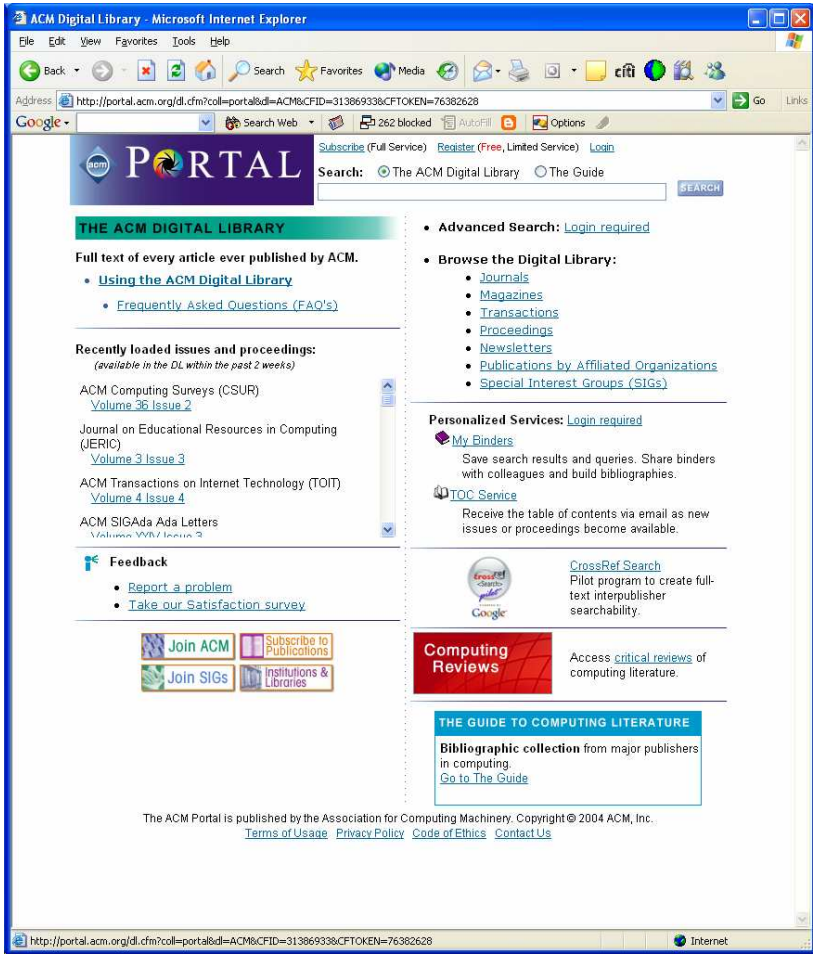
The LOCKSS P2P Digital Preservation System

Petros Maniatis,
Intel Research Berkeley
with Mary Baker (HP), TJ Giuli (Stanford),
David S. H. Rosenthal (Stanford Libraries),
Mema Roussopoulos (Harvard)

Intel Research Berkeley

Digital Preservation of Academic Materials

- Academic publishing is moving from paper to electronic media
  - Instead of purchasing paper copies, libraries rent access to on-line digital materials



Digital Preservation of Academic Materials

- Librarians are scared with good reason
  - Access depends on the fate of the publisher
  - Time is unkind to bits after decades
  - Plenty of enemies (ideologies, governments, corporations)
- Goal: Preserve access for a very long time

Digital Preservation of Other Materials?

- Librarians are scared with good reason
  - Access depends on the fate of the publisher
  - Time is unkind to bits after decades
  - Plenty of enemies (ideologies, governments, corporations)
- Goal: Preserve access for a very long time

FCW.COM

A crisis for Web preservation

Fugitive documents published on the Web are not being preserved

BY Florence Olsen
June 21, 2004

The Federal Depository Library Program has fallen behind in cataloging and preserving access to government documents published only on the Web. As a result, public access to those publications is spotty at best.

RELATED LINKS

Crawling for content

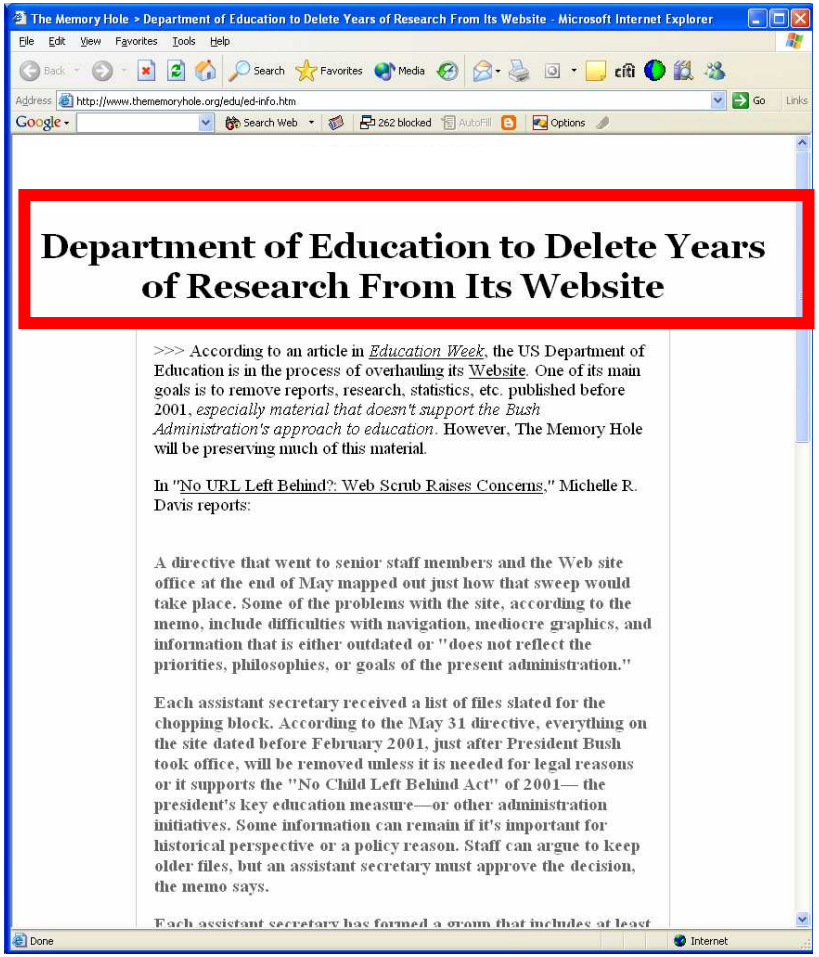
GPO Access site

Cyber Cemetery site

"GPO wants modernization money" [FCW.com, April 30, 2004]

Digital Preservation of Academic Materials

- Librarians are scared with good reason
  - Access depends on the fate of the publisher
  - Time is unkind to bits after decades
  - Plenty of enemies (ideologies, governments, corporations)
- Goal: Preserve access for a very long time



Intel Research Berkeley

The "Abstract" Preservation Problem

- Preserve access to replicas of an "Archival Unit" (AU)
  - AU may be a year-long run of a journal
- Requirements
  - Very low-cost hardware, operation and administration
  - No central control
  - Acceptable to publishers (e.g., no disruption or changes)
  - Respect for access controls
  - Independent of publisher's fate
  - A long-term horizon
- Must anticipate and degrade gracefully with
  - Undetected storage failures (bit rot, human error,…)
  - Sustained attacks

Protocol Threats

- Assume conventional platform/social attacks
- Mitigate further damage through protocol
- Top adversary goals:
  - Stealth Modification
    - Modify replicas to contain adversary's version
    - Hard to reinstate original content after large fraction of replicas are modified
  - Attrition
    - Waste peers' resources (network, application, human)
    - Storage failures overwhelm and damage the system
- Other adversary goals
  - Content theft, free-riding, unreliability, etc.

Problem Challenges

- Long-term horizon
  - A patient adversary can take 30 years to change history
- Use of crypto with long-term secrets over decades is tricky
  - Signing/encryption keys would have to be protected from disclosure
  - Verification/decryption keys would have to be preserved
    - recursive instance of the same problem
  - Content must be re-encrypted/re-signed
    - when keys expire, when hash functions break, when quantum computing takes off, etc.
  - All of the above require publisher modifications
- No affordable storage medium is perfectly reliable
  - Even if one existed, we'd still have catastrophes, human error, bit rot
- Short-term guarantees of maximum malice are difficult to maintain
  - Majority of population can be taken over for a short while (e.g., worm)
- The Internet is (still) wide-open
  - DDoS is almost a turn-key application, at least for short-term attacks

Problem Opportunities

- Digital preservation must *prevent* change, not *precipitate it*
  - Operate *no faster than necessary*, not *as fast as possible*
- Efficiency is *not* a goal; feasibility is
  - Inflate cost of operations to improve attack economics within budget
- We get massive redundancy "for free"
  - Peers (libraries) demand whole local replicas of content
  - Cannot fiddle with erasure coding, etc.
- Population is relatively stable
  - Peers expected to have repeat interactions, have social network
- Preservation need not include end-users (readers) in the loop
  - Traffic/operation patterns can be relatively stationary
  - Can model "ostensibly legitimate" workloads for anomaly detection

The LOCKSS Solution

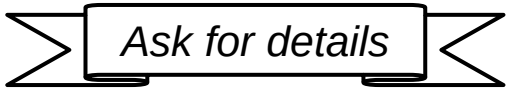
- Peer-to-peer auditing and repair system
  - For replicated archival units
  - No file sharing
- A peer periodically audits its AU replica
  - It calls an opinion poll among those with same AU
- When a peer suspects an attack, it raises an alarm for a human operator
  - Correlated failures
  - IP address spoofing
  - System slowdown
- New iteration of a deployed system

Sampled Opinion Poll

- Each peer holds for every preserved AU
  - *reference list* of peers it has discovered
  - *friends list* of peers its operator knows externally
  - *history* of interactions with others (balance of contributions)
- Periodically (faster than rate of storage failures)
  - *Poller* takes a sample of the peers in its reference list
  - Invites them to *vote*: send a hash of their replica
- Compares votes with its local copy
  - Overwhelming agreement (>70%) F Sleep blissfully
  - Overwhelming disagreement (<30%) F Repair
  - Too close to call F Raise an alarm
- To repair, the peer gets the copy of somebody who disagreed and then reevaluates the same votes

State Update

- Reference List
  - Take out voters, so that the next poll is based on different group
  - Replenish with some “strangers” and some “friends”
    - Strangers: Accepted nominees proposed by voters who agree with poll outcome
    - Friends: From the friends list
    - The measure of favoring friends is called *friend bias*
- History
  - Poller owes its voters a vote (for their future polls)
  - Detected misbehavior penalized in victim’s history



Defenses

LOCKSS Defenses

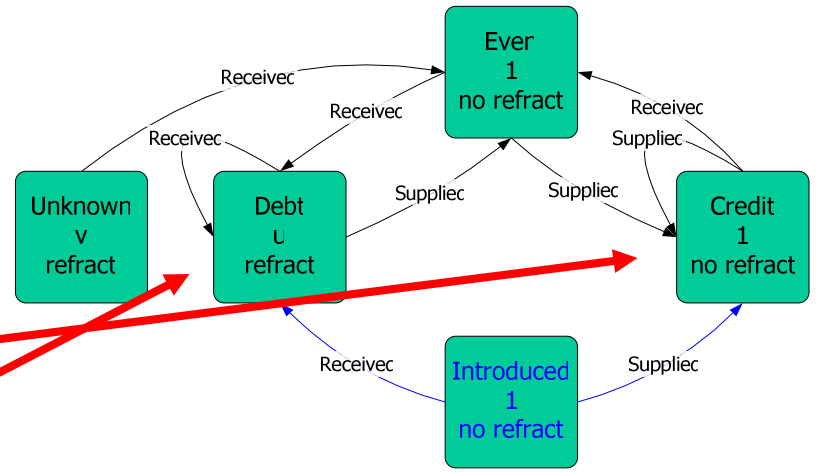
- Limit the rate of operation
- Effort balancing
- Bimodal alarm behavior
- Bias reference list with friends
- State unpredictability

Limit the rate of operation

- During initiation of new polls
  - Peers determine their rate of calling polls autonomously
  - No changes due to external stimuli
  - Adversary must wait for my next poll to attack me as a voter
- Keep poll rate constant to cap attack rate!

Limit the rate of operation

- During admission of new-poll invitations
  - "Self-clocking": accept invitations at the rate I invite others
  - Peer "goodness": local history of my interactions with others
  - Random drops for unknown or undesirable peers
  - Upper limit from local commitment schedule
- Those to whom I owe
  - Admitted
- Those who owe me
  - Subject to random drops
  - Rate limited



Ask for details

Intel Research Berkeley

Effort Balancing

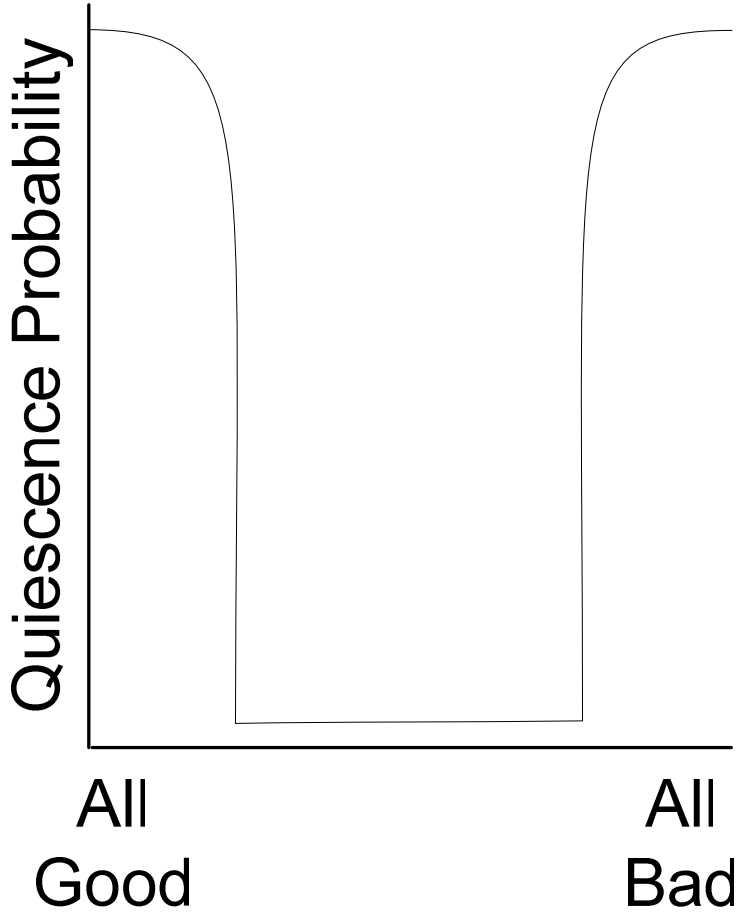
- Problem: Induced “splurge”
- In a multi-step protocol, if cause is cheap but effect expensive
  - You’ve got an Attrition vector!
- Examples
  - TCP SYN floods
    - Cheaply exhaust connection descriptors
  - SSL session initiation floods
    - Cheaply exhaust SSL-box resources
  - Email spam
    - Cheaply cause me to read message and hit DELETE

Effort Balancing

- No operational path is faster than others
  - Artificially inflate "cost" of cheap operations
  - No attack can occur faster than normal ops
  - Ostensible legitimacy costs as much as normal oper.
- Use "pricing via processing" approach
  - Memory-bound functions [Dwork 2003]
  - Poll invitation inflated to match cost of a vote
  - Vote inflated to match cost of vote validation
- Use "compliance enforcement"
  - Partial, cheaper protocol runs penalized
  - Compliance "receipt" computed during operation

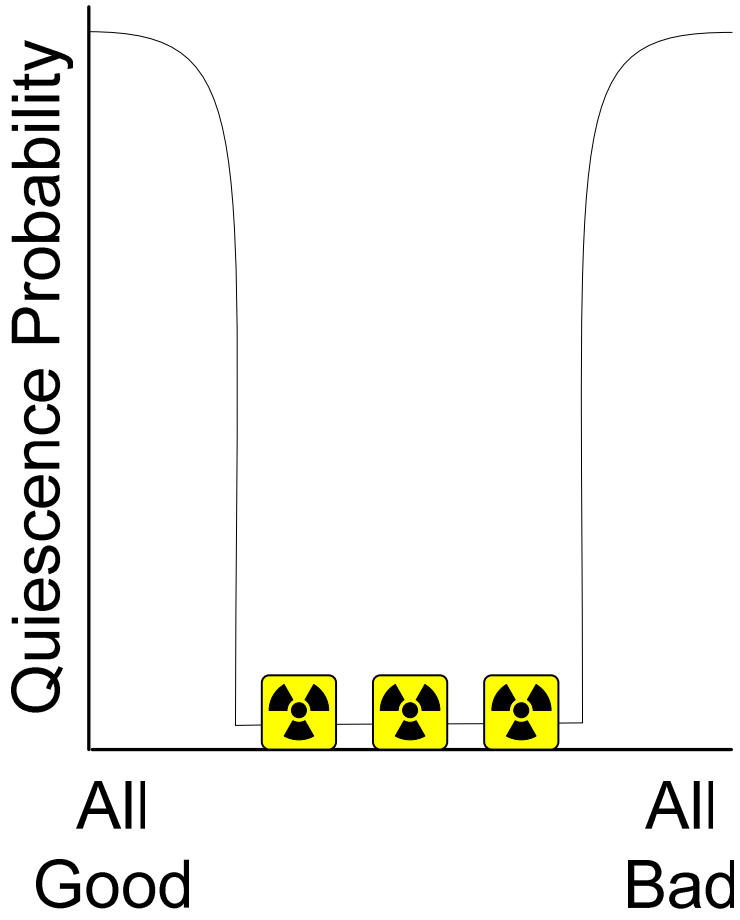
Bimodal Alarm Behavior

- When most replicas are the same, no alarms



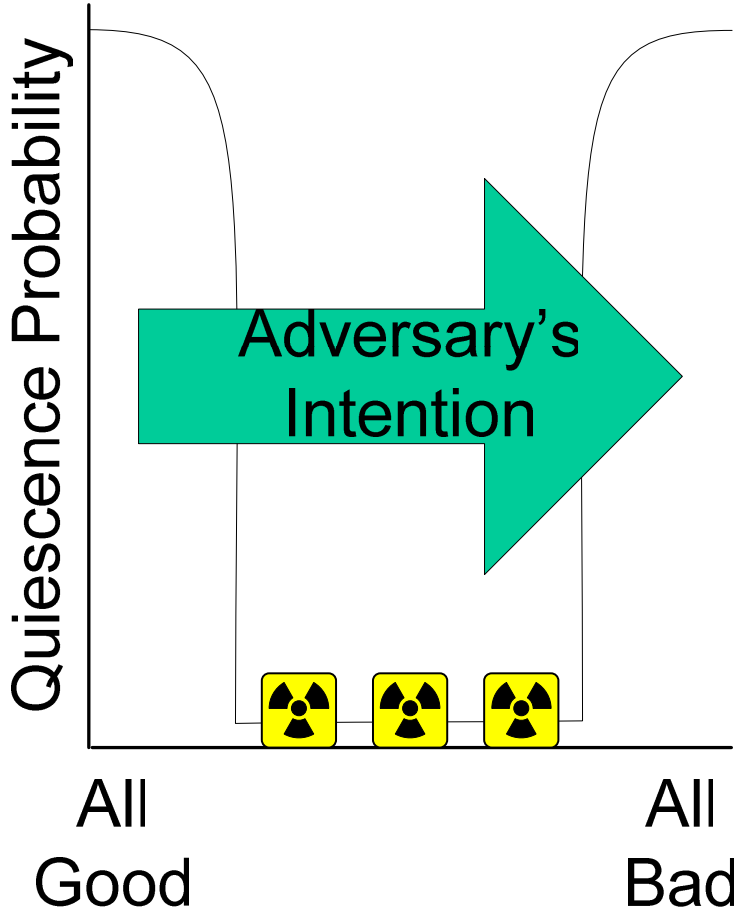
Bimodal Alarm Behavior

- When most replicas are the same, no alarms
- In between, alarms are very likely



