A Key Management Scheme for *secure* Multicast

Michel Abdalla , UC San Diego
Yuval Shavitt, Bell Labs, Lucent
Avishai Wool, Bell Labs, Lucent

Current Key Mgmt Schemes

[Wallner, Harder, Agee 9/98] - Internet draft

[Wong, Gouda, Lam - Sigcomm 98]

[Mittra - Sigcomm 97] - Iolus

- Tree structure: groups are partitioned to form a key hierarchy.
- A tree per session.
- Cost (key distributions): $k \log k$,

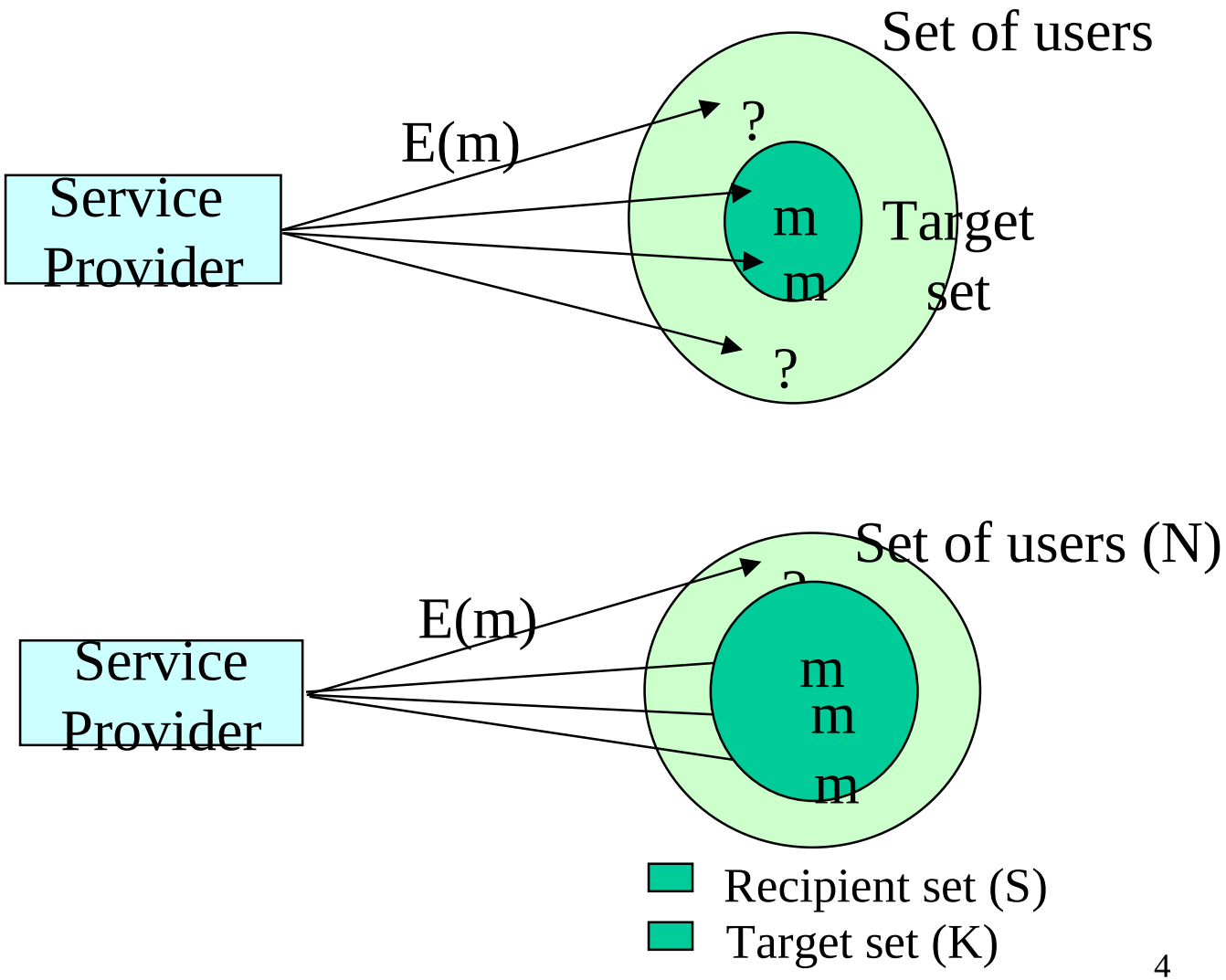
  k = number of receivers.

Why Encryption?

- In many cases:
  - To force customers to pay for a service
  - No real secrets
- Examples:
  - TV/Radio transmissions
  - stock quotes
  - business news
  - e-coupons

Can we trade some security for performance?

Relax the requirements



Requirements

- Efficiency
  - number of decryption key transmissions.
  - number of keys per user.
- Effectiveness
  - non members of the target set can decrypt the message with low probability - no incentive to avoid payment.

New parameter: Redundancy

- The redundancy (***f***) is the guaranteed maximum ratio between the number of recipients and the size of the target set.

$$f = \max_{K} \frac{|S(K)|}{|K|}$$

Our suggestion

- Invest once in a key structure that includes all the receivers in the network.
- Use the key structure to distribute decryption keys to members of a group.
- Allow a limited number of non-members to receive the decryption keys.
- Cost (key distributions): $<k$,

  k = receiver number.

A simulation study

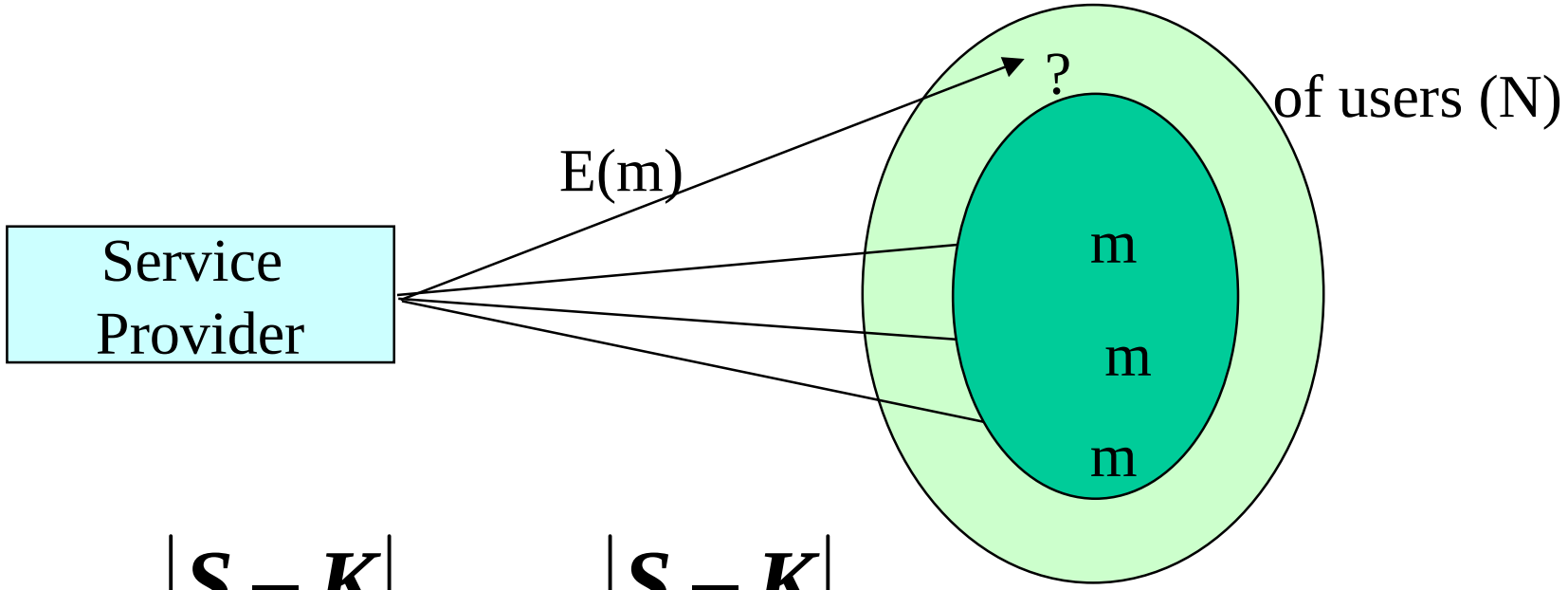
- Study new redundant key allocation schemes.
- Evaluate average case behavior of schemes.

