Accountability in Distributed Systems

Nimit Singhania

WPE II Presentation

Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

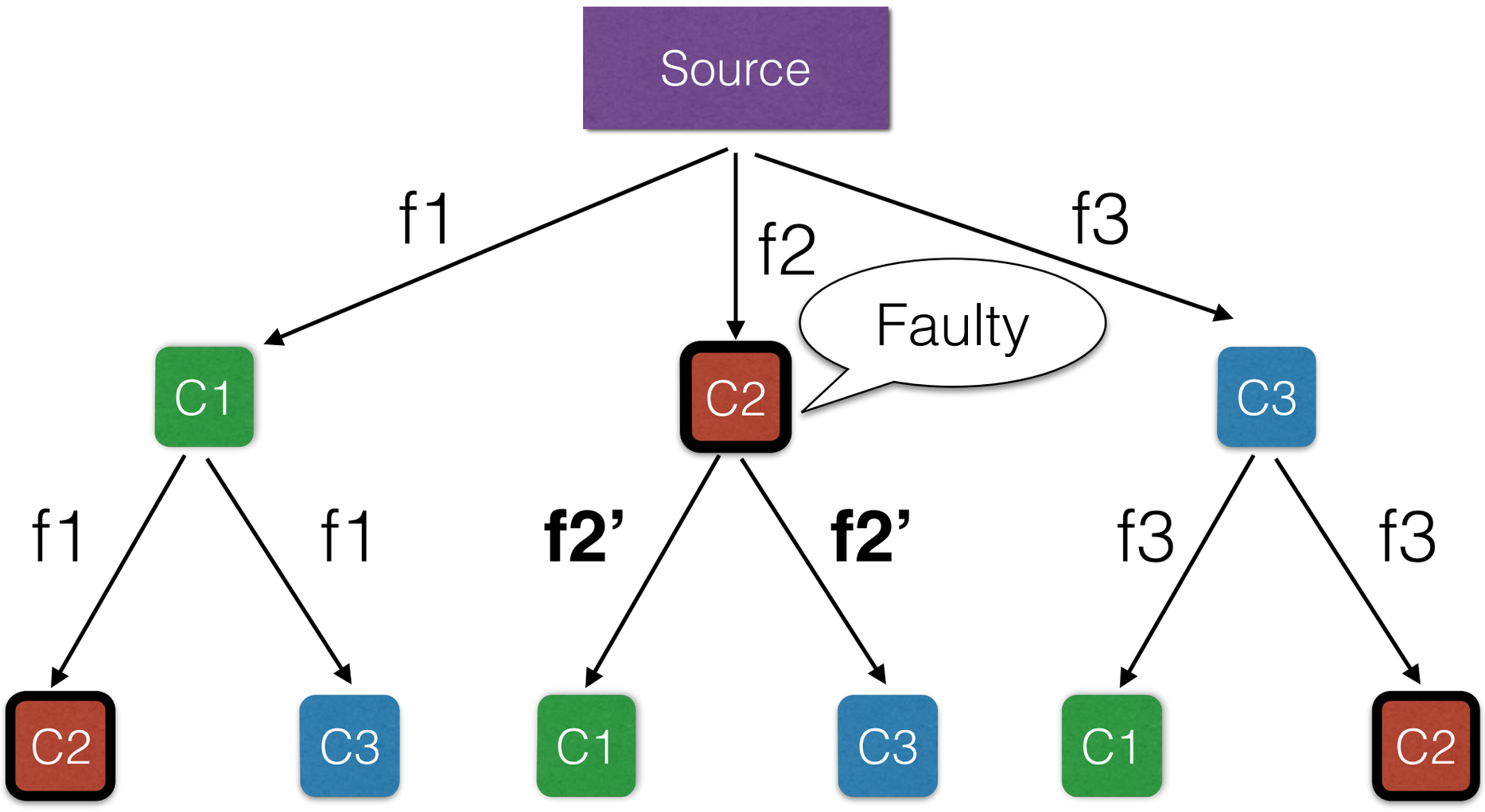
Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

Problem of fault detection

- Finding faults in Distributed Systems can be difficult
  - Localizing and isolating the faulty nodes
    - Presence of untrusted nodes
      - Such nodes might avoid detection
      - Need evidence to assign blame
- Accountability protocol can help detect such faults

Example - CDN



No way to detect C2 as faulty

Problem of fault detection

- Scenarios
  - C2 blames Source of sending **f2’**
  - C2 blames C1 and C3 of lying
  - C2 claims to follow protocol even though it received f2 and sent **f2’**

