packet with rumors picked uniformly at random.

GO Heuristic

- **Recipient selection:**
  - Pick node *d* biased towards higher group traffic.
- **Content selection:**
  - Compute the *utility* of including rumor *r*
    - Probability of *r* infecting an uninfected host when it reaches the target group.
  - Pick rumors to fill packet with probability proportional to utility.

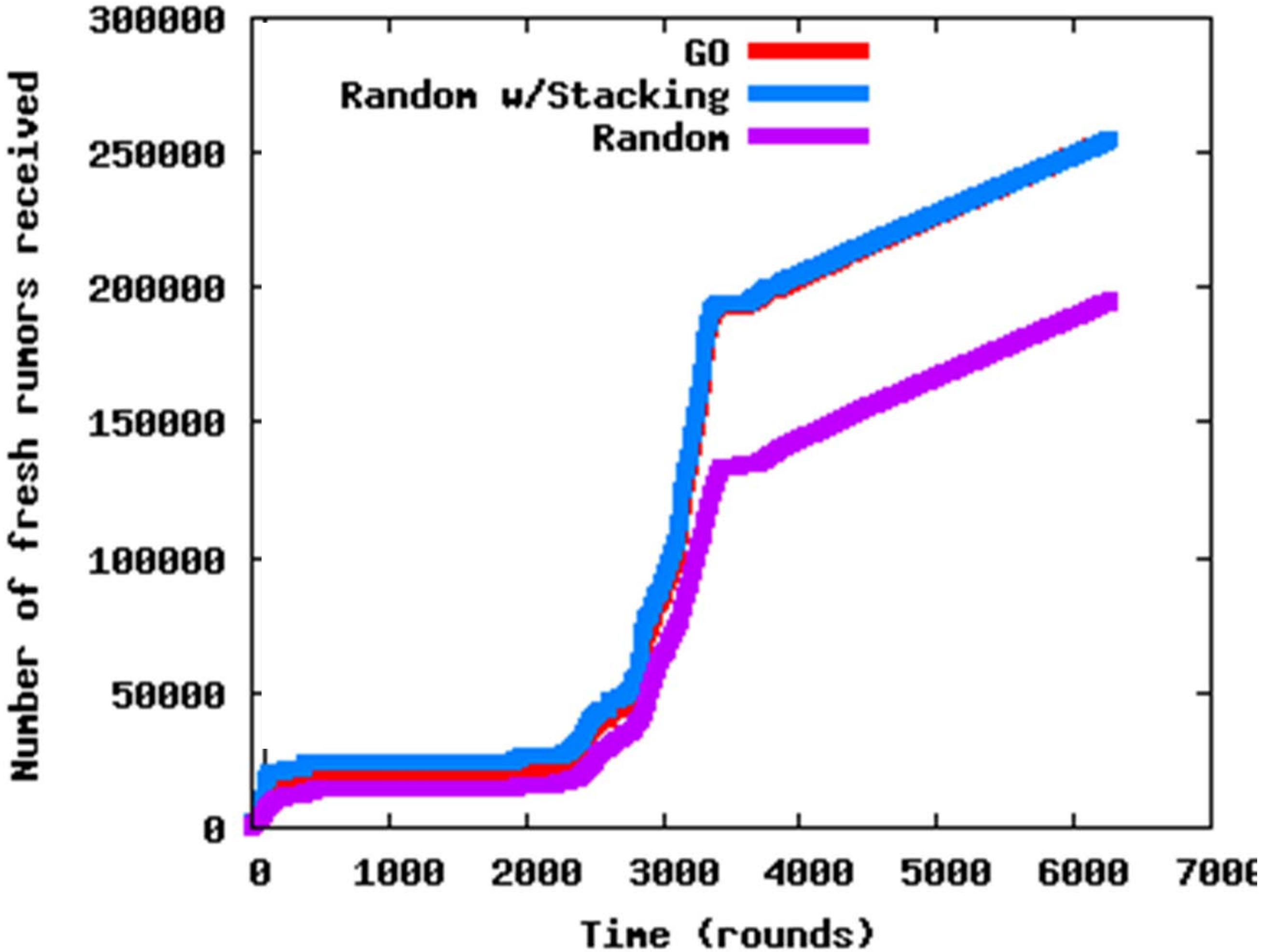
GO Heuristic



- Compute the *utility* of including rumor *r*
    - Probability of *r* infecting an uninfected host when it reaches the target group.
- Pick rumors to fill packet with probability proportional to utility.