  (We also have a theoretical lower bound.)
- … as a function of the target set (K) size.
- Use a redundancy factor of ***f=2.***

Measures of efficiency

- Number of decryption key transmissions ***(t)***
- Actual redundancy ***(f_a)***
- Opportunity ***(η)***

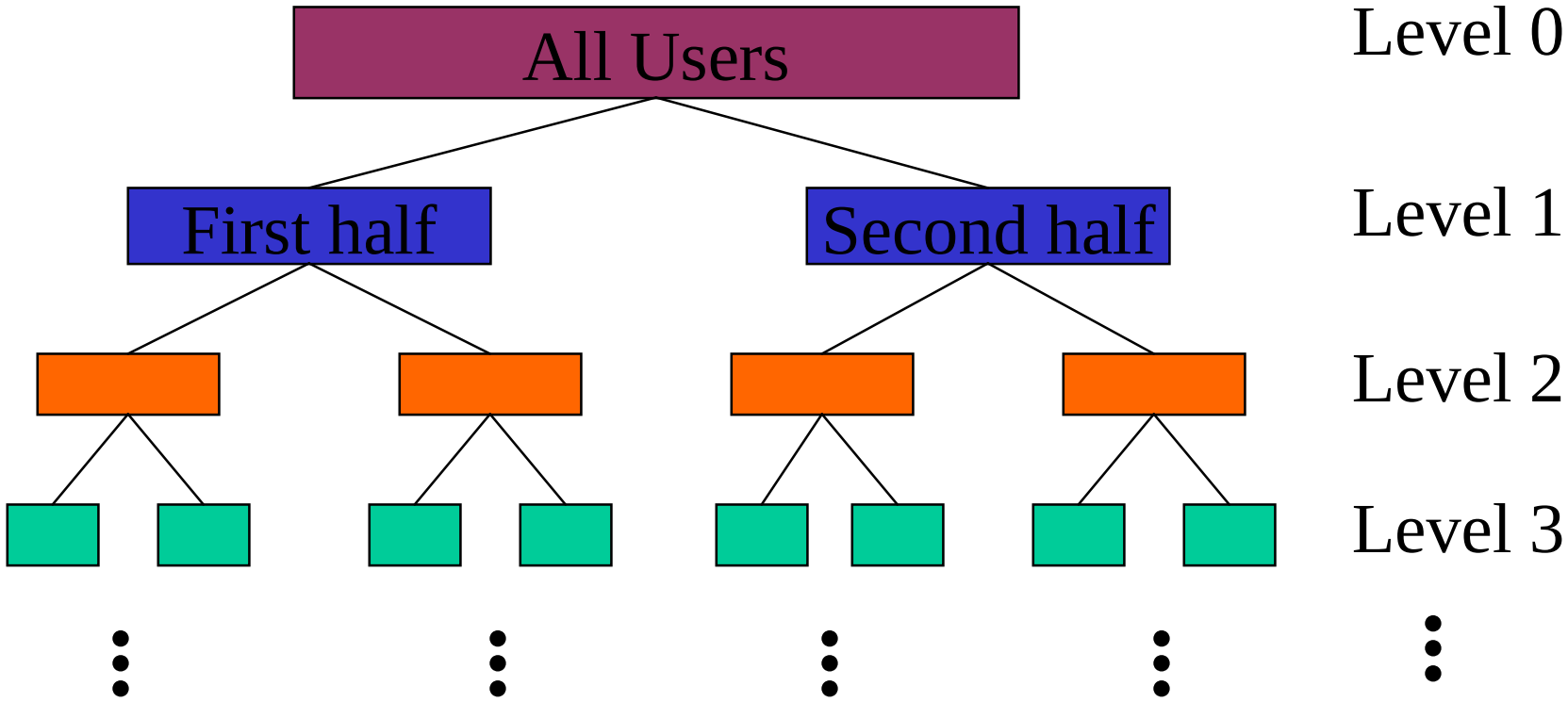
Parameters' definition



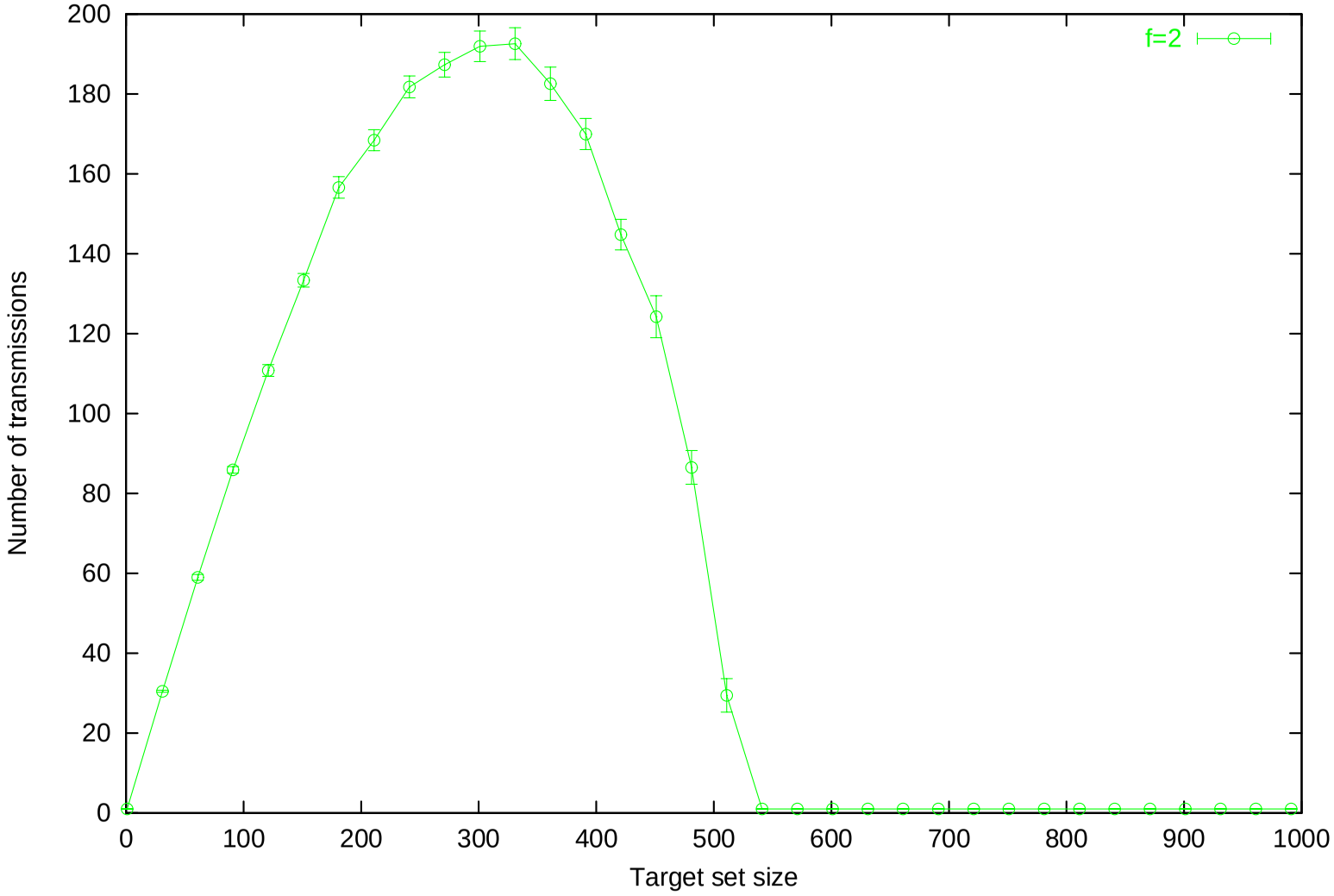
$$f_a = \frac{|S - K|}{|K|} \qquad \eta = \frac{|S - K|}{|N - K|}$$