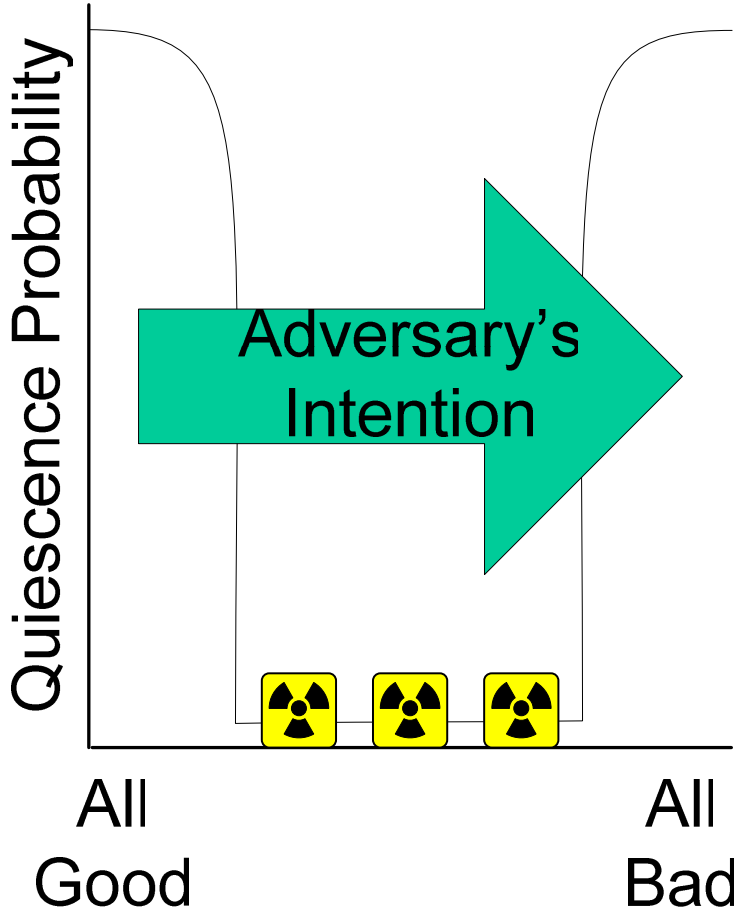
Bimodal Alarm Behavior

- When most replicas are the same, no alarms
- In between, alarms are very likely
- To get from mostly correct to mostly wrong replicas, system must pass through “moat” of alarming states:
  - Damaged peers voting with undamaged peers



Bimodal Alarm Behavior

- Rate limitation helps!
  - Arrow speed slow
  - Arrow speed unbiased
- Adversary victim of his own success
  - Damaged peers call inconclusive polls



Bias Reference List with Friends

- Take advantage of social network
- High friend bias favors friends
  - Reduces the effects of Sybil attacks
    - Single entity posing as multiple identities
  - But offers easy targets (friends) for focused attack
- Low friend bias favors strangers
  - It offers Sybil attacks free reign
    - Bad peers nominate bad; good peers nominate some bad
    - Recently termed an *Eclipse* attack [Singh2004]
  - Makes focused attack harder, since adversary can predict less of the poll sample
- Adversary foothold in reference lists never 100%
  - "Moat" of alarming states hard to jump over
- Goal: strike a balance

State Unpredictability

- If you don't know whom I'm inviting, you can't target them (e.g., DDoS them)
  - Randomize order of obtaining votes for a single poll
  - Spread vote solicitations over months
  - If solicitation fails, try again later; do not replace
    - Replacement would allow bias of voter pool
- If you don't know I'm damaged, you don't know if refusing me a repair hurts me
  - Asking for repairs reactively signals desperation
  - Instead, ask for repairs (even unneeded) proactively

Evaluation

Intel Research Berkeley

Evaluation Methodology

- Model very powerful, realistic adversaries
  - Network control, computational power
- Identify major goals of adversary attacks
- Implement rational adversary strategies
  - Policies on when to defect and how
- Measure the impact of each strategy
  - locally (on readers)
  - globally (on document survival)
- Use the Narses protocol simulator
  - Simplified network model (flow level, no congestion)
  - Large time scales

Metrics

- Worst-case access failure probability
  - Reader obtained bad replica
- Preservation failure probability
  - Majority of replicas damaged coherently
  - Change of historic record
- Others
  - Friction
    - Defensive cost increase due to attacks
  - Cost ratio
    - Attack cost vs. defense cost
  - Delay ratio
    - Increase in mean time between successful polls at a peer

Adversary Model

- Unconstrained resources
  - Purchased (cheap) or spoofed (cheaper) identities
  - Limitless computational power (though cannot break crypto)
- Network control
  - Shuts down fraction of network for variable time periods (pipe stoppage)
  - Edge presence only (i.e., core routers are correct)
- Subversion
  - Can initially take over fraction of peer population
  - Through platform exploits, bribery, broken kneecaps
  - Taken over peer is subverted for good
- Perfect coordination
  - Instantaneous communication with and control of minions
  - Flawless content preservation
- Perfect knowledge of peers' operational parameters
  - Including resource commitment schedules

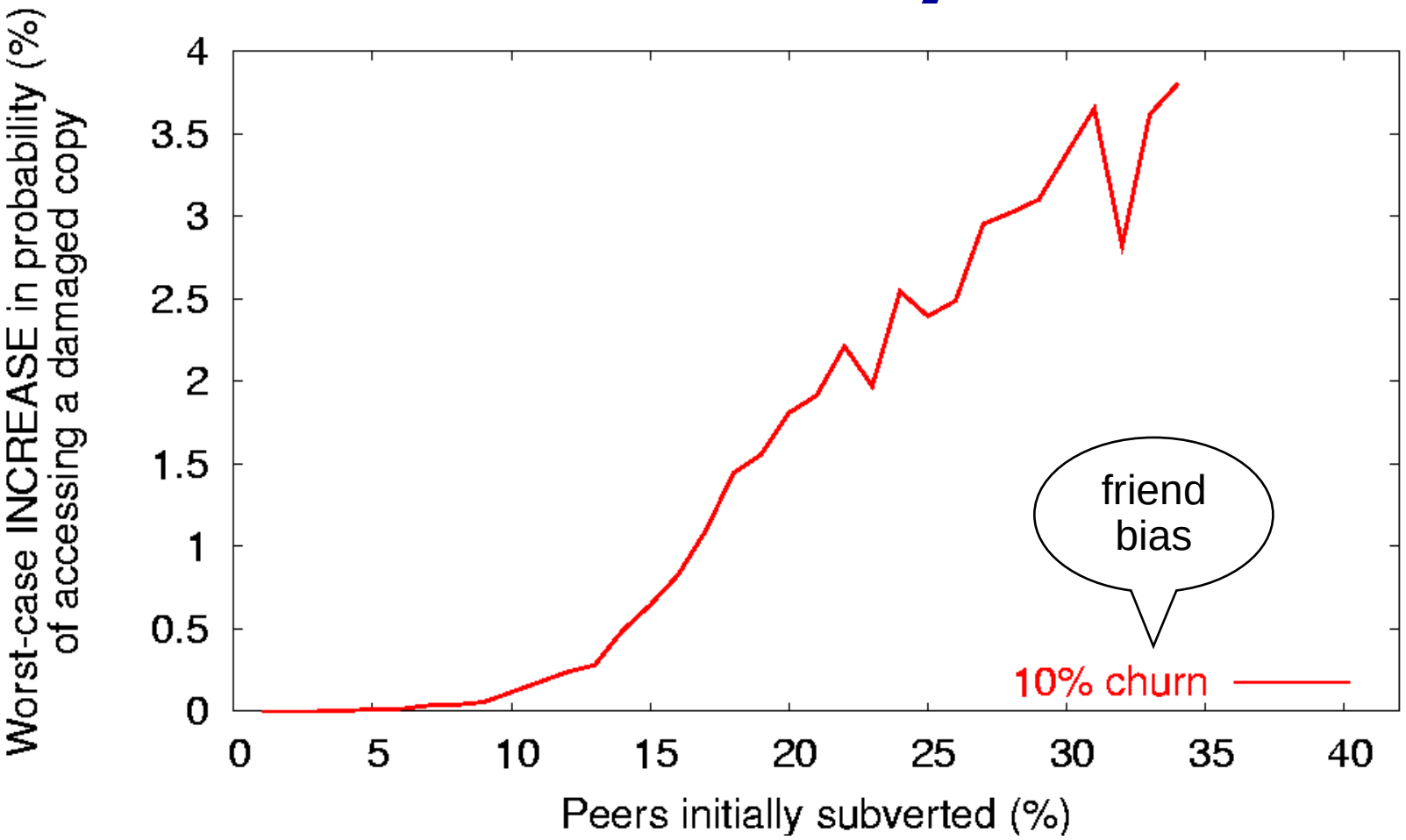
Stealth Modification Strategy

- Goal:
  - Replace good replicas with bad replicas
  - Avoid detection (i.e., alarms)
- First he lurks
  - Increase his foothold in unsubverted peers' reference lists
  - Follow the protocol, but always nominate bad peers
- Then he attacks
  - When adversary underrepresented in poll
    - he votes with correct replica to avoid an alarm
  - When well represented
    - Convince unsubverted peer its replica is damaged
    - Supply a damaged "repair" copy of the AU
- Large foothold reduces likelihood of detection

Stealth Modification Setup

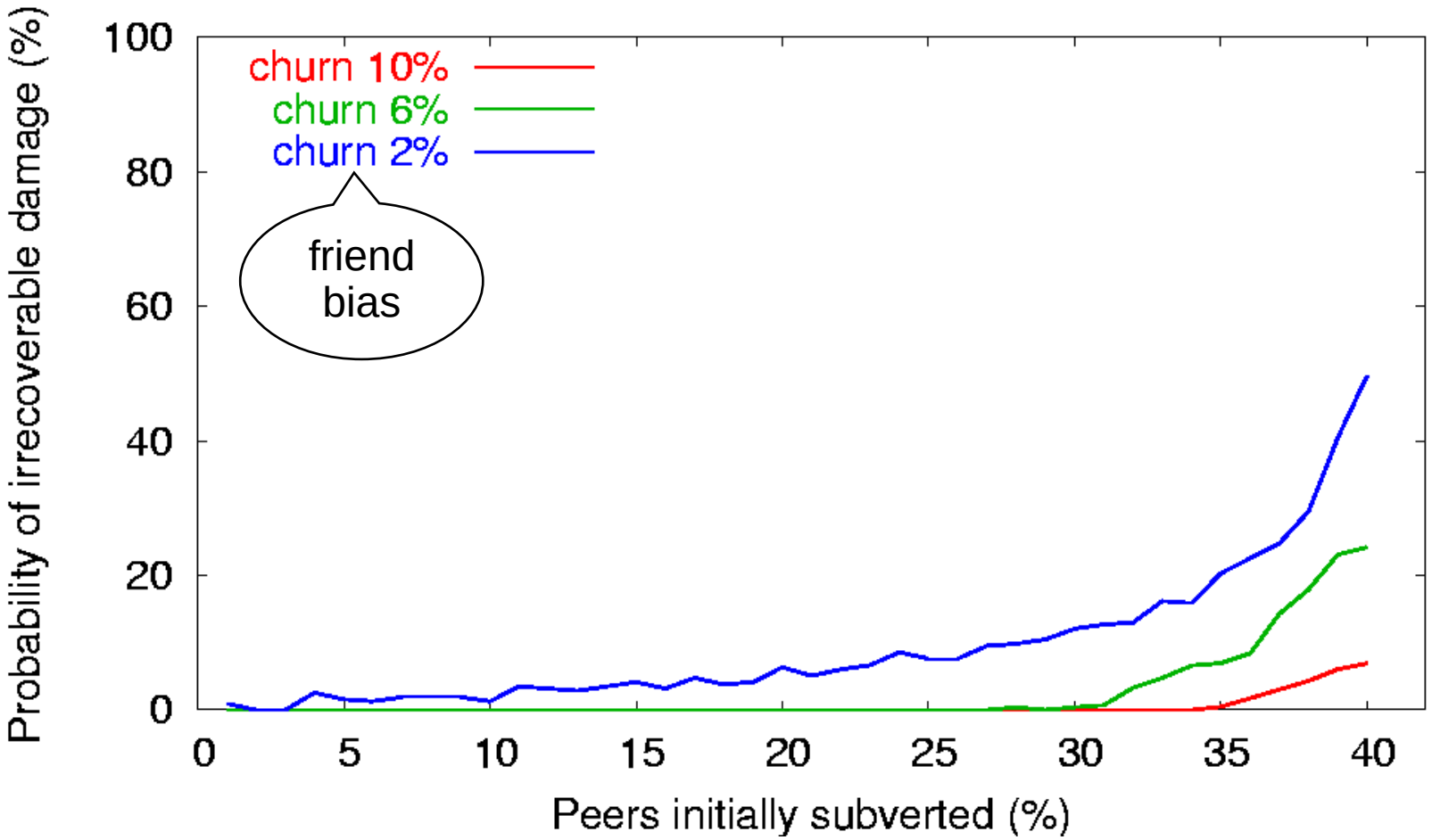
- 1000 original peers, clustered friends lists
- Initially, 0 – 40% are subverted for good
- Adversary lurks for up to 20 years
- Attacks for up to 10 more years
- Report worst-cases over ~200 runs per data point

Stealth Modification: Marginal Access Failure Probability



Intel Research Berkeley

Stealth Modification: Preservation Failure Probability



Intel Research Berkeley

Attrition Strategies

- Goal
  - Waste peers' resources so as to slow down audits
  - If peers are busier, they are harder to schedule

1 Pipe-stoppage
  - DDoS varying fractions of population for varying time
  - Affected population runs / participates in no audits

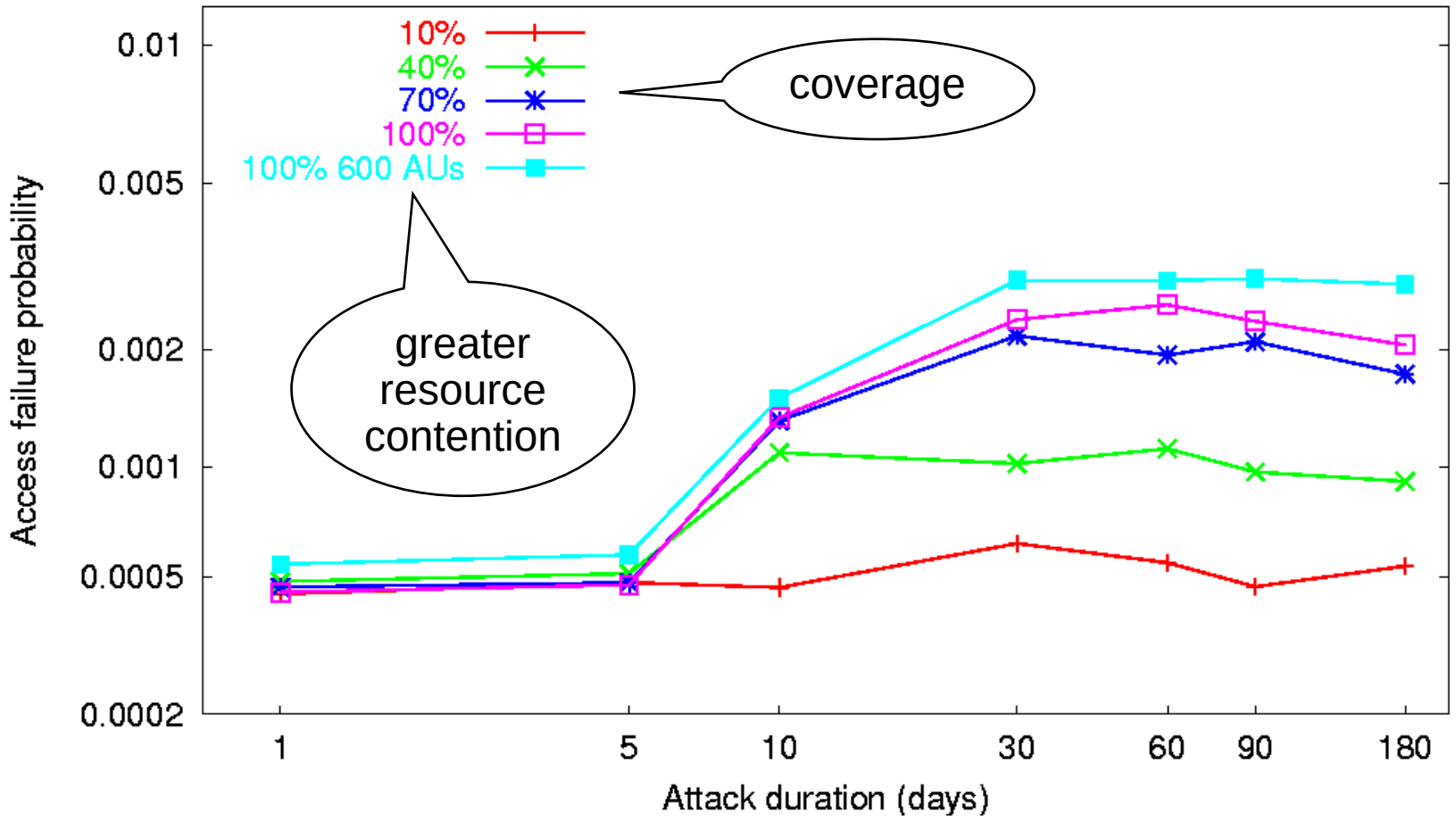
2 Request floods
  - Anomalously high-rate request traffic to overcommit peers
  - Affected population must exercise heavy admission control

3 Ostensible Legitimacy
  - (Locally undetectable) low-rate request traffic
  - Different points of defection
  - Computation required (*effortful* attack)

