

# **AVMEM**: Availability-Aware Overlays for Management Operations in Non-cooperative Distributed Systems

---

**Ramsés Morales**, Brian Cho and Indranil Gupta

---

Dept. of Computer Science  
University of Illinois at Urbana-Champaign



# Outline

- Introduction
  - New operations
- Design Goals
- **AVMEM**

# Introduction

- ▶ New operations for distributed management applications
  - P2P
  - Grid
  - PlanetLab
- ▶ Availability-based operations
  - Multicast
  - Anycast
  - **Availability: fraction of time a node is online**

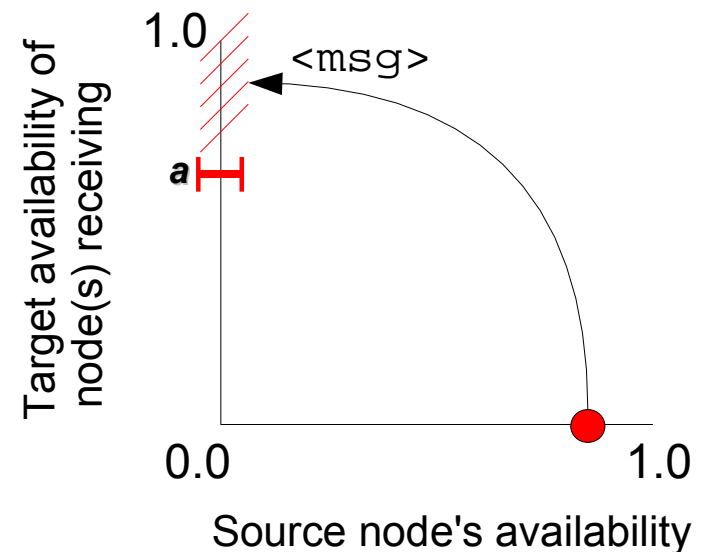
# Motivation: Availability-based Multicast & Anycast

## ▶ Threshold-multicast, threshold-anycast

- to nodes with availability  $> a$ ,  $a \in [0, 1)$

## ▶ Useful for

- Control operations
  - ▶ “Distinguished” peer selection
    - [Liang 2006; Lo 2005; Min 2006]
- Data operations
  - ▶ Multicast reliability  $\propto$  availability
    - [Pongthawornkamol 2006]



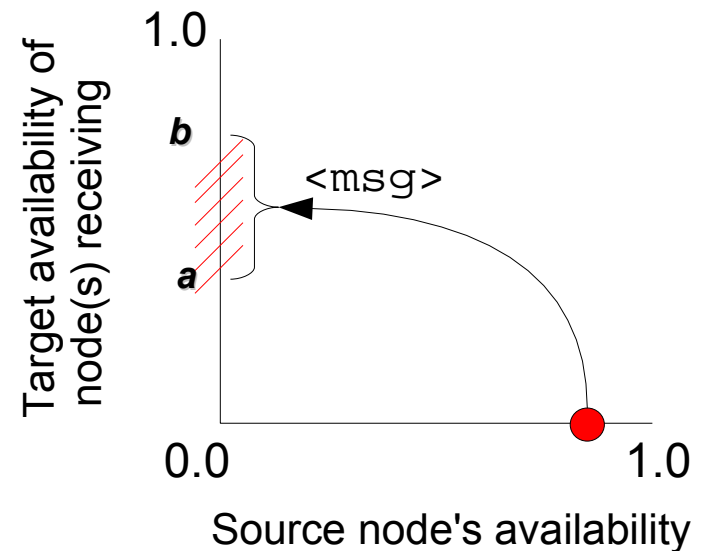
# Motivation: Availability-based Multicast & Anycast

## ▶ Range-multicast, range-anycast

- to nodes with availability in range  $[a, b] \subseteq [0, 1]$

## ▶ Useful for

- Node characteristic fingerprint
- Replica placement
  - ▶ [Bhagwan 2004; Chun 2006]
- Grid instance placement
  - ▶ [Cappello 2005]



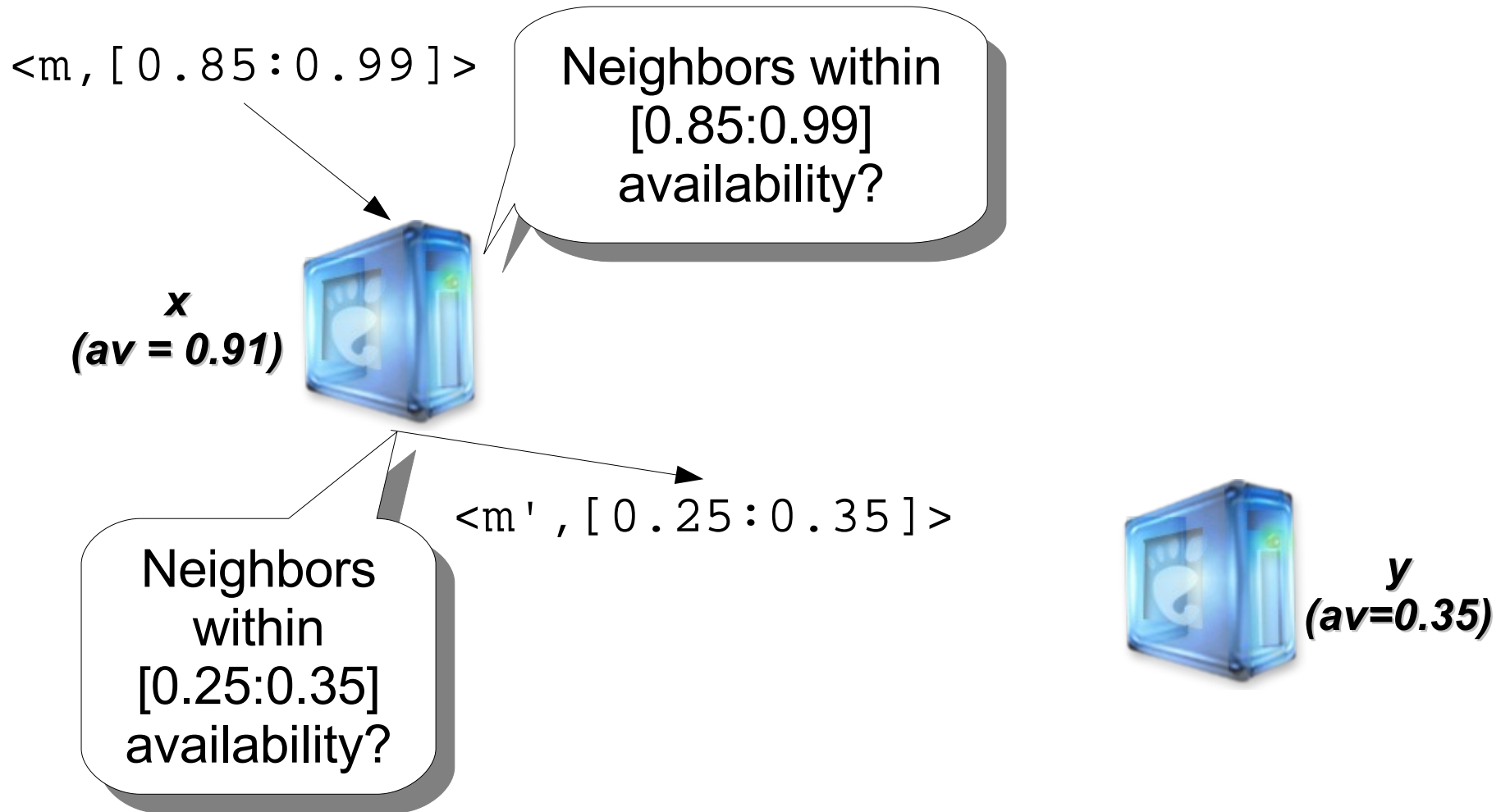
# Outline

- Introduction
  - Issues
- Design Goals
- **AVMEM**

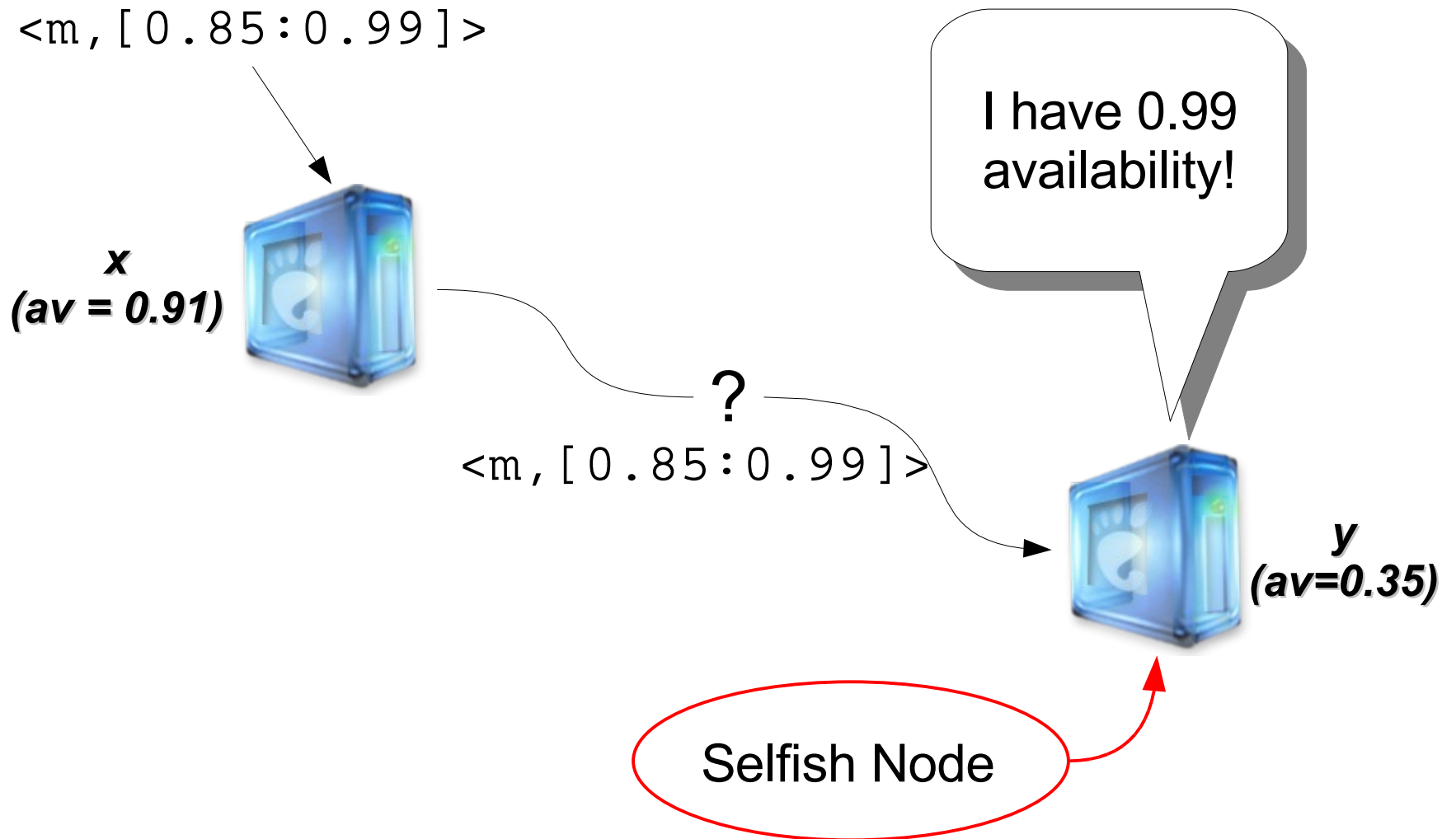
# Introduction: Issues

- ▶ Difficult to predict **availability variations** across nodes and time
  - **Availability**: fraction of time a node is online

# Introduction: Issues



# Introduction: Issues



# Outline

- Introduction
  - Possible solutions
- Design Goals
- **AVMEM**

# Possible Solutions

- ▶ Centralized
- ▶ DHT-based [Rowstron 2001; Stoica 2001]
  - ***nodeID = f(av(x))***
- ▶ Overlays for range-queries
  - Skip-trees, Voronoi, Segment trees, others
    - ▶ [Aspnes 2003; Harvey 2003; Shu 2005; Sheng 2006]

# Outline

- Introduction
- Design Goals
- **AVMEM**

# Design Goals

- ▶ **Availability-based operations** for management operations
- ▶ Overlay where **neighbor selection** is based on **node availability**

# Design Goals

## ▶ $M(x, y) \in \{0, 1\}$

- Depends on:
  - ▶  $av(x), av(y)$
  - ▶  $y$ 's `<ip, port>`
  - ▶  $x$ 's `<ip, port>`