Recipient set (S)
Target Set (K)

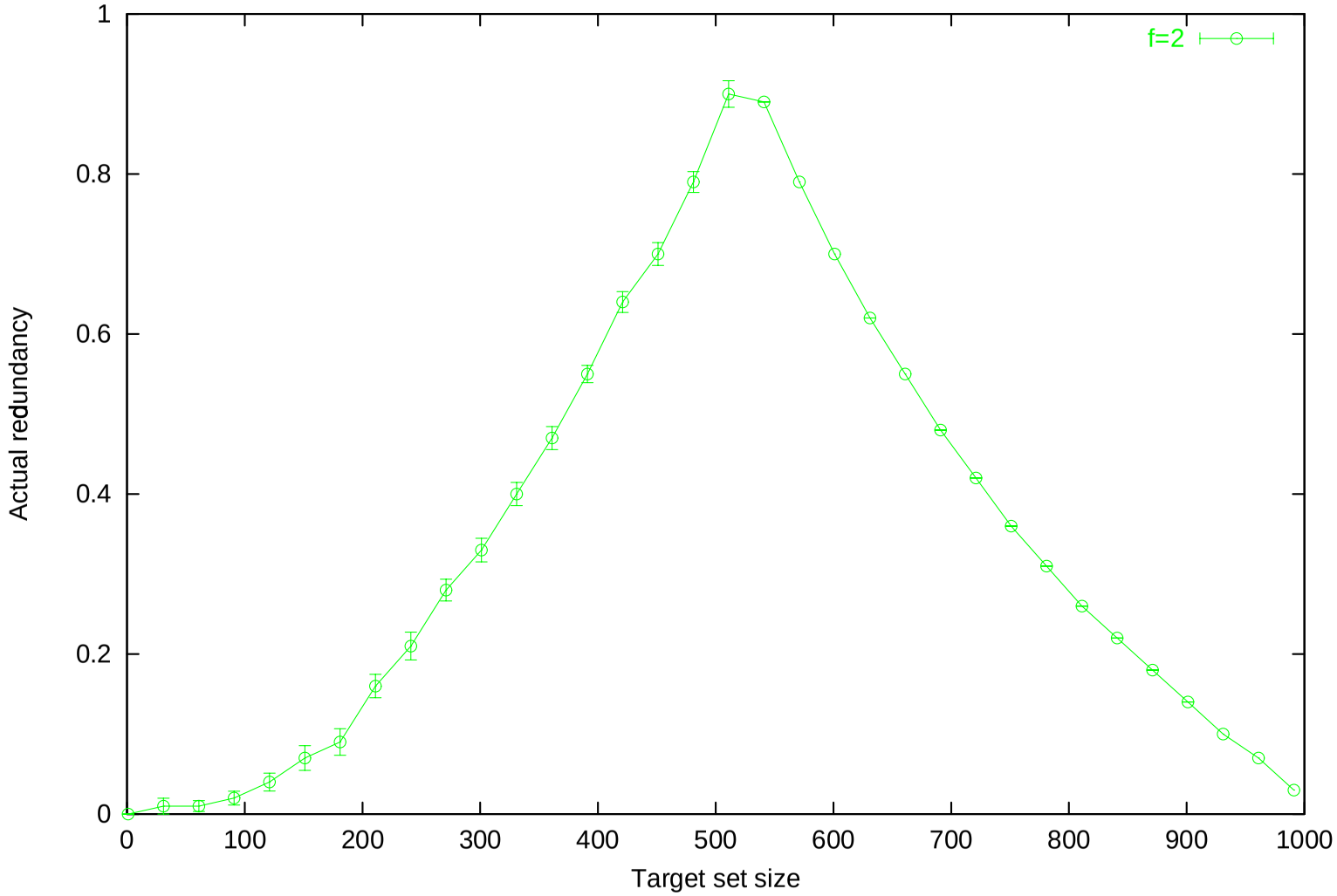
A tree-based scheme



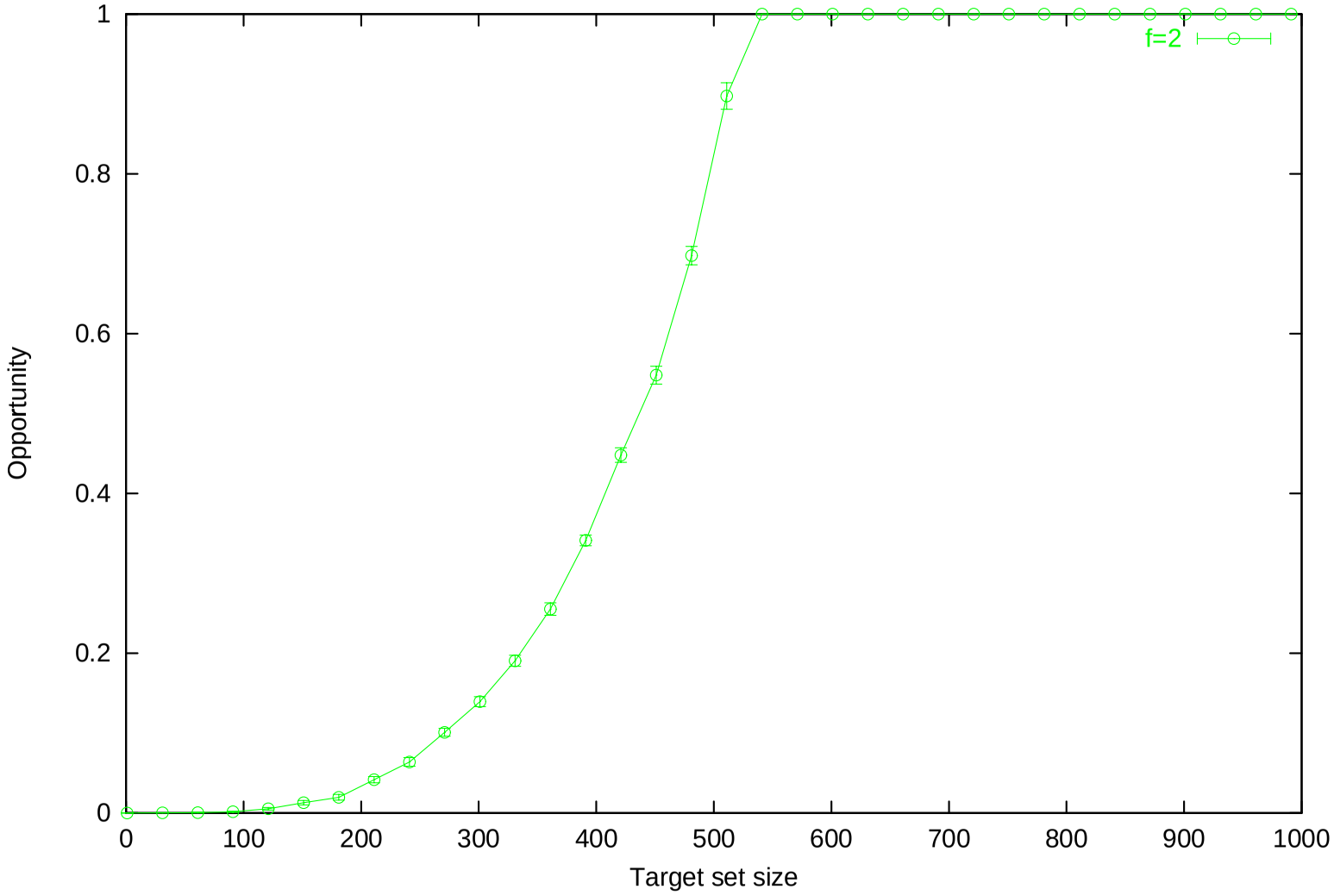
Number of transmissions, *t*



Actual redundancy, f_a

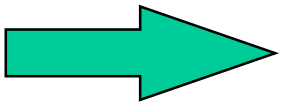