Attrition Setup

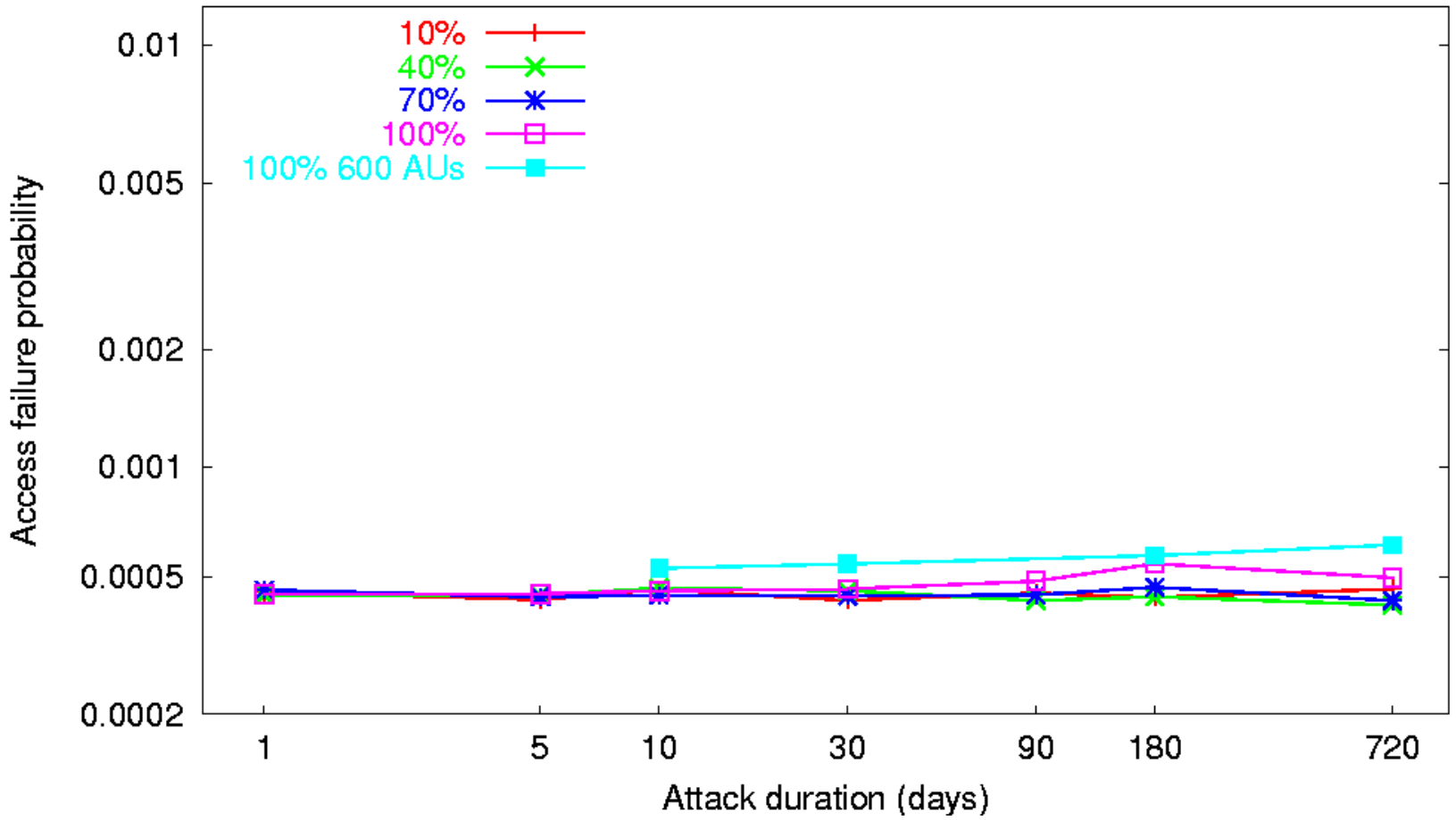
- 100 peers
- Each peer preserves up to 600 AUs
- Adversary external to population
- Total of 2 simulated years
- Attack coverage 10%-100% of population
- Attack duration up to 6 months at a time
- 30-day reprieve between attacks

Pipe stoppage: Access Failure Probability



Intel Research Berkeley

Request Flood: Access Failure Probability



Intel Research Berkeley

Results Summary

- Stealth modification
  - Preservation guaranteed for up to 35% subversion
  - Beyond 35%, preservation increasingly probable
  - Access failure minimal for up to 35% subversion
- Attrition
  - Worst case access failure due to pipe stoppage (still less than stealth modification)
  - Request flooding, brute-force, etc., easier to resist

Status

- Production version for academic publishing
  - In use at over 100 libraries worldwide
  - Publishers of over 2000 titles on board
  - Many otherwise-doomed humanities titles
  - Presented protocol to be deployed by end of 2004
  - Development team at Stanford Libraries
  - Funding from NSF, Mellon Foundation
- Separate installations
  - Internally at NY Public Library (9/11 archives)
  - Chinese Academy of Sciences
  - Others brewing
- http://www.lockss.org/

US Participants

Amherst College, Bloomsburg University of Pennsylvania, Brigham Young University, Carnegie Mellon University, Case Western Reserve University, Columbia University, Cornell University, Creighton University Health Sciences Library, Dartmouth College, Emory University, Florida Center for Library Automation, Florida State University, George Mason University, Georgetown University, Georgia Institute of Technology, Georgia State University, Harvard University, Harvard University (Countway Library), Indiana University, Iowa State University, Johns Hopkins University, Library of Congress, Los Alamos National Laboratory, Michigan State University, MOBIUS (Missouri) Consortium, National Agricultural Library, New York Public Library, New York University, North Carolina State University, Ohio State University, Pennsylvania State University, Portland State University, Princeton University, Purdue University, Saint Anselm College, Stanford University, The College of William and Mary, University of California Berkeley, University of Connecticut, University of Illinois at Chicago, University of Iowa, University of Kentucky, University of Maryland (HSHSL), University of Michigan, University of Minnesota, University of Nevada Reno, University of Oklahoma (Health Science Center), University of Pennsylvania, University of Texas (Health Science Center at San Antonio), University of Texas Austin, University of Utah, University of Virginia, University of Washington, University of Wisconsin-Madison, University of Tennessee, Vanderbilt University, Virginia Tech, Wellesley, Wesleyan University, Yale University

Participants

Amherst College, Bloomsburg University of Pennsylvania, Brigham Young University, Carnegie Mellon University, Case Western Reserve University, Columbia University, Cornell University, Creighton University Health Sciences Library, Dartmouth College, Duke University?, Emory University, Florida Center for Library Automation, Florida State University, George Mason University, Georgetown University, Georgia Institute of Technology, Georgia State University, Harvard University, Harvard University (Countway Library), Indiana University, Iowa State University, Johns Hopkins University, Library of Congress, Massachusetts Institute of Technology, Michigan State University, MOBIUS (Missouri), National Agricultural Library, New York Public Library, New York University, Ohio State University, Pennsylvania State University, Princeton University, Purdue University, Rice University, The College of William and Mary, University of Connecticut, University of Illinois at Chicago, University of Kentucky, University of Maryland (HSHSL), University of Minnesota, University of Nevada Reno, University of Oklahoma (Health Sciences Center), University of Pennsylvania, University of Texas Health Science Center at San Antonio), University of Texas Austin, University of Utah, University of Virginia, University of Washington, University of Wisconsin-Madison, University of Tennessee, Vanderbilt University, Virginia Tech, Wellesley, Wesleyan University, Yale University

You are welcome to join us!

Next Steps

- Analysis
  - Hybrid attacks
    - e.g., stealth modification during DDoS weakening
  - Evaluation and mitigation of spoofing
  - Formal treatment of small portions of protocol
    - minimization of adversary influence
- More applications
  - Distributed Internet archive
  - Government documents (GPO)
  - Federal depository library system (copyrighted materials)
  - Scientific datasets
    - NASA simulations
    - network tracefiles
- Put a replica at the Duke University Libraries?

Conclusions

- LOCKSS P2P digital preservation system
- Opinion polls to audit replicated content
- On-going research and practice
  - Rate limiting
  - Effort balancing
  - Bimodal alarm behavior
  - Friend bias
  - State unpredictability
- Promising results
  - Resistant to attacks for low adversary strength
  - Degrades gracefully for stronger adversaries
- Thank you!

Criticism Prefetch

1. How generally applicable is any of this?
    1. Who cares about libraries anyway?
    2. Isn't preservation a special type of application?
    3. Isn't RAID the solution to it all?
2. What about other attacks?
    1. For stealth adversaries?
    2. For attrition adversaries?
    3. For combined adversaries?
    4. Where's the proof of security?
3. Haven't we heard all of this before?
    1. Hasn't X already talked about rate limiting, admission control, desynchronization, client puzzles, caching, indirection, etc.?
4. What about
    1. Cosmic rays to slow down CPUs
    2. Tsunami waves for network partitioning via flooding attacks
    3. Raise global inflation to make maintenance expensive
    4. Misspelling the word "locks"

Popular Yet Flawed Alternatives

- Use super-fabulous RAID
  - Can be complementary, but alone cannot ensure survivability when failures do occur (e.g., human error)

Scaling

Backup

Intel Research Berkeley

Scaling Considerations

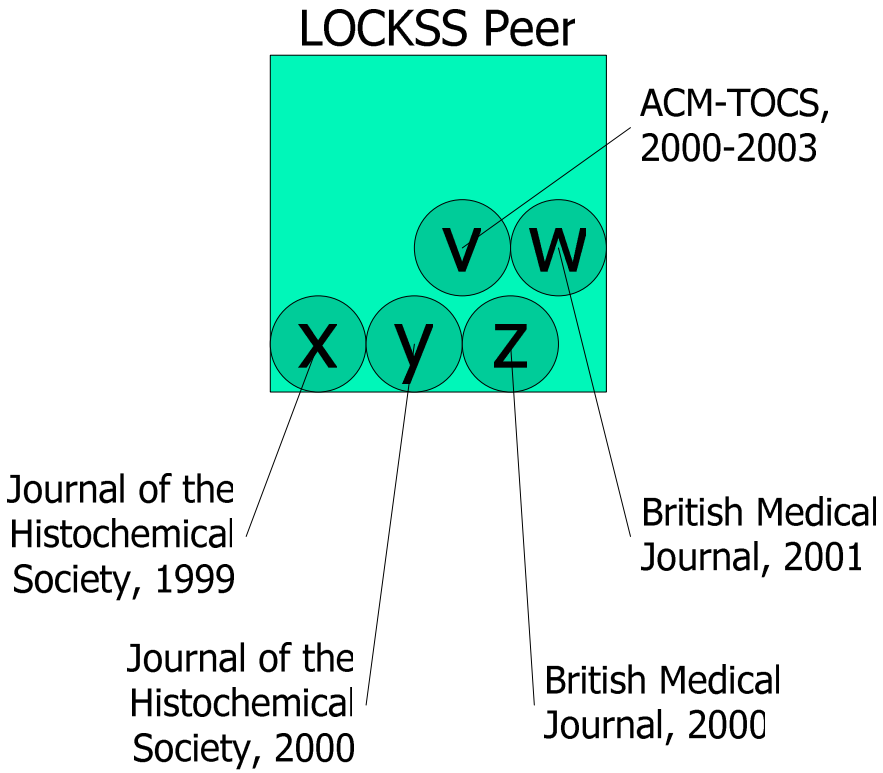
- Large libraries have about 5000 electronic titles, most with short history
- Median title/year size is 300 MBytes, max is 2.5 GBytes
- For storage and polls, each title/year costs no more than $5/year
  - Only a drop in the ocean of susbscriptions
- Rack of ~10 PCs can cover large collections