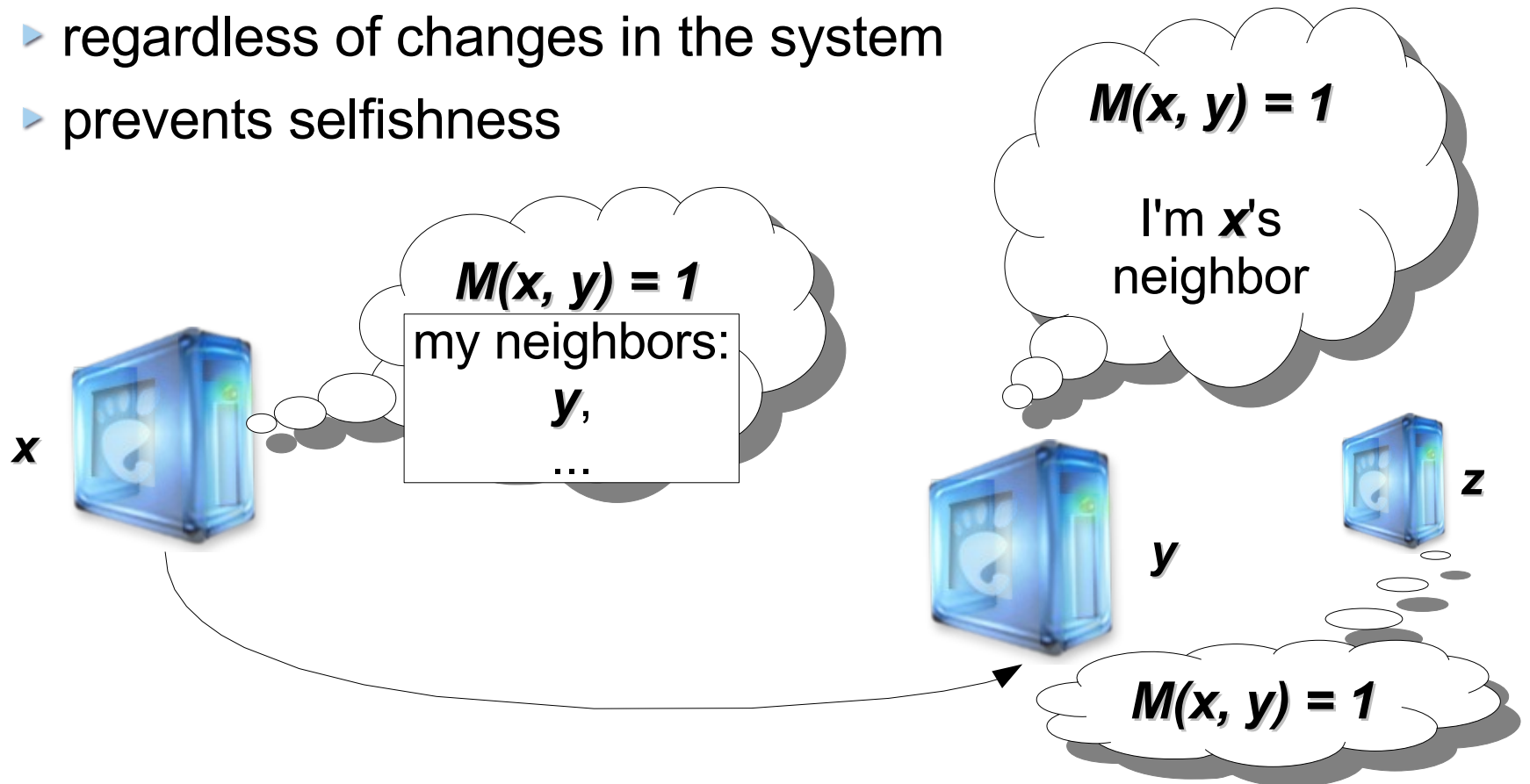


# Design Goals

## ▶ $M(x, y) \in \{0, 1\}$

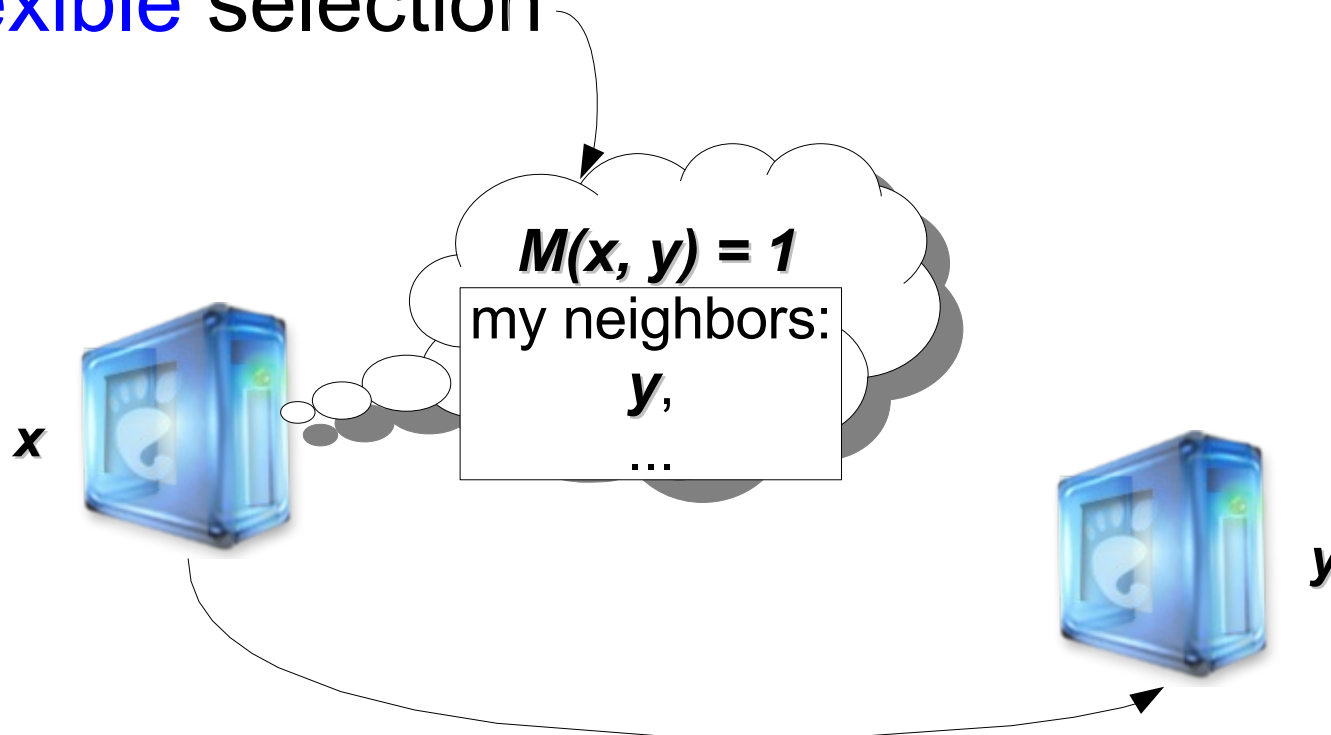
- Consistent

- ▶ regardless of changes in the system
- ▶ prevents selfishness



# Design Goals

- ▶ Small number of neighbors
- ▶ Randomized neighbors
- ▶ Flexible selection

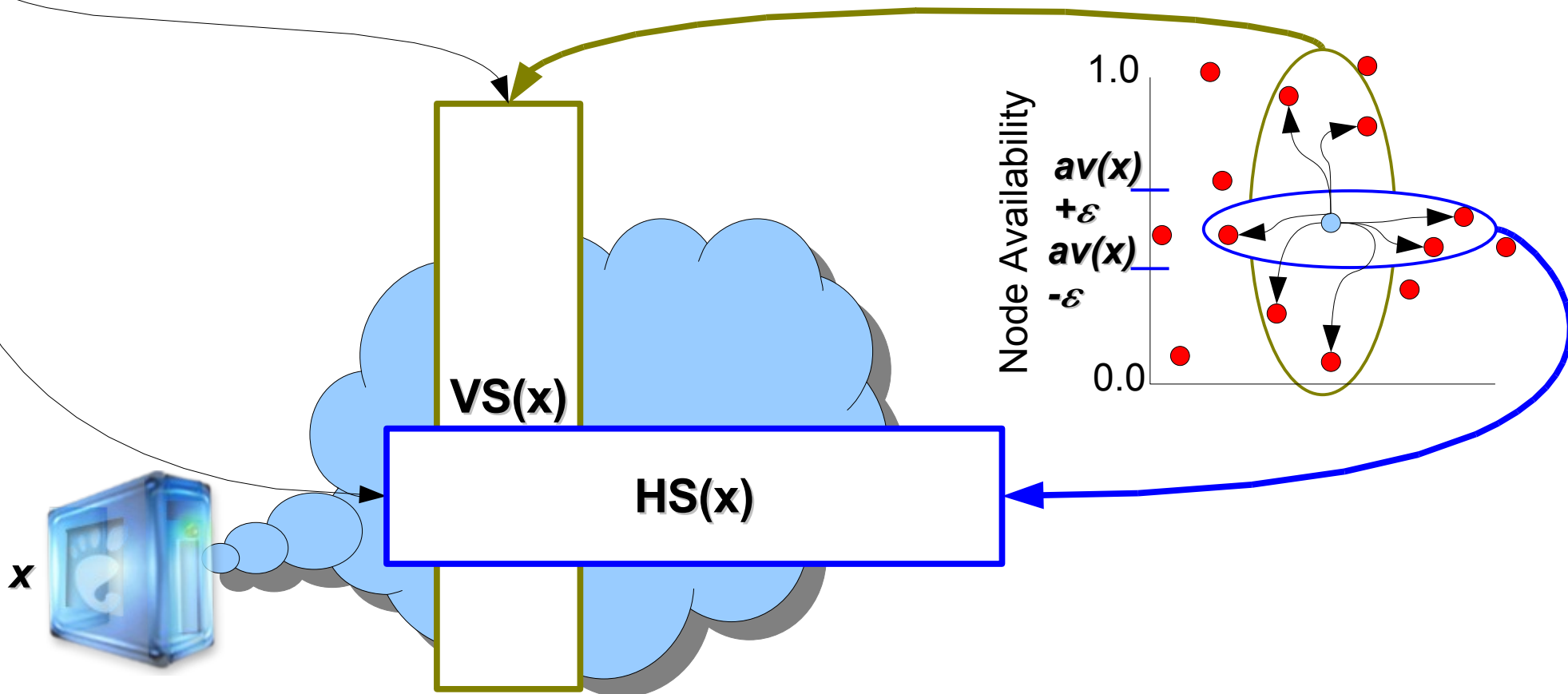


# Outline

- Introduction
- Design Goals
- **AVMEM**
  - Membership graph predicates

# AVMEM: Membership Graph Predicates

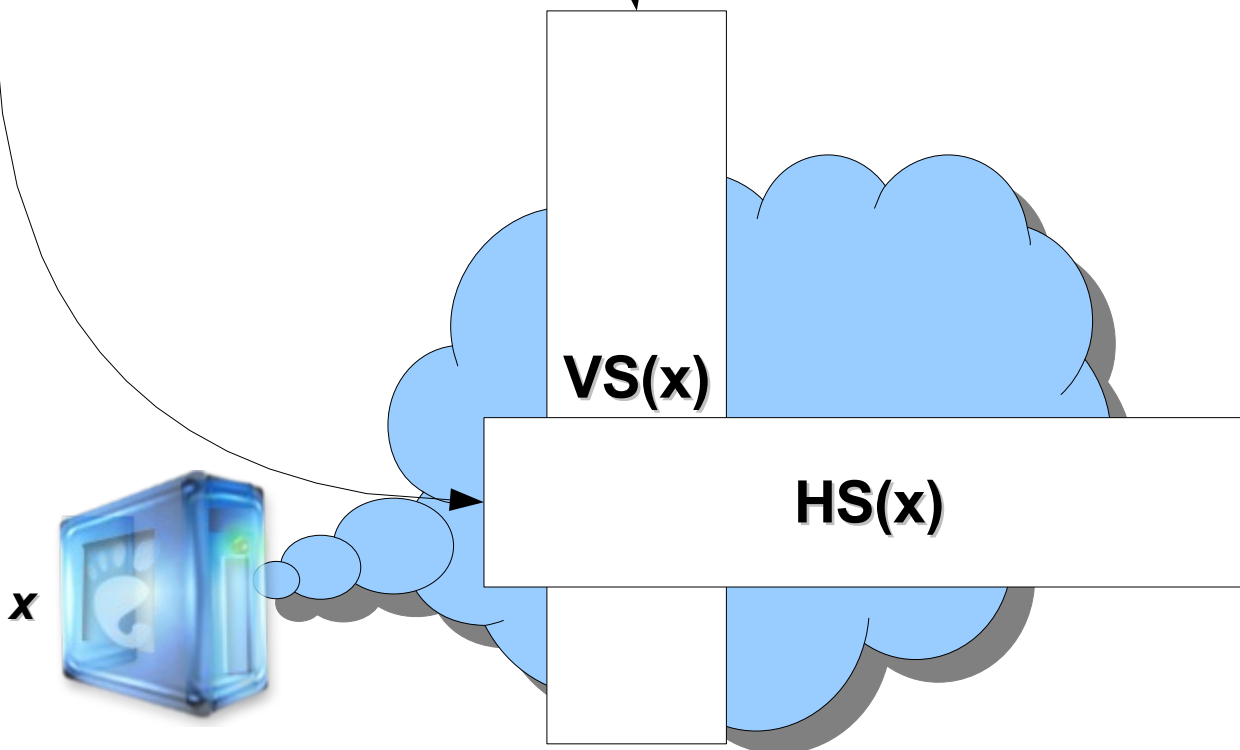
- ▶ Some nodes whose av. in  $[av(x) - \varepsilon, av(x) + \varepsilon]$
- ▶ Some other nodes



# **AVMEM**: Membership Graph Predicates

$$M(x, y) \equiv \{ H(id(x), id(y)) \leq f(av(x), av(y)) \}$$

$$M(x, y) = 1 \text{ iff } x \rightarrow y$$

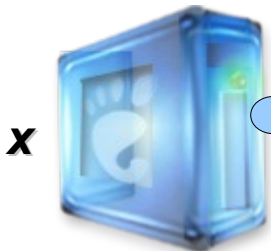


# AVMEM: Membership Graph Predicates

$$M(x, y) \equiv \{ H(id(x), id(y)) \leq f(av(x), av(y)) \}$$

System-wide pred.