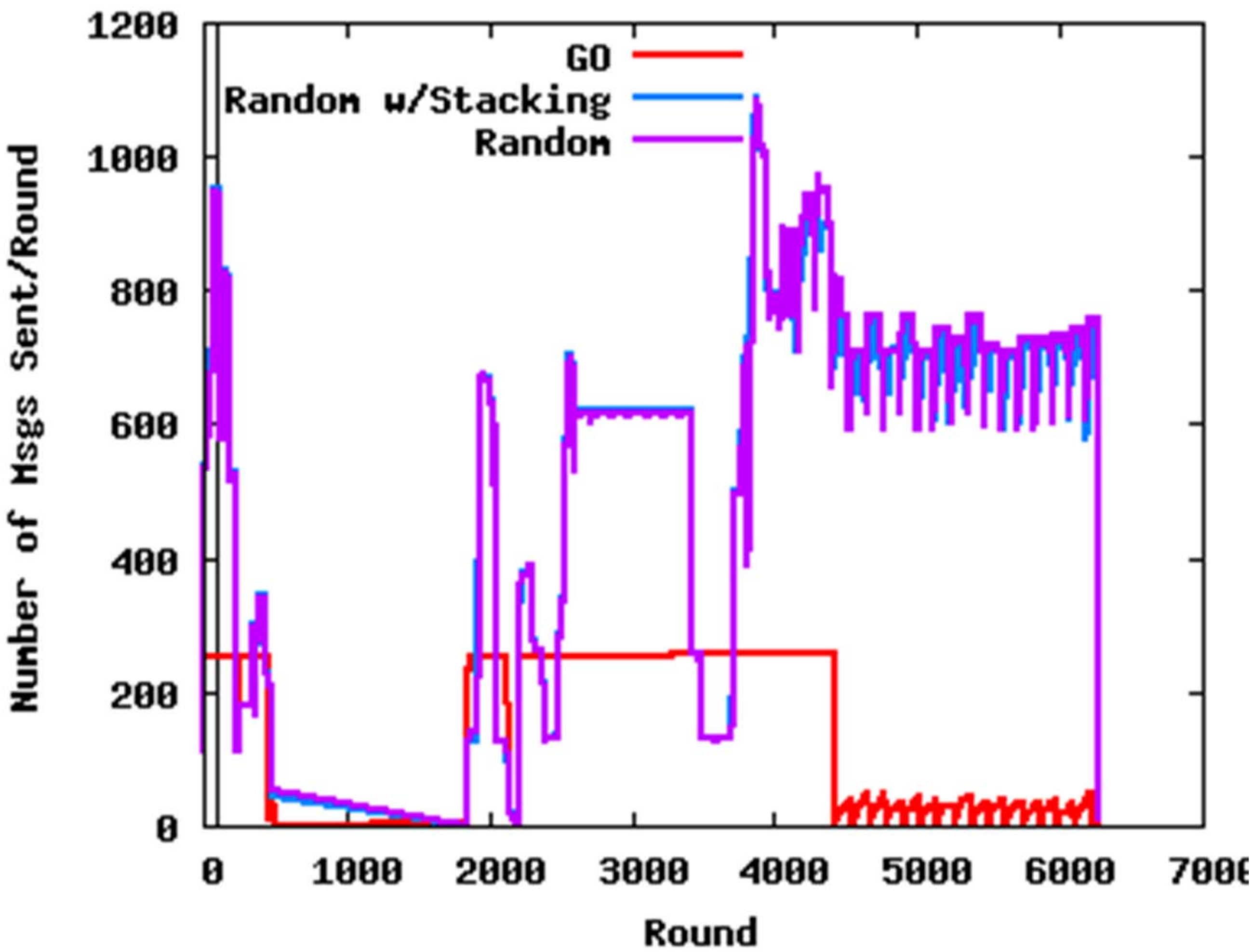
Evaluation

- IBM Websphere trace (1364 groups)



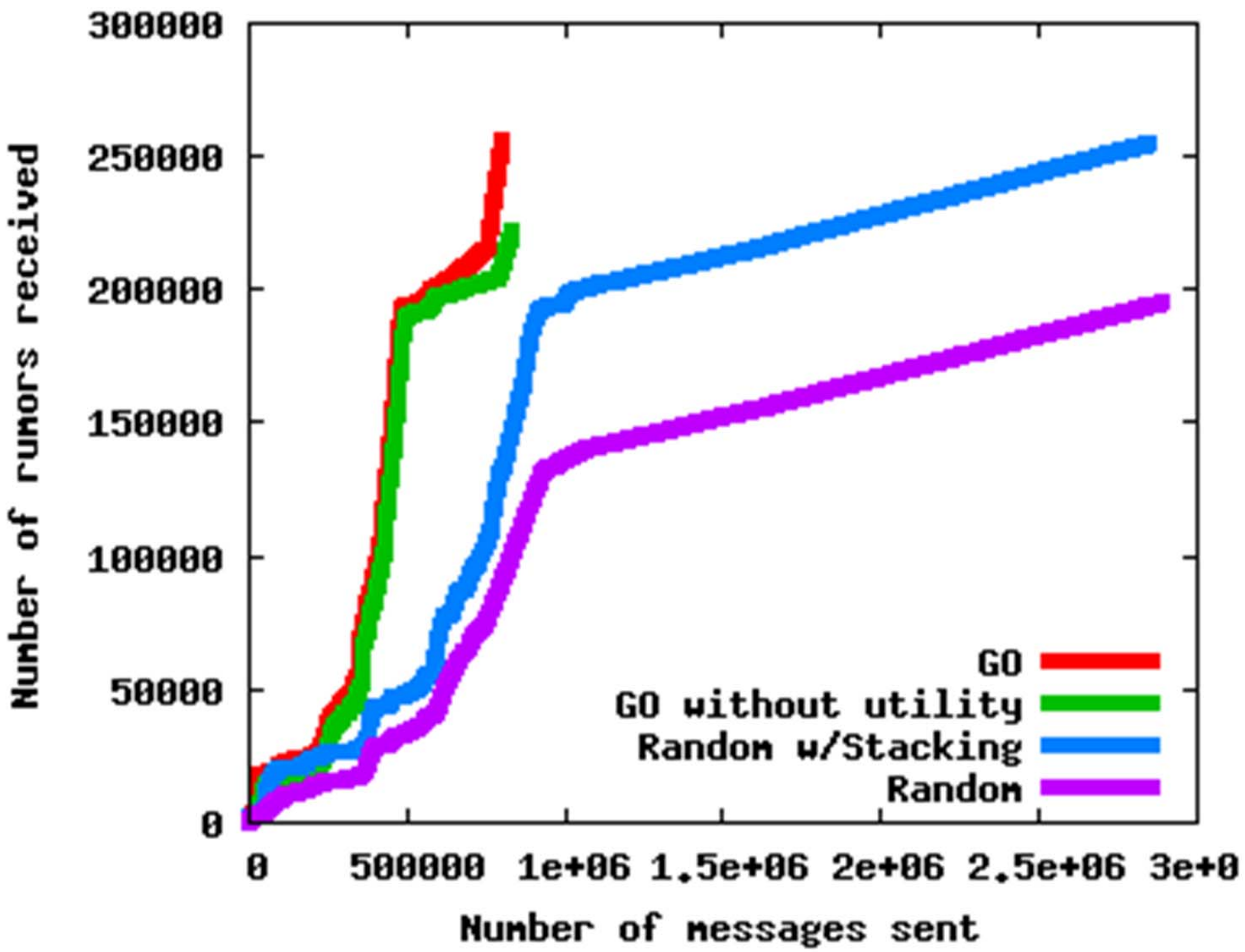
Evaluation

- IBM Websphere trace (1364 groups)