Detail on Opinion Poll Protocol

Backup

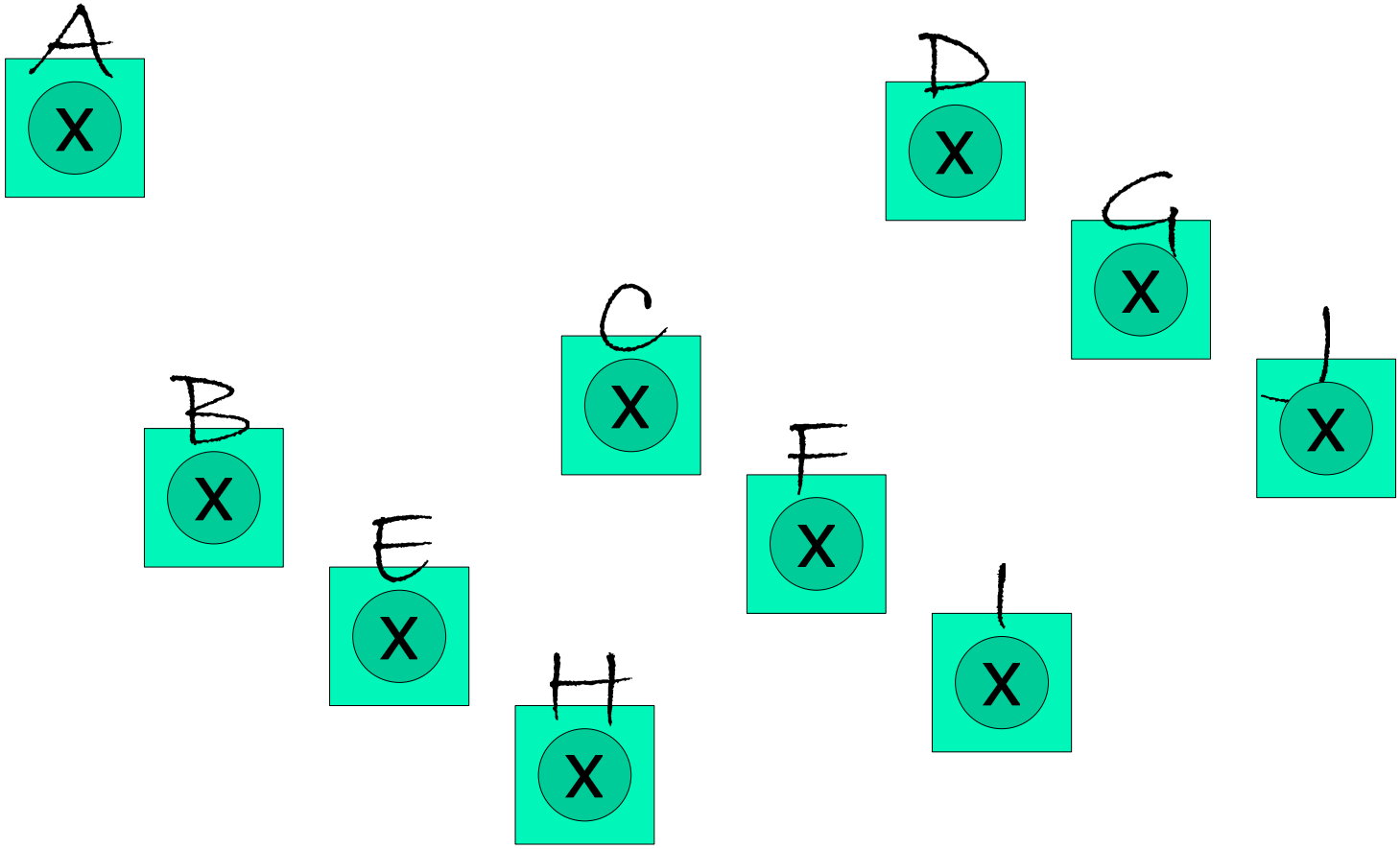
Intel Research Berkeley

LOCKSS Overview



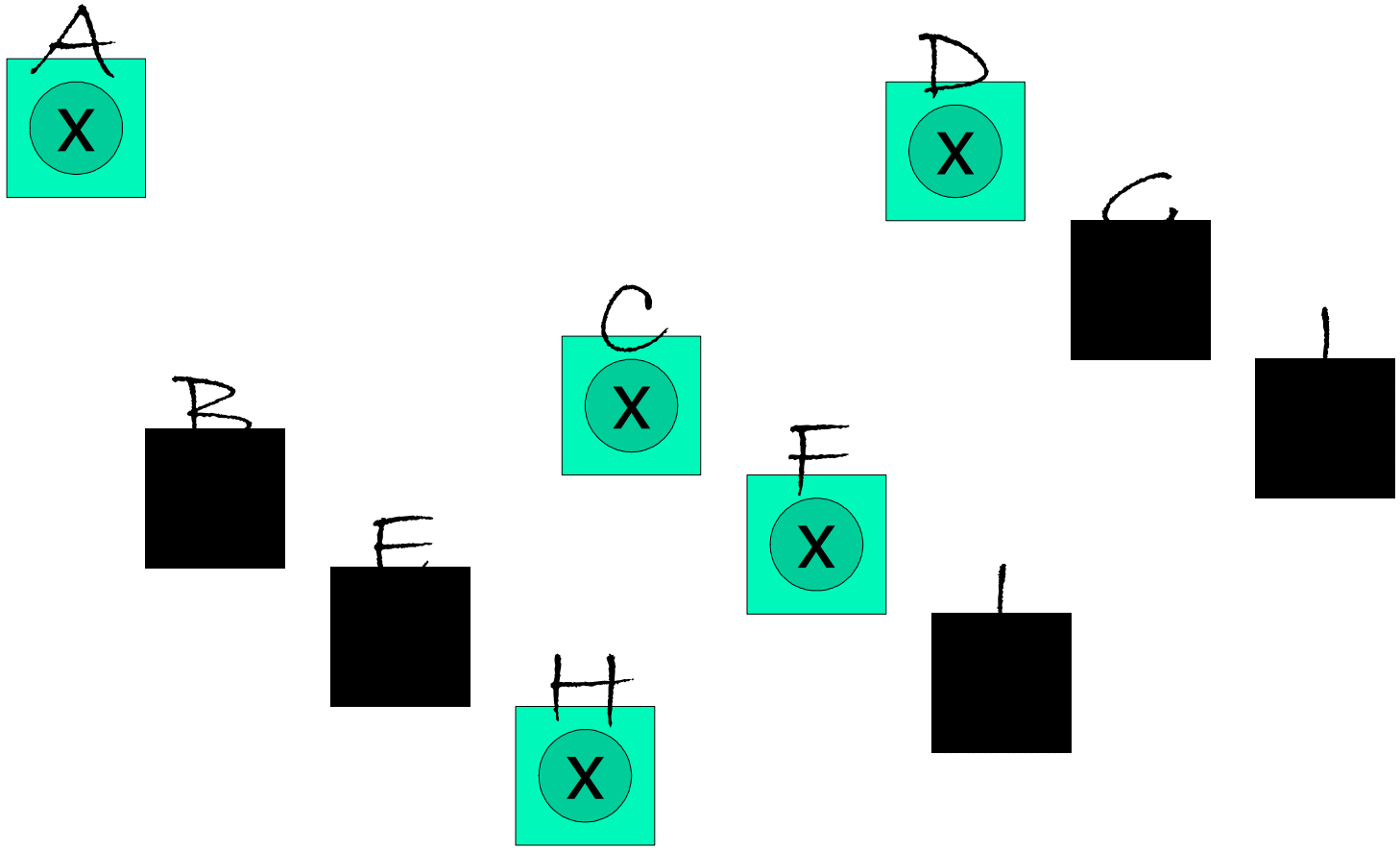
Intel Research Berkeley

LOCKSS Overview



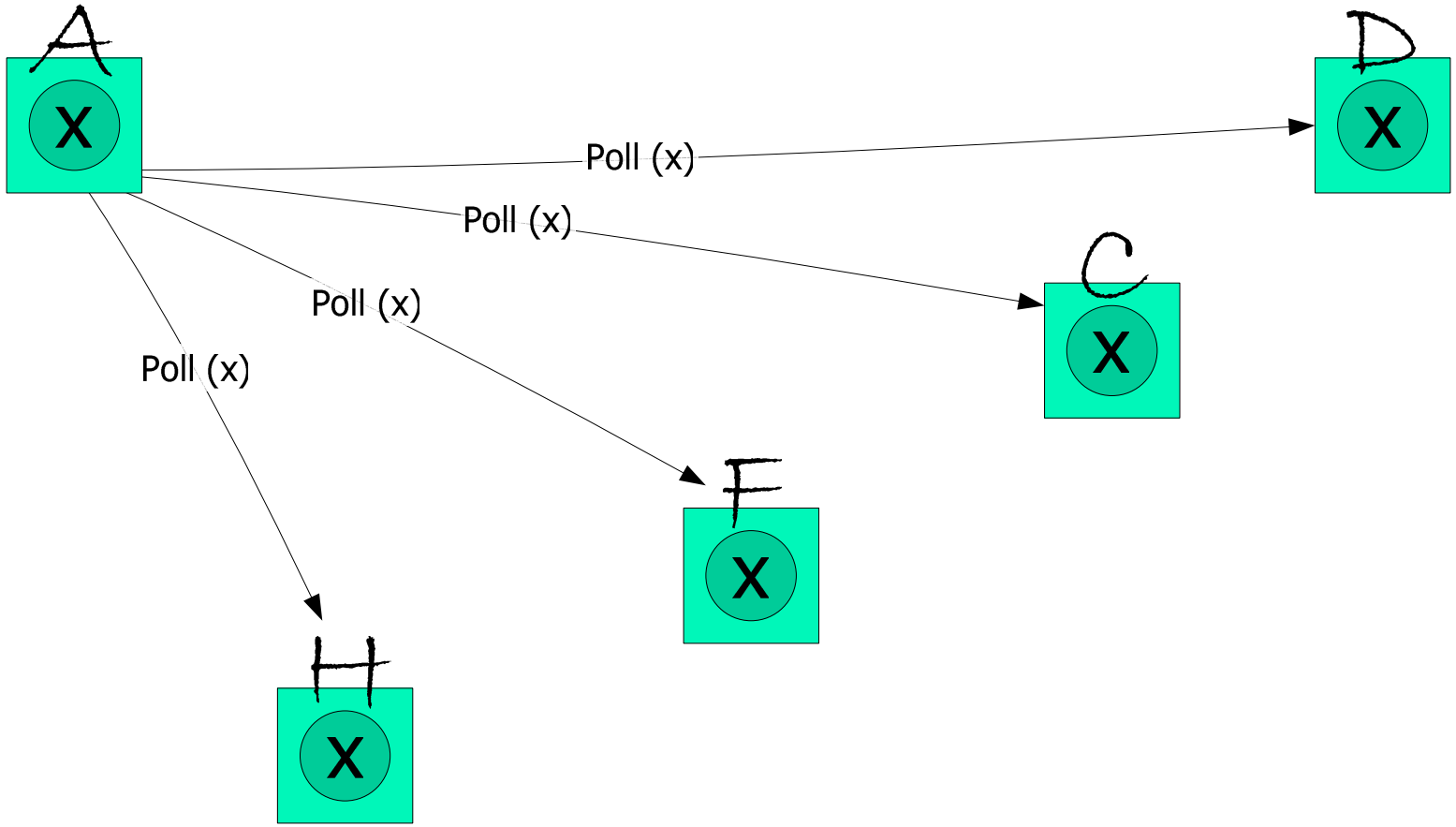
Intel Research Berkeley

LOCKSS Overview



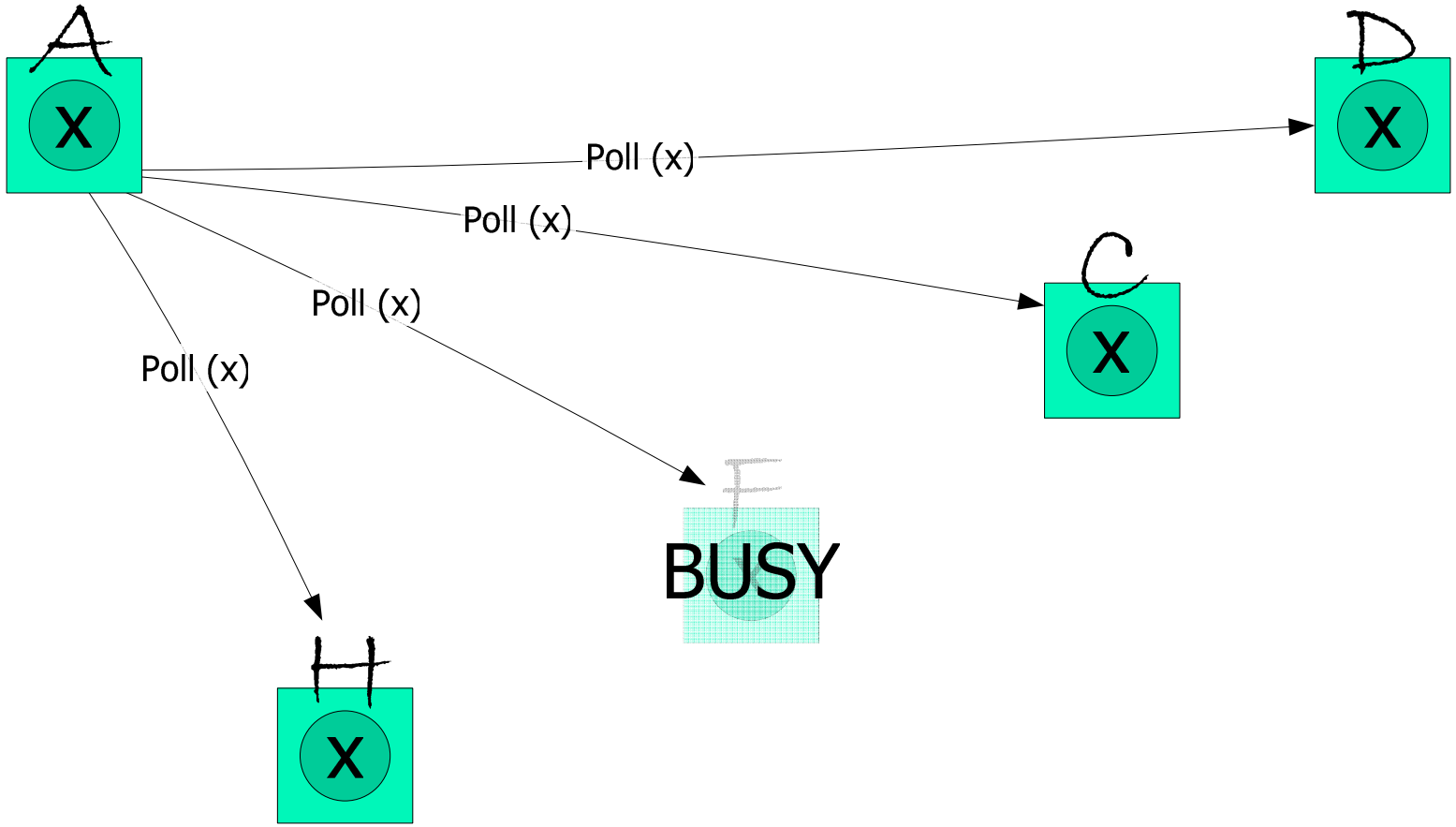
Intel Research Berkeley

LOCKSS Overview



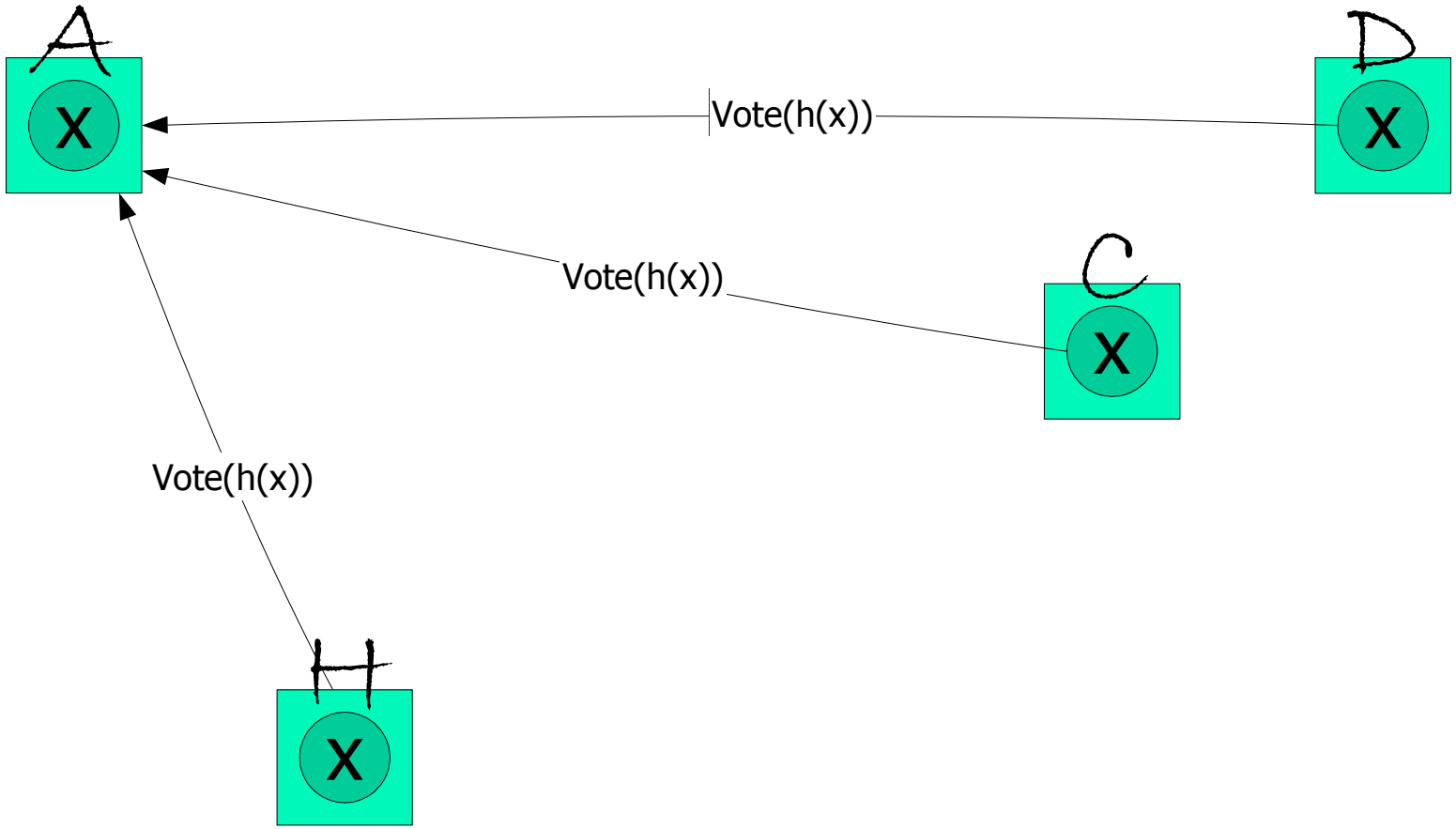
Intel Research Berkeley

LOCKSS Overview



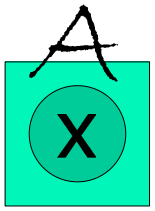
Intel Research Berkeley

LOCKSS Overview



Intel Research Berkeley

LOCKSS Overview



vC =? h(x)

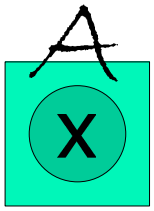
vD

vH

v...

Intel Research Berkeley

LOCKSS Overview



vC = h(x) ✓

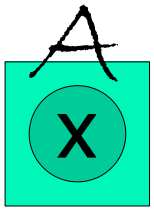
vD

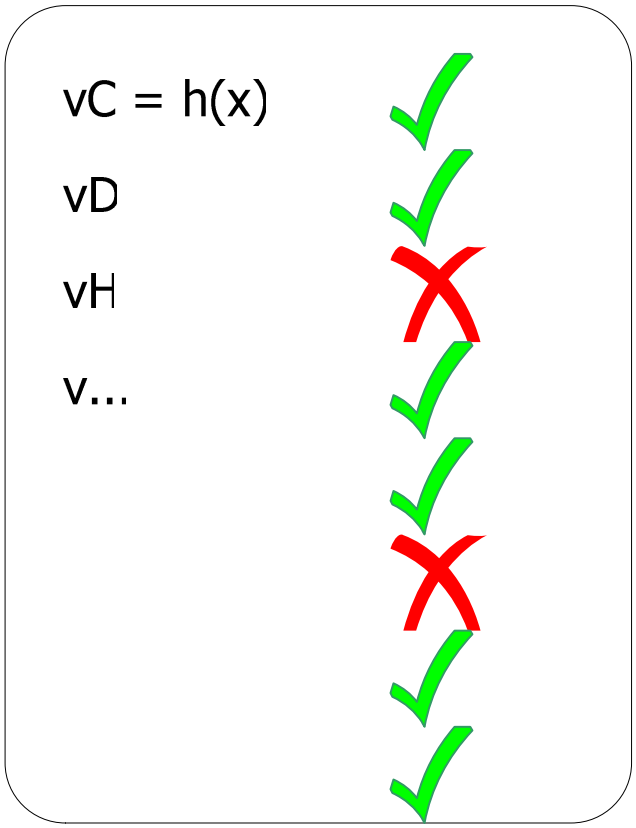
vH

v...

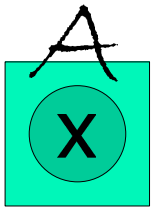
Intel Research Berkeley

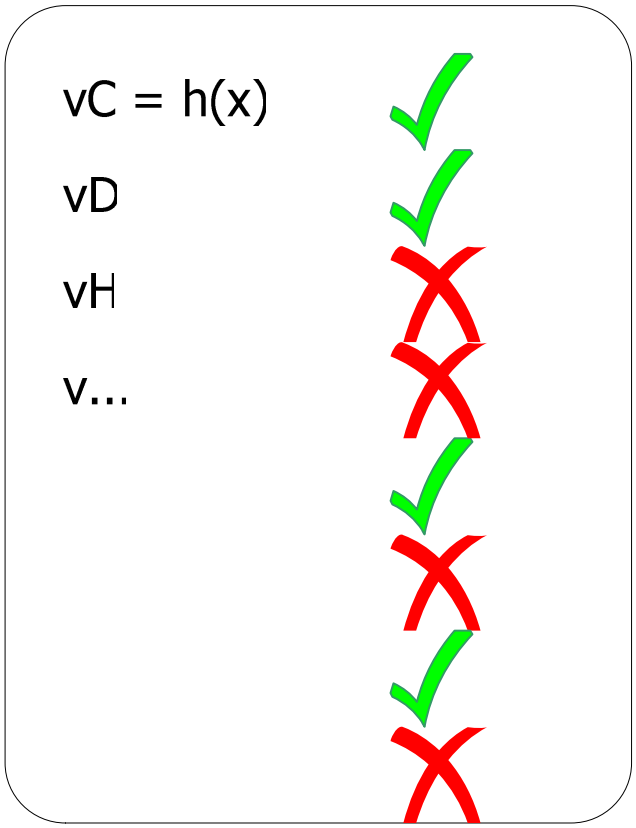
LOCKSS Overview



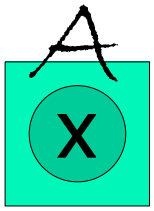


LOCKSS Overview





LOCKSS Overview



vC = h(x)