MD5, SHA1, etc.,  
normalized to [0, 1]



x

VS(x)

HS(x)

# Outline

- Introduction
- Design Goals
- **AVMEM**
  - Family of availability predicates

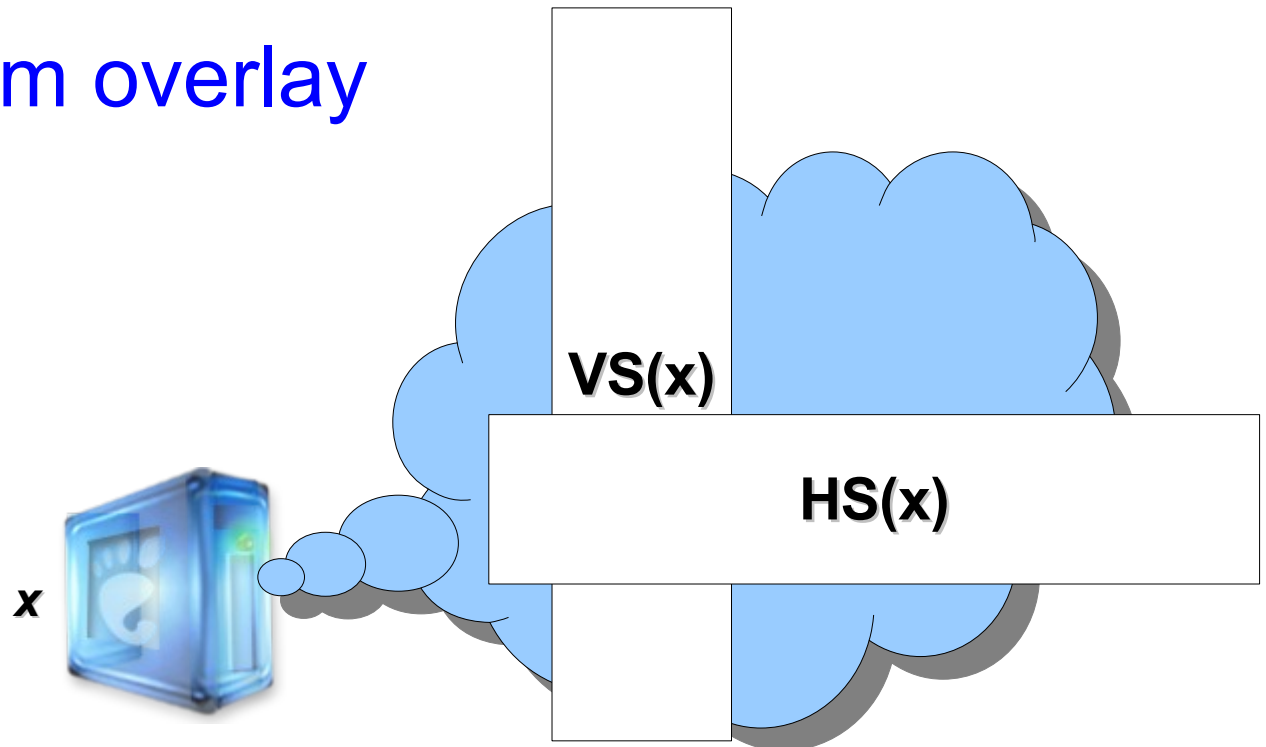
# **AVMEM**: Family of Availability-Aware Predicates

- ▶ Vertical sub-predicate:
  - Constant vertical sliver
  - Logarithmic vertical sliver
  - Logarithmic-decreasing vertical sliver
- ▶ Horizontal sub-predicate:
  - Constant horizontal sliver
  - Logarithmic-constant horizontal sliver

# **AVMEM**: Constant Vertical/Horizontal Sliver

$$d_1 = O\left(\frac{\log(N^*)}{N^*}\right)$$

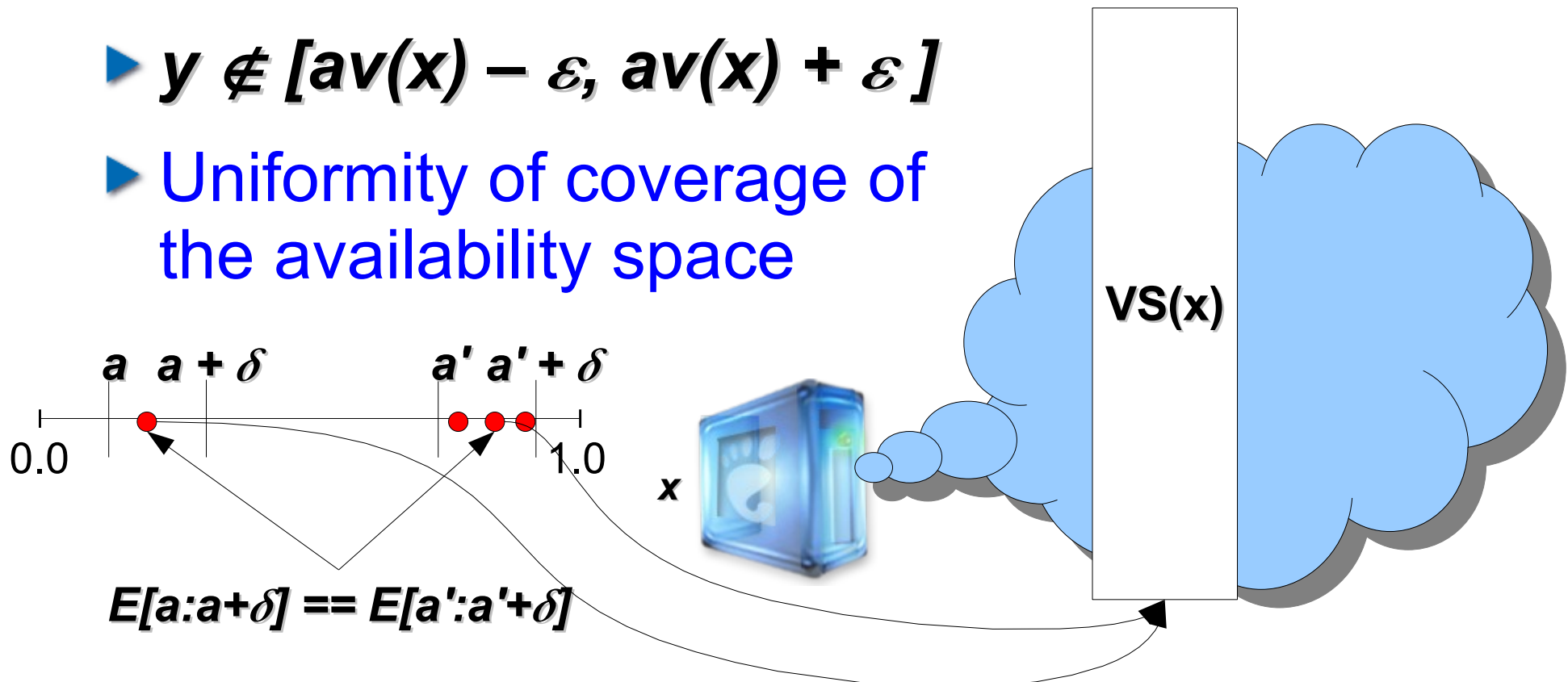
- ▶ Uniformly random overlay



# AVMEM: Logarithmic Vertical Sliver

$$\min\left(\frac{c_1 \cdot \log(N^*)}{N^* \cdot p(av(y))}, 1.0\right)$$

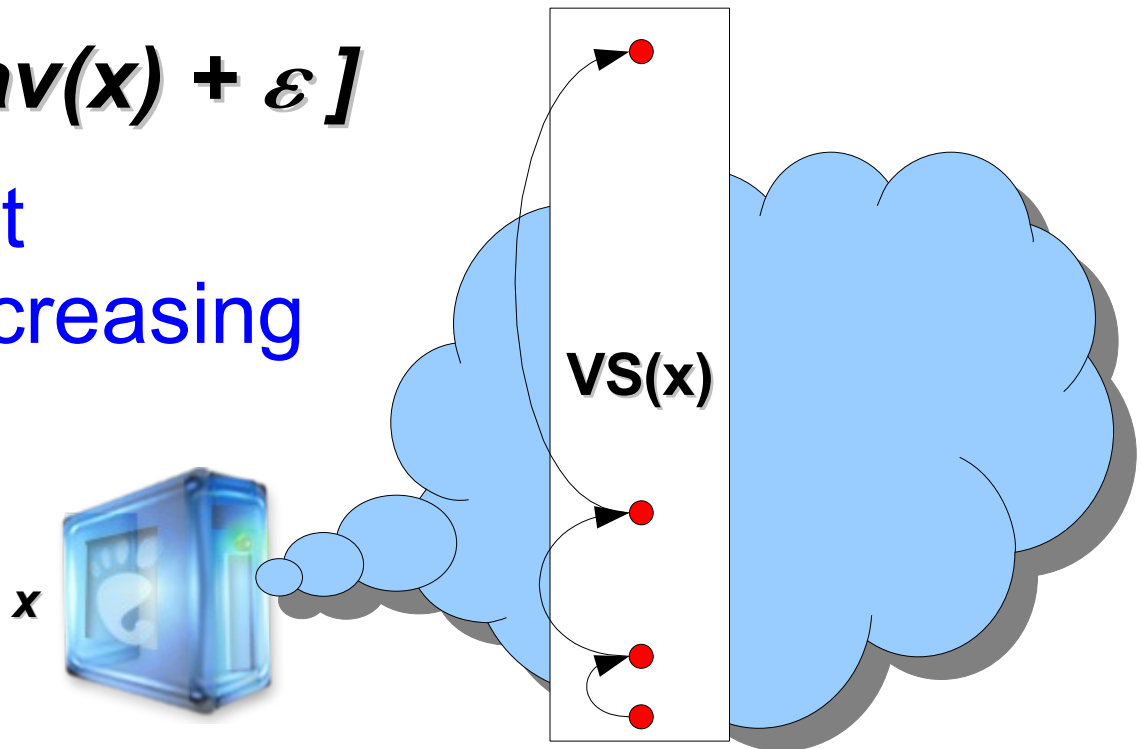
- ▶  $y \notin [av(x) - \varepsilon, av(x) + \varepsilon]$
- ▶ Uniformity of coverage of the availability space



# AVMEM: Logarithmic-decreasing Vertical Sliver

$$\min\left(\frac{c_1 \cdot \log(N^*)}{N^* \cdot p(av(y)) \cdot |av(y) - av(x)|}, 1.0\right)$$

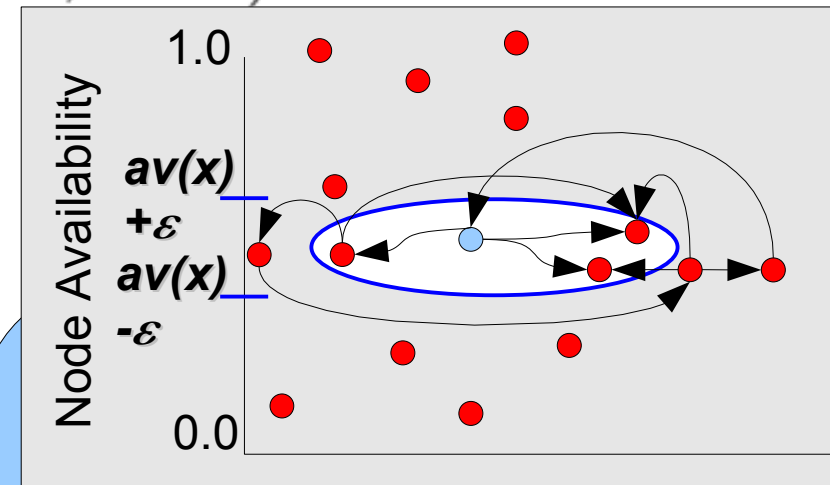
- ▶  $y \notin [av(x) - \varepsilon, av(x) + \varepsilon]$
- ▶ Neighbors are at exponentially increasing distance from  $\mathbf{x}$



# AVMEM: Logarithmic Horizontal Sliver

$$\min\left(\frac{c_2 \cdot \log(N_{av(x)}^*)}{N_{av(x)}^{*min}}, 1.0\right)$$

- ▶  $y \in [av(x) - \varepsilon, av(x) + \varepsilon]$
- ▶ Sub-overlay of nodes in  $[av(x) - \varepsilon, av(x) + \varepsilon]$  is connected w.h.p.