Opportunity, η



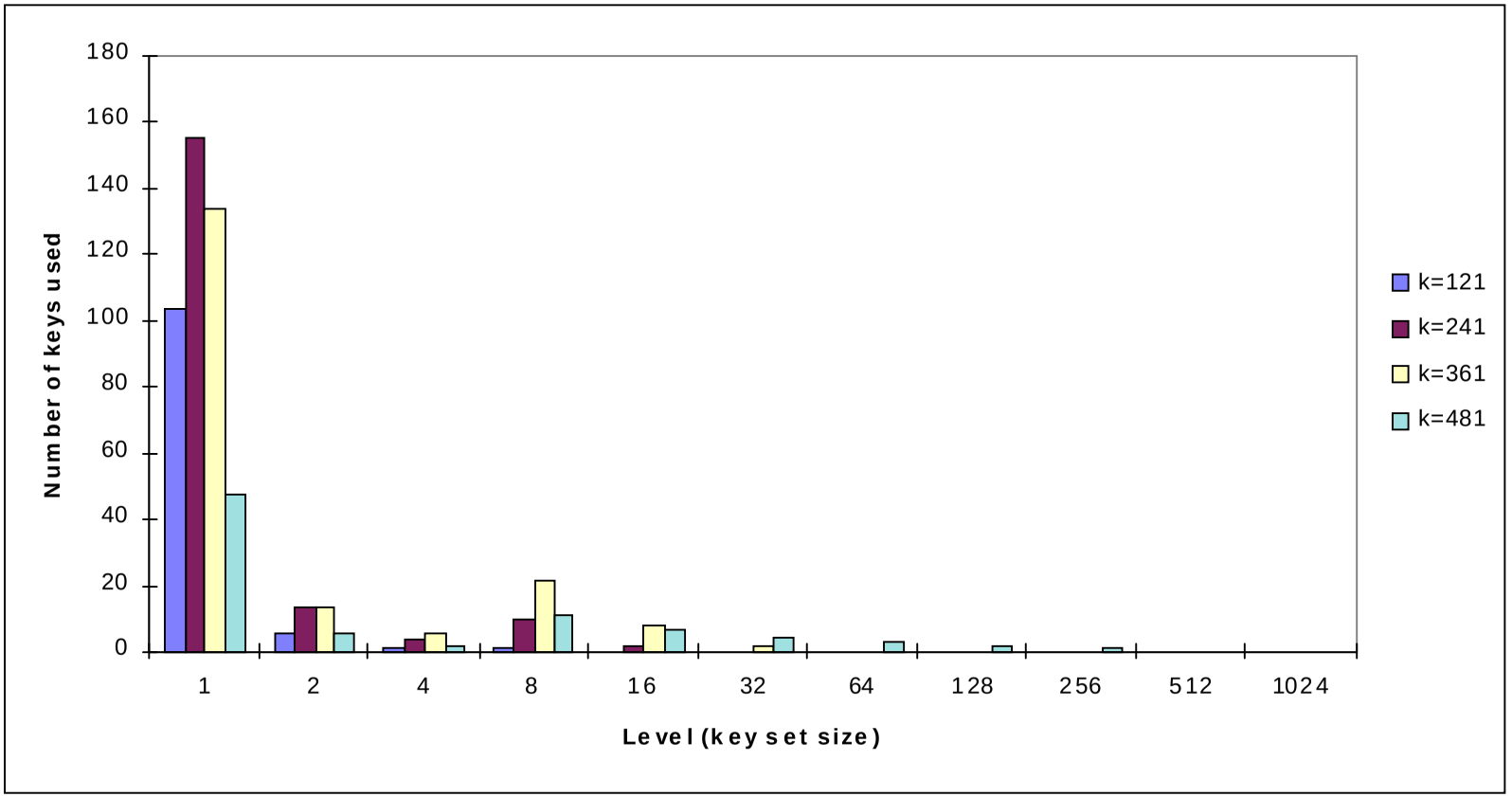
Augmenting the basic tree

- More than ***log(n)*** keys per user:
  - How should we distribute the extra keys?
  - In which levels are keys more effective?