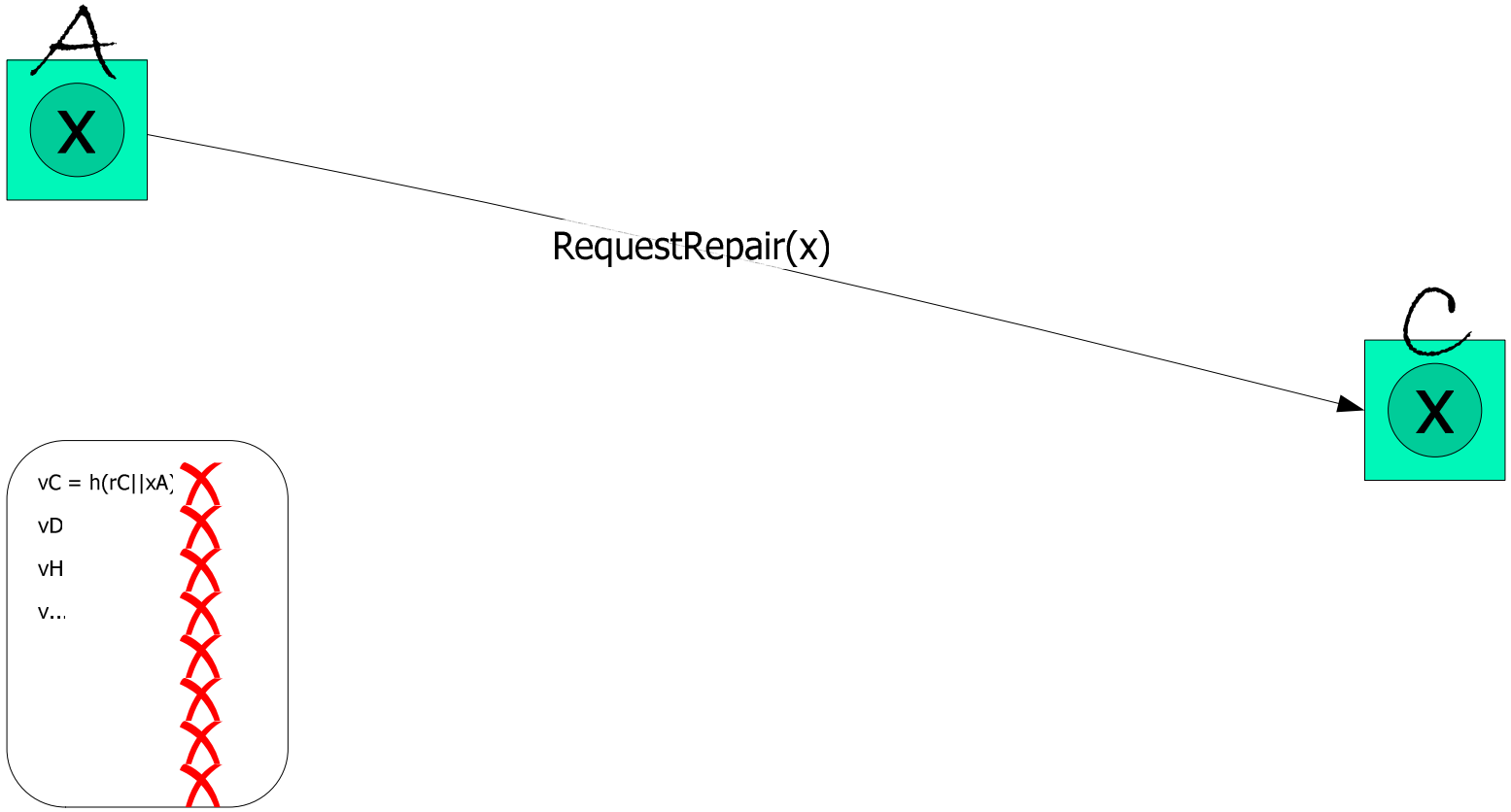
vD

vH

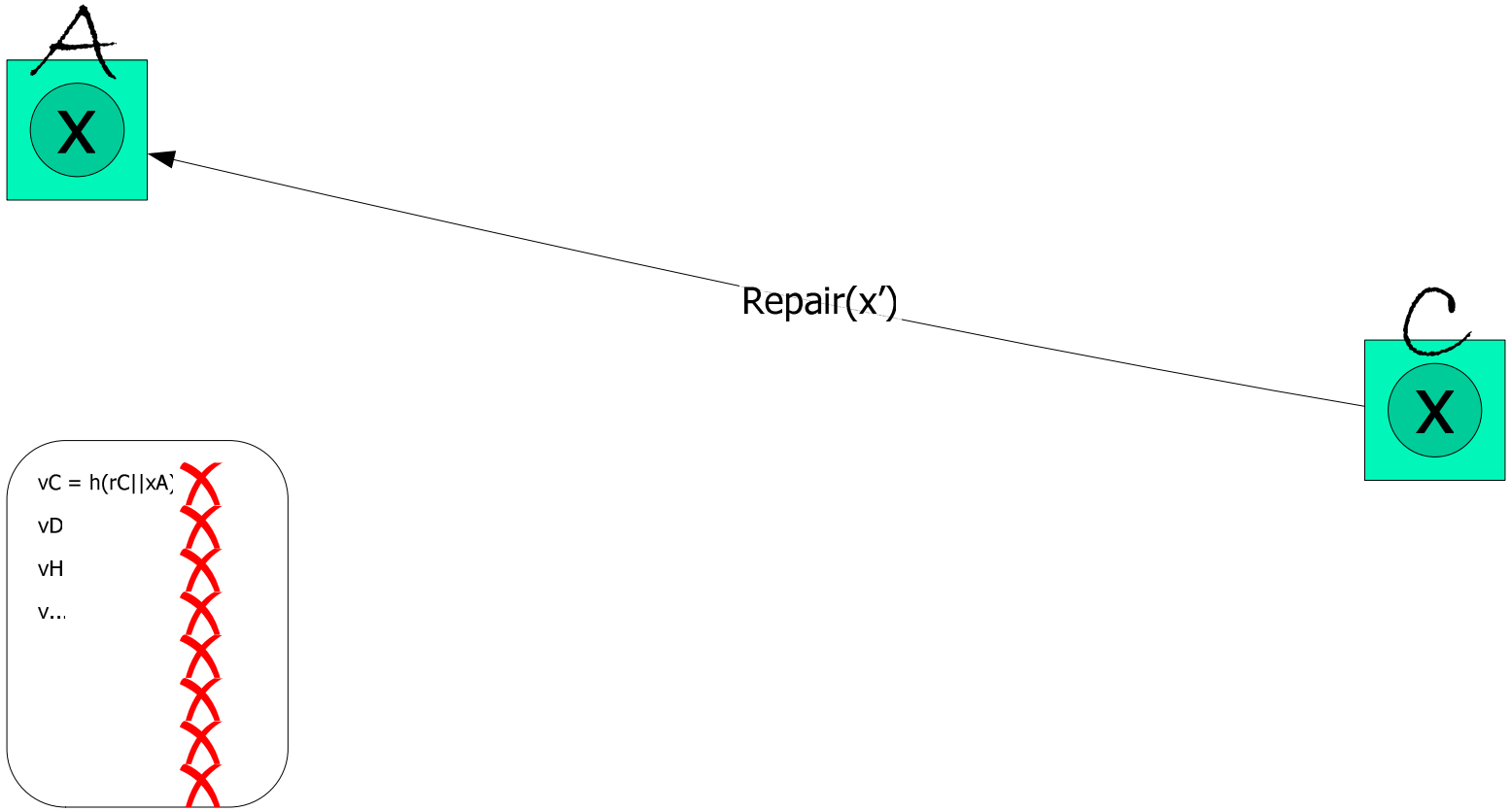
v...

Intel Research Berkeley

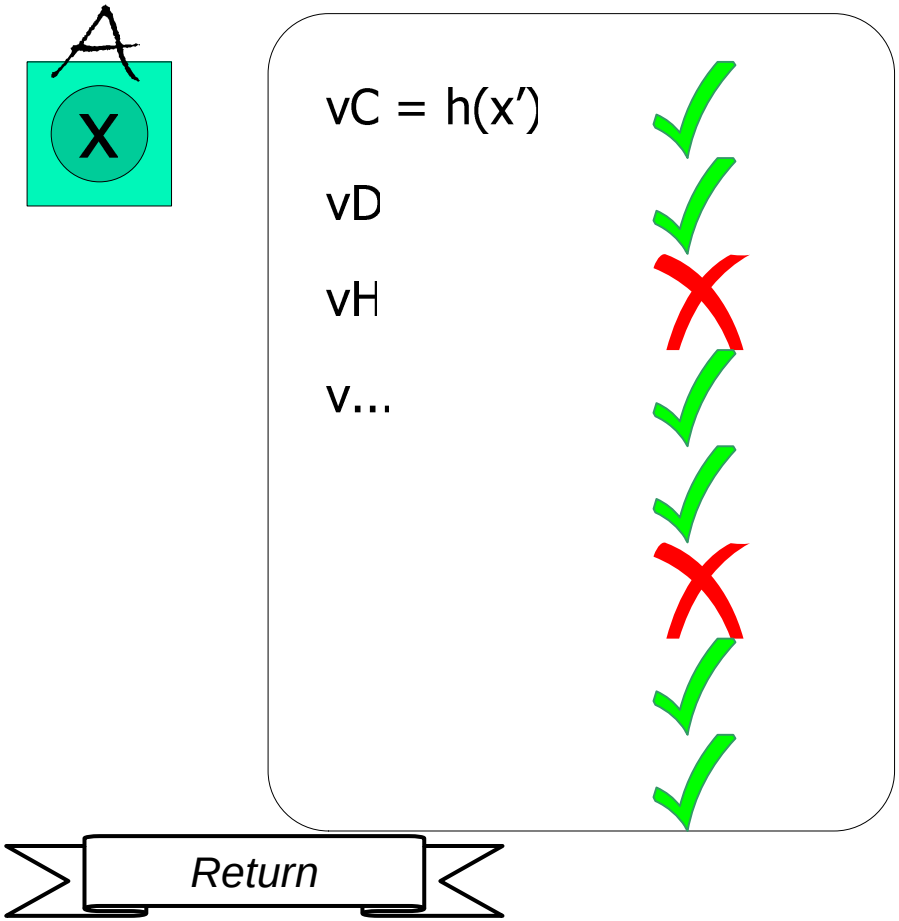
LOCKSS Overview



LOCKSS Overview



LOCKSS Overview



Intel Research Berkeley

Discovery of Likely Voters

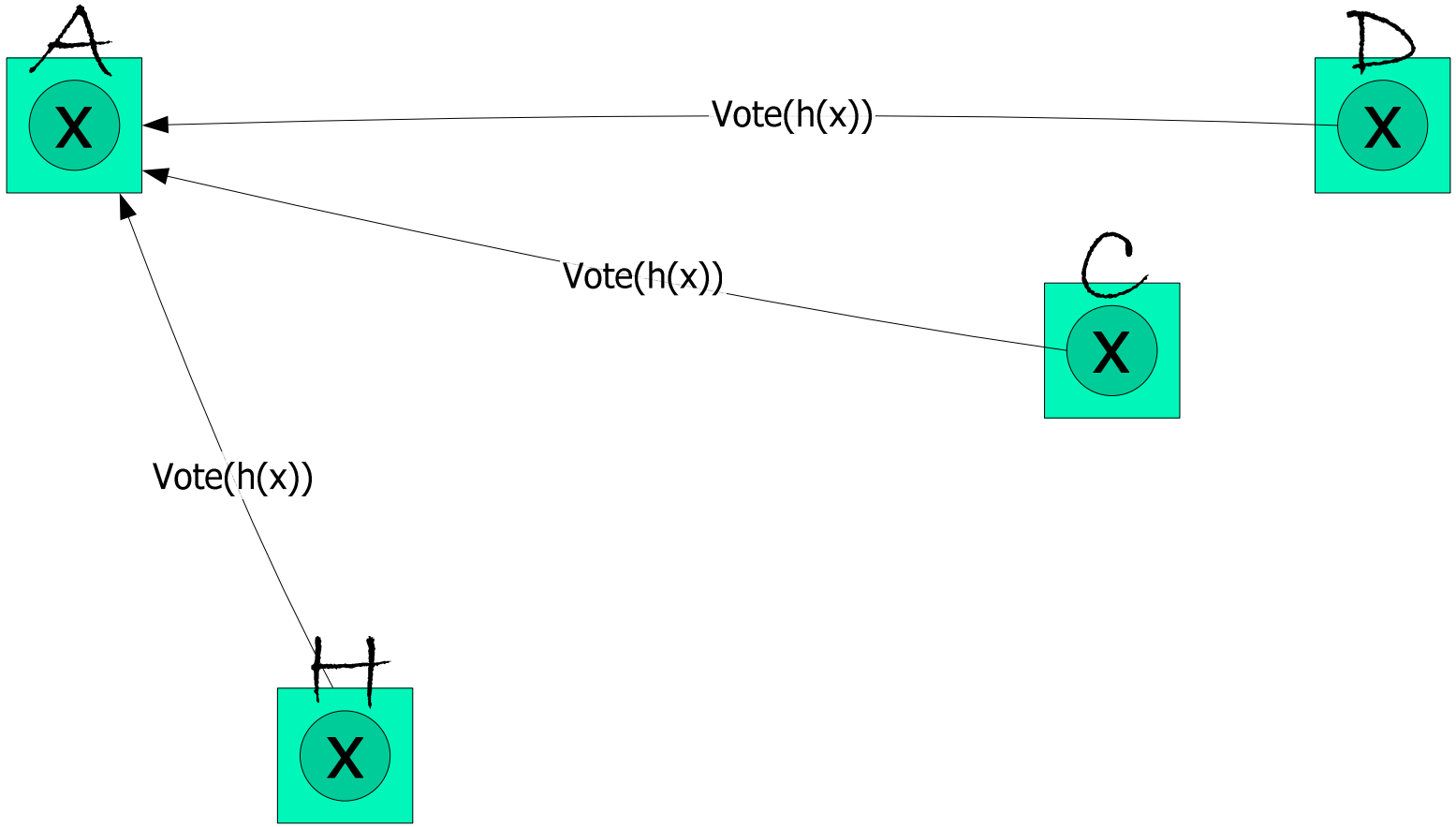
Backup

Intel Research Berkeley

Discovery

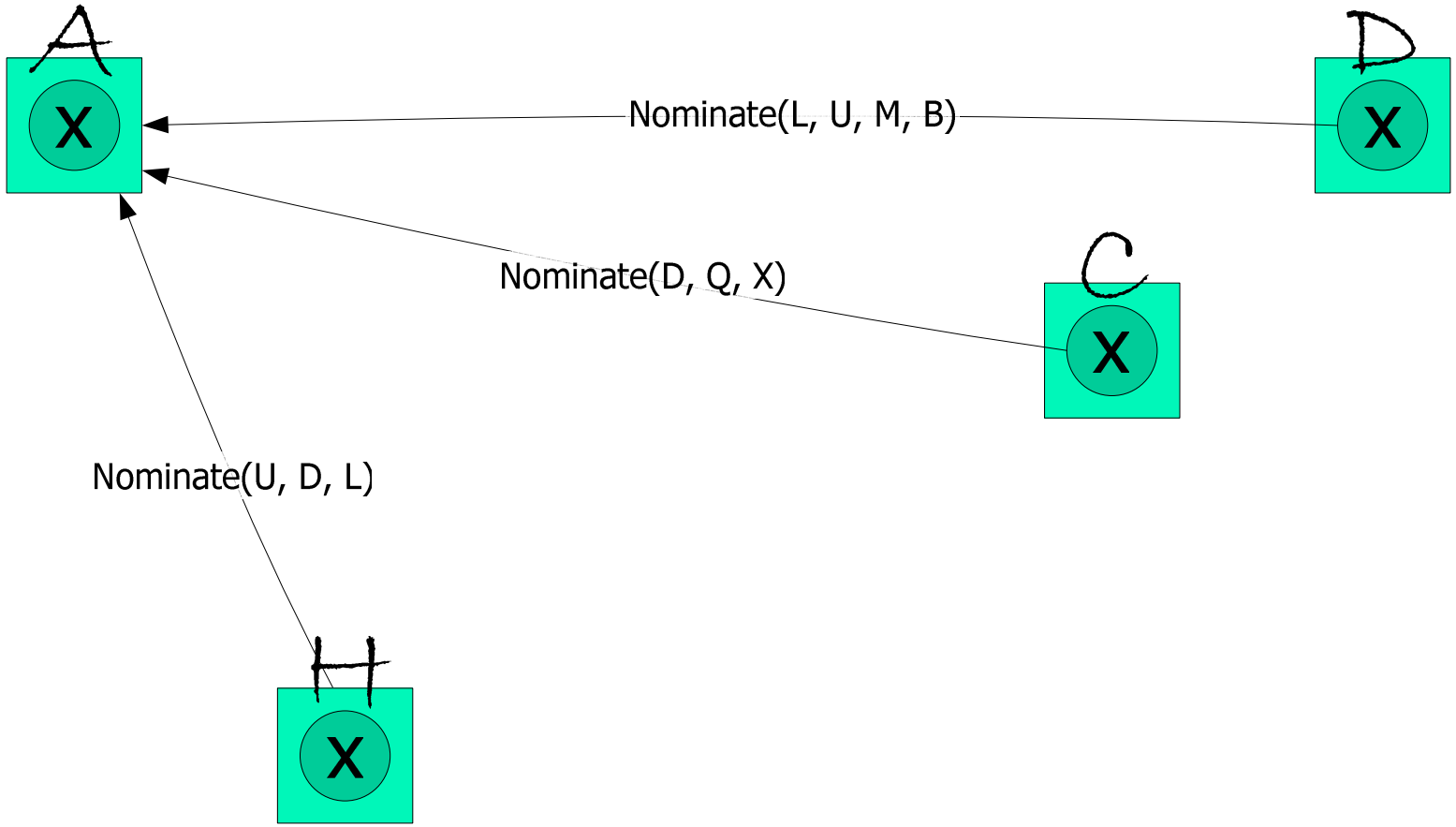
- Participants in a poll nominate peers for inclusion in the caller's reference list
  - They can pick nominees however they want
- Caller invites some nominees into the poll
  - Equal number from each nominator
  - Enough to keep the reference list populated
  - Nominees vote in same way as original invitees
- Nominee's vote is **only** used to determine acceptance
  - Nominee is accepted only if it votes with "correct" result
  - Nominee vote is **not** counted in vote tabulation

LOCKSS Overview



Intel Research Berkeley

LOCKSS Overview



Intel Research Berkeley

Admission Control

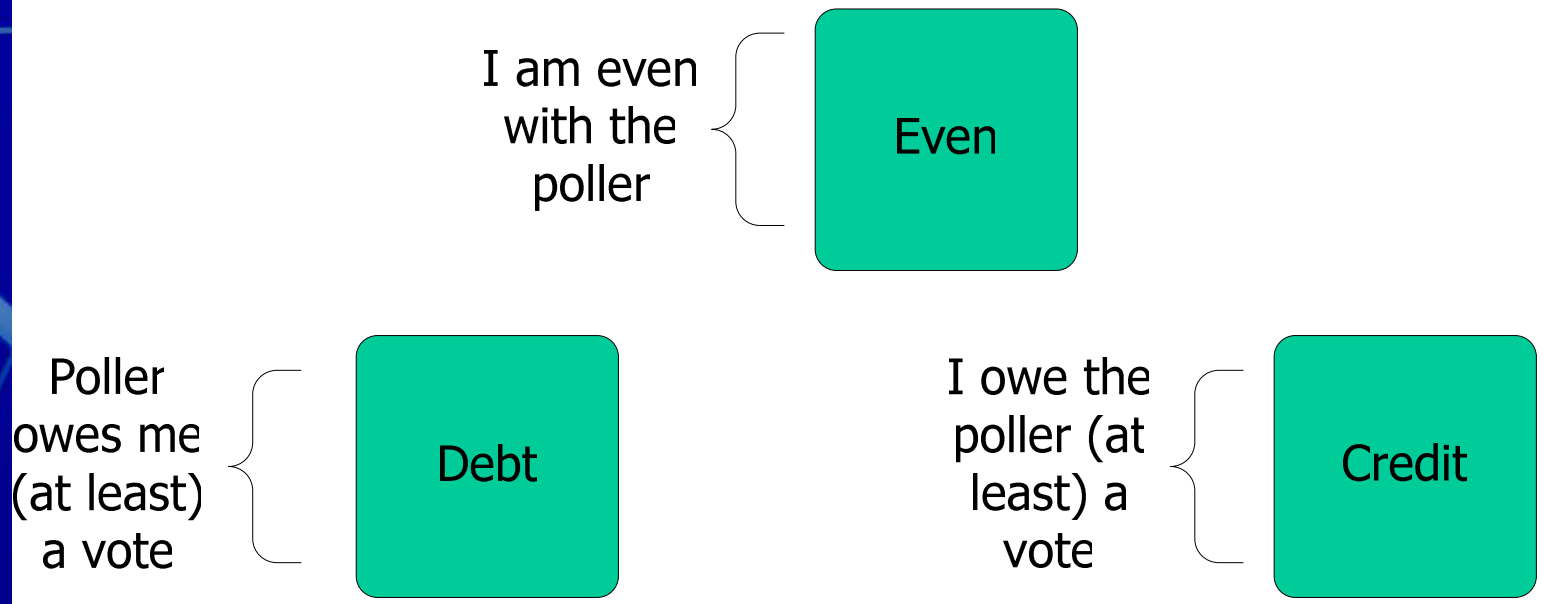
Backup

Intel Research Berkeley

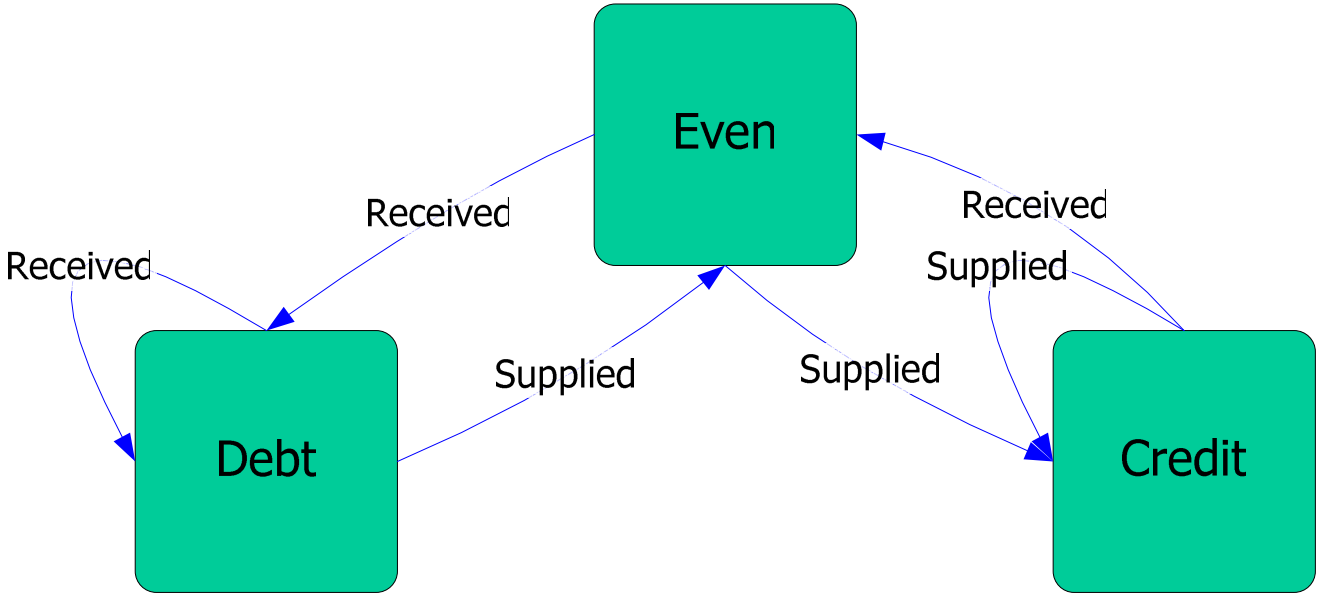
Protocol Flood Defenses

- Cap rate at which requests are handled
  - To limit overhead of request admission
- At lower rates, favor requests from peers who
  - Appear to be following the protocol and
  - Have a history in the system
  - To ensure that "good" requests are admitted
- Tools
  - "Subjective reputation" i.e., history
    - Keep track of my interactions with others
  - Self clocking
    - Favor peers who operate at my own rate of operation
  - Newcomer pays
    - New identities are penalized
  - Introductions
    - Penalty waived if newcomer introduced by a good peer