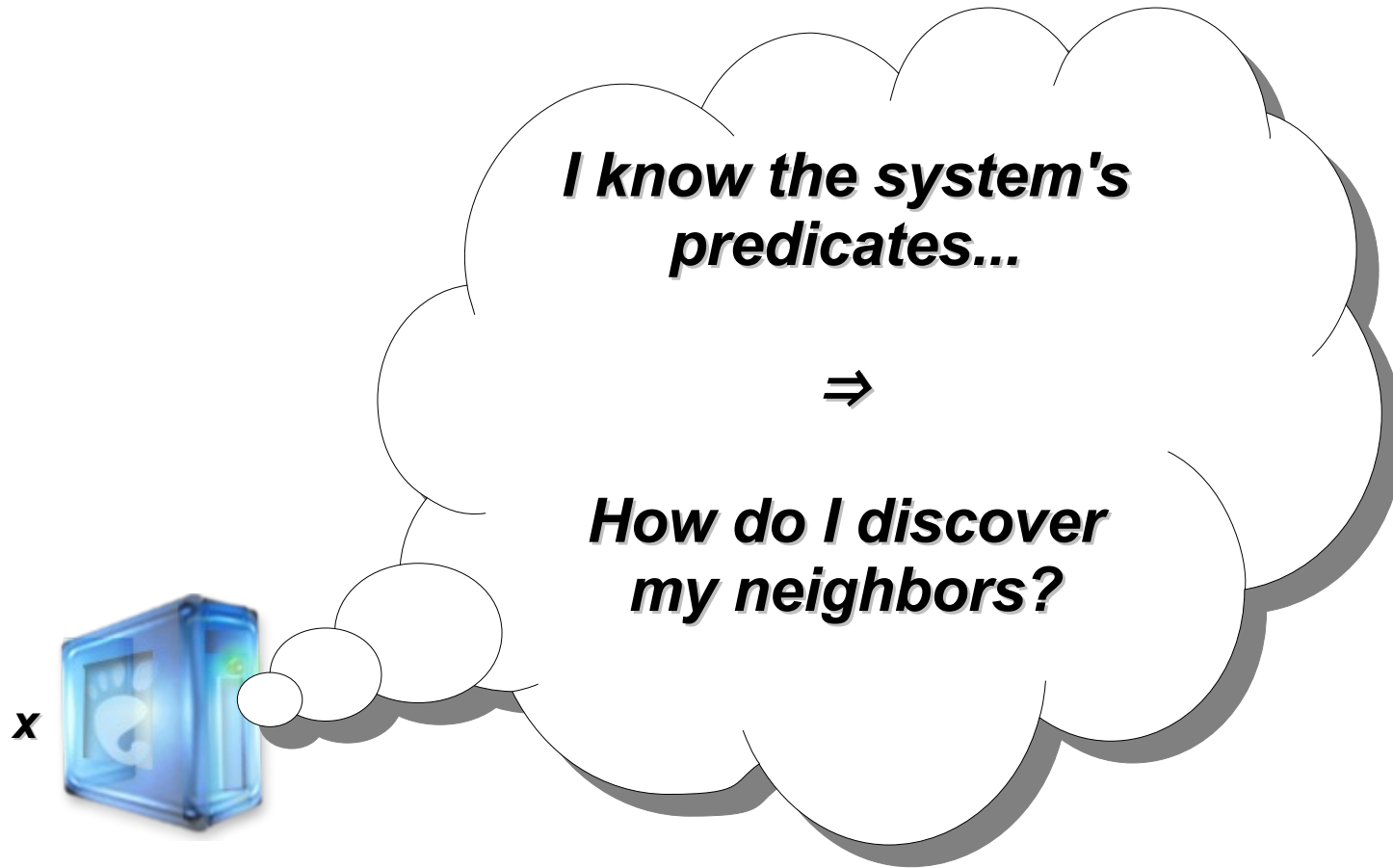


HS(x)

# Outline

- Introduction
- Design Goals
- **AVMEM**
  - Membership maintenance

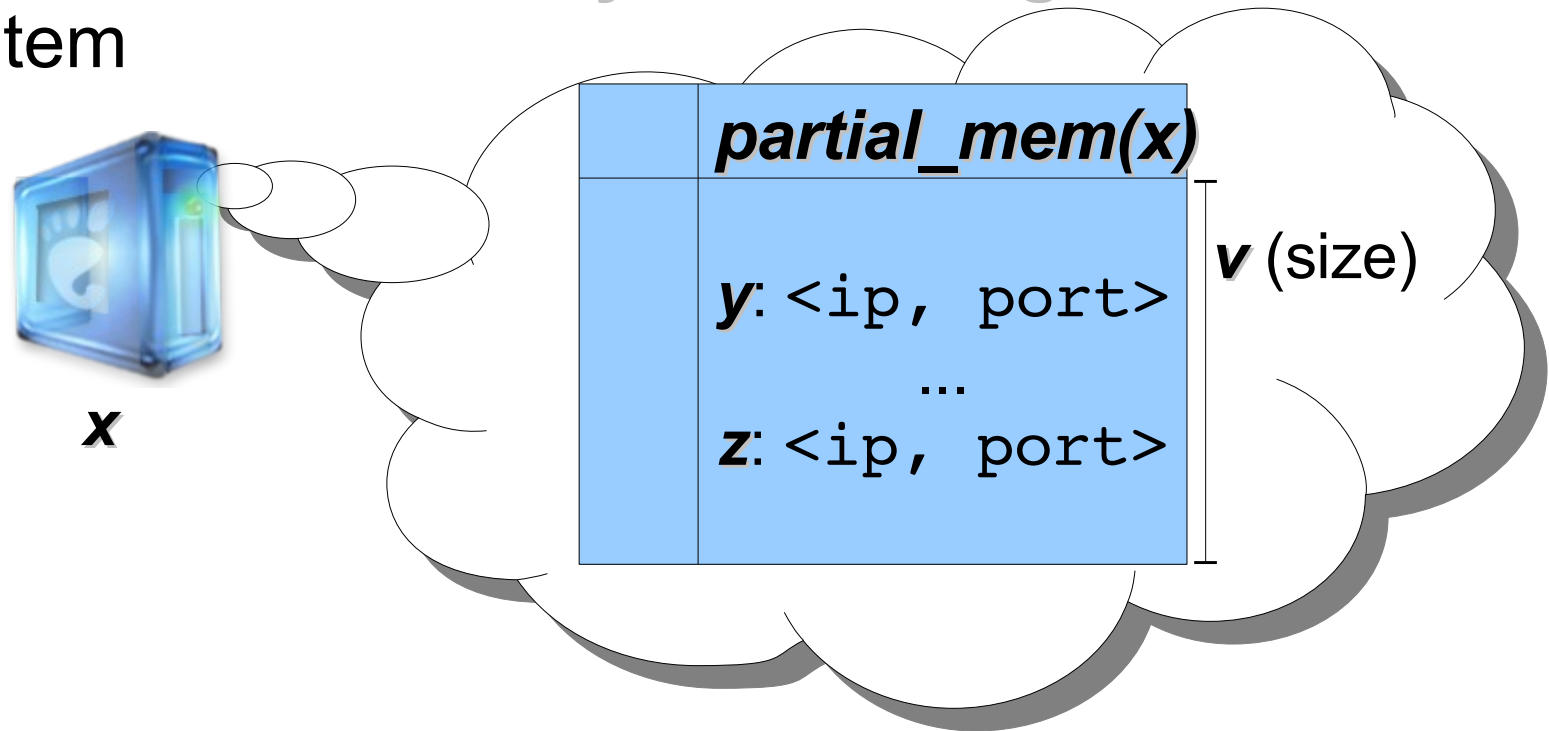
# **AVMEM:** Membership Maintenance



# AVMEM: Membership Maintenance

## ► Discovery Sub-Protocol:

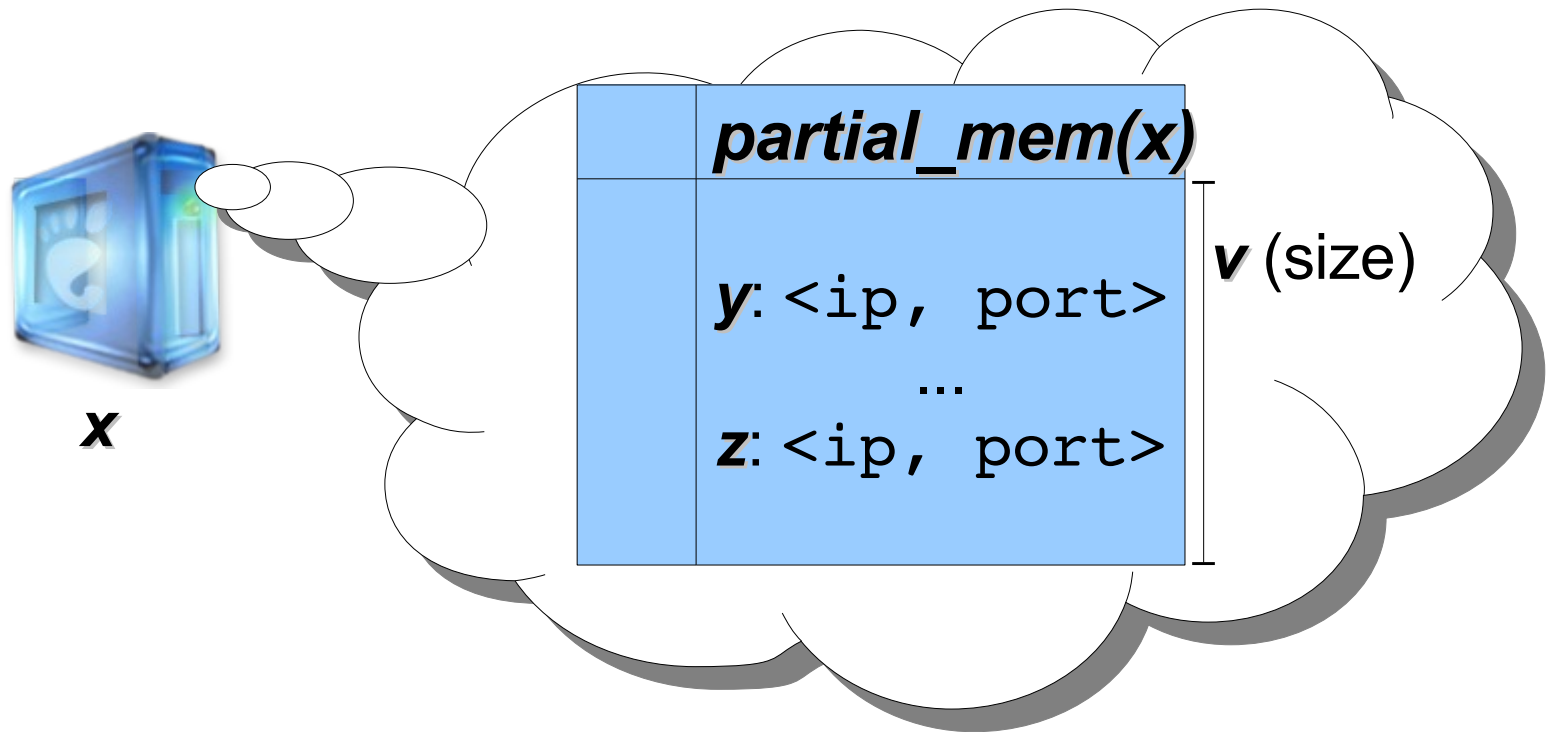
- Needs randomly changing partial membership list
  - [Ganesh 2003 (SCAMP)], [Jelasity 2005 (T-Man)], [Voulgaris 2005 (CYCLON)]
- Needs an **AV**ailability **MON**itoring [ICDCS '07] system