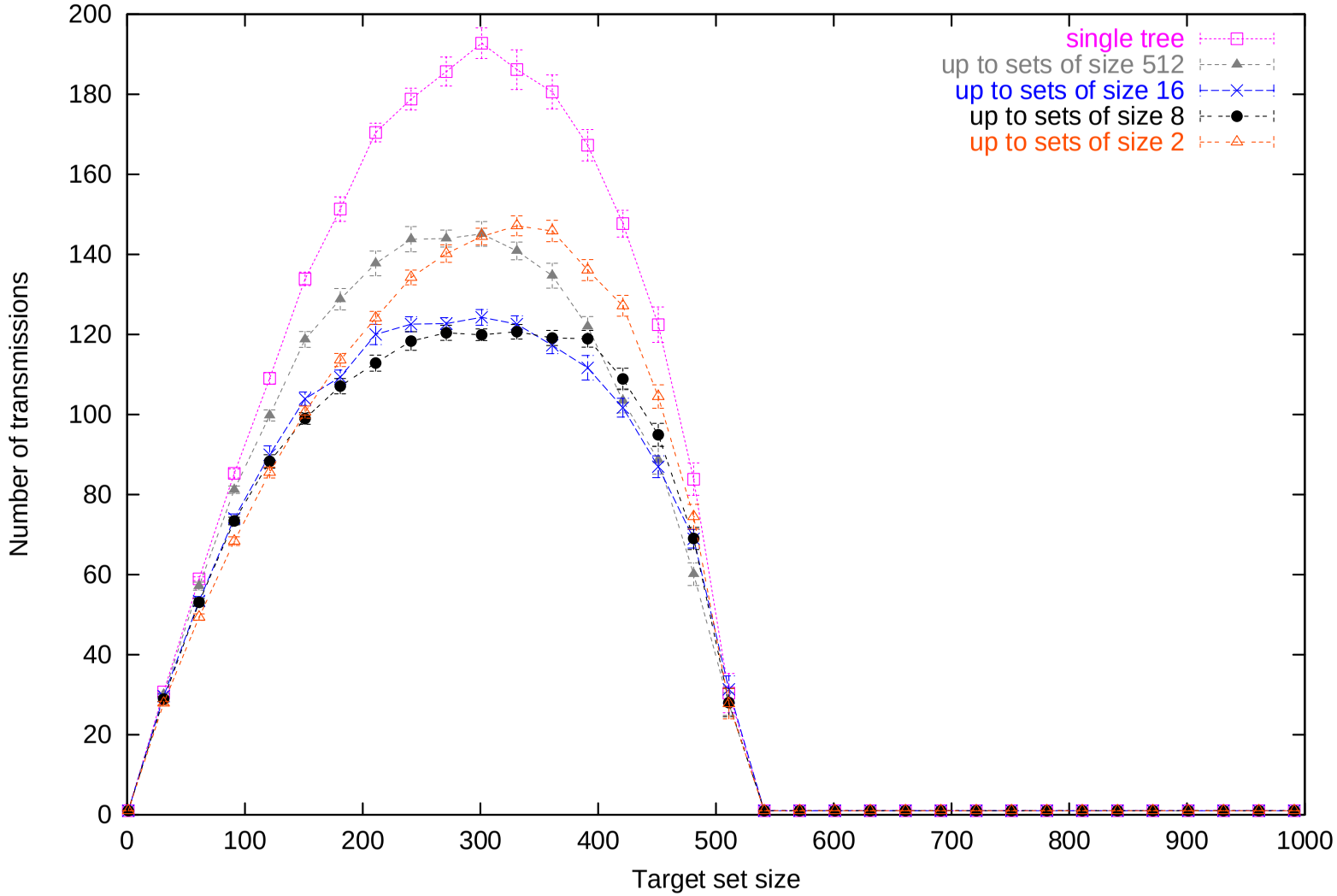


Histogram

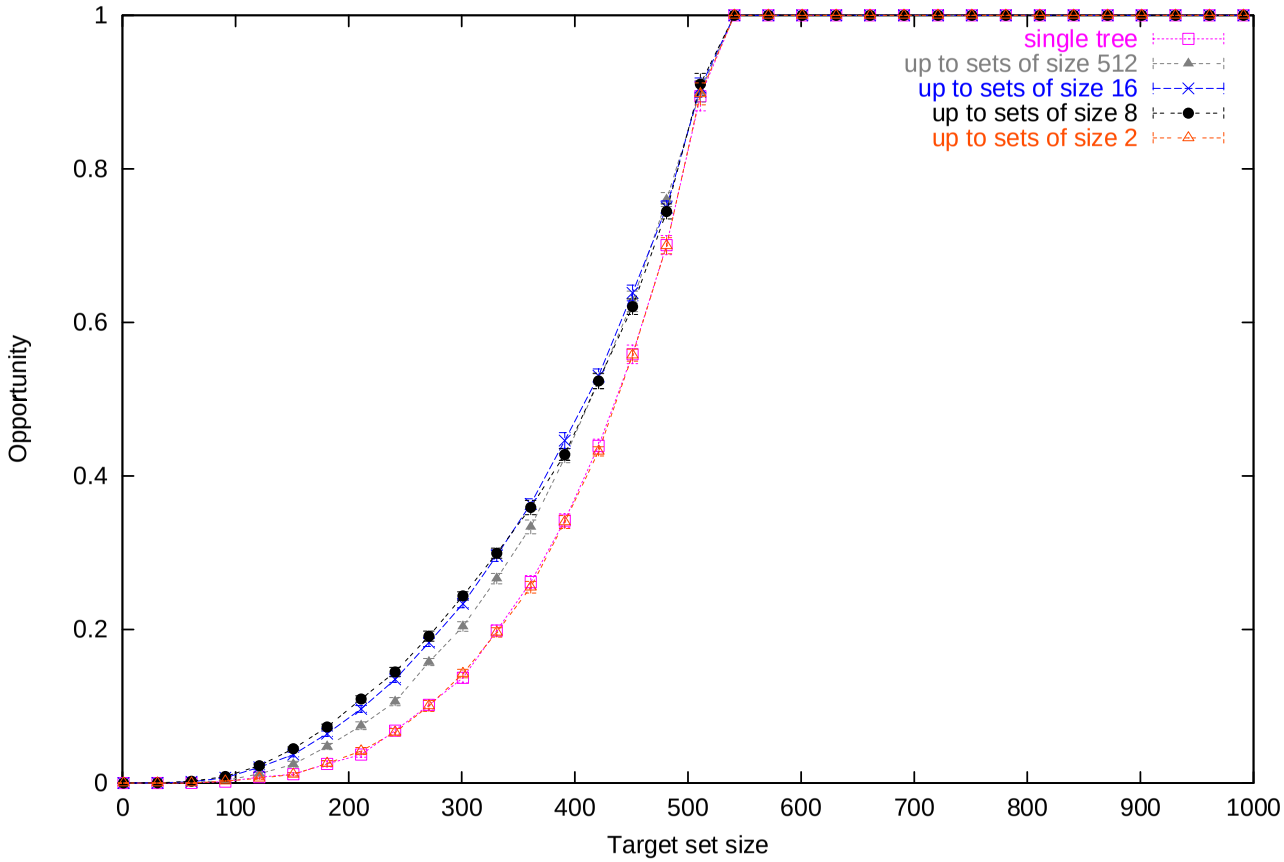
Histogram for the basic tree



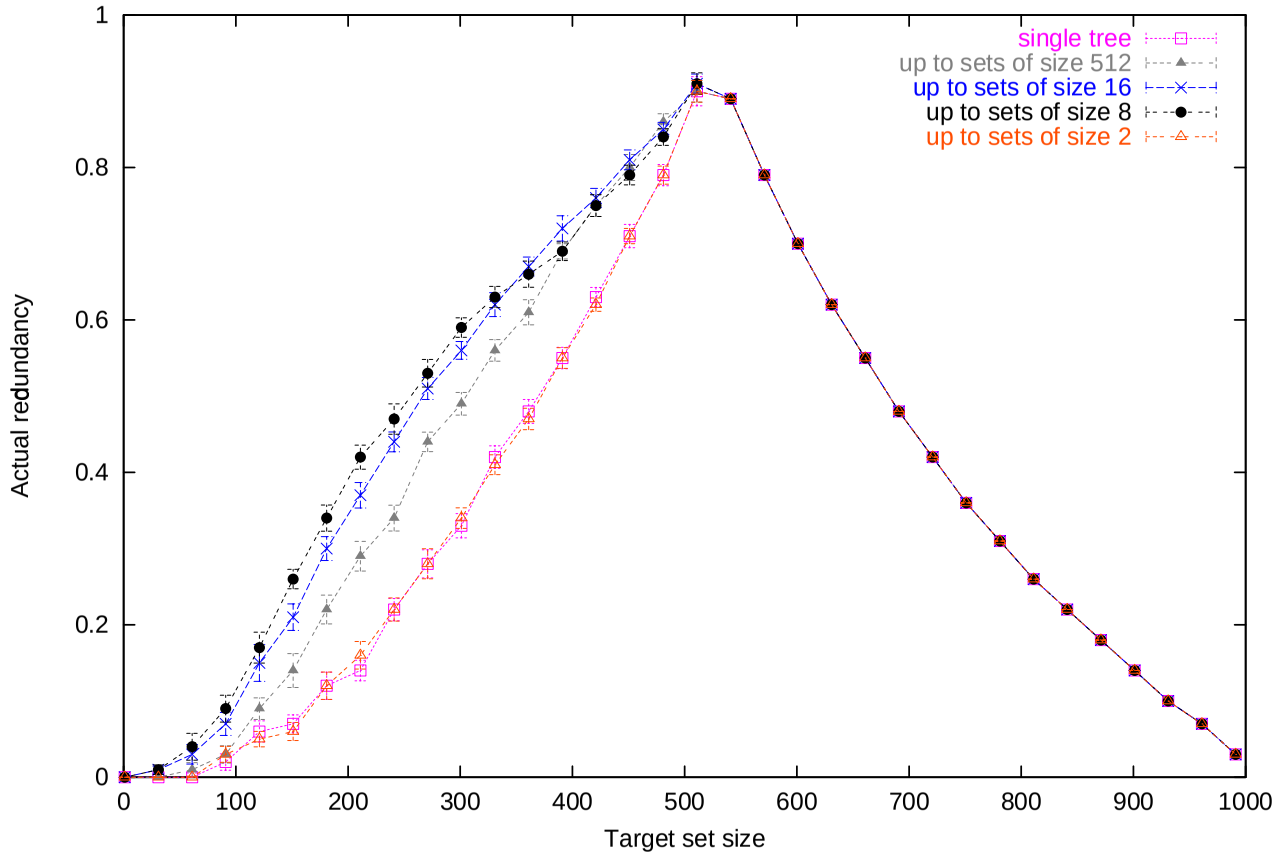
t, with *2log(n)* keys



η, with ***2log(n)*** keys

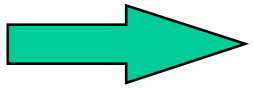


Basic tree augmented by 9 extra keys (actual redundancy)