Evaluation

- IBM Websphere trace (1364 groups)



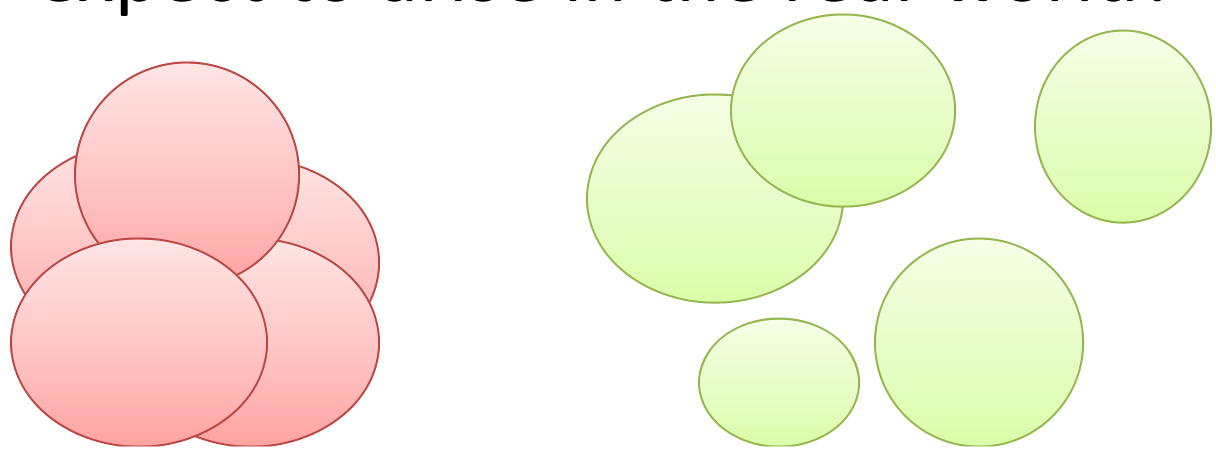
Talk Outline

- Dr. Multicast (**MCMD**)
  - Group scalability in IP Multicast.
- Gossip Objects (**GO**) platform
  - Group scalability in gossip.
- Affinity
  - GO+MCMD optimizations based on group overlaps.
  - Explore the properties of overlaps in data sets.
- Conclusion

Affinity

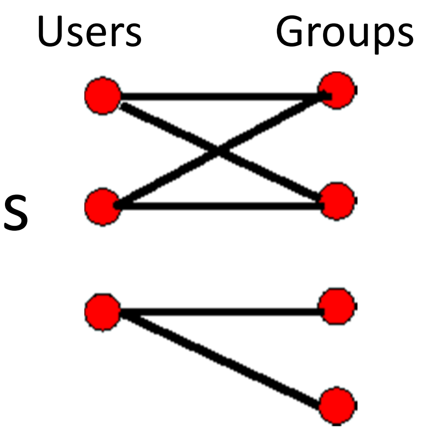
- Both **MCMD** and **GO** have optimizations that depend on pairwise group overlaps (*affinity)*.

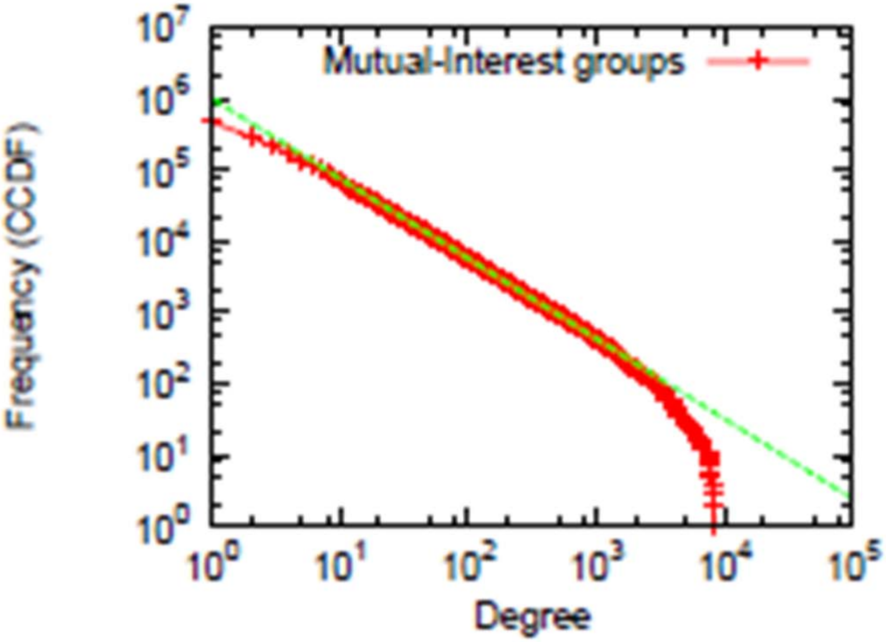
- What degree of affinity should we expect to arise in the real-world?