# AVMEM: Membership Maintenance

## ▶ Discovery Sub-Protocol:

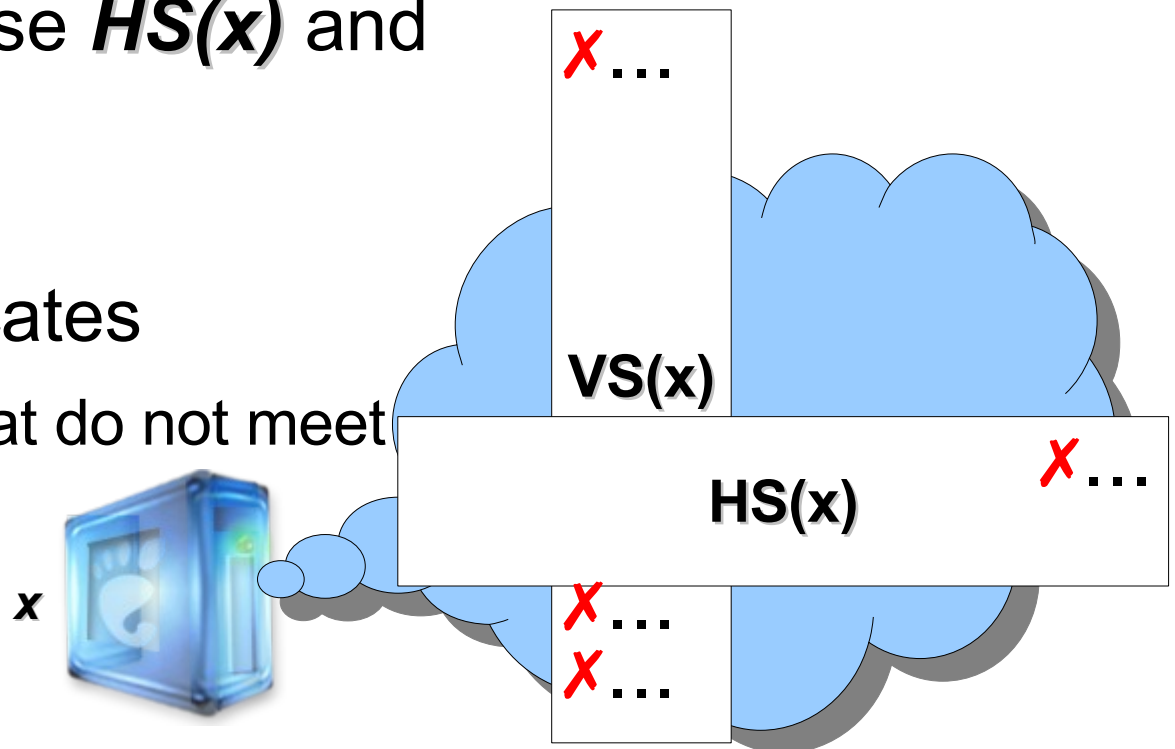
- Periodically traverse *partial\_mem(x)*
  - ▶ Exclude nodes already in  $HS(x) + VS(x)$
  - ▶ Predicates are applied to nodes in partial membership



# AVMEM: Membership Maintenance

## ▶ Refresh sub-protocol:

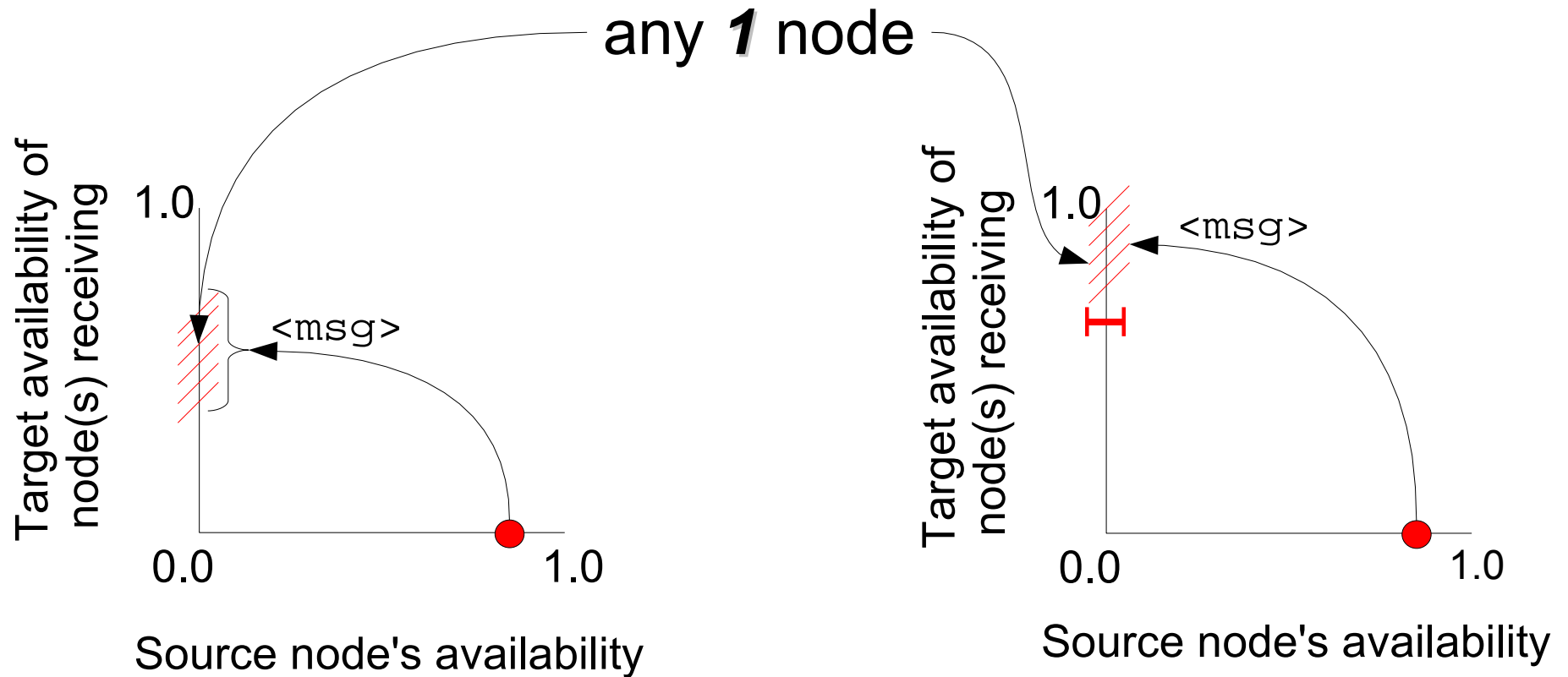
- Entries can become stale
  - ▶ Nodes can change availability
- Periodically traverse **HS(x)** and **VS(x)**
  - ▶ 20 min
- Recompute predicates
  - ▶ Drop neighbors that do not meet predicates



# Outline

- Introduction
- Design Goals
- **AVMEM**
  - Management operations

# AVMEM: Threshold/Range- Anycast



# AVMEM: Threshold/Range- Anycast

## ▶ Greedy Forwarding:

- Select as next hop **AVMEM**-neighbor with availability closest to target
- TTL (=6)

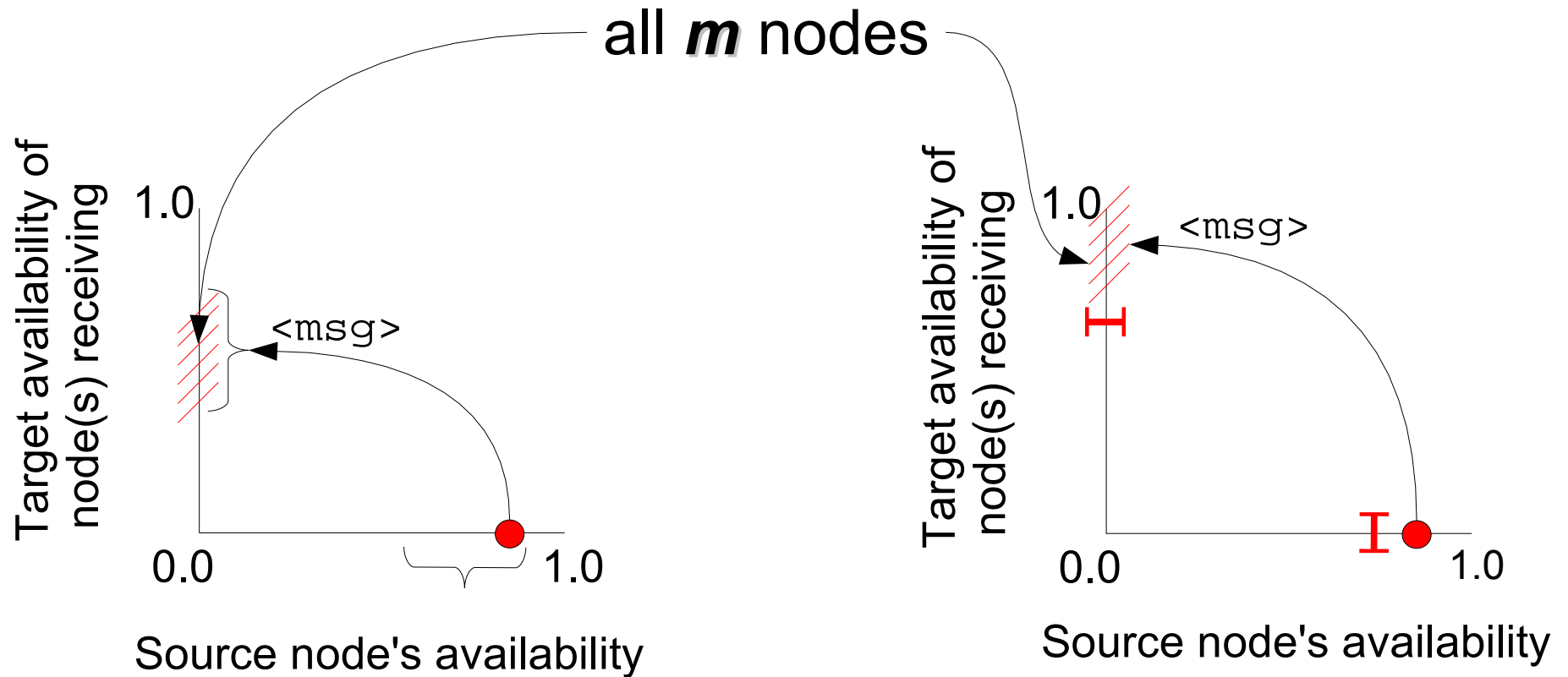
## ▶ Retried Greedy Forwarding:

- Try next-best **AVMEM**-neighbor if best is unresponsive
- up to  $k$  retries

## ▶ Simulated Annealing:

- Choose random **AVMEM**-neighbor with  $p = \exp(-\mathit{delta} / \mathit{ttl})$ , otherwise greedy

# AVMEM: Threshold/Range-Multicast



# **AVMEM**: Threshold/Range-Multicast

## ▶ Flooding:

- Forwards to **AVMEM**-neighbors inside range
- Ignore duplicate messages

## ▶ Gossiping:

- Once per period, forward to **fanout AVMEM**-neighbors inside range (excluding previously chosen neighbors)
- Repeat  **$N_g$**  times
- **$N_g * fanout = \log(N^*)$**  [Ganesh 2003]
- Ignore duplicate messages

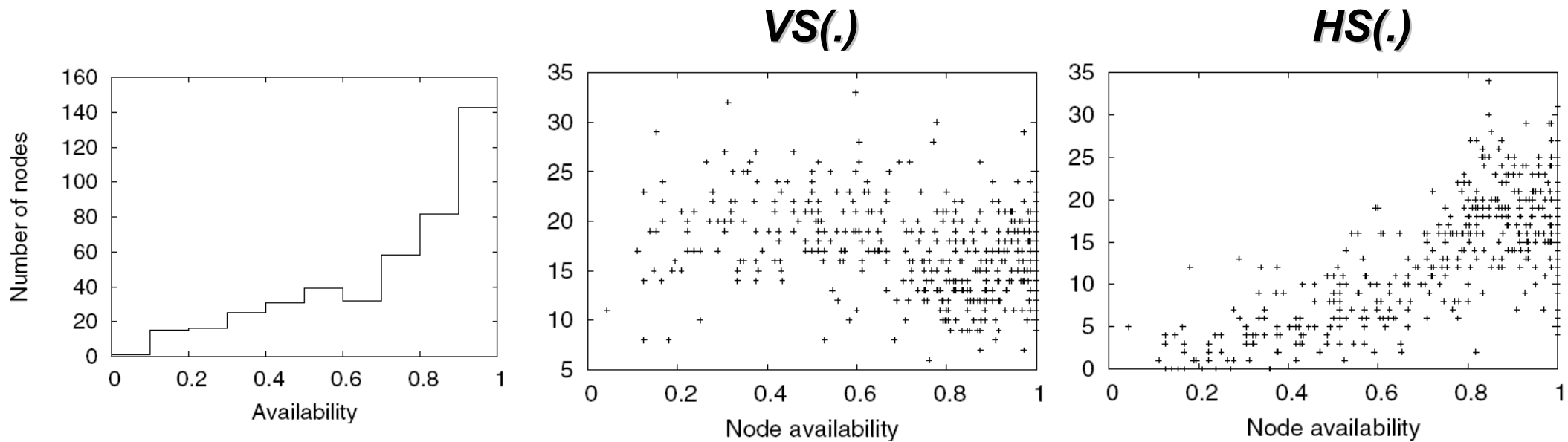
# Outline

- Introduction
- Design Goals
- **AVMEM**
  - Experiments

# Experimental Setup

- ▶ Implemented in C
- ▶ Use Overnet Availability traces [Bhagwan 2003]
  - Injected as collected
  - 2440 nodes
  - Around 550 nodes online at any time
- ▶ Logarithmic Vertical Sliver
- ▶ Logarithmic Horizontal Sliver