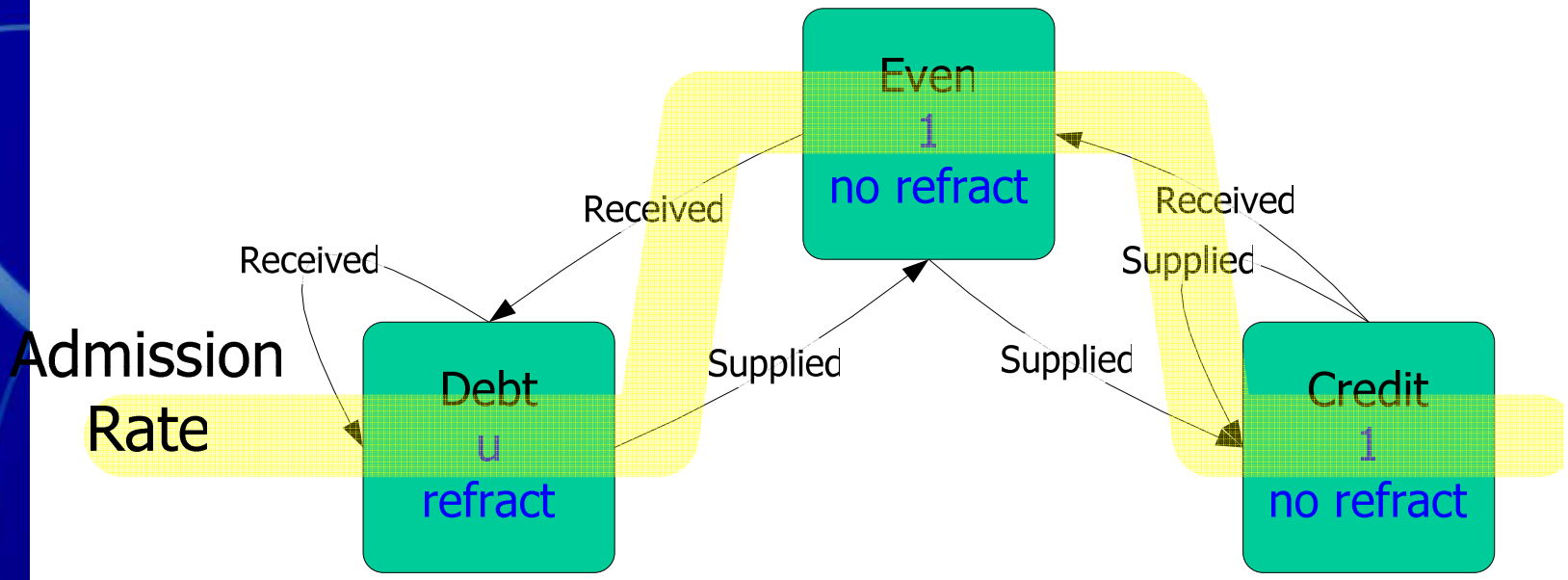
Admission Control



Admission Control

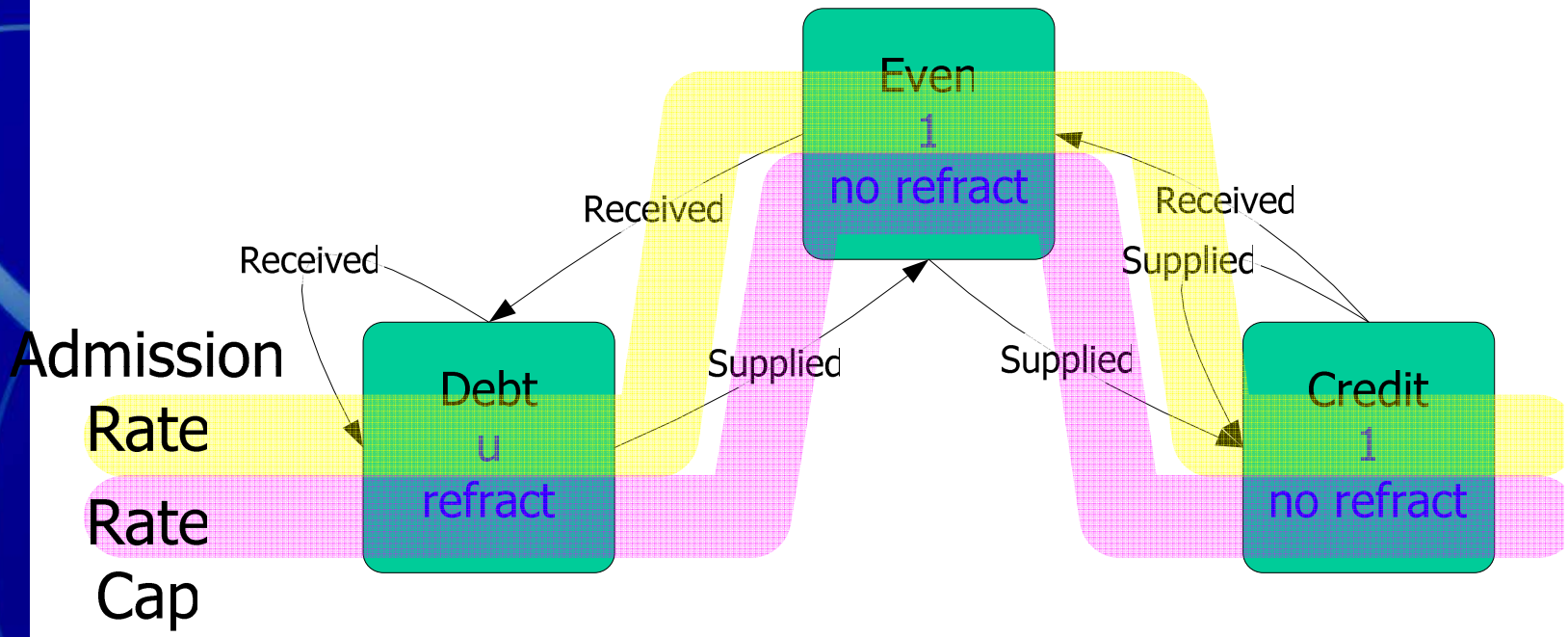


Admission Control



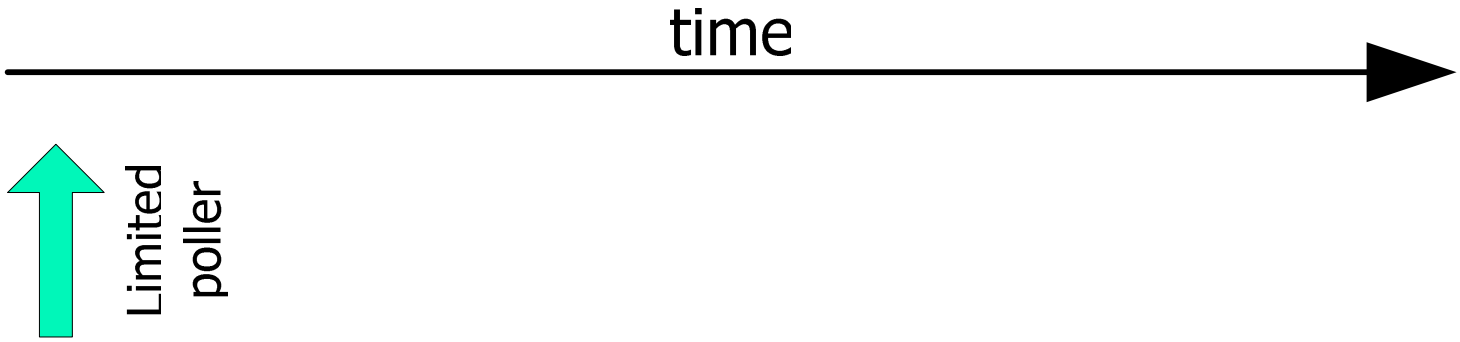
Intel Research Berkeley

Admission Control

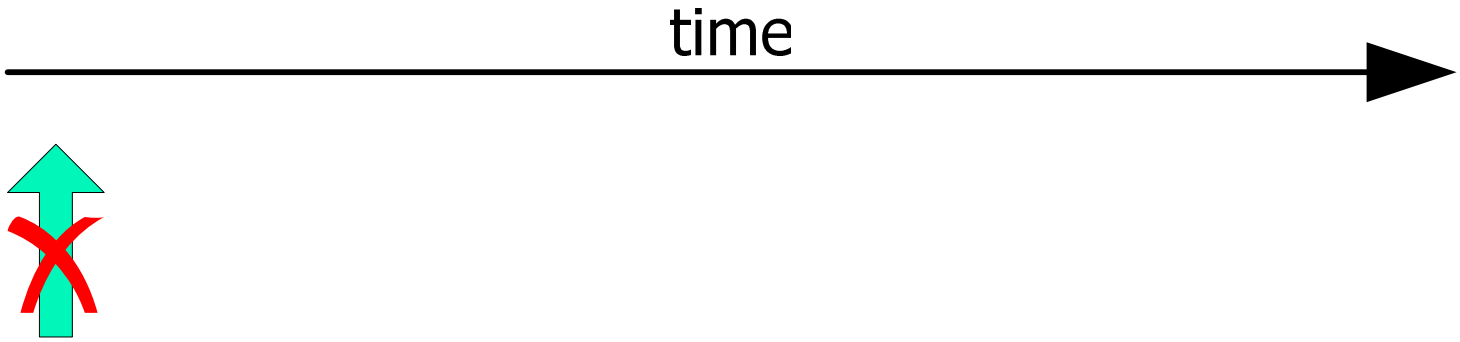


Intel Research Berkeley

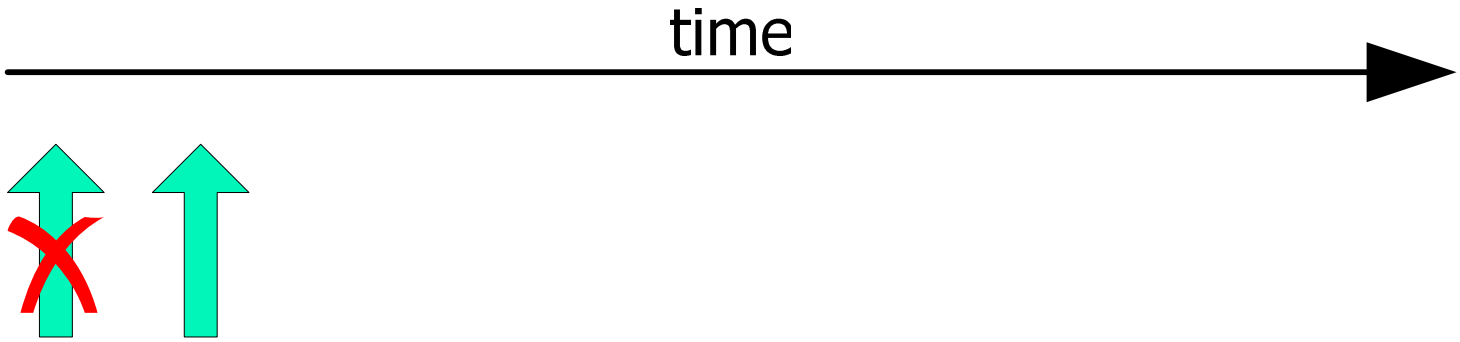
(Refractory Period)



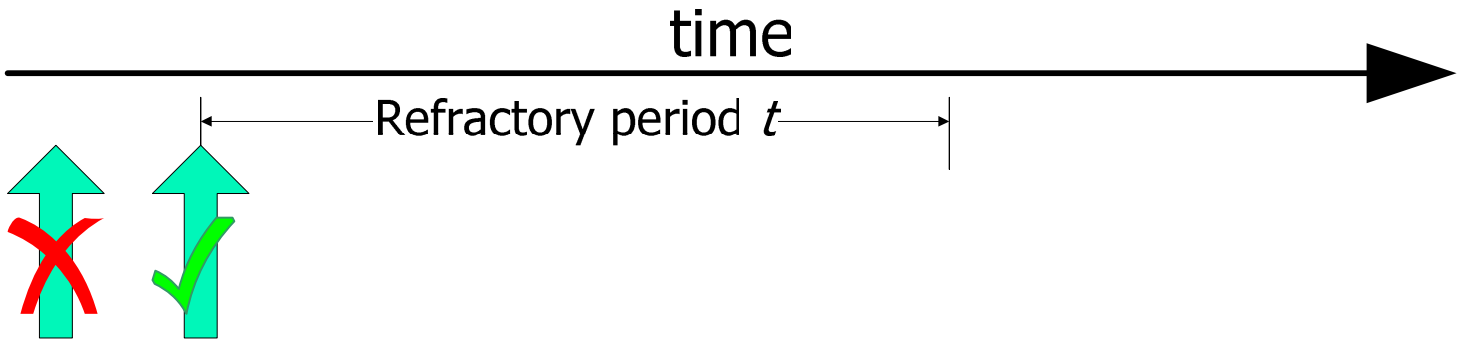
(Refractory Period)



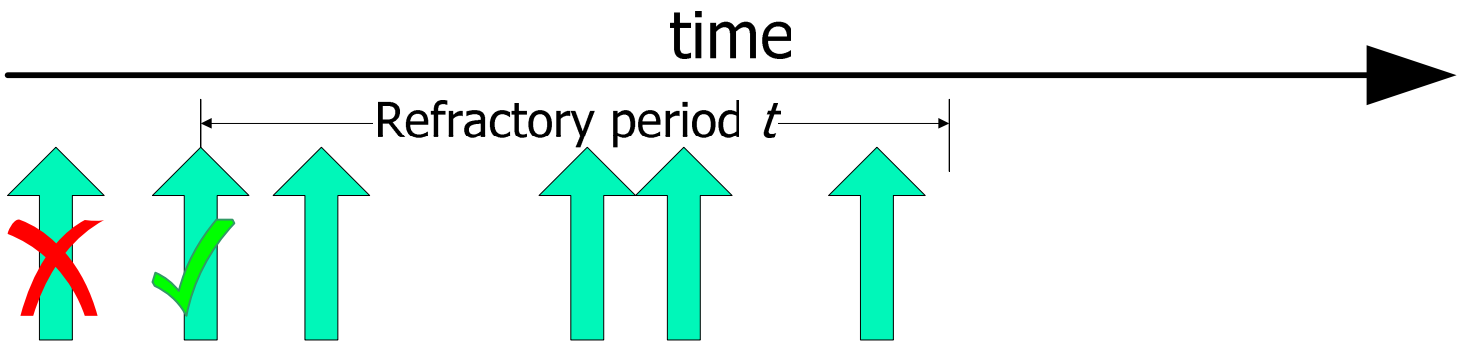
(Refractory Period)



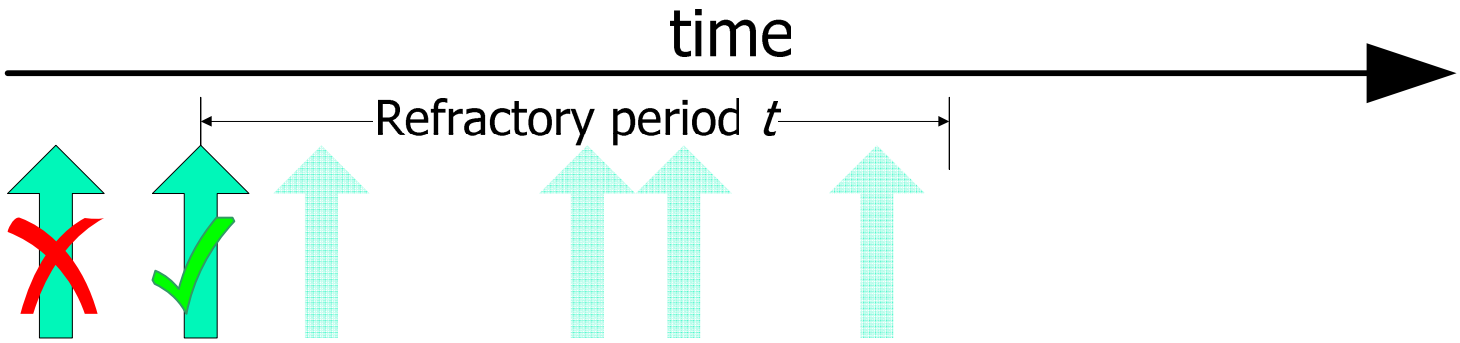
(Refractory Period)