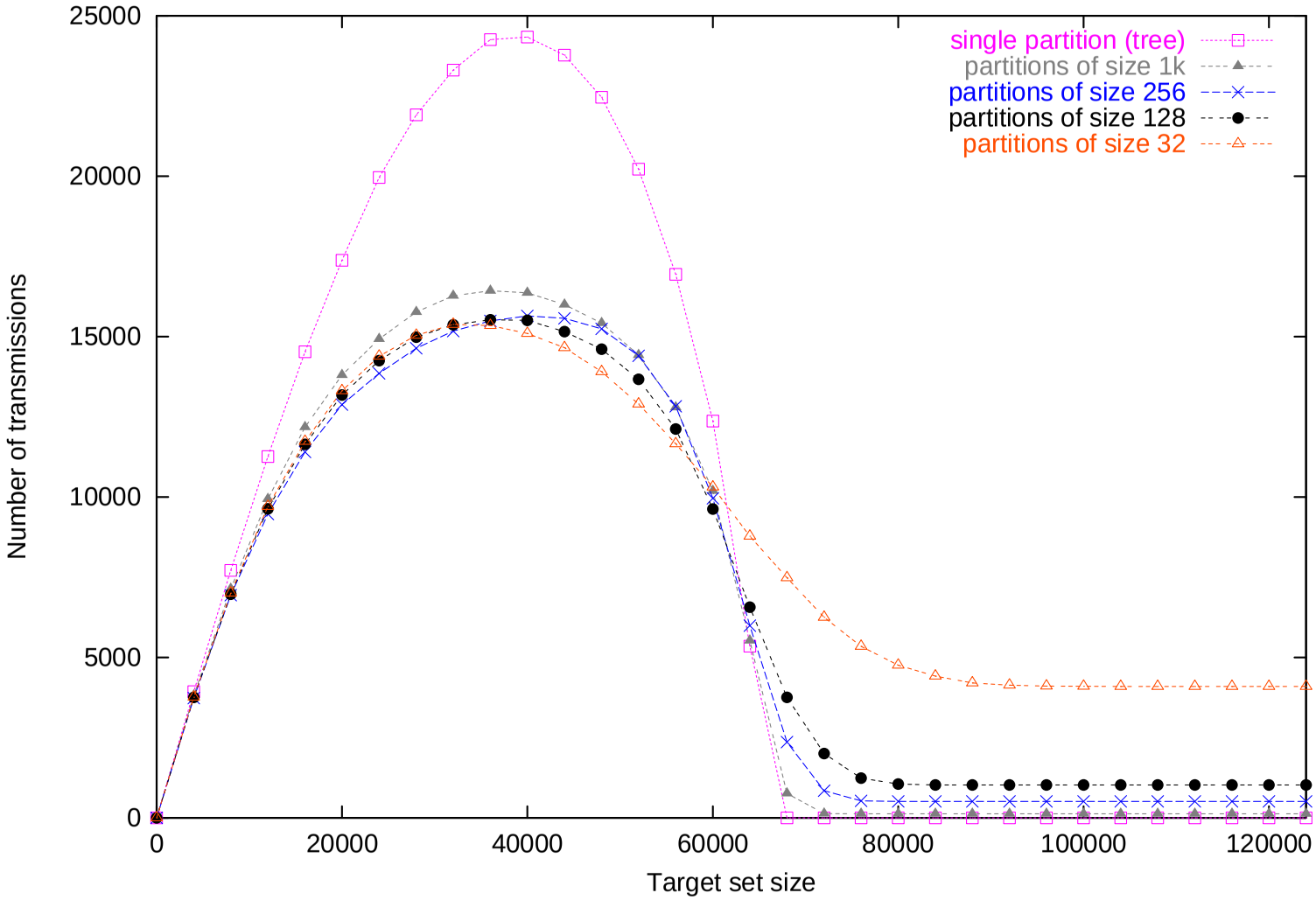
Partitioning

- Since high-level keys are not so effective, why don’t we get rid of them?

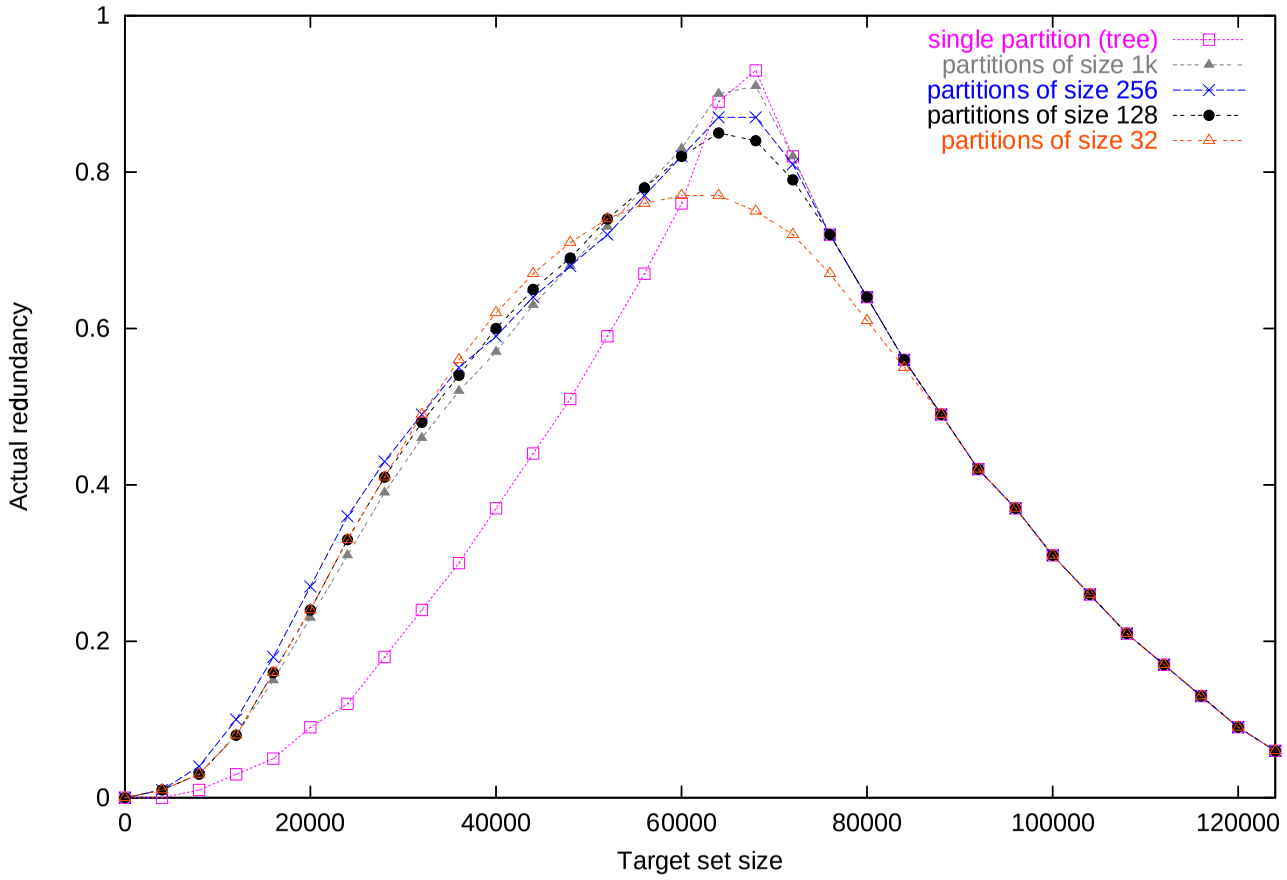
   Partitioning.

- Decreases the total number of keys
  - Allows to add more keys in lower levels.
- Scalability
  - Larger populations, easy to add users.