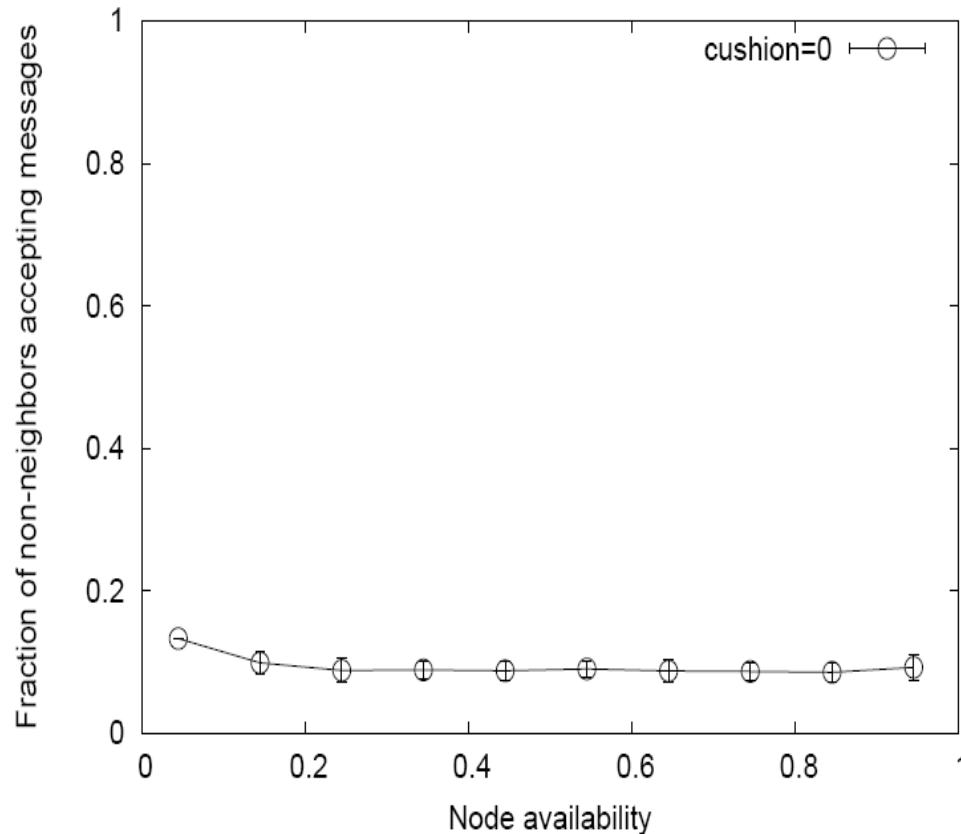
# Sliver micro-benchmark



- ▶ ***VS(.)*** size is uncorrelated to availability
- ▶ Increase in ***HS(.)*** is sublinear

# Flooding Attack

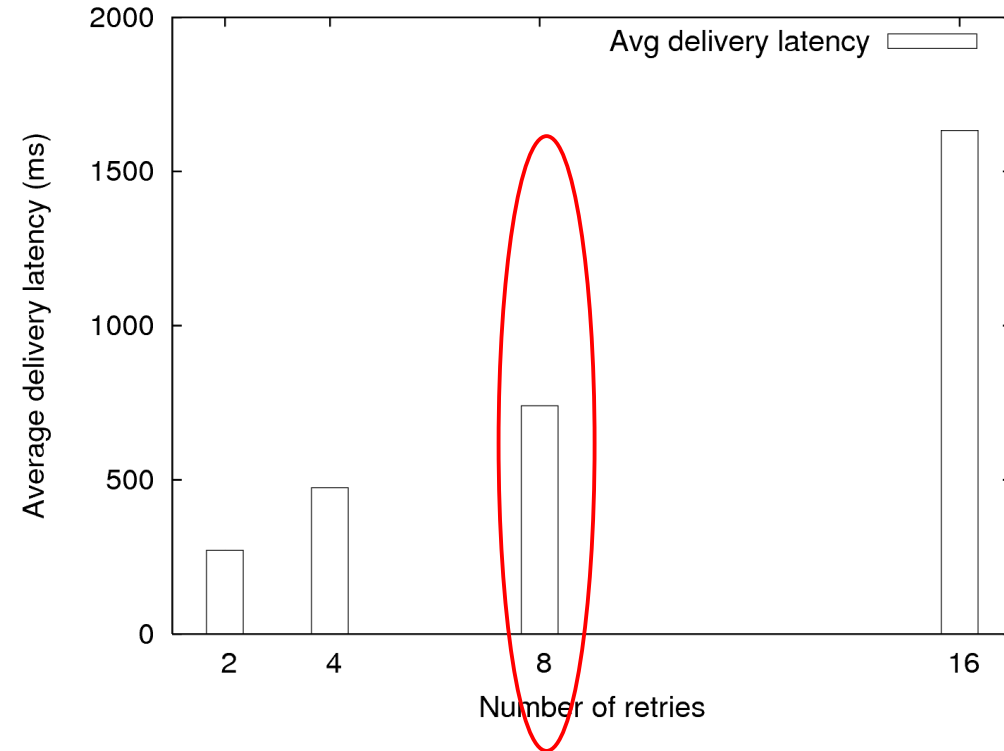
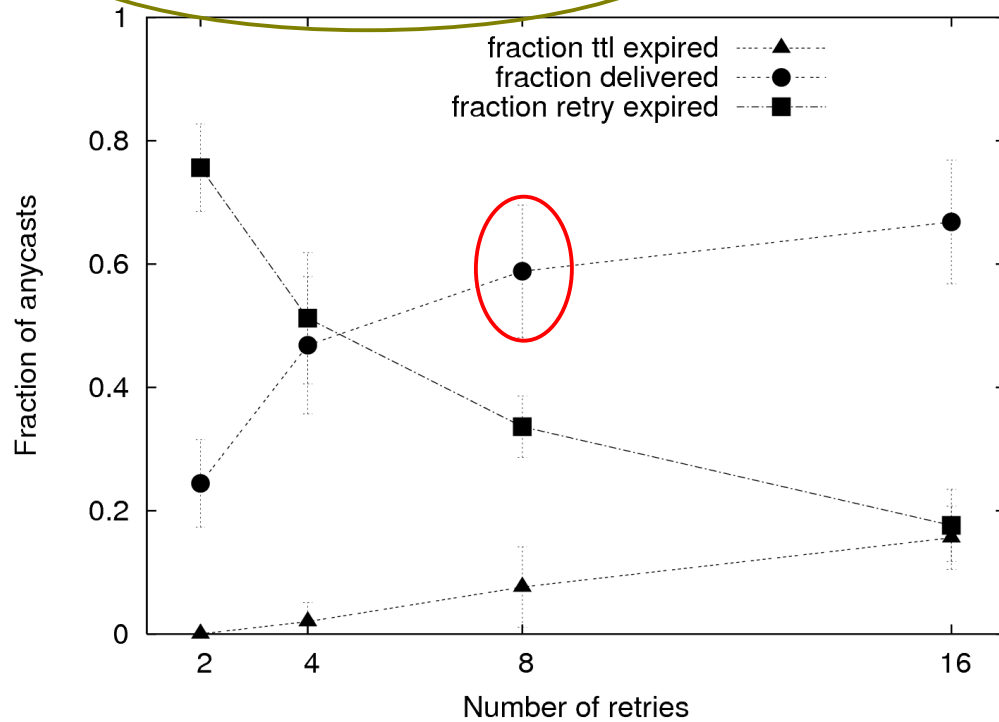
Malicious node contacting **AVMEM**-neighbors and **NON-AVMEM**-neighbors



- ▶ Fewer than 10% accept (attempted) flood
- ▶ Independent of node's availability

# Retried Greedy Anycast

node in HIGH availability  
sending to [0.15, 0.25]

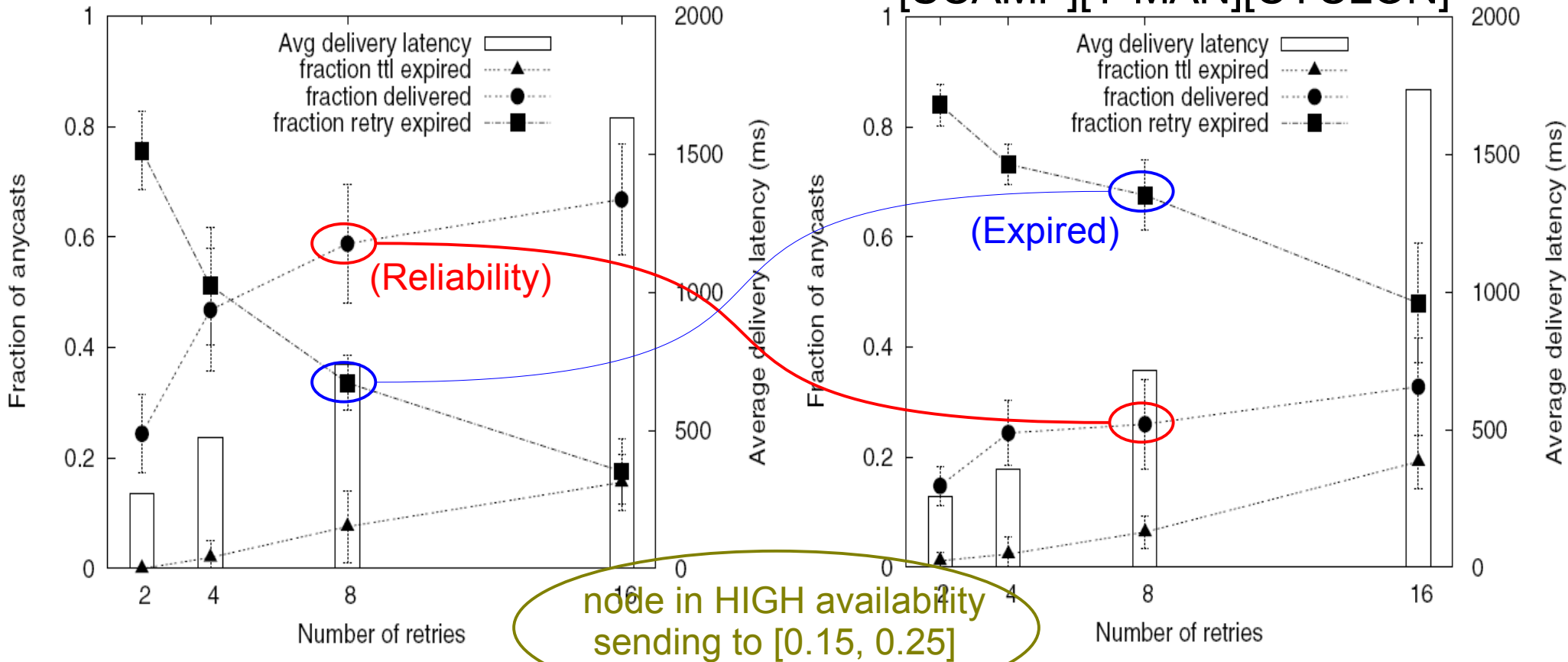


- ▶ Reliability improves in harsh scenario
- ▶ Reliability improves faster than latency increase