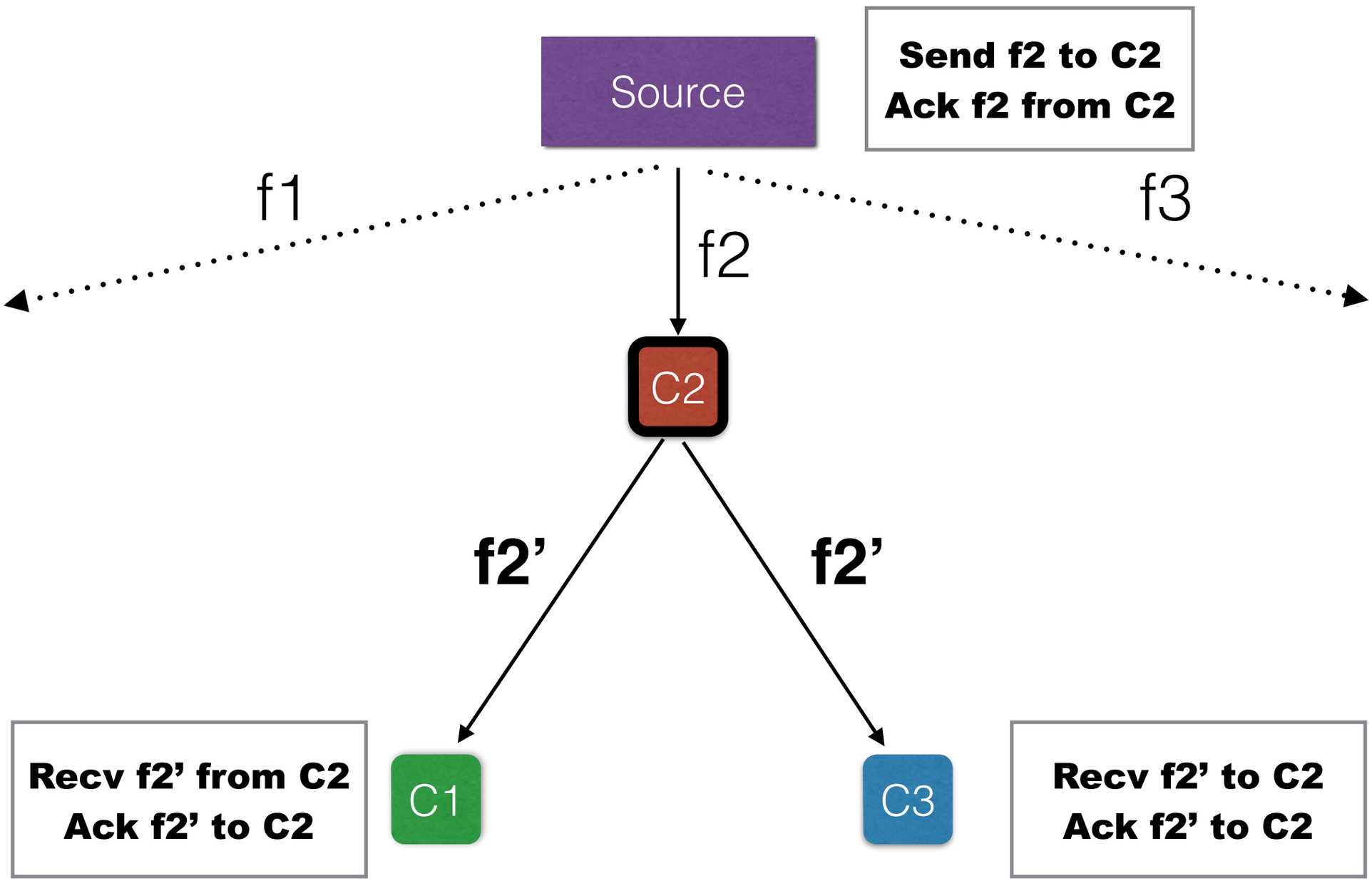
Problem of fault detection

- Scenarios
  - C2 blames Source of sending **f2’**
    - Source needs evidence of sending f2 to C2
  - C2 blames C1 and C3 of lying
    - C1 and C3 need evidence of receiving f2’ from C2
  - C2 claims to follow protocol even though it received f2 and sent **f2’**
    - Other nodes need to inspect C2 to detect that it breached protocol

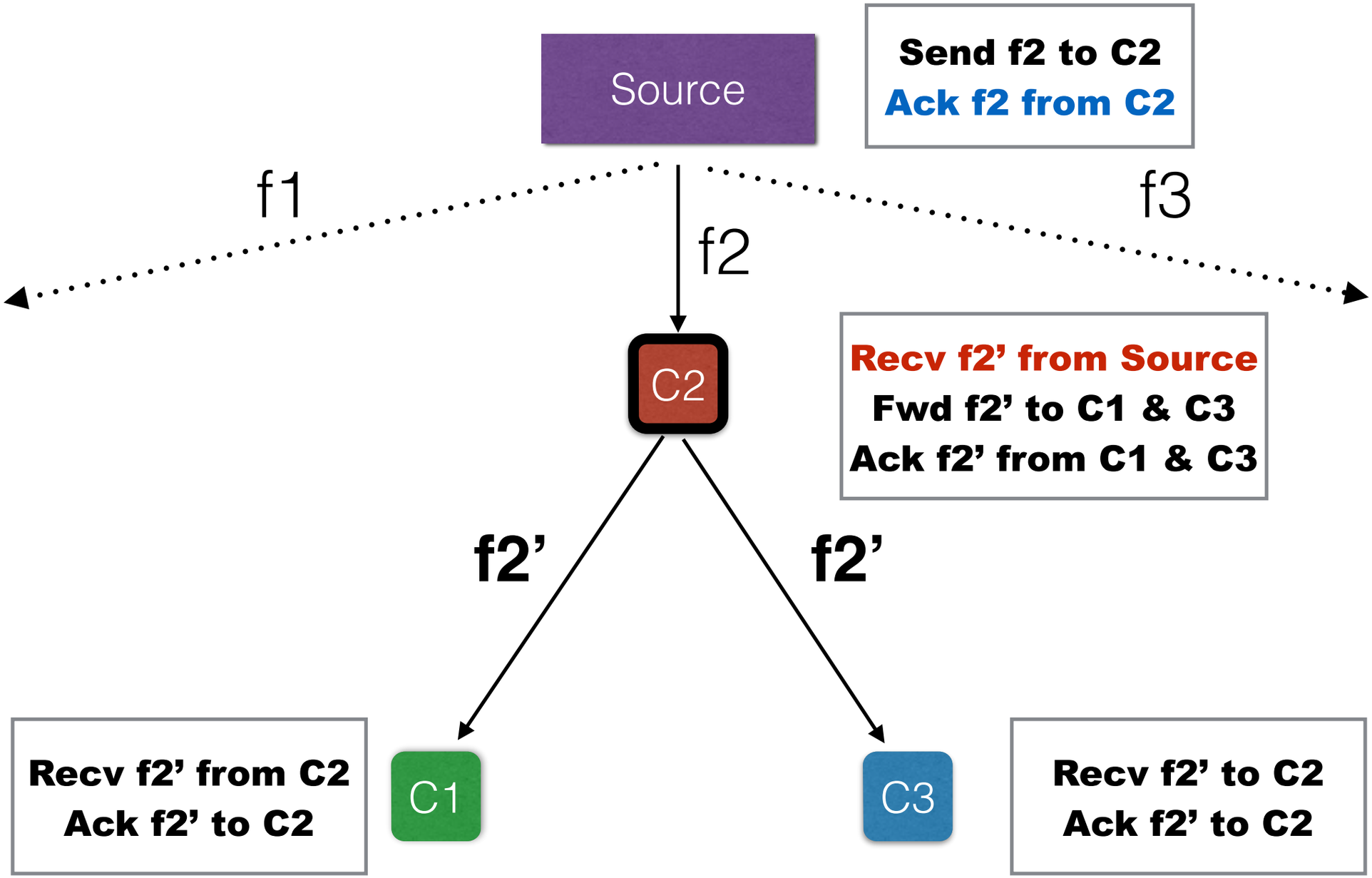
Solution

- Accountability protocol
  - Each node collects evidence about their correctness
  - Evidence inspected by other nodes
  - If evidence is *incorrect*, node is *faulty*
- Ensures that a detectably faulty node is eventually detected
- Ensures correct node is not *falsely implicated*