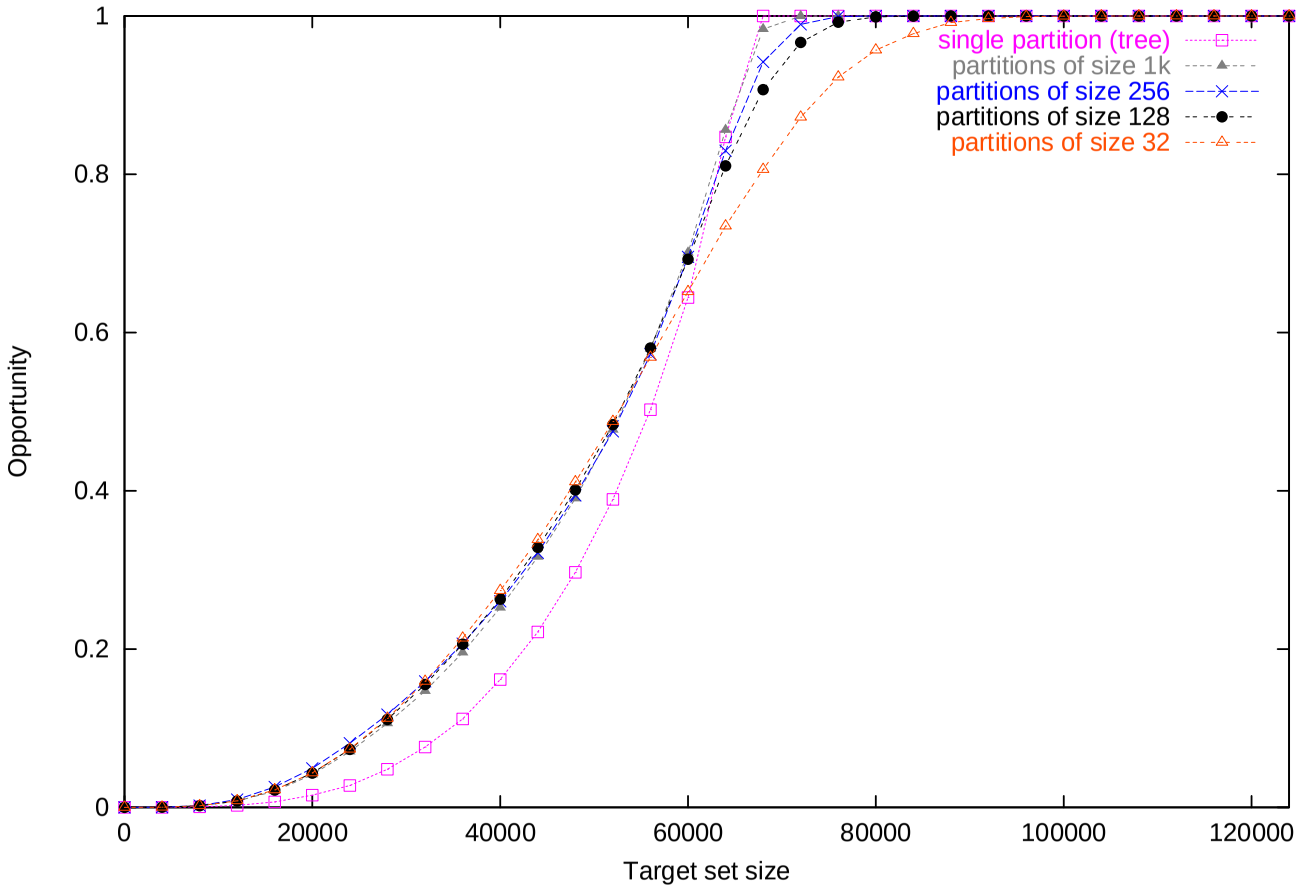
t, with partitioning, $n=2^{17}=128k$



Partitioning N=128k (redundancy)



Partitioning N=128k (opportunity)



Maintaining the global structure

- Prepare a new partition with virtual receivers
- A new receiver replaces a virtual receiver and it is assigned its group keys.
- When a receiver is deleted - it becomes a zombie.
- A partition with too many zombies is deleted and the non-zombies are assigned to virtual receivers in a new partition - amortize the rekeying cost among many delete operations.

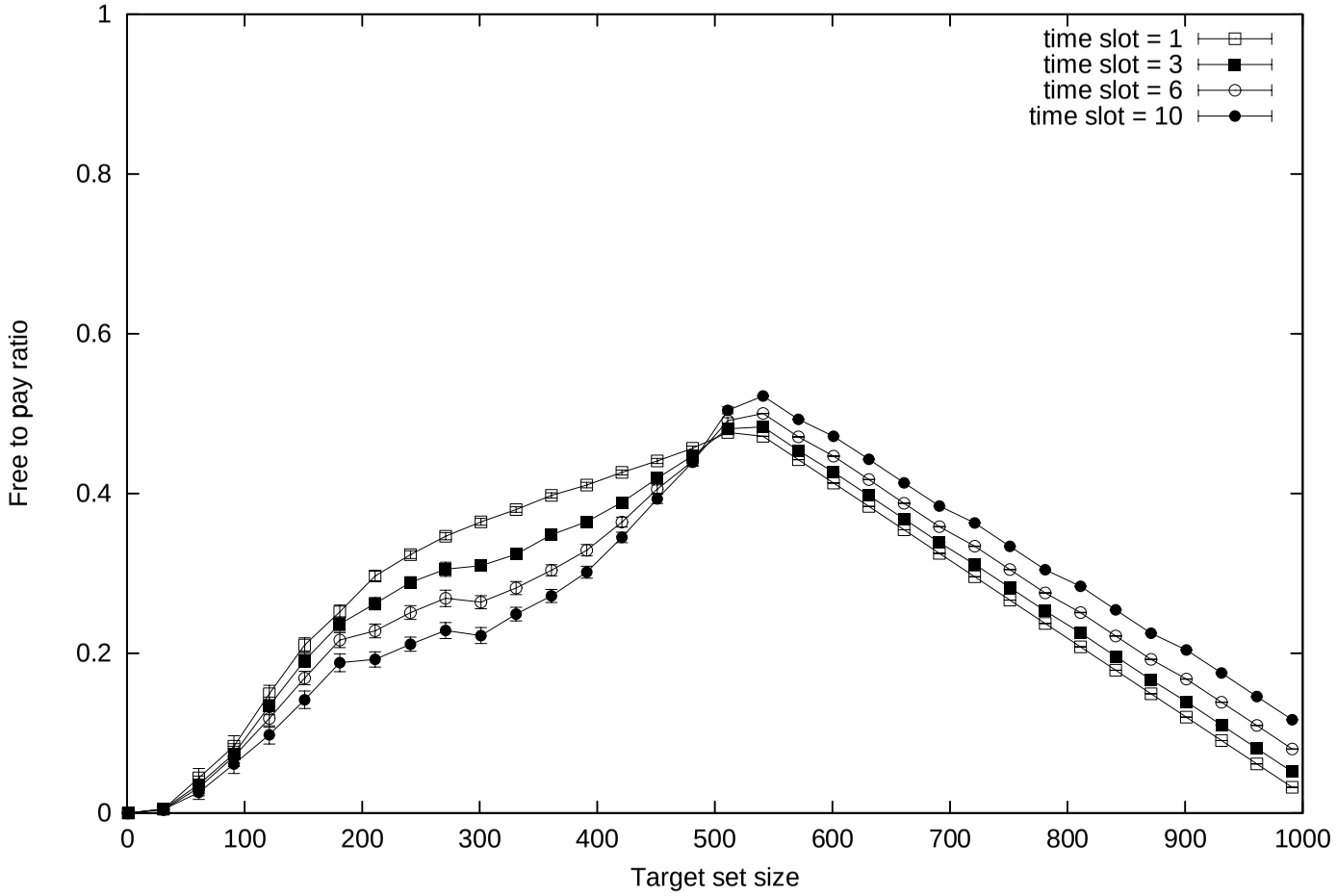
Dynamic group membership

- When users leave, the redundancy requirement may not be violated.
- A small number of joining users can receive the decryption key using their private key.
- Rekeying cost: less than *k*.