# Retried Greedy Anycast

## Random Graph [SCAMP][T-MAN][CYCLON]

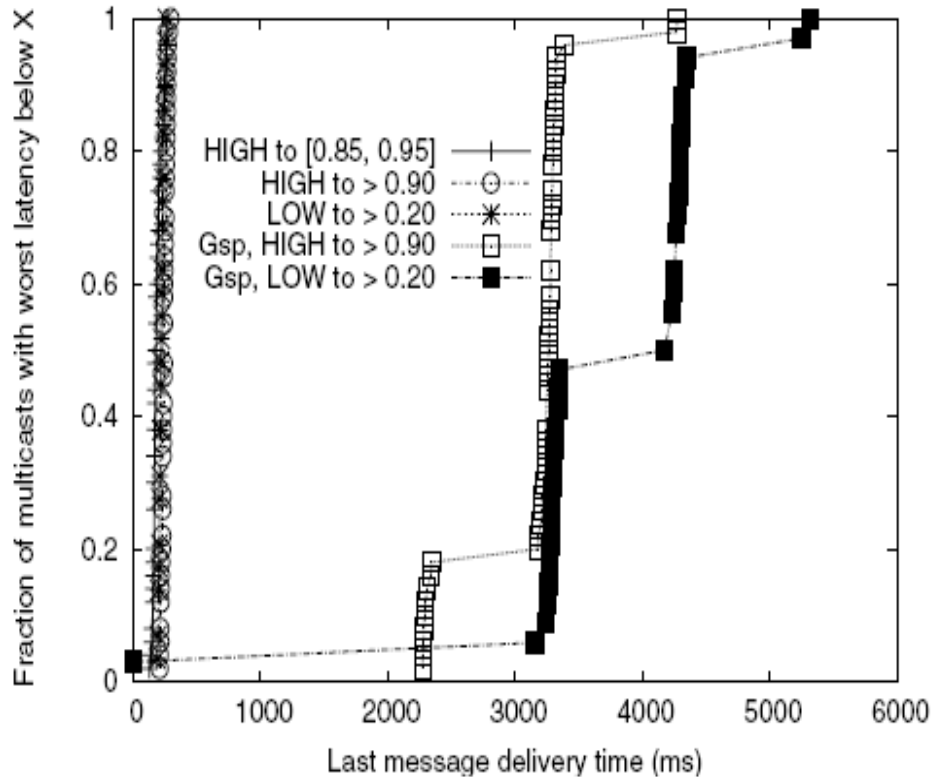
### AVMEM



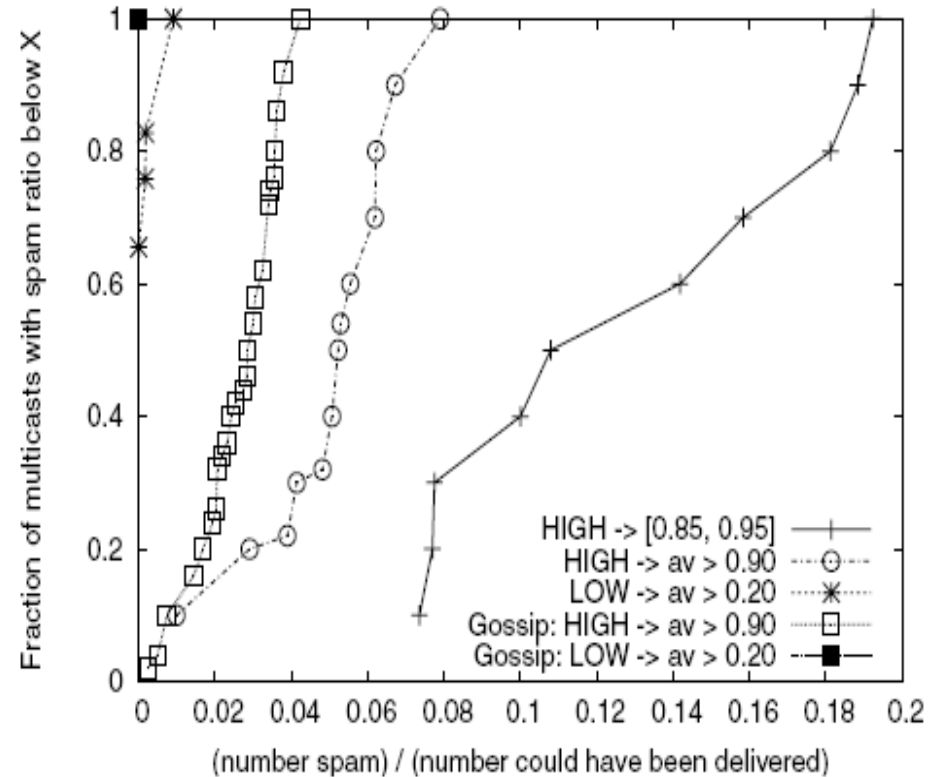
► Using **AVMEM** gives better reliability than & comparable latency to using random overlay

# Multicast Efficiency

## Latency



## SPAM



- ▶ Flooding is fast  $\leq 300\text{ms}$
- ▶ Gossip saves BW
- ▶ SPAM is low  $< 0.2$

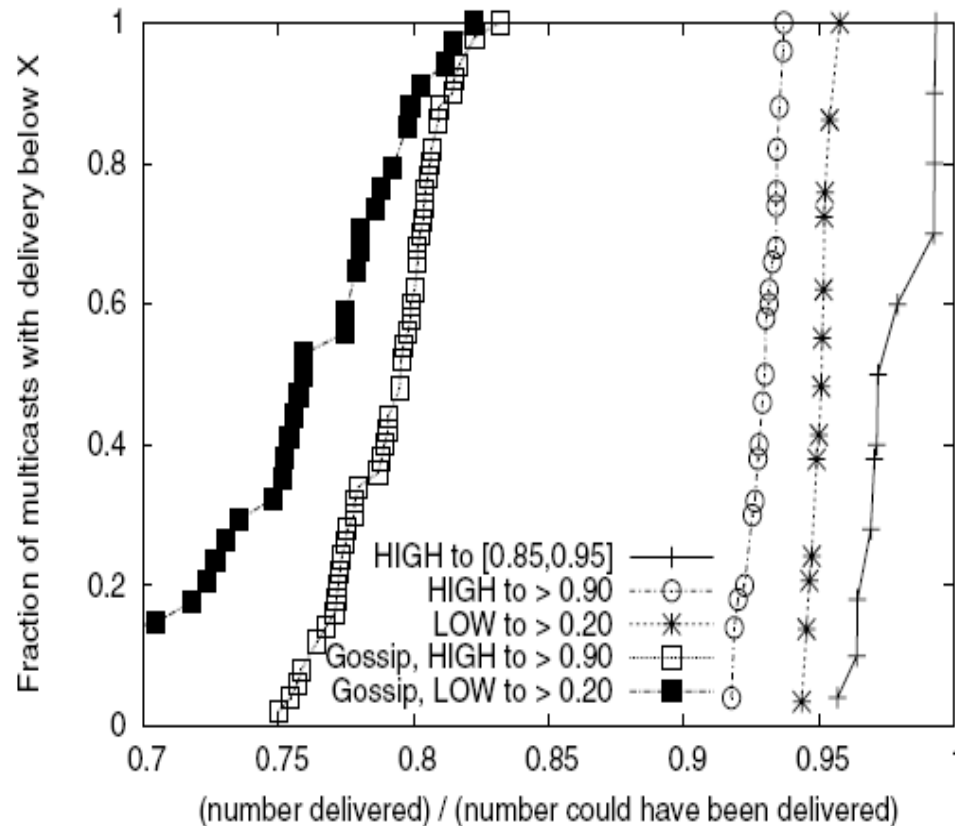
simulated latency in [20ms, 80ms]

# Conclusion

- ▶ **AVMEM** is first availability-aware overlay
- ▶ **AVMEM** overlay construction is scalable
- ▶ Management operations -- range/threshold - anycast/multicast -- can be implemented
  - reliably
  - efficiently



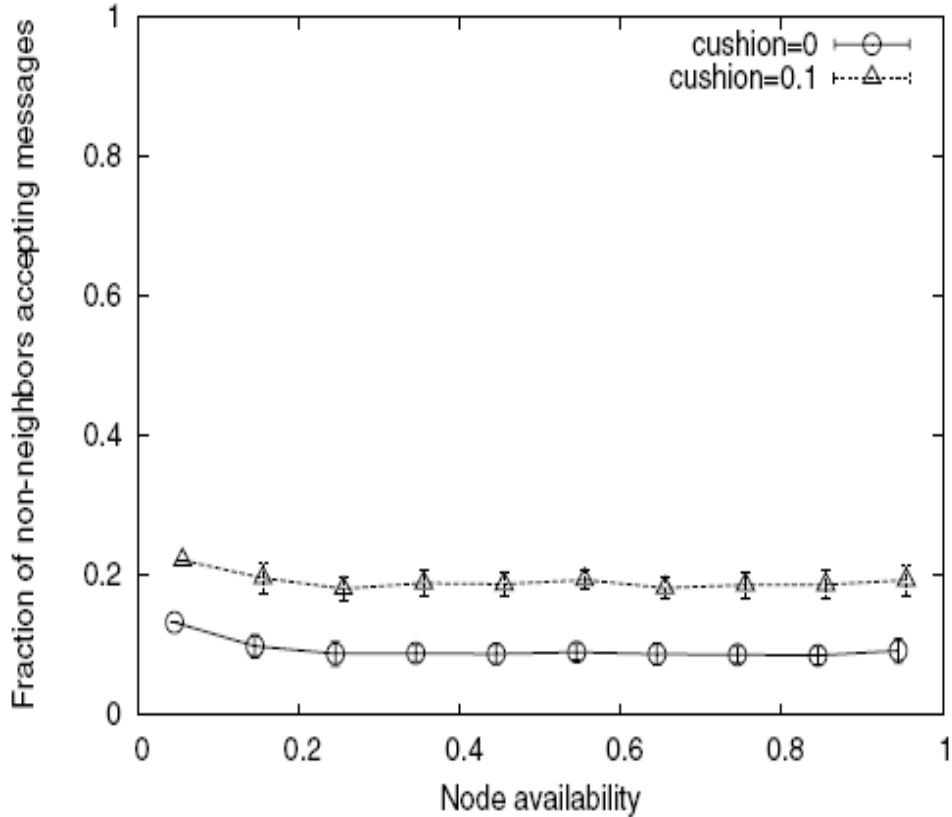
# Multicast Reliability



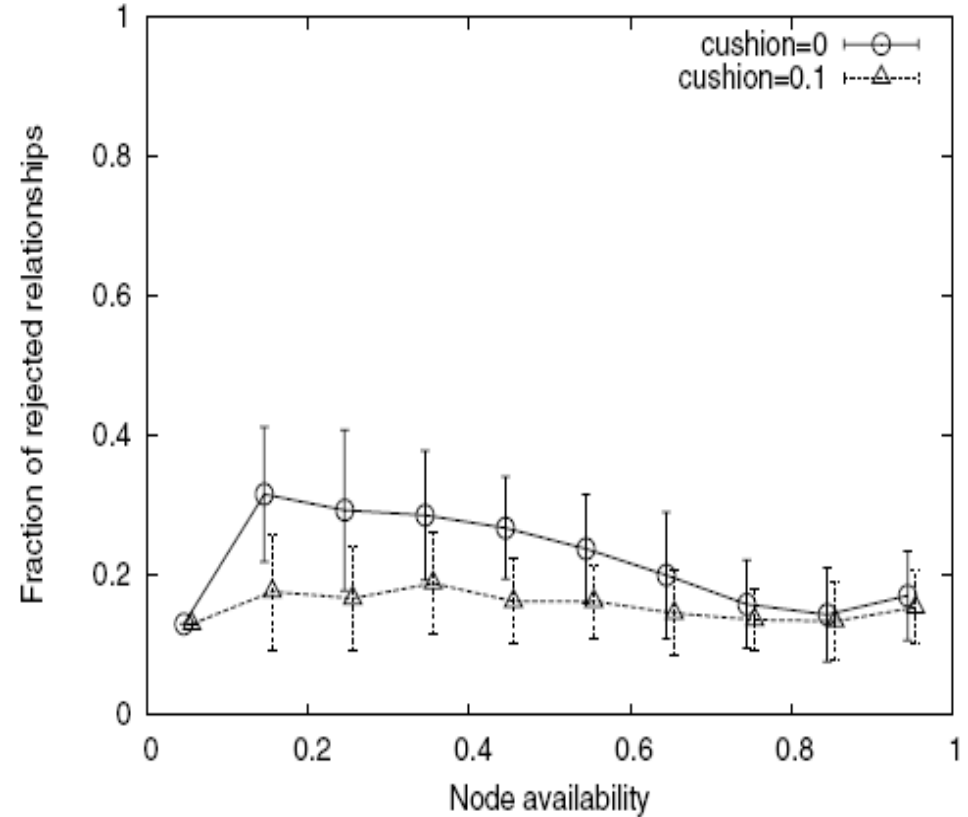
- ▶ Gossip > 70% reliability
- ▶ Flooding > 90% reliability

# Flooding Attack

## Non-peers accepting incoming

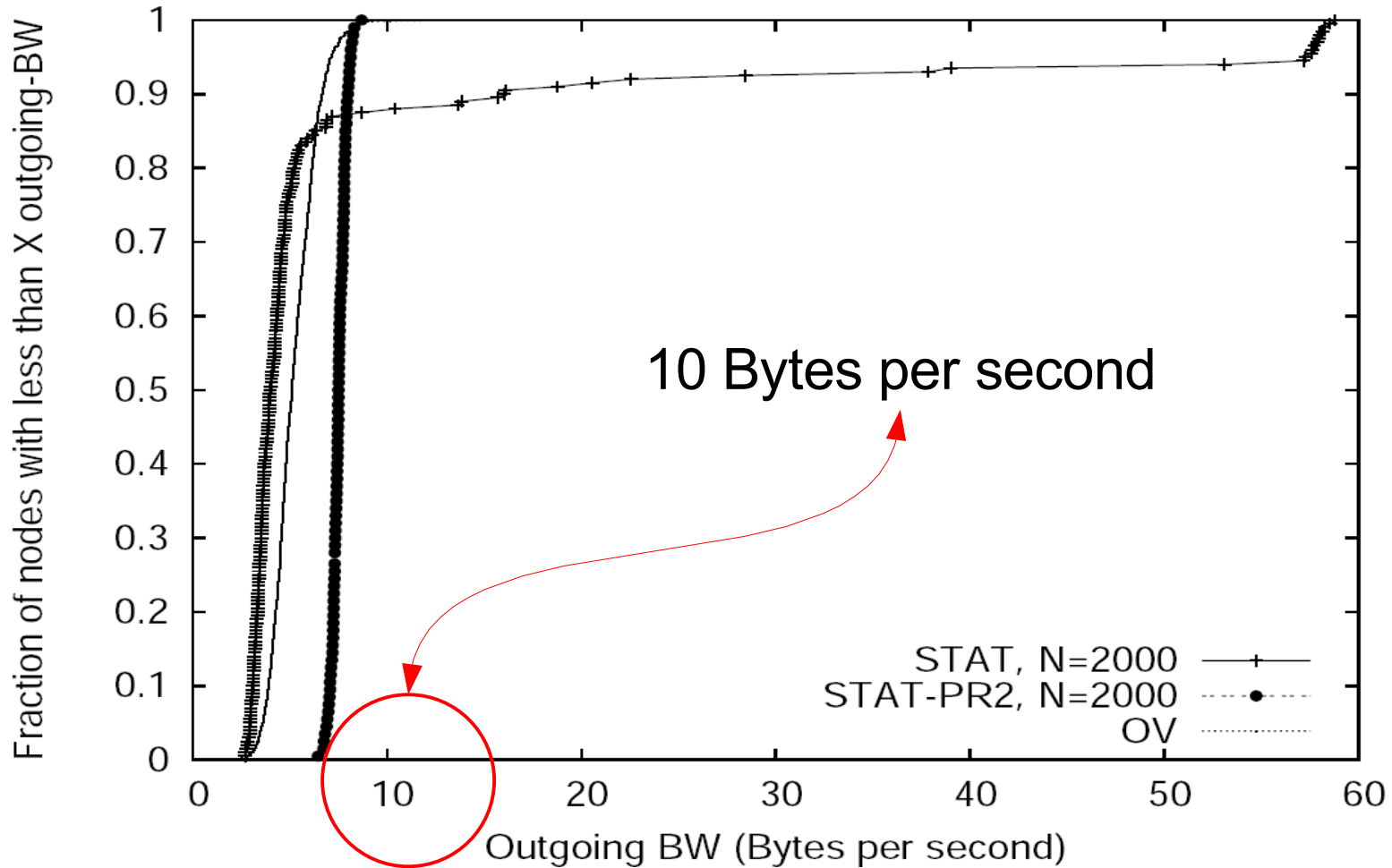


## Legitimate peer rejection



- ▶ Fewer than 10% accept incoming selfish msg
- ▶ Legitimate rejection rate is low
- ▶ Independent of node's availability