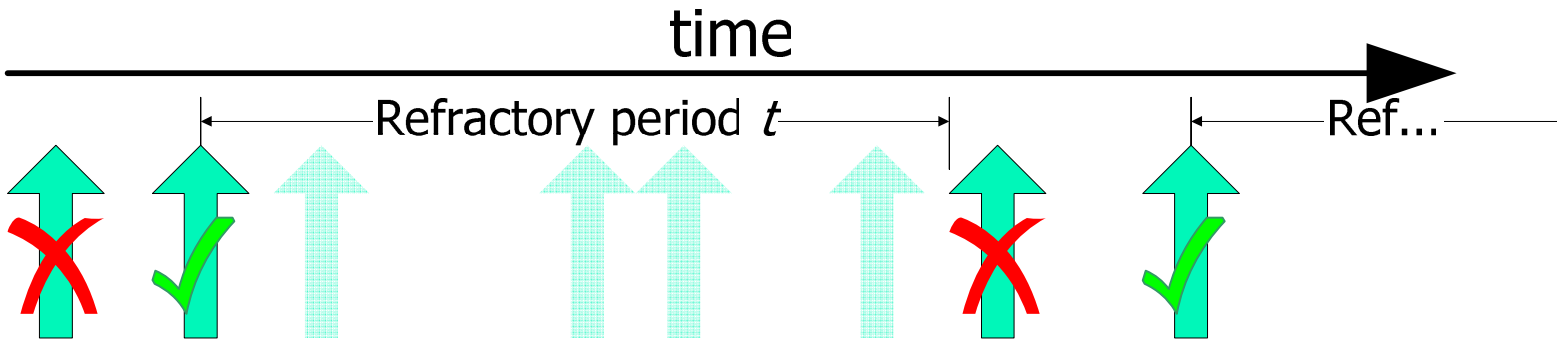
(Refractory Period)



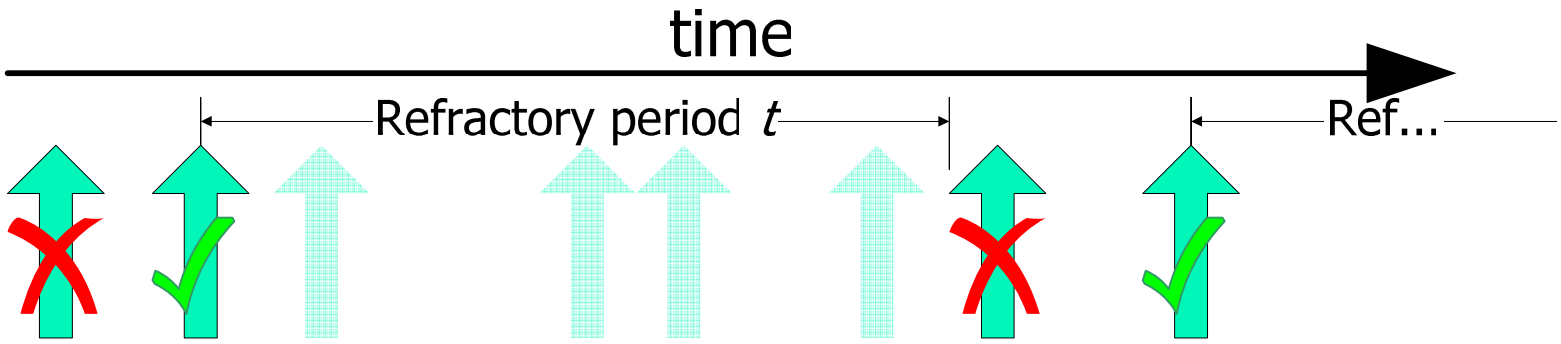
(Refractory Period)



(Refractory Period)

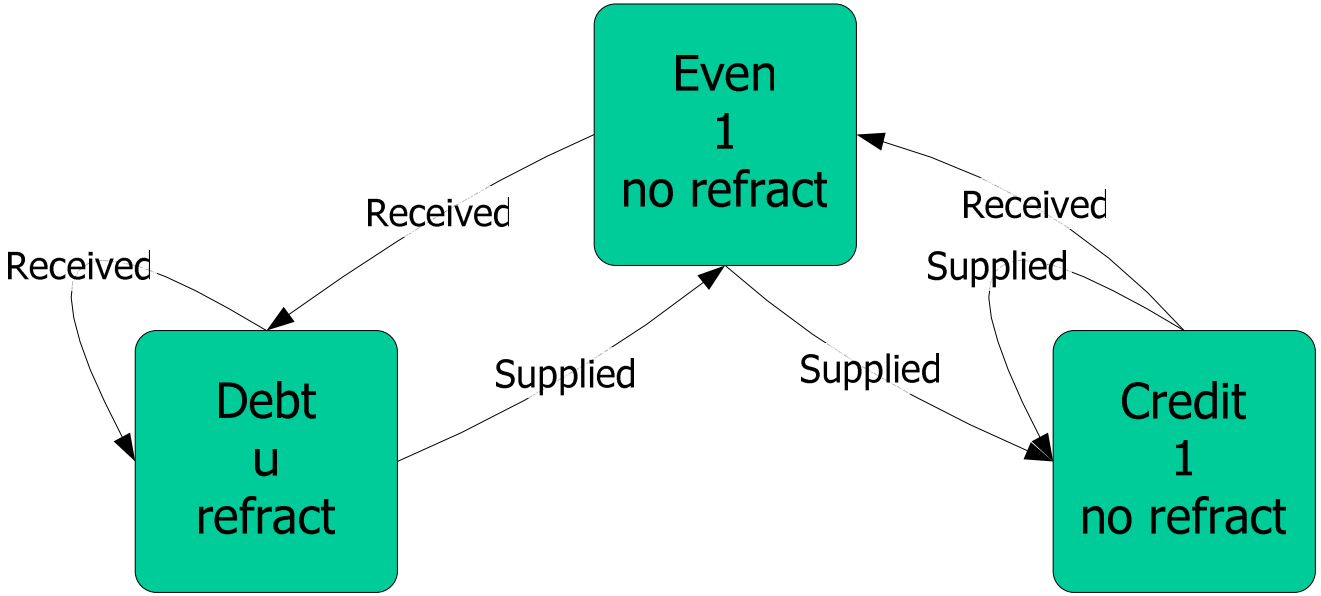


(Refractory Period)

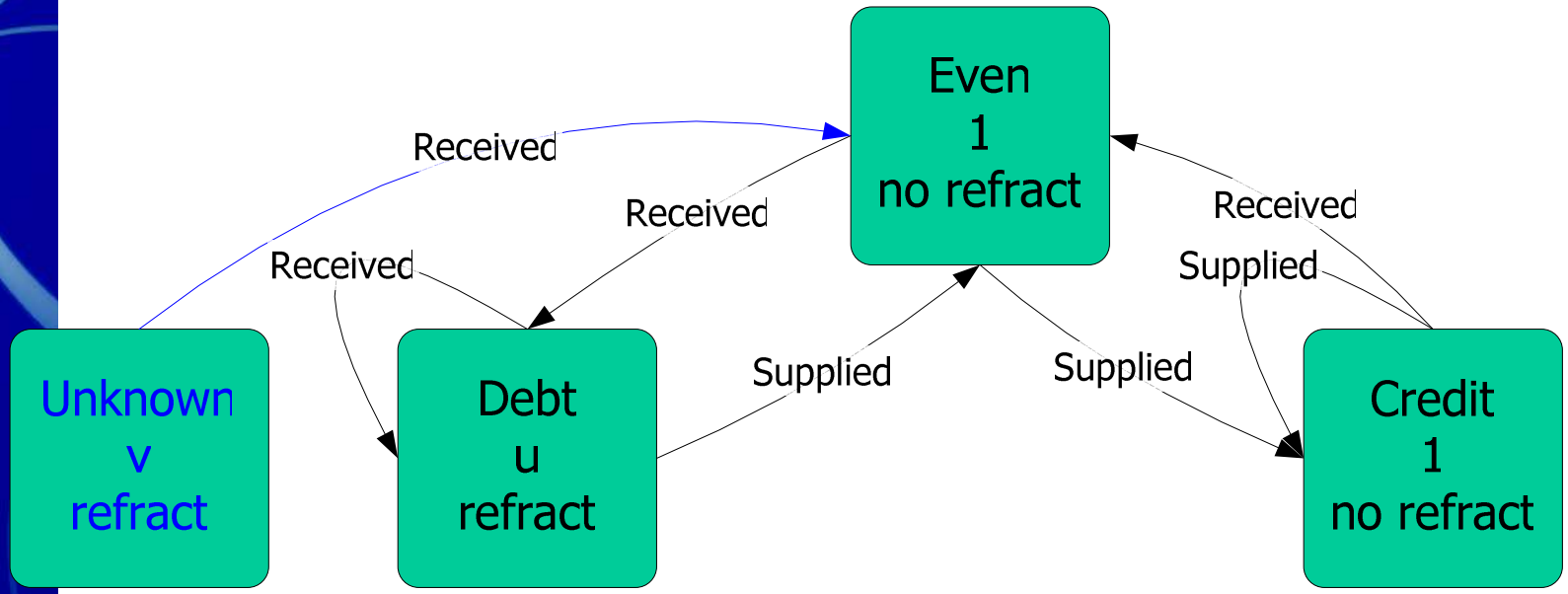


$$\frac{1}{t} = \text{Max request rate}$$

Admission Control



Admission Control



Intel Research Berkeley

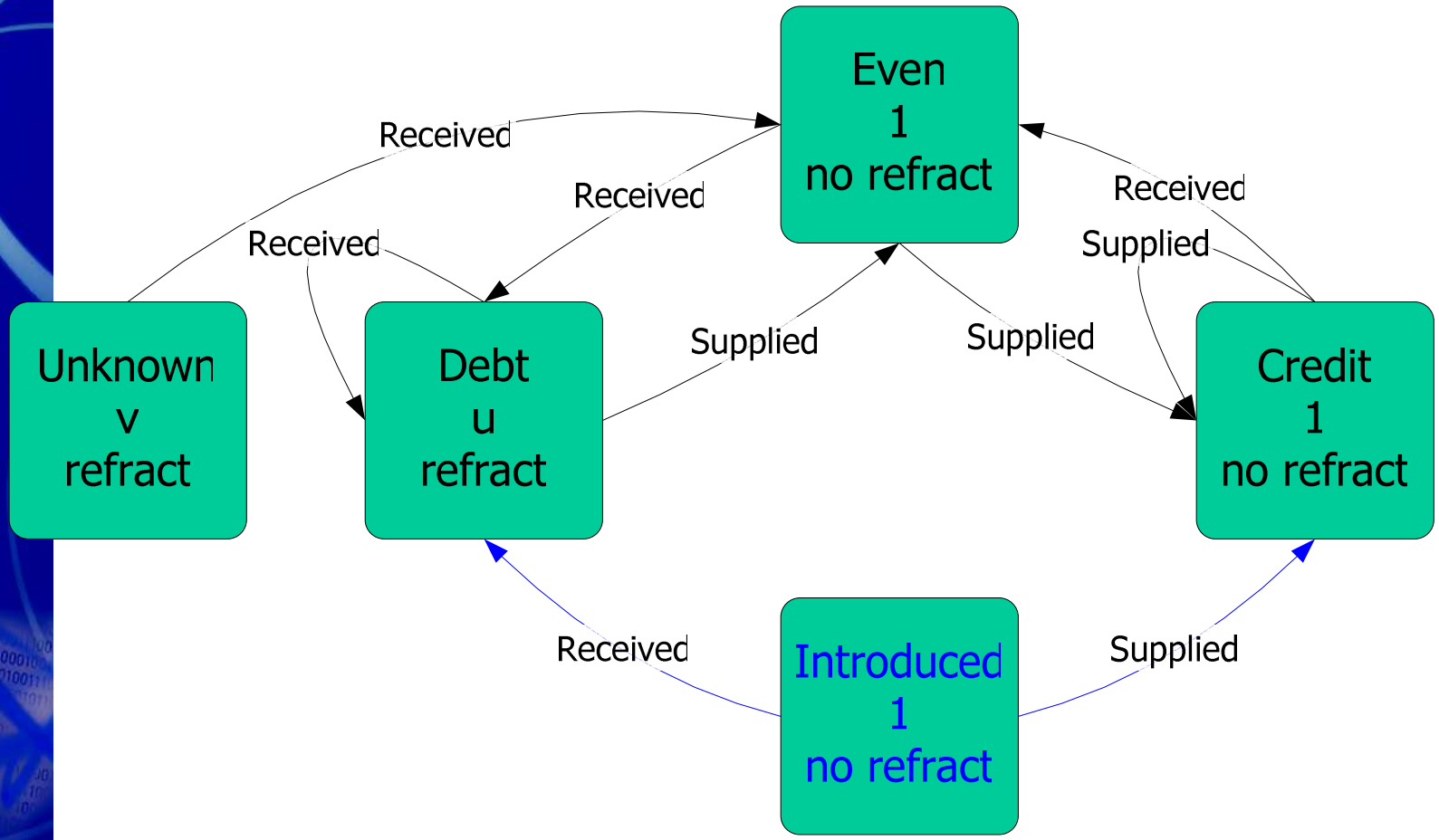
Properties so far

- Those who operate at my rate get a pleasant experience
- Those who operate more slowly are treated the same way
  - Great behavior + Bad behavior != Good behavior!
- Those who try to operate faster suffer drops until they slow down

But

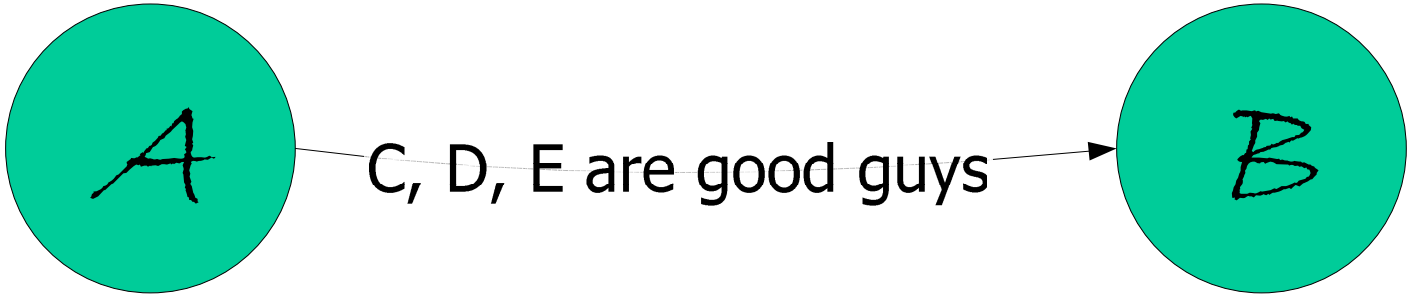
- Adversary can push me into refractory period
  - Effectively stopping discovery of new peers
  - What if who I think is a “newcomer” is a venerable, trusted peer?

Admission Control

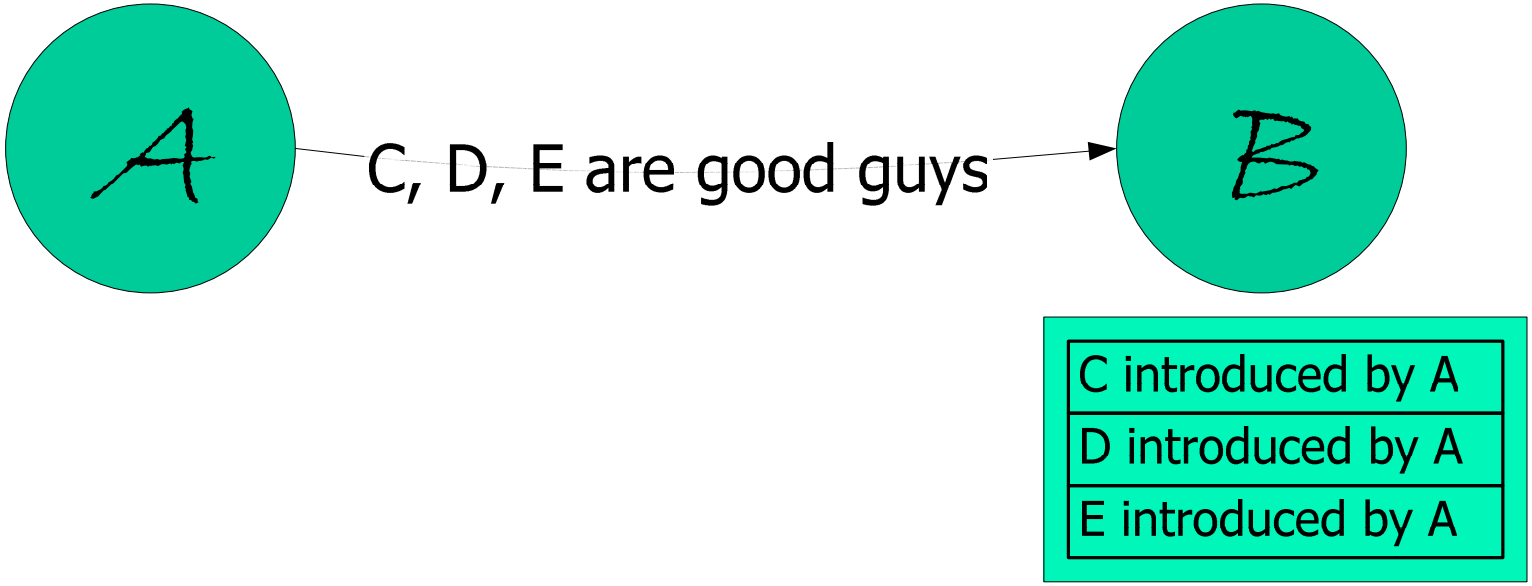


Intel Research Berkeley

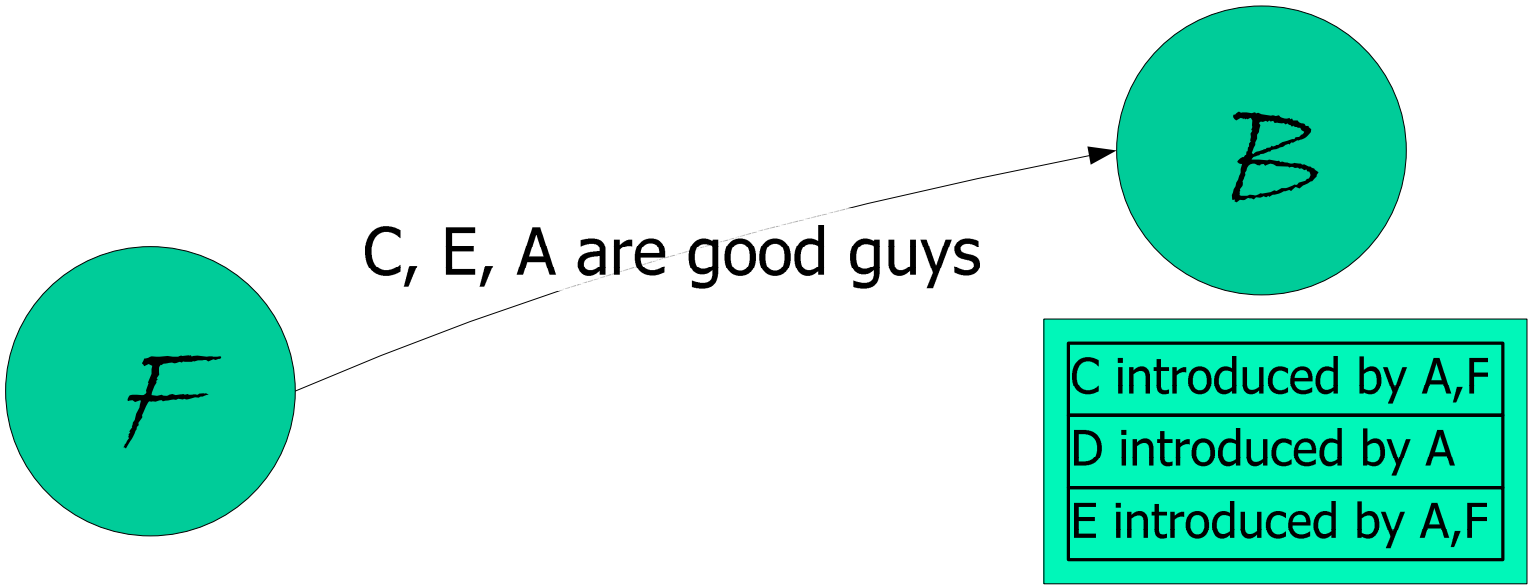
(Introductions)