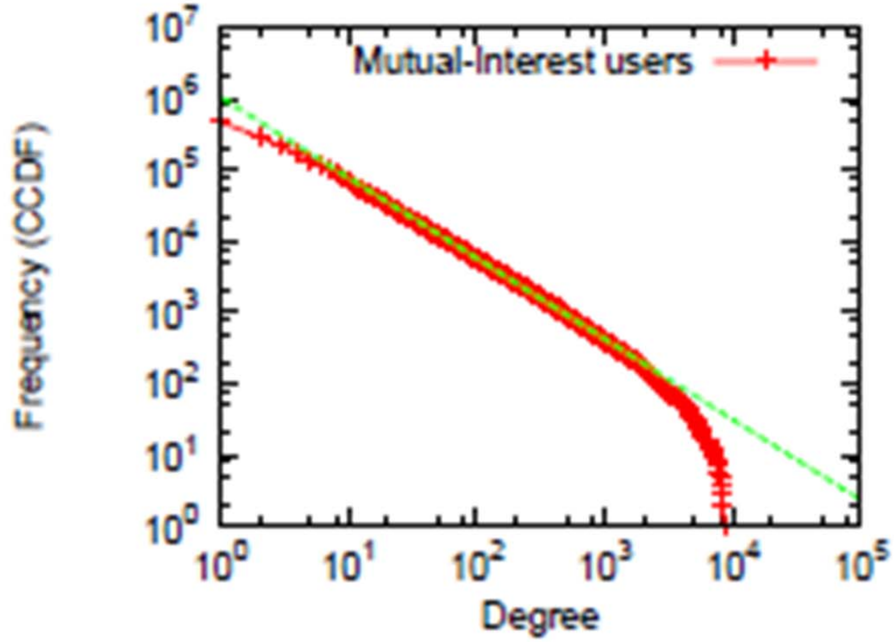


Data sets/models

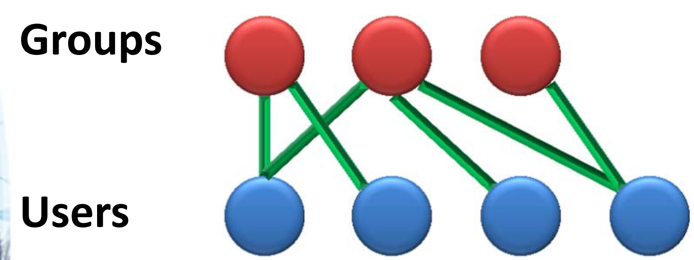
- What's in a ``group'' ?
- Social:
  - Yahoo! Groups
  - Amazon Recommendations
  - Wikipedia Edits
  - LiveJournal Communities
  - ***Mutual Interest Model***
- Systems:
  - IBM Websphere
  - ***Hierarchy Model***





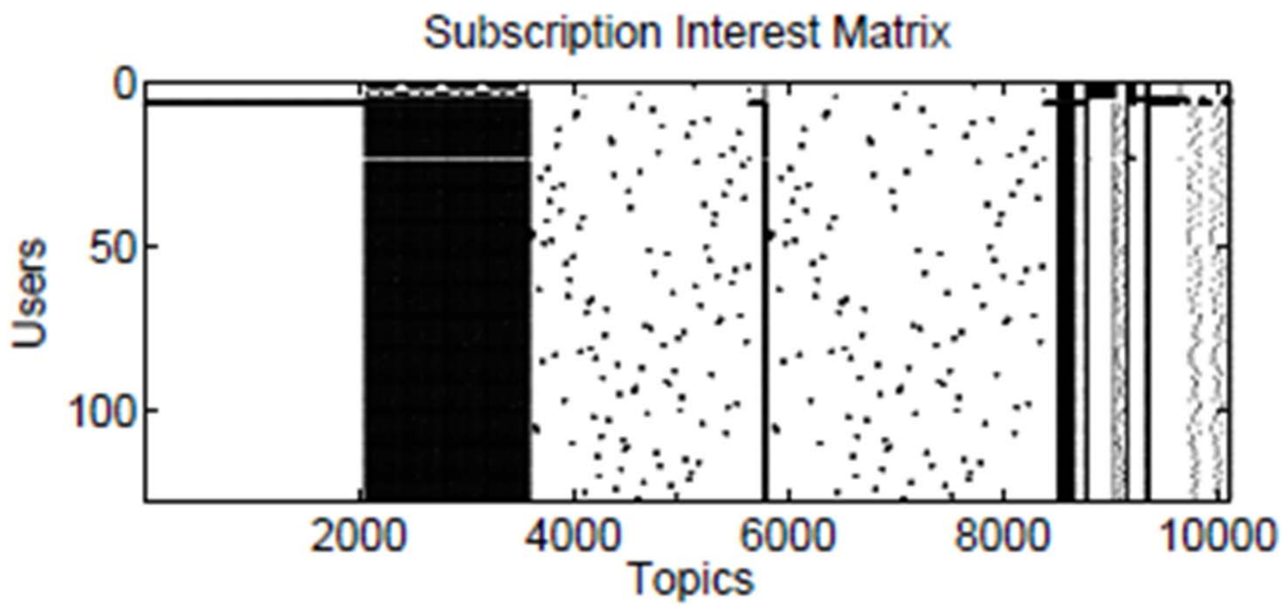


- ***Mutual Interest*** model:
  - Preferential attachment for bipartite graphs.



Systems Data Set

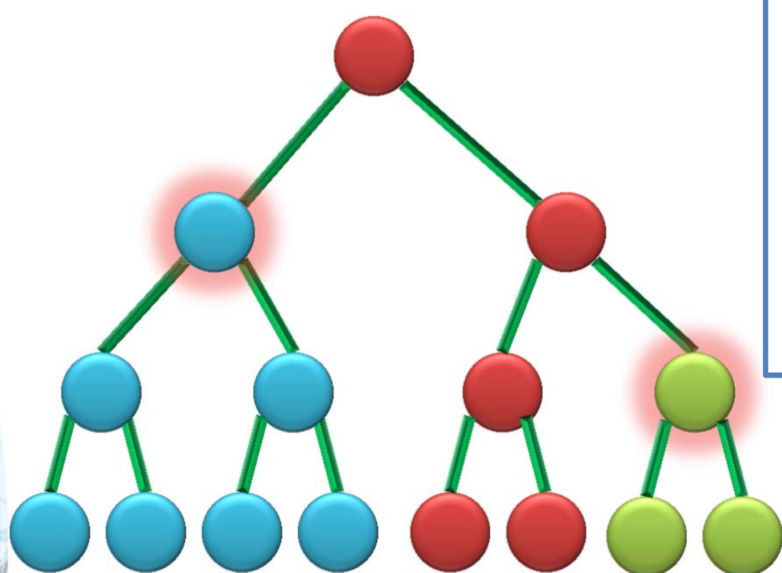
- IBM Websphere has remarkable structure!