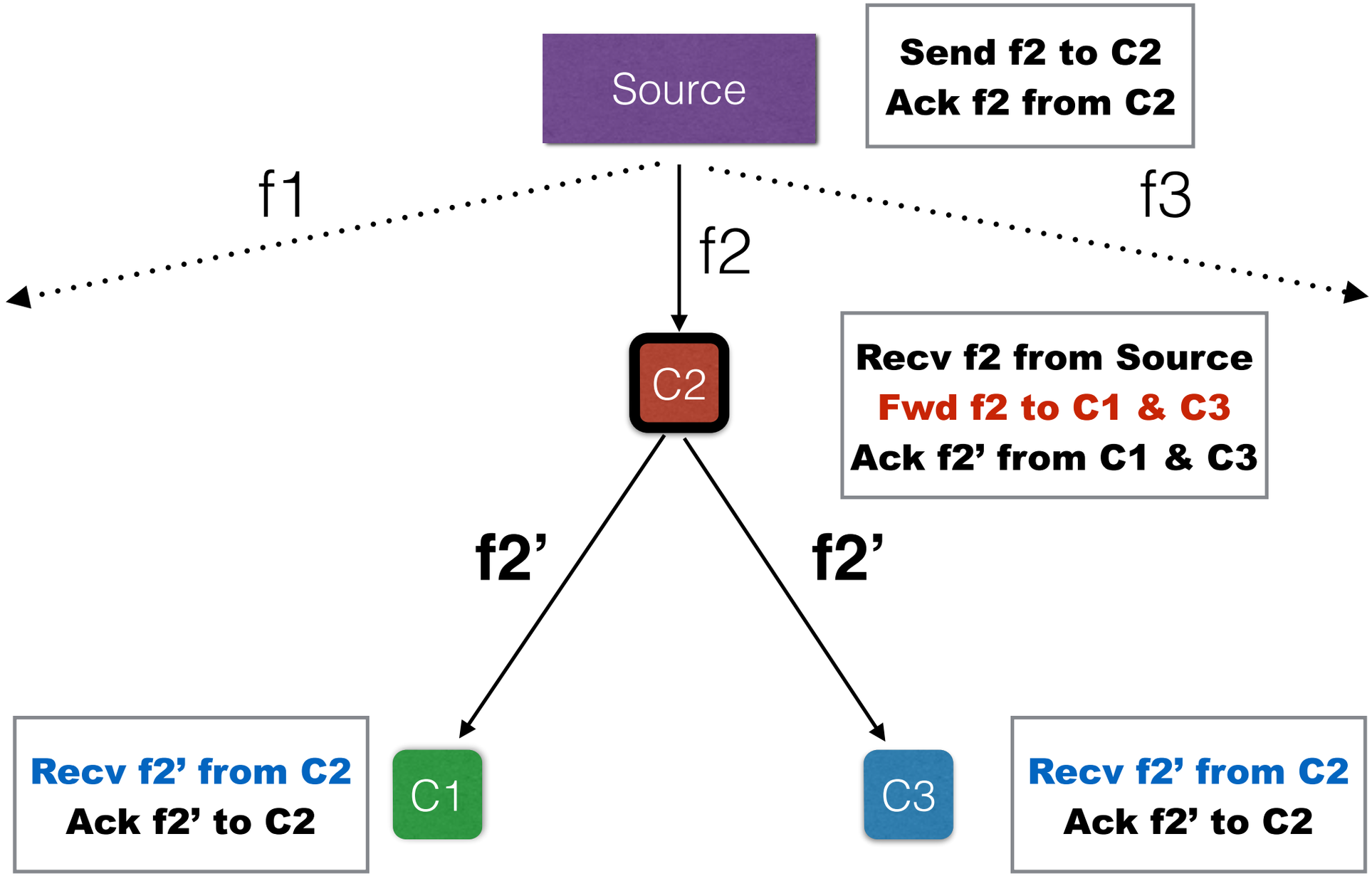
Solution


Source
Send f2 to C2
Ack f2 from C2
f1
f2
f3
C2
f2'
f2'
Recv f2' from C2
Ack f2' to C2
C1
C3
Recv f2' to C2
Ack f2' to C2