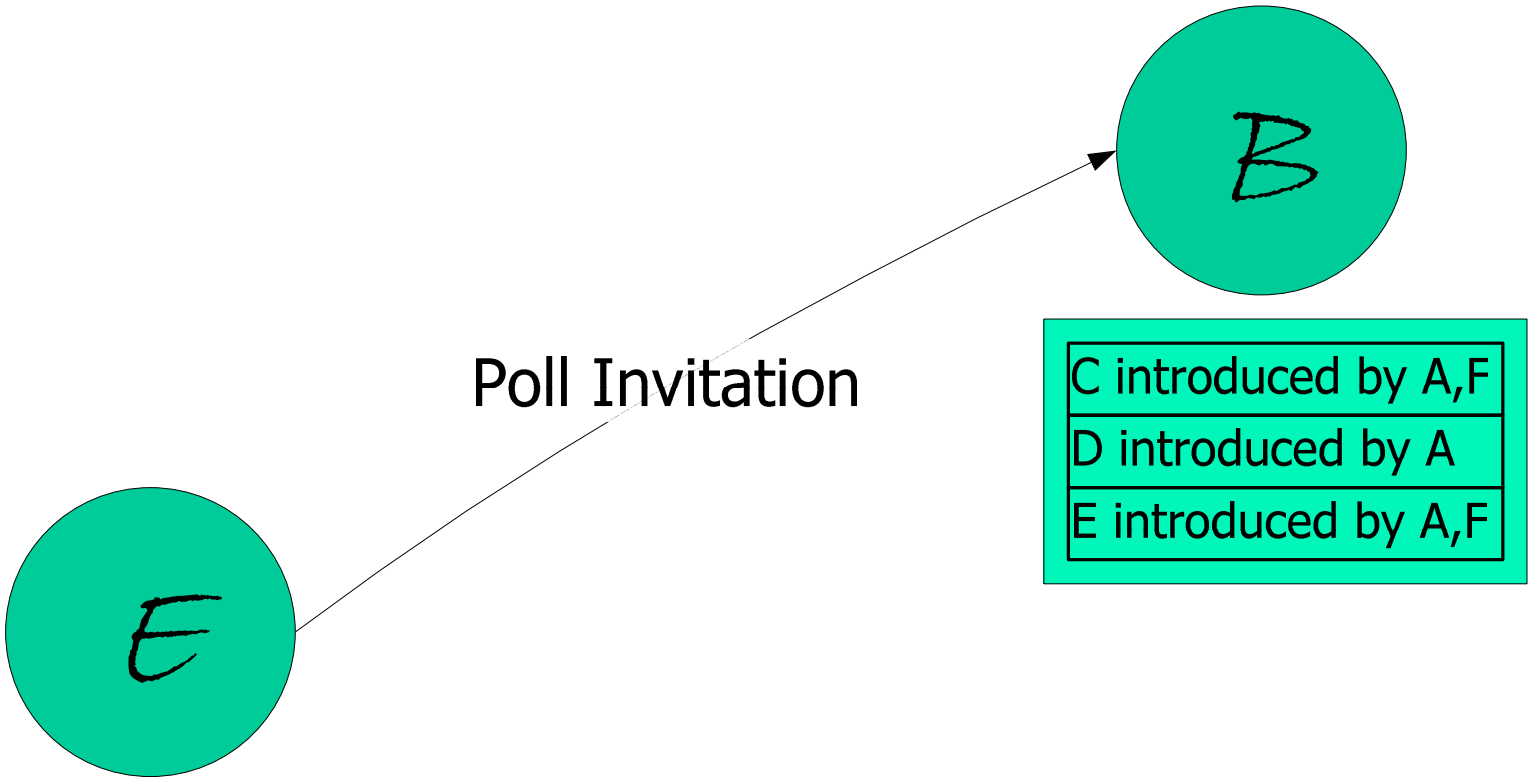
(Introductions)



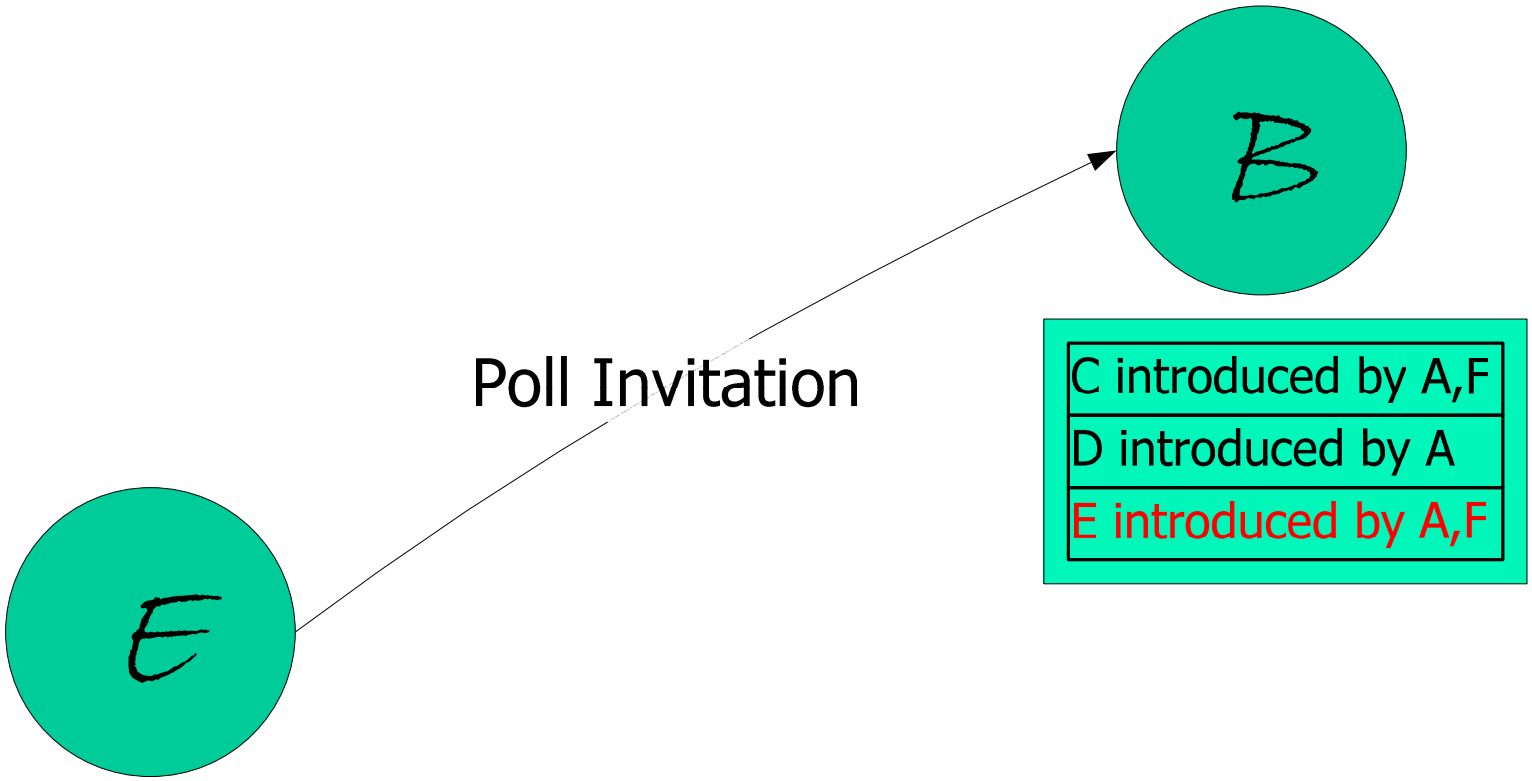
(Introductions)



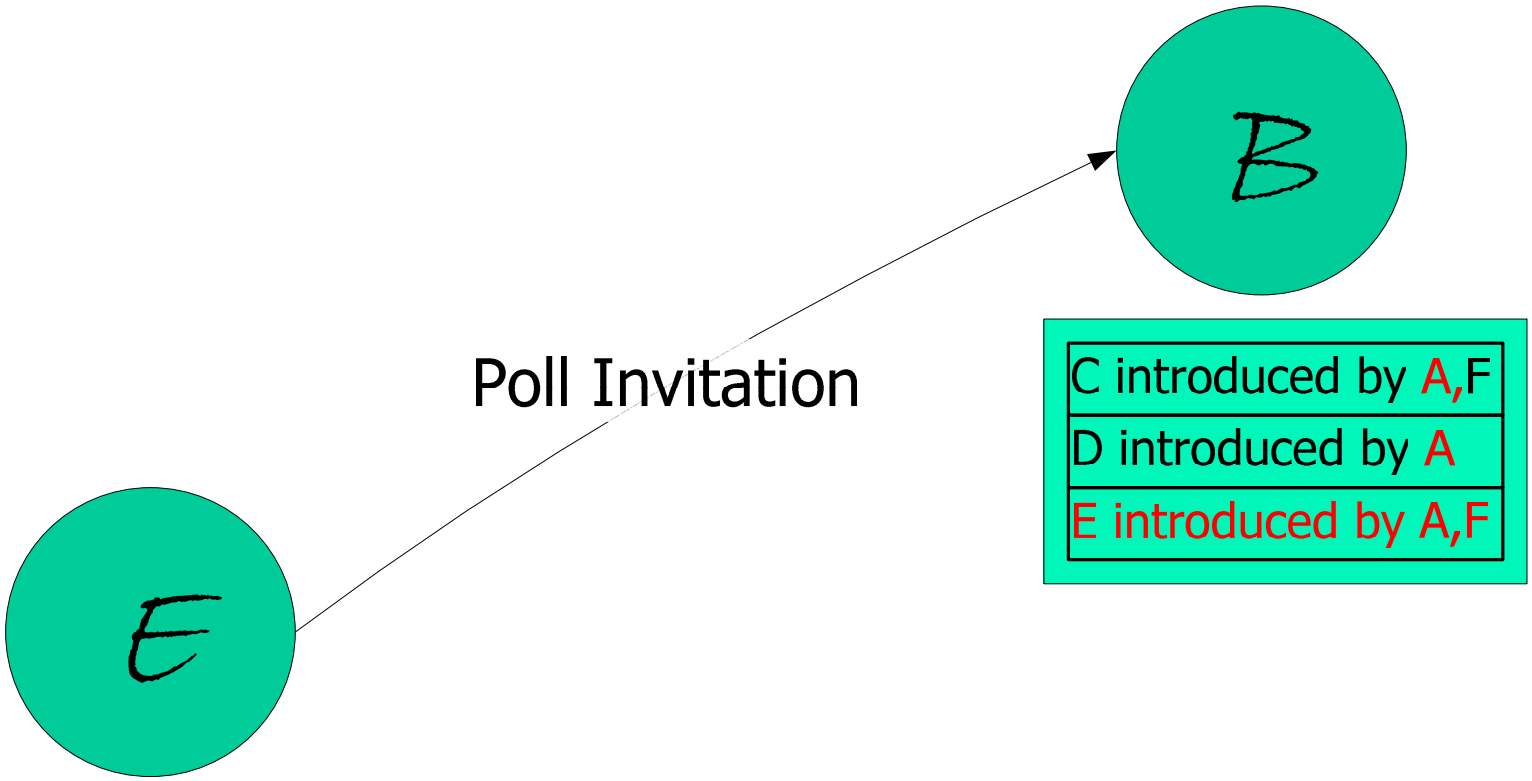
(Introductions)



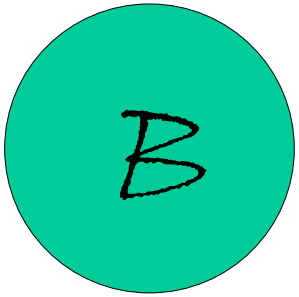
(Introductions)



(Introductions)



(Introductions)

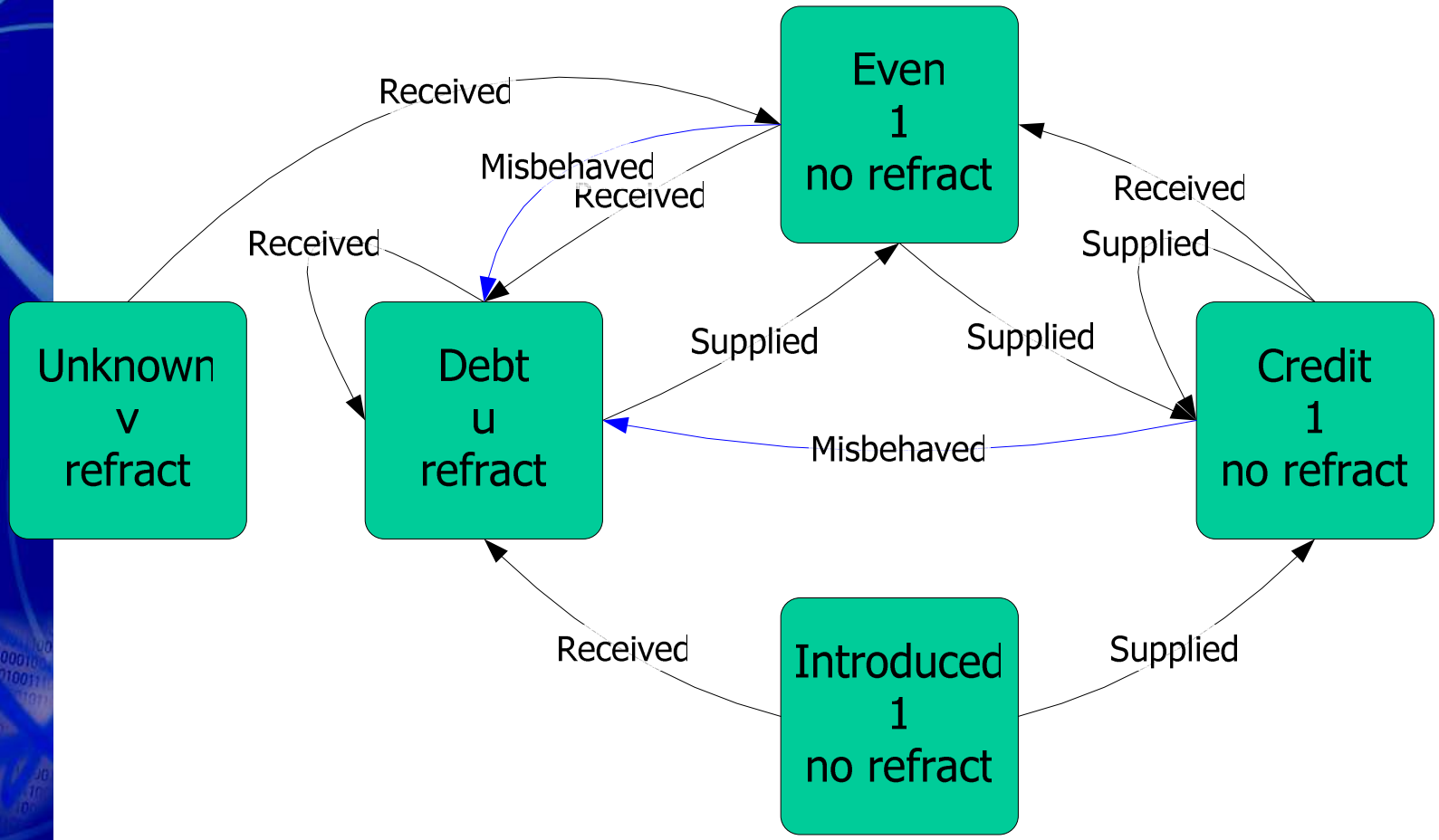


C introduced by F

Introduction Abuse

- To abuse introductions and bypass admission control, the adversary must
  - Have even/credit peer minions producing intros
  - Have kamikaze peer minions consuming intros
  - But, rate of intro production is “self clocked”
  - Therefore, rate of kamikaze attacks is “self clocked”
- Introductions effectively increase the debit-credit distance by a small constant

Admission Control



Intel Research Berkeley

Implications

- Subjective reputation makes sense with repeat interactions
  - In digital preservation, population changes slowly
  - True for “infrastructural” services, e.g., PlanetLab, OpenHash, i3, etc.
- Reputation system as good as authentication system
  - We rely on trusted routing infrastructure
  - Stronger authentication complementary

Return

Rate Limited Repairs

Backup

Intel Research Berkeley

Problem: Repair as Exceptional Service

- Poller requests repairs only when it needs them
  - Repairs are “out-of-the-ordinary”
- Adversary can request lots of them
  - To waste resources
- Repair requests “signal” desperation
  - Adversary can be well-behaved with votes, but
  - Stiff me when I really need a repair
- Must make repair a first-order operation
  - To be able to rate-limit it as well
  - To squelch the desperation signal

Continuous Repairs

- Operate in a continuous repair regime
  - After every poll, peer requests a fixed number of repairs
  - If it doesn't need any, it makes them up at random
- Now "good behavior" means
  - I supply a vote I agreed to give, and
  - All subsequent repairs (up to fixed number)
- Partial compliance is penalized via admission control

Continuous Repair Implications

- The good
  - Adversary doesn't know when refusing a repair has impact
  - Fortuitous side-effect: peers now know a repair budget for rate limiting
- The not-entirely-good
  - Upper bound on repairs means minimum time-to-repair completion
    - If too many repairs needed, must repair over multiple polls
  - Lower bound on repairs means on-going repair overhead
    - Every poll requires the transfer of some content blocks even at the best of times
  - This is the price of compliance enforcement!