- Typical for real-world systems?
  - Only one data point.

Systems Data Set

- Distributed systems tend to be hierarchically structured.
- **Hierarchy model**
  - Motivated by Live Objects.



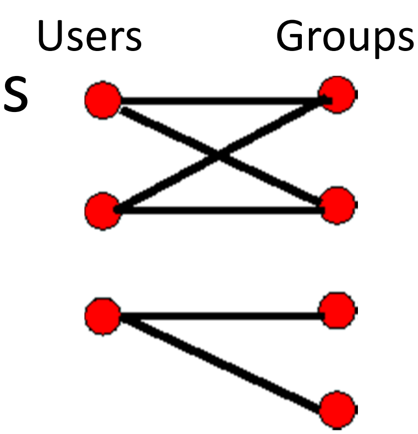
Thm: Expect a pair of users to overlap in

$$\Theta\left(\frac{n}{\log^2 n}\right)$$

groups .

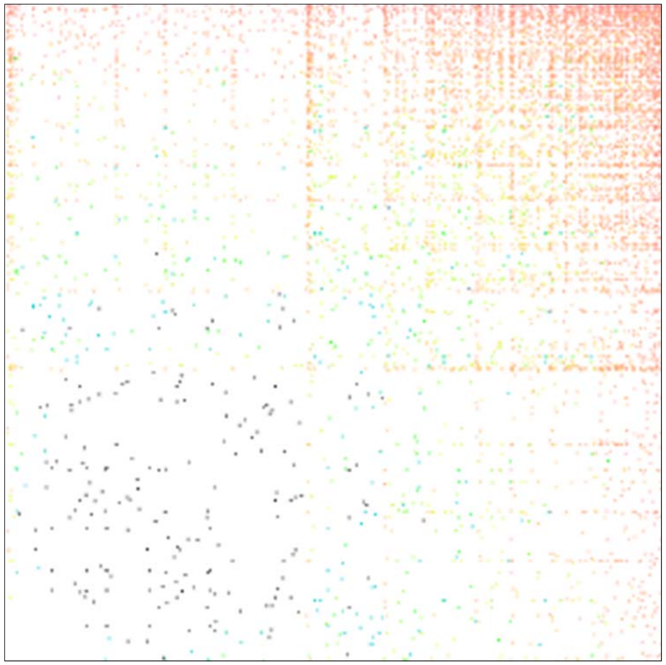
Data sets/models

- Social:
  - Yahoo! Groups
  - Amazon Recommendations
  - Wikipedia Edits
  - LiveJournal Communities
  - ***Mutual Interest Model***
- Systems:
  - IBM Websphere
  - ***Hierarchy Model***

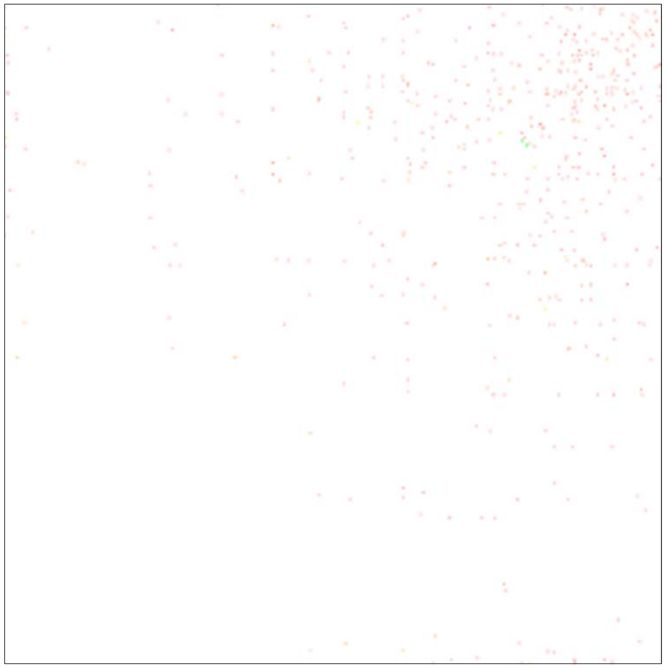


Group similarity

- **Def:** Similarity of groups *j,j'* is $\mathrm{SIM}(j, j') = \dfrac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}$.