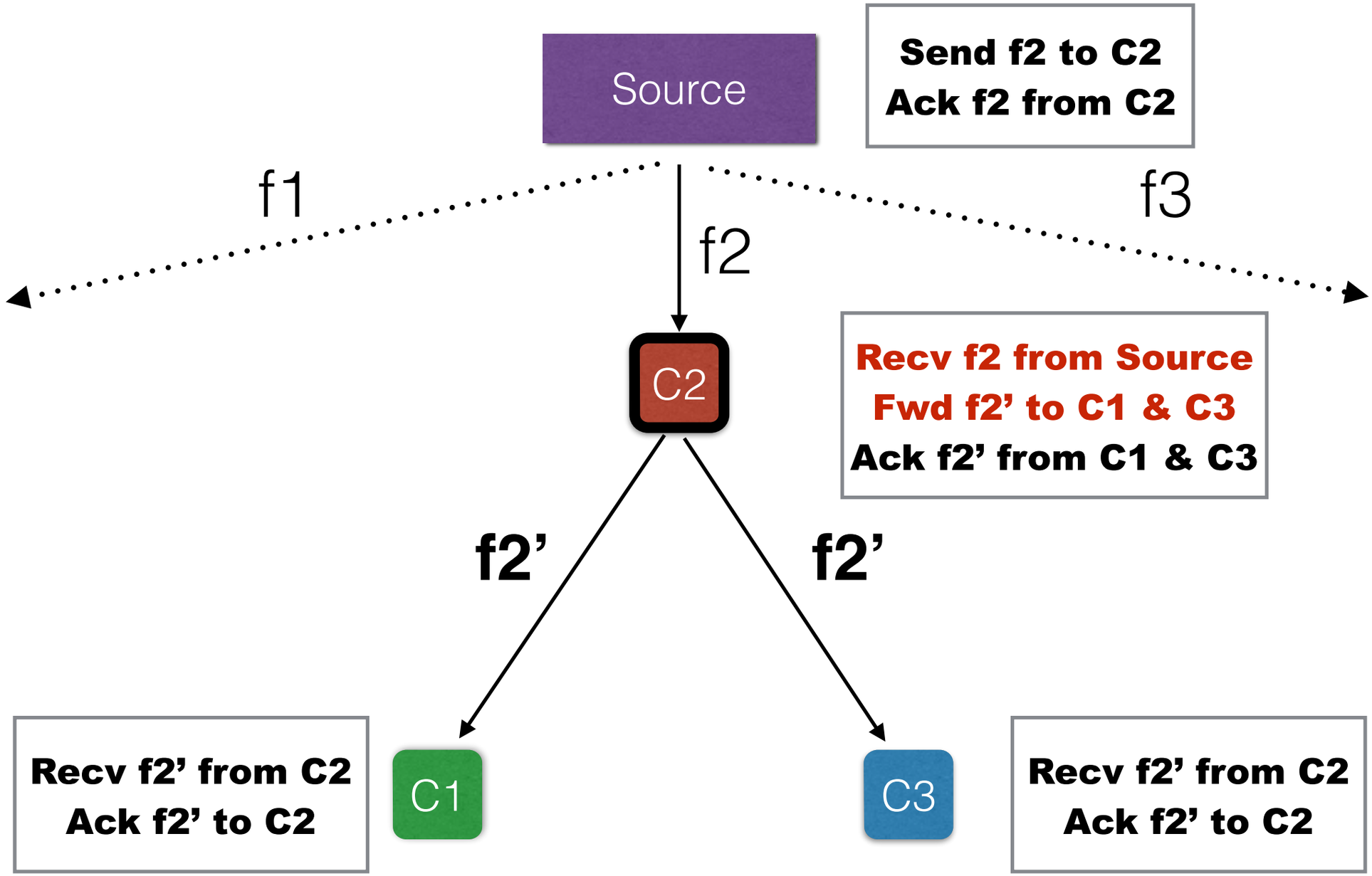
Scenario 1


Source
Send f2 to C2
Ack f2 from C2
f1
f2
f3
C2
Recv f2' from Source
Fwd f2' to C1 & C3
Ack f2' from C1 & C3
f2'
f2'
Recv f2' from C2
Ack f2' to C2
C1
C3
Recv f2' to C2
Ack f2' to C2

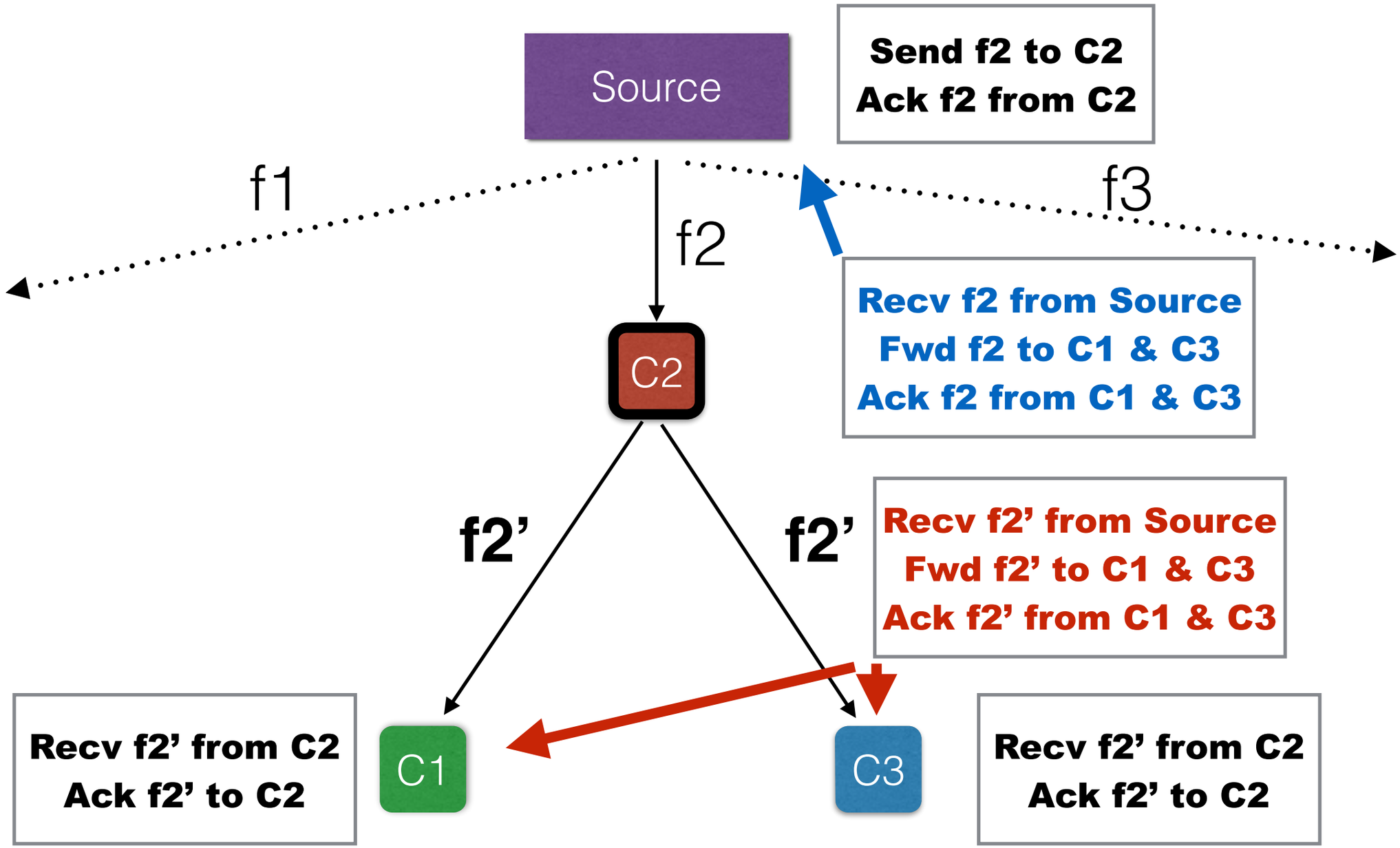
Scenario 2


Source
Send f2 to C2
Ack f2 from C2
f1
f2
f3
C2
Recv f2 from Source
Fwd f2 to C1 & C3
Ack f2' from C1 & C3
f2'
f2'
Recv f2' from C2
Ack f2' to C2
C1
C3
Recv f2' from C2
Ack f2' to C2

Scenario 3


Source
Send f2 to C2
Ack f2 from C2
f1
f2
f3
C2
Recv f2 from Source
Fwd f2' to C1 & C3
Ack f2' from C1 & C3
f2'
f2'
Recv f2' from C2
Ack f2' to C2
C1
C3
Recv f2' from C2
Ack f2' to C2

Forking Attack



Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

Aspects of Accountability Protocol

- Notion of Correctness
- Evidence Collection
- Evidence Inspection
- Probabilistic Guarantees
- Fault Detection Power

Notion of Correctness

- Correctness properties of a node
- Used to verify evidence provided by a node
- CDN - Node is *correct* if it follows the required protocol of forwarding files

Evidence Collection

- Two components of evidence
- Self-Correctness
  - Evidence of satisfying correctness properties
    - CDN - Log of sequence of actions performed
- Mutual-Correctness
  - Evidence of correct interaction with other nodes
  - CDN - Signed receipts of sending or receiving files