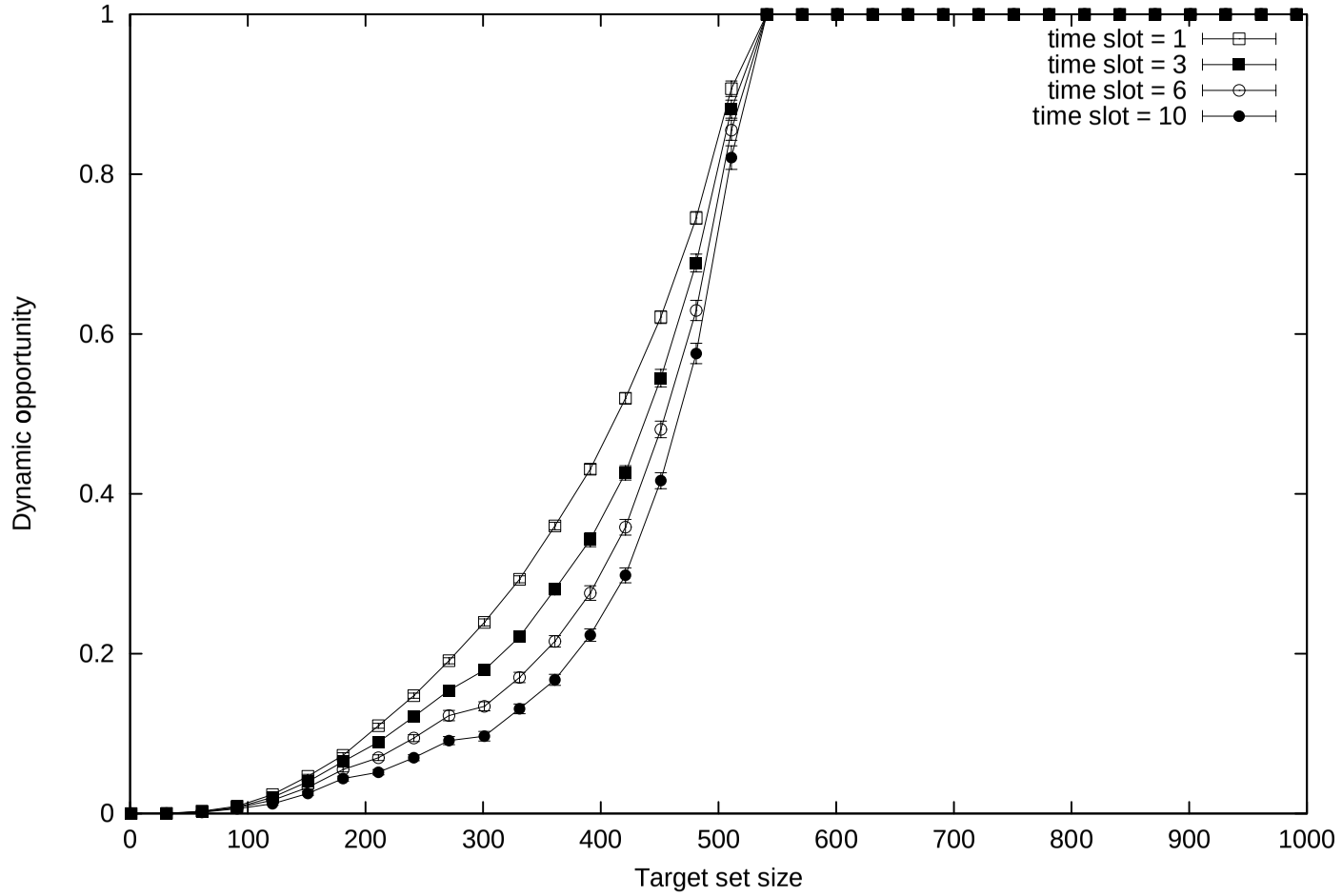
Experimenting with dynamic membership

- Time is slotted. Rekey at every slot.
- Target group size, k, is constant.
- τ = membership change rate for a slot.
- Performance measures:
  - $\rho(i)$ = free riders to paying customer ratio for i slots combined.
  - $\eta_d(i)$ = dynamic opportunity

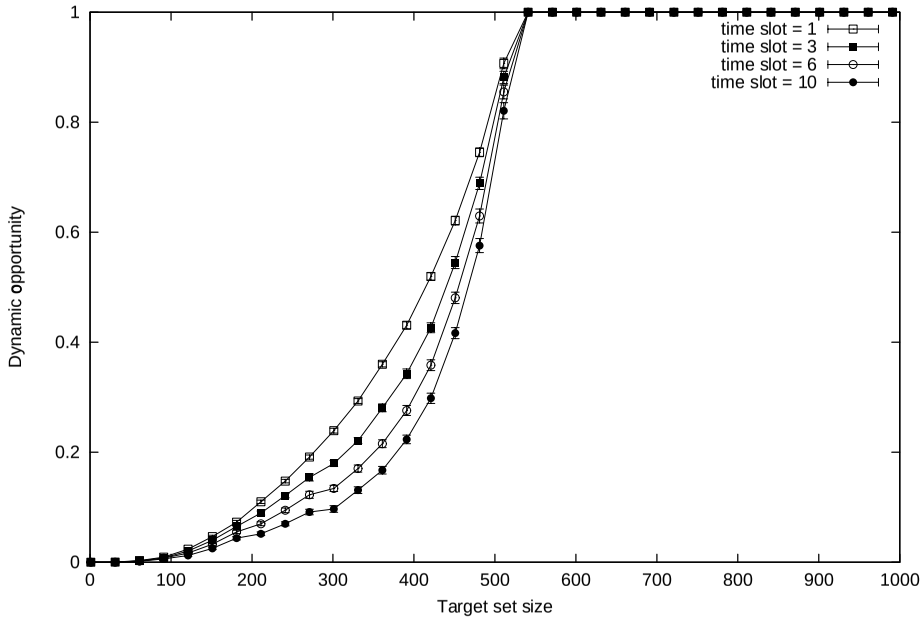
ρ(*i*) for *i*=1, 3, 6, 10, τ=0.01

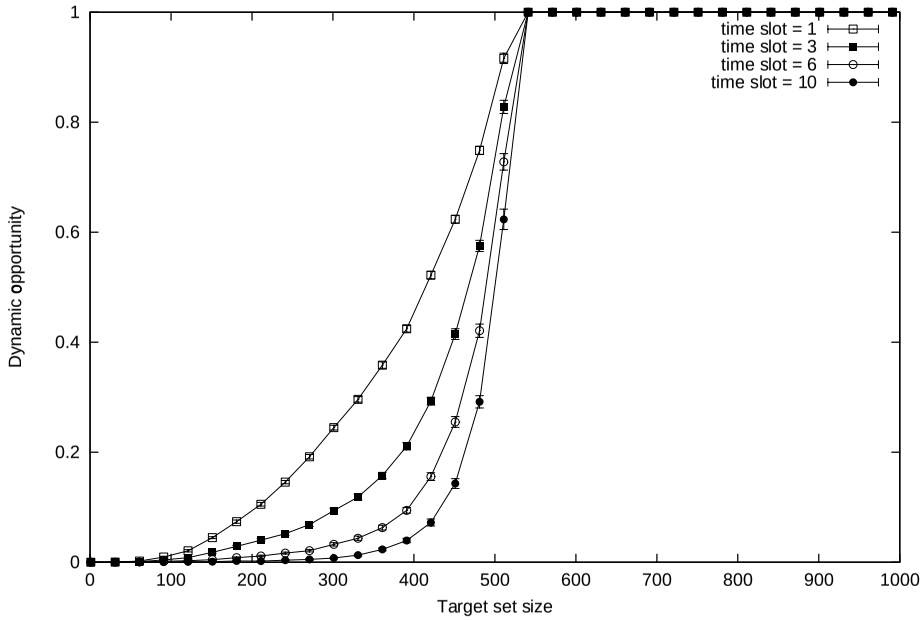


$\eta_d(i)$ for i=1, 3, 6, 10, τ=0.01



$\eta_d(i)$ for i=1, …, 10, τ=0.01, 0.1





The dynamic opportunity drops fast

Comparison with existing schemes

- enhanced global tree
- building the tree is amortized over all connection
- building a new group with k members: $< k$
- cost of Δk joins: at most Δk
- cost of Δk leaves: ak, $a<1$

- tree per program
- no prior investment
- building a new group with k members: $k \log k$
- cost of Δk joins: $\Delta k \log k$
- cost of Δk leaves: $\Delta k \log k$

Conclusions

- The enhanced tree is efficient and practical.
- Partitions are effective for large populations.
- Partitions of size $\sqrt{n}$ usually offer a good tradeoff.
- Dynamic membership extinguishes the hope for a free ride.