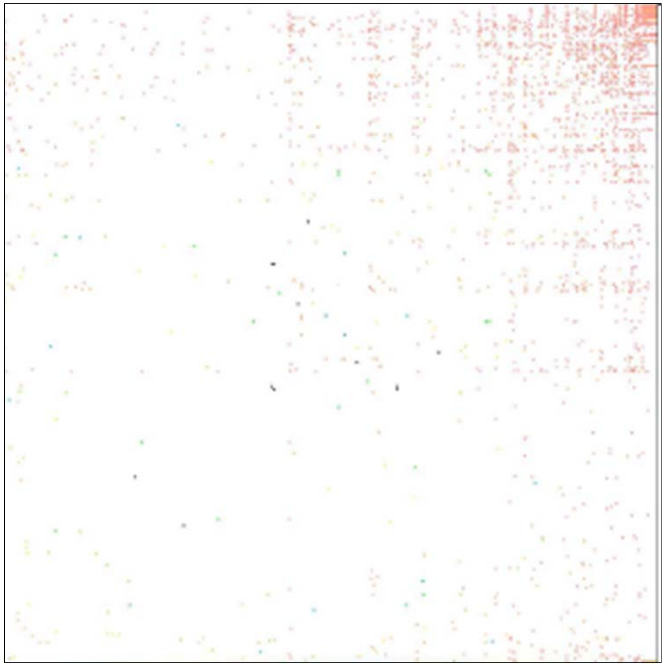
Wikipedia



LiveJournal

Group similarity

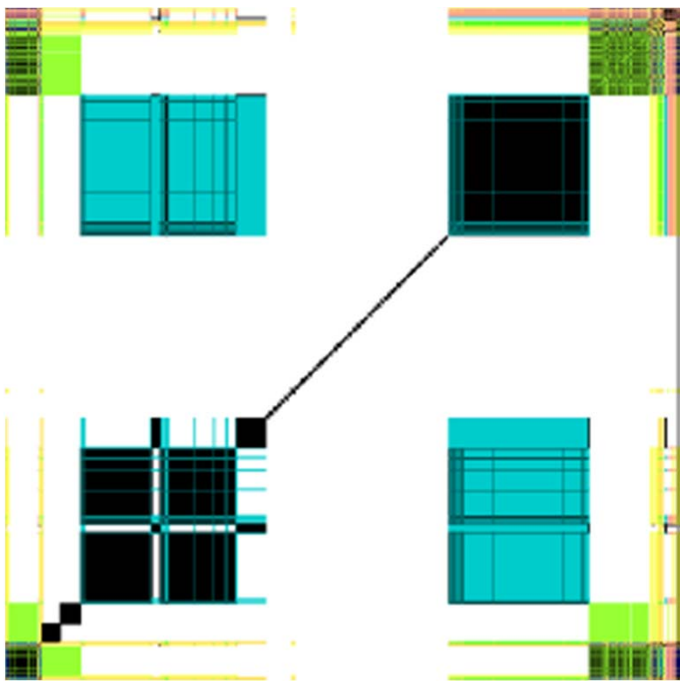
- **Def:** Similarity of groups *j,j'* is $\mathrm{SIM}(j,j') = \dfrac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}$.



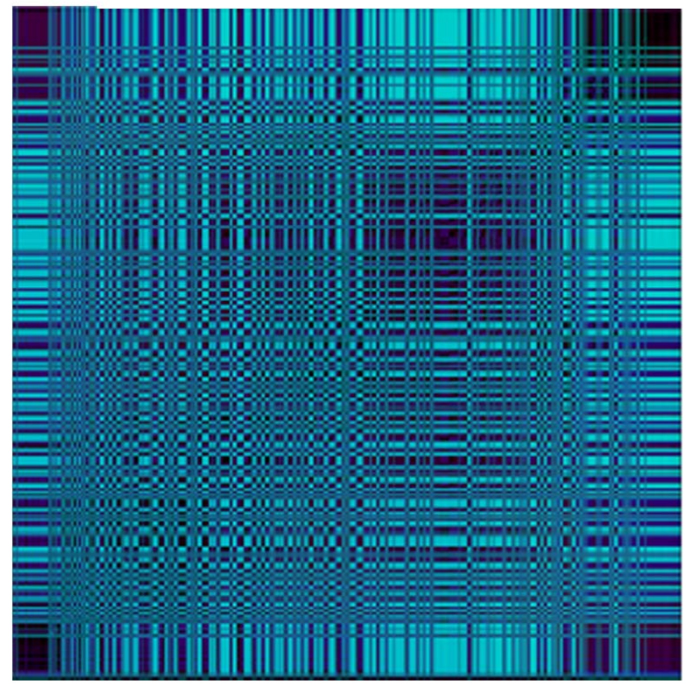
Mutual Interest Model

Group similarity

- **Def:** Similarity of groups *j,j'* is $\mathrm{SIM}(j,j') = \dfrac{|G_j \cap G_{j'}|}{\max\{|G_j|, |G_{j'}|\}}.$



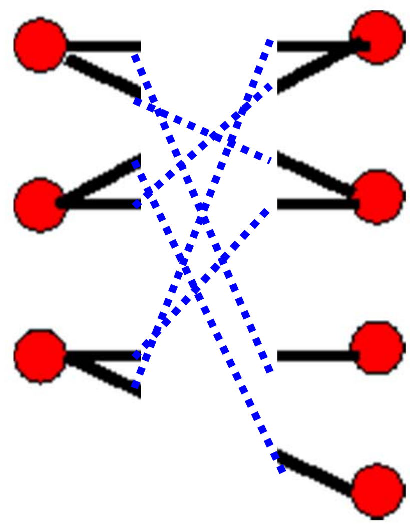
IBM Websphere



Hierarchy model

Baseline overlap

- Is the similarity we see a real effect?
- Consider a random graph with the same degree distributions as a *baseline*.
- **Spokes model:**



Baseline overlap

- Plot difference between data and *Spokes*

$$\Delta(j, j') = \mathrm{SIM}_{\Gamma}(j, j') - \mathrm{SIM}_{\hat{\Gamma}}(j, j').$$

- At most 50 samples per group size pair.

Data set/model	Avg. Δ value
Wikipedia	- 0.004
Amazon	0.031
Yahoo! Groups	0.000
Mutual Interest Model	0.006
IBM Websphere	**0.284**
Hierarchy Model	**0.358**

Looking pretty random

Conclusions

- Group communication important, but group scalability is lacking.

- **Dr. Multicast** harnesses IPMC in data centers.
  - **Impact:** HotNets paper + NSDI Best Poster award.
  - Solution being adopted by CISCO and IBM.

Conclusions

- **GO** provides group scalability for gossip.
  - **Impact:** LADIS paper + Invited to the P2P Conference.
  - Platform will run under the Live Objects framework.