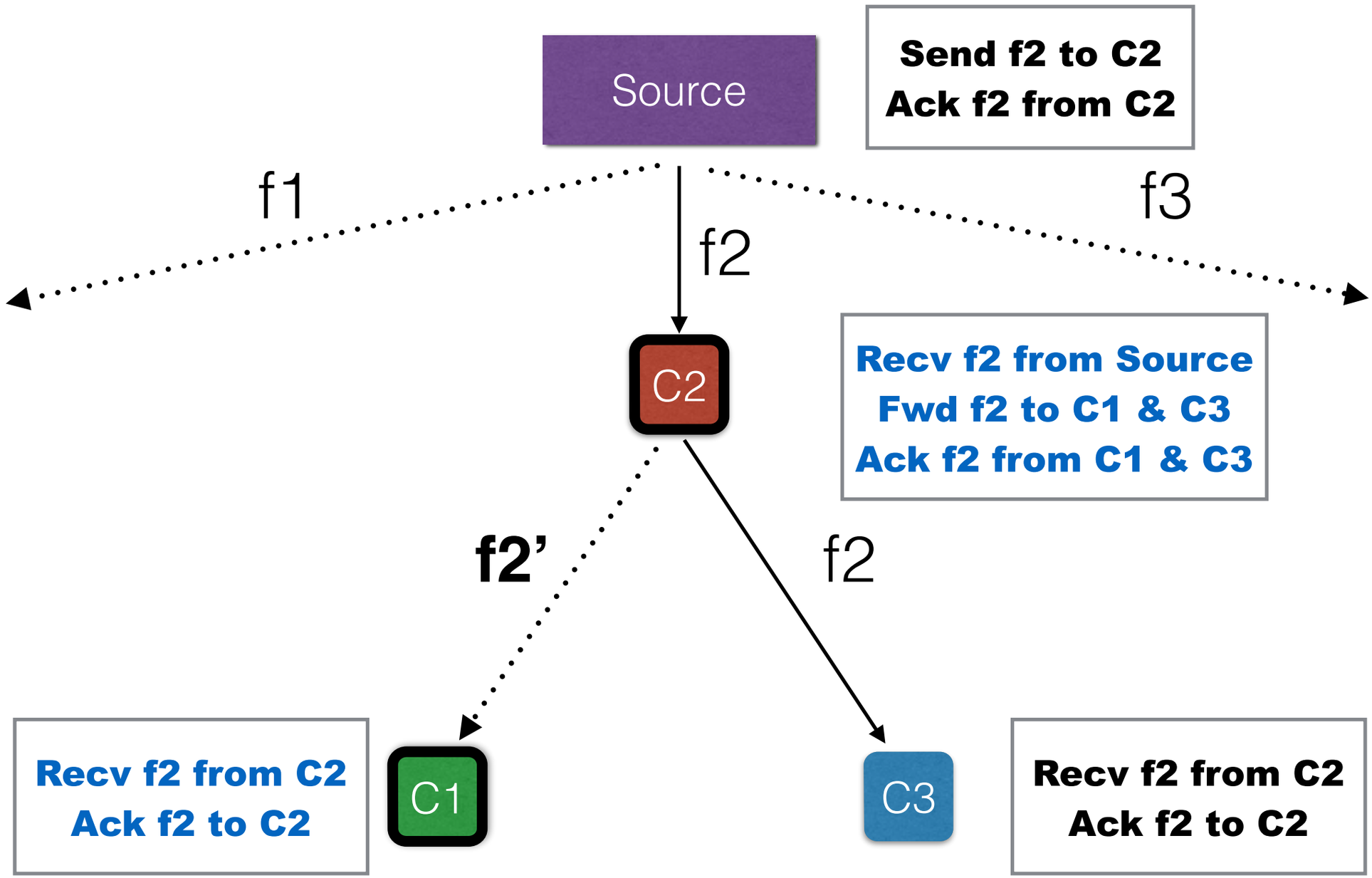
Evidence Inspection

- Consistency
  - Check if the evidence by a node is unique
  - Check if interaction with other nodes is correct
    - CDN - Detects fault when C2 blames Source, C1 or C3
- Audit
  - Check if the evidence satisfies correctness properties
  - CDN - Detects fault when C2 gives incorrect sequence of actions
- Challenge/Response - What if a node does not respond?

Probabilistic Guarantees

- Overhead of Evidence Collection & Inspection *huge*
  - Huge # of messages exchanged
  - Significant computation required
- Can be reduced with Prob. Guarantees of Fault Detection
- CDN - *Randomly* check transmission of some files from a sequence of files

Fault Detection Power



C1 and C2 give fake evidence to get away

Outline

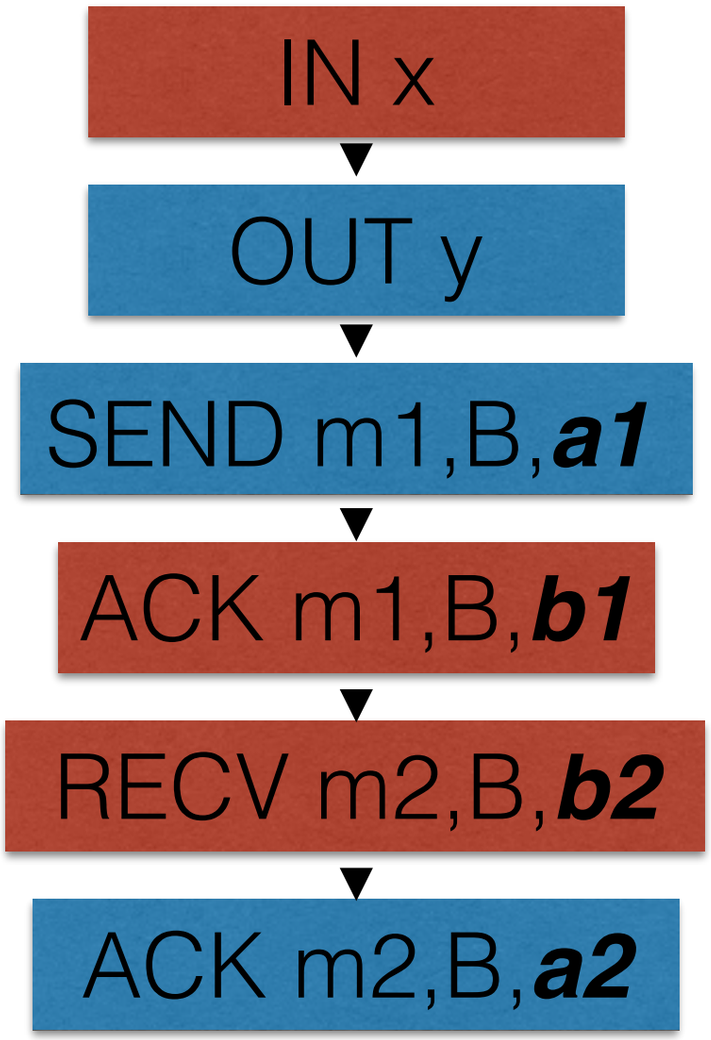
- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