# Bandwidth



► BW is **uniform**, and **low**

$$= p(av(y)) da \cdot N^* \times \frac{c_1 \cdot \log(N^*)}{N^* \cdot p(av(y))} = c_1 \cdot \log(N^*) da$$

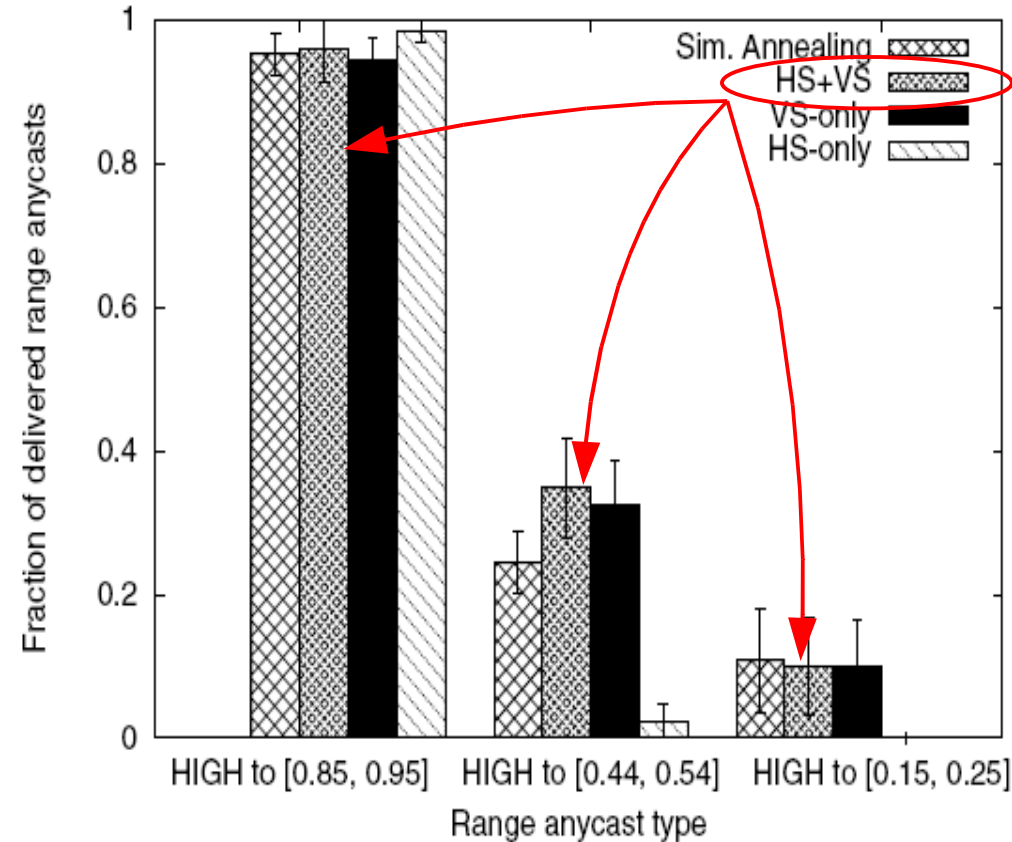
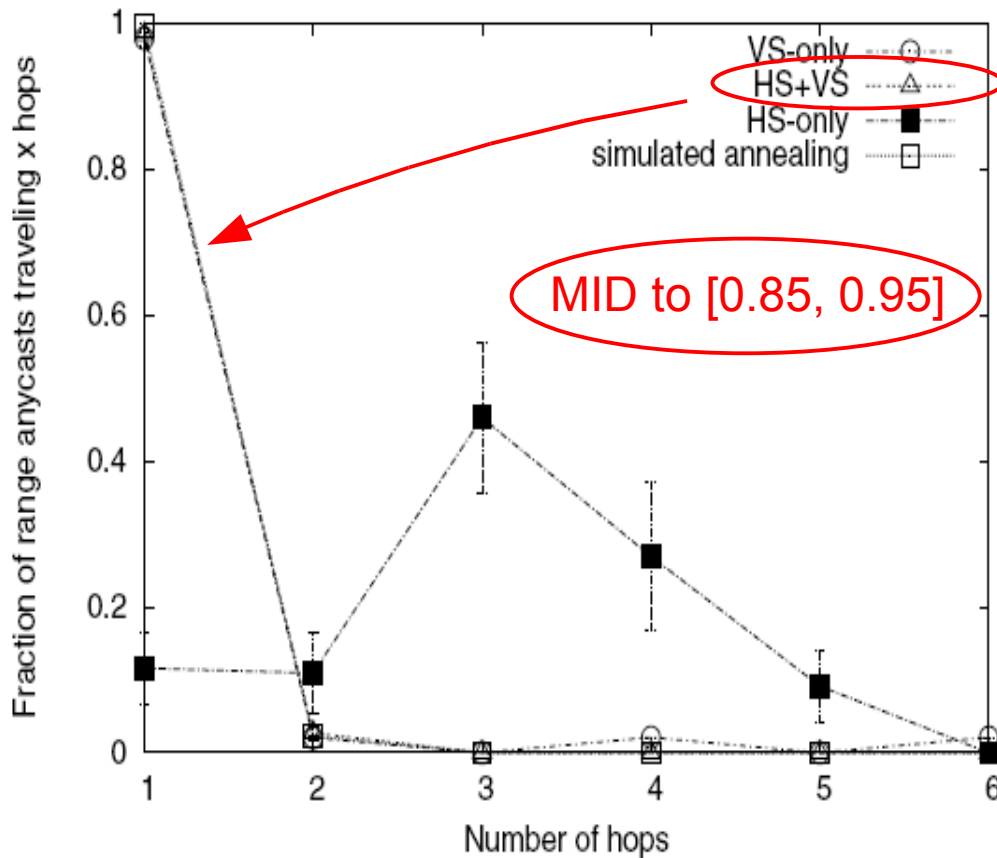
$$\begin{aligned}
&= \int_{av(x)}^{av(x)+\epsilon} \left( c_2 \cdot \frac{\log(N_{av(y)}^*)}{N_{av(y)}^{*min}} \times (N^* \cdot p(a)) \right) da \\
&= \frac{c_2 \cdot \log(N_{av(y)}^*)}{N_{av(y)}^{*min}} \cdot N_{av(x)}^{*+} \\
&\geq c_2 \cdot \log(N_{av(y)}^*), \text{ (since } N_{av(x)}^{*+} \geq N_{av(y)}^{*min} \text{)} \\
&\geq c_2 \cdot \log(N_{av(x)}^{*+}), \text{ (since } N_{av(y)}^* \geq N_{av(x)}^{*+} \text{)}
\end{aligned}$$

$$\begin{aligned}
&\geq \left(1 - \left(1 - \frac{c_2 \cdot \log(N^*)}{N_{av(x)}^{*+}}\right)^{N_{av(x)}^{*+}}\right) \times \left(1 - \left(1 - \frac{c_2 \cdot \log(N^*)}{N_{av(x)}^{*-}}\right)^{N_{av(x)}^{*-}}\right) \\
&\geq (1 - e^{-c_2 \cdot \log(N^*)}) \cdot (1 - e^{-c_2 \cdot \log(N^*)}) \\
&\geq \left(1 - \frac{2}{(N^*)^{c_2}}\right)
\end{aligned}$$

$$= \int_0^{av(x)-\epsilon} c_1 \cdot \log(N^*) da + \int_{av(x)+\epsilon}^1 c_1 \cdot \log(N^*) da \leq c_1 \cdot \log(N^*)$$

$$\leq \int_{av(x)-\epsilon}^{av(x)+\epsilon} \left( c_2 \cdot \frac{\log(N_{av(x)}^*)}{N_{av(y)}^{*min} \cdot \epsilon} \times (N^* \cdot p(a)) \right) da = c_2 \cdot \frac{\log(N_{av(x)}^*)}{N_{av(y)}^{*min}} \times N^*$$

# Anycast Delivery



► Greedy using HS + VS is fast and reliable

# AVMEM: Membership Maintenance

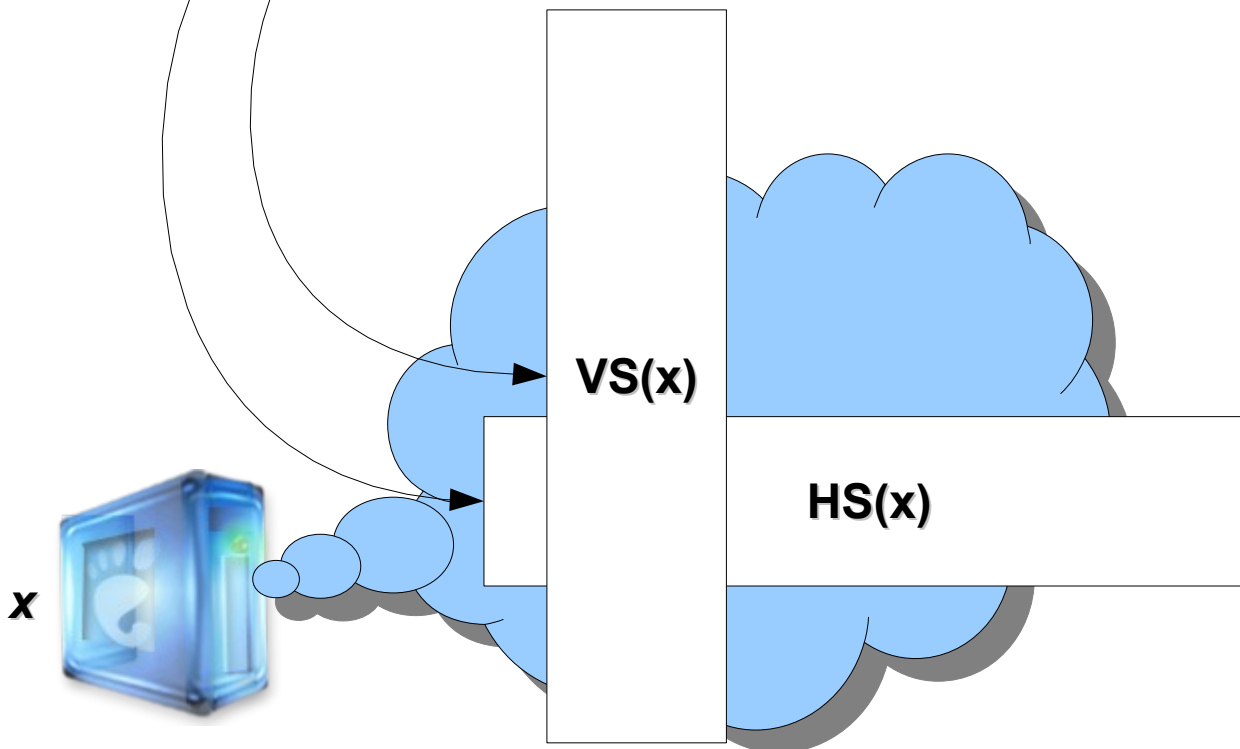
## ▶ Discovery Sub-Protocol:

- OPTIMALITY:
- time for  $y$  to appear in  $CV(x)$  is  $O(N/cvs)$
- minimize  $f(v) = cvs + N/cvs$
- $cvs = O(\sqrt{N})$ 
  - ▶  $N = 100,000$
  - ▶  $cvs = 320$

# AVMEM: Membership Graph Predicates

$$M(x, y) \equiv \{ H(id(y), id(x)) \leq f(av(x), av(y)) \}$$

$$\left. \begin{array}{l} hs(av(x), av(y), p(.)) \\ vs(av(x), av(y), p(.)) \end{array} \right\} \begin{array}{l} \text{if } |av(x) - av(y)| < \varepsilon \\ \text{otherwise} \end{array}$$



# AVMEM: Membership Graph Predicates

$$M(x, y) \equiv \{ H(id(y), id(x)) \leq f(av(x), av(y)) \}$$

$$\left. \begin{array}{l} hs(av(x), av(y), p(.)) \\ vs(av(x), av(y), p(.)) \end{array} \right\} \begin{array}{l} \text{if } |av(x) - av(y)| < \varepsilon \\ \text{otherwise} \end{array}$$

$p(a)da$ , fraction of nodes with availability between  $a$  and  $(a - da)$