- Characterizing and exploiting group affinity in systems is exciting current and future work.

Publications

GO: Platform Support For Gossip Applications.

With Ken Birman, Qi Huang, Deepak Nataraj. **LADIS** ‘09. Invited to **P2P** '09.

Adaptively Parallelizing Distributed Range Queries.

With Adam Silberstein, Brian Cooper, Rodrigo Fonseca. **VLDB** ’09.

Slicing Distributed Systems.

With Vincent Gramoli, Ken Birman, Anne-Marie Kermarrec, Robbert van Renesse. **PODC** ’08 (short). In ***IEEE Transactions on Computers*** 2009.

Dr. Multicast: *Rx* for Data Center Communication Scalability.

With Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, Yoav Tock. **Hotnets** ‘08. **LADIS** ‘08. **NSDI** ‘08 (Best Poster).

Hyperspaces for Object Clustering and Approximate Matching in P2P Overlays.

With Bernard Wong, Emin Gun Sirer. **HotOS** ‘07.

[redacted]

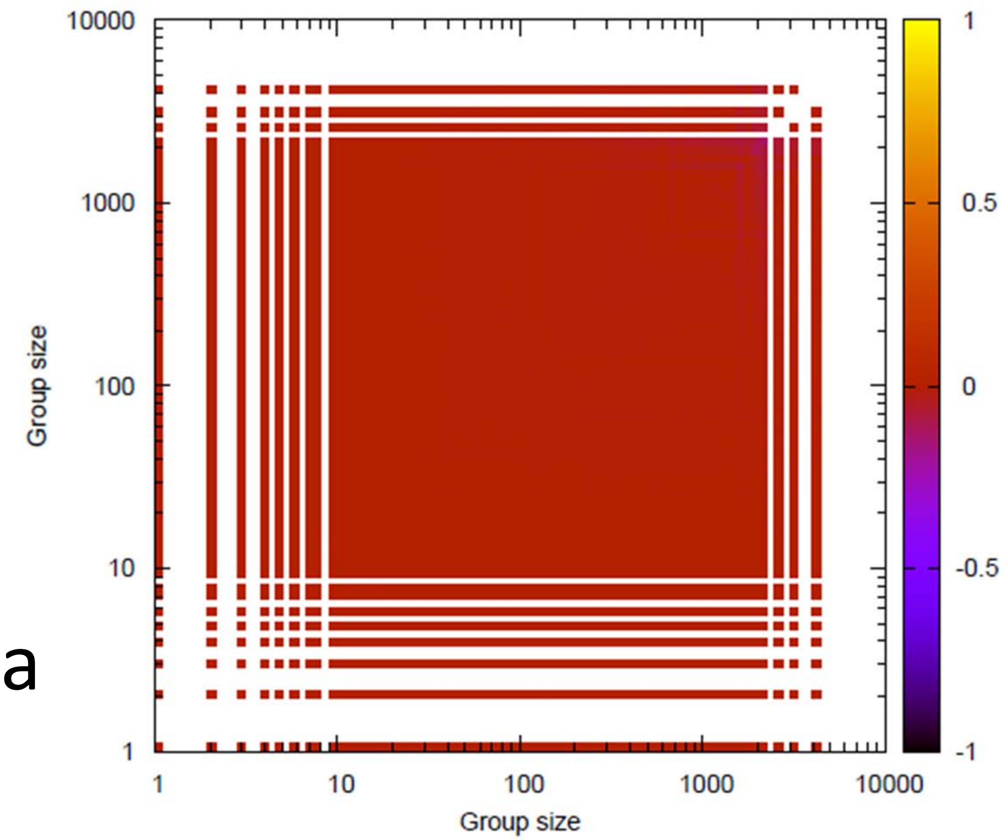
Baseline overlap

- Plot difference between data and *Spokes*

$$\Delta(j, j') = \mathrm{SIM}_{\Gamma}(j, j') - \mathrm{SIM}_{\hat{\Gamma}}(j, j').$$

- **Cell:** Avg. Δ over particular group sizes.