PeerReview

- A general accountability protocol
  - Applicable to systems where nodes follow a deterministic protocol
  - Assumes each node can sign messages that can be used as irrefutable evidence
- **Notion of Correctness**: A node is *correct*
  - if it follows a deterministic protocol
  - A reference implementation can be used to replay execution

PeerReview - Evidence Collection

- Self-correctness: Sequence of inputs/ outputs to a node
- Mutual-correctness: *Authenticators* attached to messages and their acks
- a1, b1, a2, b2 are authenticators
- **a1** - Unique hash value of events up to (SEND m1, b) signed by A



Log of node A

PeerReview - Evidence Inspection

- *Witness Set(j)*
  - A set of nodes responsible for inspecting node *j*
- Consistency - Authenticators from *j* are forwarded to its witness set
  - Check if all authenticators are accounted for in *j*'s log entries
  - Forward authenticators from *j*'s log to corresponding witnesses
- Audit - Each witness compares log entries against output from the reference implementation
- Challenge/Response - Node marked as suspected if it does not respond

PeerReview - Prob. Guarantees

- Message Complexity of Consistency: $O(w^2)$
  - w = # of witnesses in witness set
- For complete guarantee, w > # of faulty nodes
- if w allowed to have all faulty nodes,
  - $w = O(\log n)$
  - Message Complexity = $O(\log^2 n)$

Peer Review - Fault Detection Power

- Commission Faults - Node sends incorrect message
  - Caught during consistency or audit by witnesses
- Omission Faults - Node refuses to respond
  - Suspected by witnesses
- Non-Observable Faults - No incorrect messages received by correct nodes
  - Faulty nodes can get away by giving fake evidence

Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

CATS

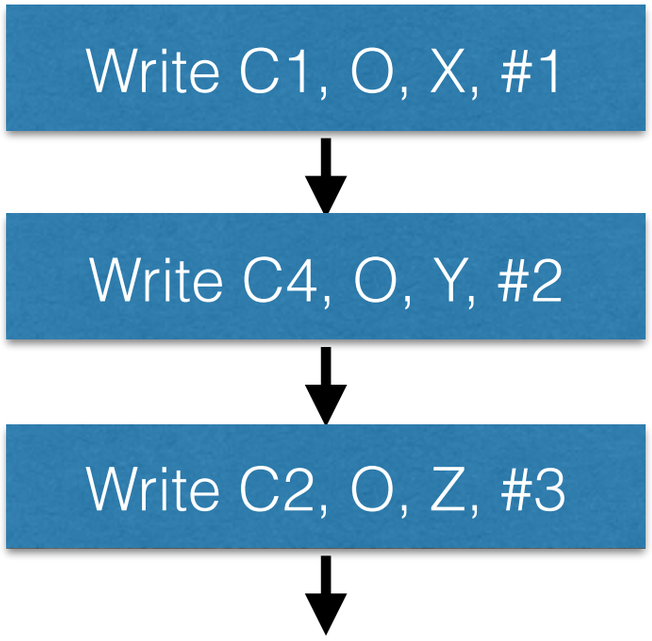
- Accountability protocol for network storage
  - Server maintains a set of shared objects
    - Clients can read and write on them
- Helps detect server faults or client misbehavior
- Assumes access to a trusted publishing medium and that clients can sign messages like PeerReview

CATS - Notion of Correctness

- Server is *correct* if
  - Executes writes from authorized clients
  - Applies writes in order
  - Reads return values of latest writes
  - Writes are visible to all authorized clients

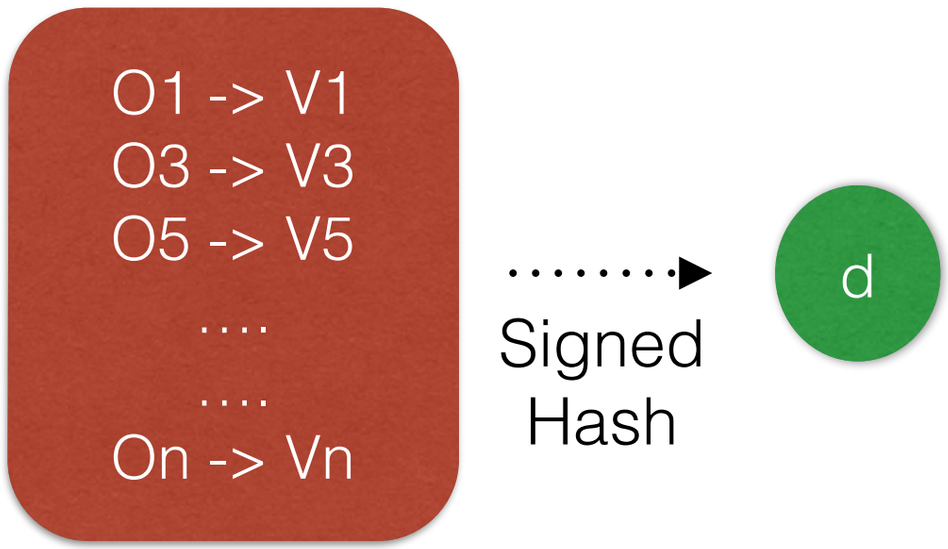
CATS - Evidence Collection

- Action Histories
- Sequence of writes on an object



Action History for object O

- State Digests
  - Signed hash over contents of server
  - Can verify if correct values of objects used for digest

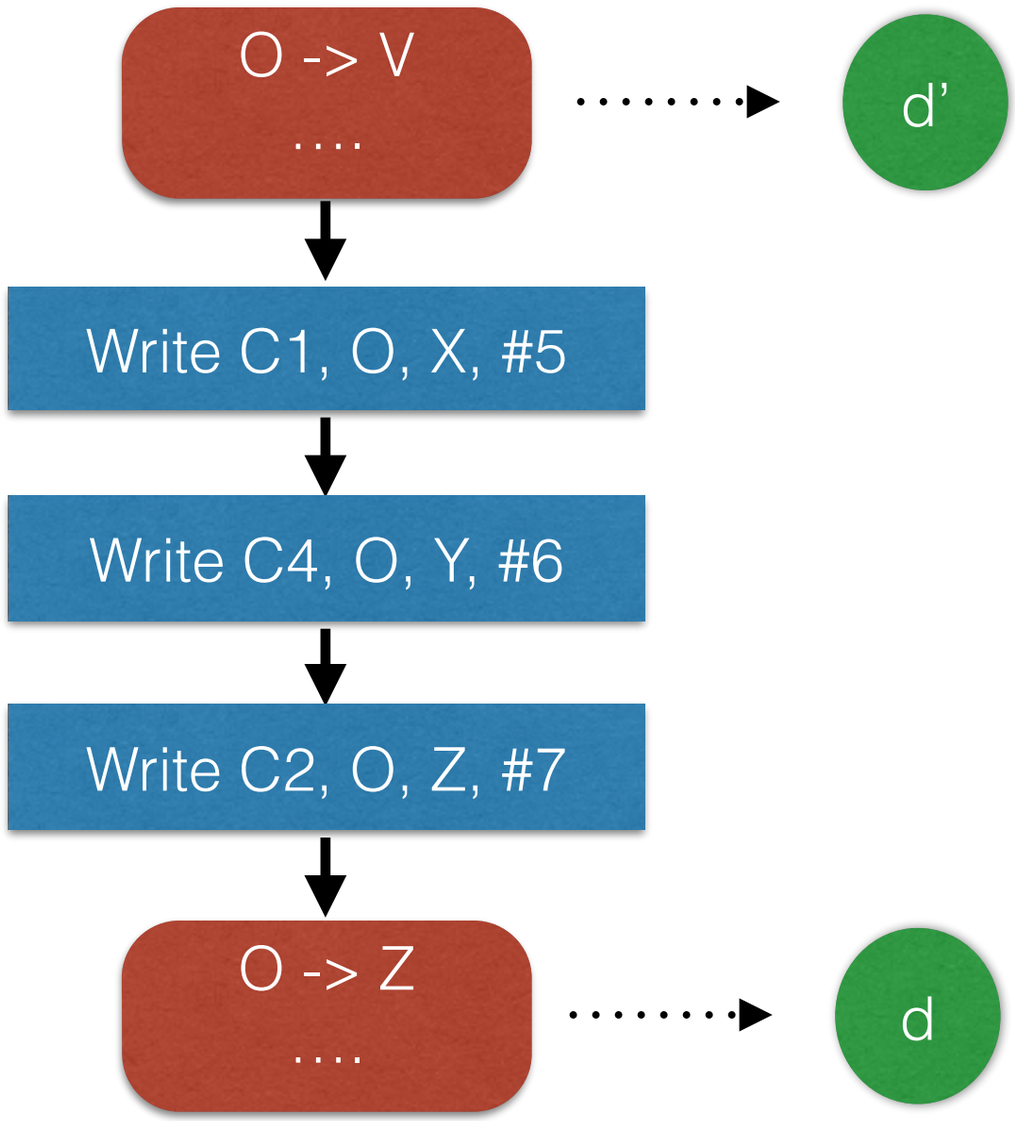


CATS - Evidence Inspection

- Consistency
  - State digests periodically published to public medium
  - Commits Server to unique view (forking attack not possible)
  - Clients check if their requests are consistent with digests

CATS - Evidence Inspection

- Audit
- Check if digests are computed correctly
- Digests can be checked relative to previous digests
- All correctness properties checked



CATS - Probabilistic Guarantees

- Checking all digests in a span of time - computationally expensive

  - *Randomly* select k digests to audit from an interval of time

  - *Randomly* select some objects to audit

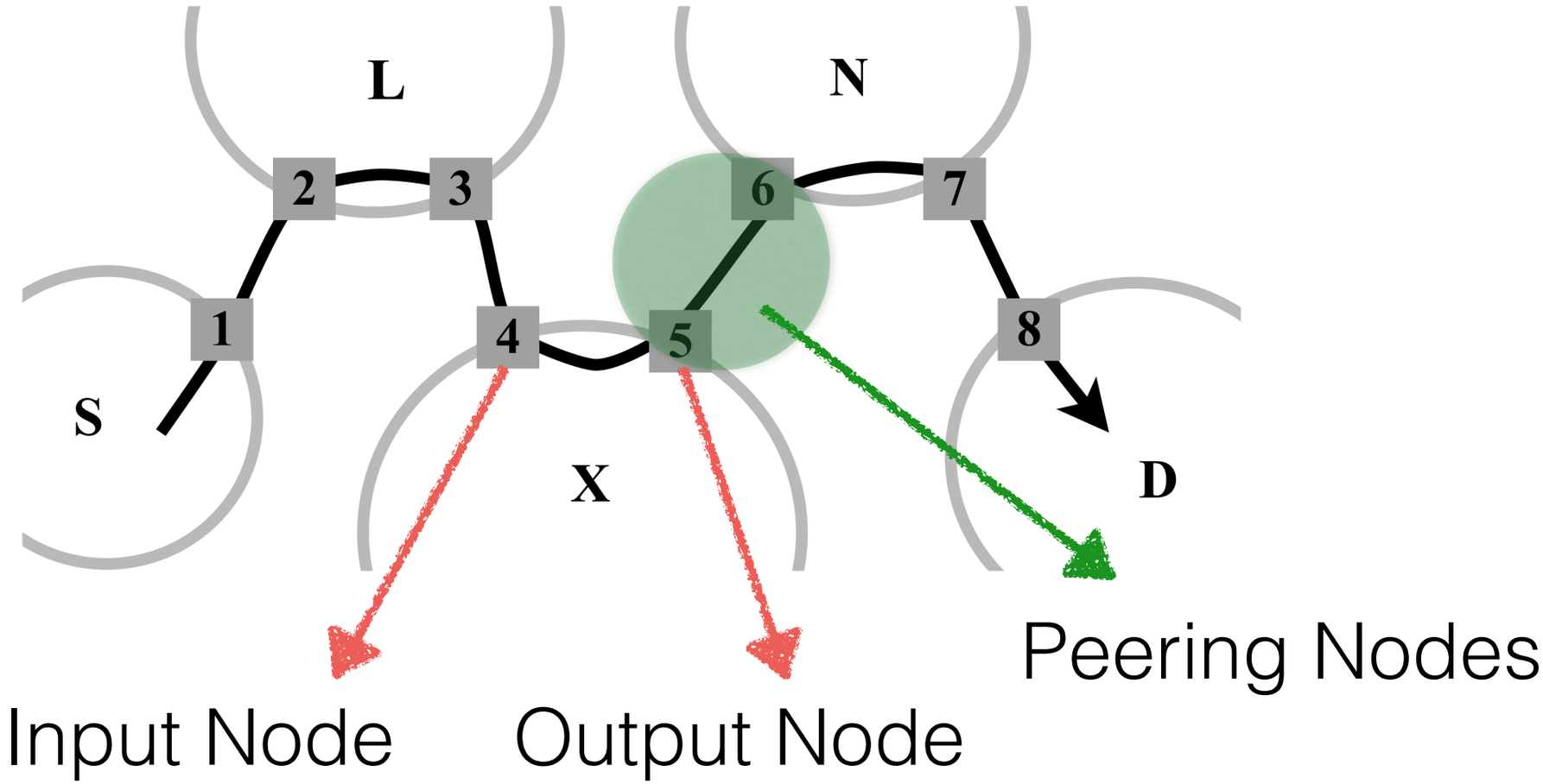
Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