Wikipedia



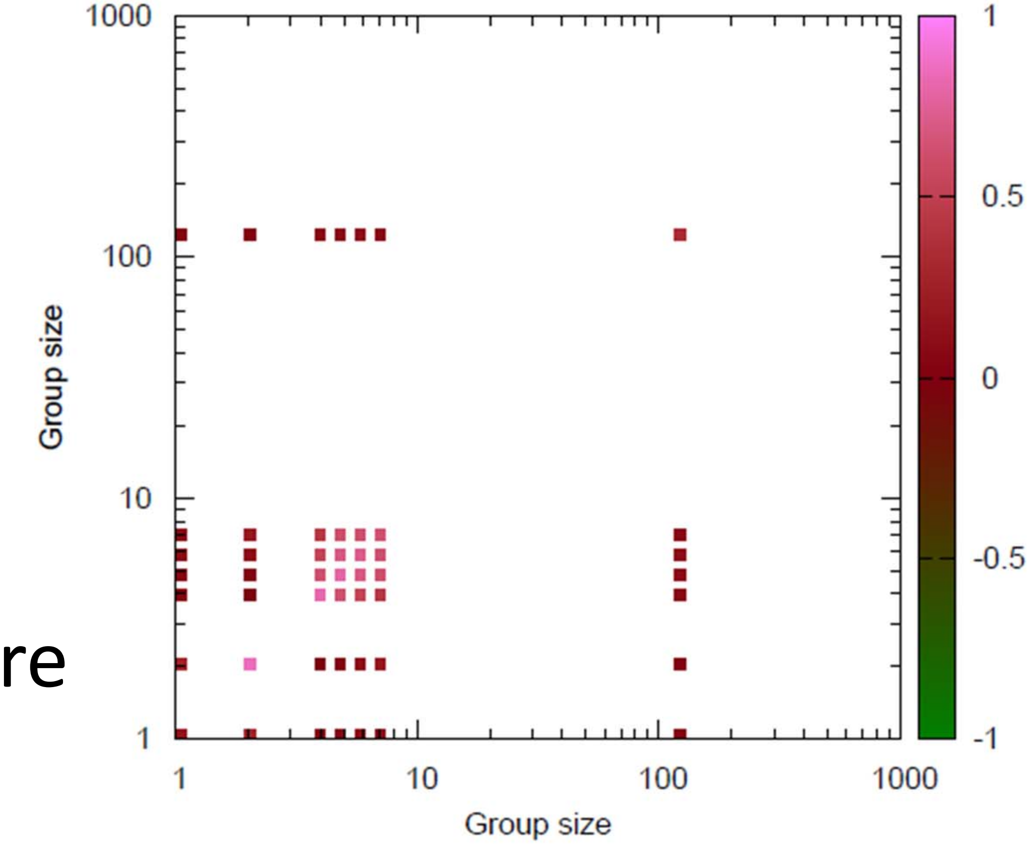
Baseline overlap

- Plot difference between data and *Spokes*

$$\Delta(j, j') = \mathrm{SIM}_{\Gamma}(j, j') - \mathrm{SIM}_{\hat{\Gamma}}(j, j').$$

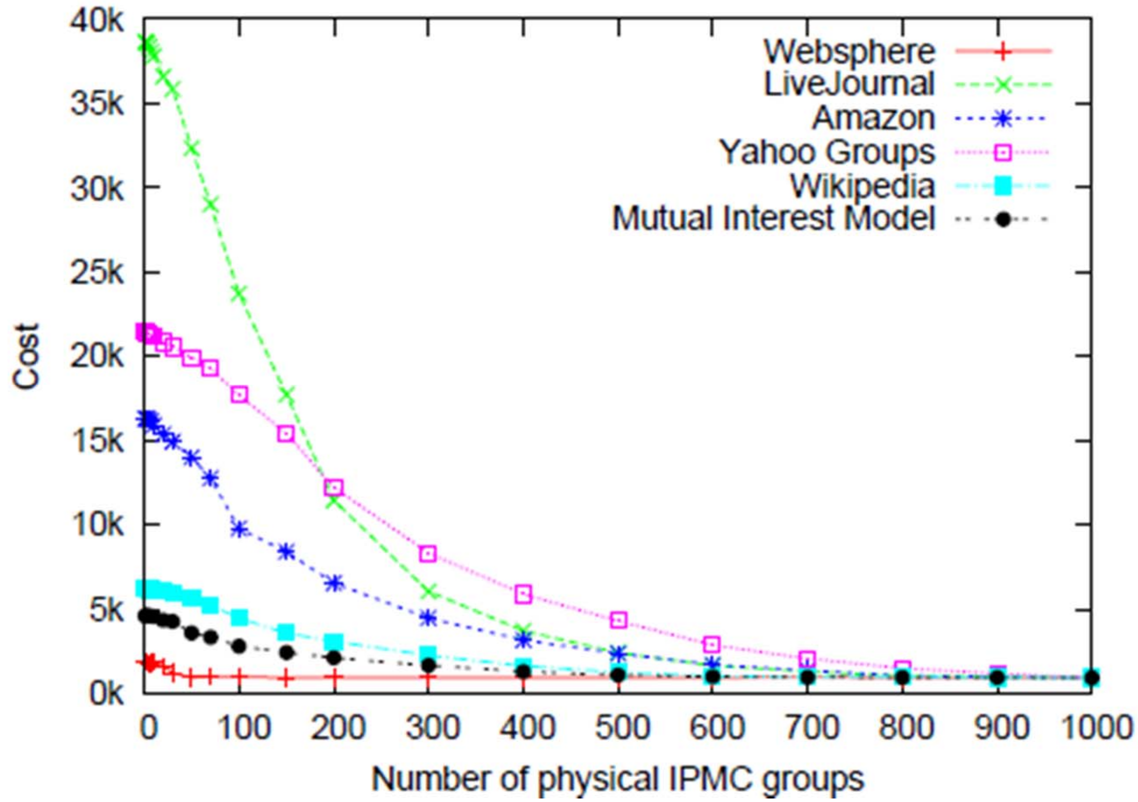
- **Cell:** Avg. Δ over particular group sizes.

Websphere



Affinity results

- Social affinity pretty random.
- Websphere has substantial overlaps.
- MCMD Heuristic does well in all cases:



Conclusions

- Group communication important, but group scalability is lacking.
- Dr. Multicast harnesses IPMC in data centers.
  - **Impact:** HotNets paper + NSDI Best Poster award.
  - Solution being adopted by CISCO and IBM.
- **GO** provides group scalability for gossip.
  - **Impact:** LADIS paper + Invited to the P2P Conference.
  - Platform will run under the Live Objects framework.
- Characterizing and exploiting group affinity in systems is exciting current and future work.

Publications

- **GO: Platform Support For Gossip Applications.**
  With Ken Birman, Qi Huang, Deepak Nataraj. **LADIS** '09. Invited to **P2P** '09.
- **Adaptively Parallelizing Distributed Range Queries.**
  With Adam Silberstein, Brian Cooper, Rodrigo Fonseca. **VLDB** '09.
- **Slicing Distributed Systems.**
  With Vincent Gramoli, Ken Birman, Anne-Marie Kermarrec, Robbert van Renesse. **PODC** '08. In *IEEE Transactions on Computers* 2009.
- **Dr. Multicast: *Rx* for Datacenter Communication Scalability.**
  With Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, Yoav Tock. **Hotnets** '08. **LADIS** '08. **NSDI** '08 (Best Poster).
- **Hyperspaces for Object Clustering and Approximate Matching in P2P Overlays.**
  With Bernard Wong, Emin Gun Sirer. **HotOS** '07.