Network Professional

- Internet - provides no guarantees on performance
  - A *path* consists of multiple administrative *domains*
- Accountability protocol to measure domain performance on paths
  - Helps identify low performing domains and debug performance problems
- Assumes packets can be lost, reordered or delayed but not modified or inserted

Network Professional - Terminology



Network Professional - Notion of Correctness

- Performance of domains
  - Loss Rate - Amount of packet loss experienced
  - Delay - Average delay experienced by packets
- No Absolute notion of correctness
- *Link Correctness* - link between peering nodes faulty if packets lost, reordered or delayed beyond Δ
  - Used to check consistency

Network Professional - Evidence Collection

- Evidence: Receipts on packets at input and output nodes
    - *Packet Ids* and *Timestamps*
- Generating receipts on each packet - huge overhead
- Protocol allows tuning of overhead
    - At the expense of probably approximately correct measurements
- Each node samples a subset of packets based on *future incoming traffic*
    - If subset of packets is known a-priori, then nodes can bias performance
    - Some packets are labeled as markers and used to select older packets

Network Professional - Consistency

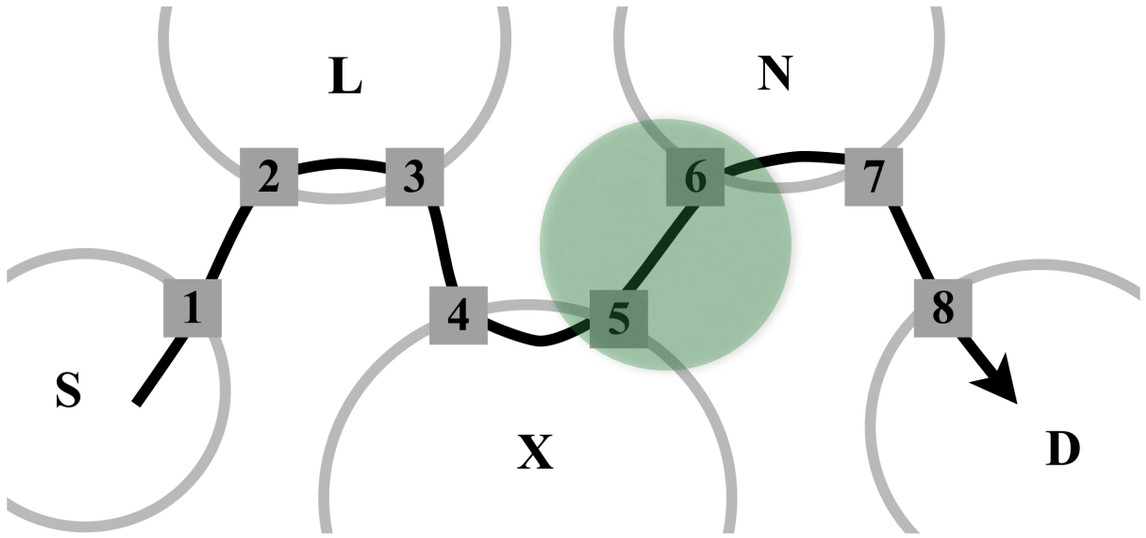
- Assumes receipts are transmitted correctly to all nodes and regulator (No forking attack)
- Checks correctness of link between peering nodes
- Ensures measurements are consistent with neighboring domains
- Checks receipts for common subset of packets sampled
  - No loss should be observed
  - Delay not more than Δ

Network Professional - Audit

- Computes measurements between input and output nodes *i* and *j*
- *Si* = subset of packets sampled at *i* and must be sampled at *j*
- *Sj* = subset of packets actually sampled at *j*
  - Loss Rate = (|*Si*| - |*Sj*|)/|*Si*|
  - Effect of reordering cancels out
- Delay computed using packets sampled at both nodes

Network Professional - Fault Detection Power

- A single node can not generate significantly biased receipts
- Pair of peering nodes can do so
  - For example 5 and 6 decrease time stamps by x
  - Suppose 4 and 7 are honest
  - N's delay increased
- One of the colluding domains will be at loss



Outline

- Why Accountability?
- Aspects of Accountability
- PeerReview
- CATS
- Network Professional
- Comparison between protocols

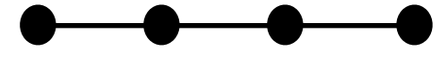
Comparison

- Notion of Correctness
  - PeerReview - Implicitly by requiring nodes to follow the protocol
  - CATS - Explicitly via high-level correctness requirements on the service
  - Network Professional - Quantitatively via loss and delay measurements
    - Link Correctness defined to check consistency

Comparison

- Evidence
- Peer Review - Sequence of messages sent and received
- CATS - Sequence of messages + Periodic State Snapshots
- Network Professional - Independent receipts on packets

Comparison

- Evidence Inspection - Audit
  - PeerReview - Complete execution needs to be replayed
  - CATS - Execution split into smaller sequences by digests 
    - Some digests selected for audit
    - Auditor signals which digests to audit
- Network Professional - Receipts on packets generated independently
  - Some packets are selected for computing measurements
  - Implicitly told for which packets to generate receipts 

Conclusion

- Accountability can help detect faults in systems
- Consists of evidence collection and inspection
- Evidence checked for consistency and correctness
- Overhead can be reduced at the expense of probabilistic guarantees

References

- K. Argyraki et al. Verifiable Network-Performance Measurements, Co-Next ’10

- A. Haeberlen et al. The Fault Detection Problem, OPODIS ’09

- A. Haeberlen et al. PeerReview: Practical Accountability for Distributed Systems, SOSP ’07

- A. R. Yumerefendi. Strong Accountability for Network Storage, TOS ’07

Questions?