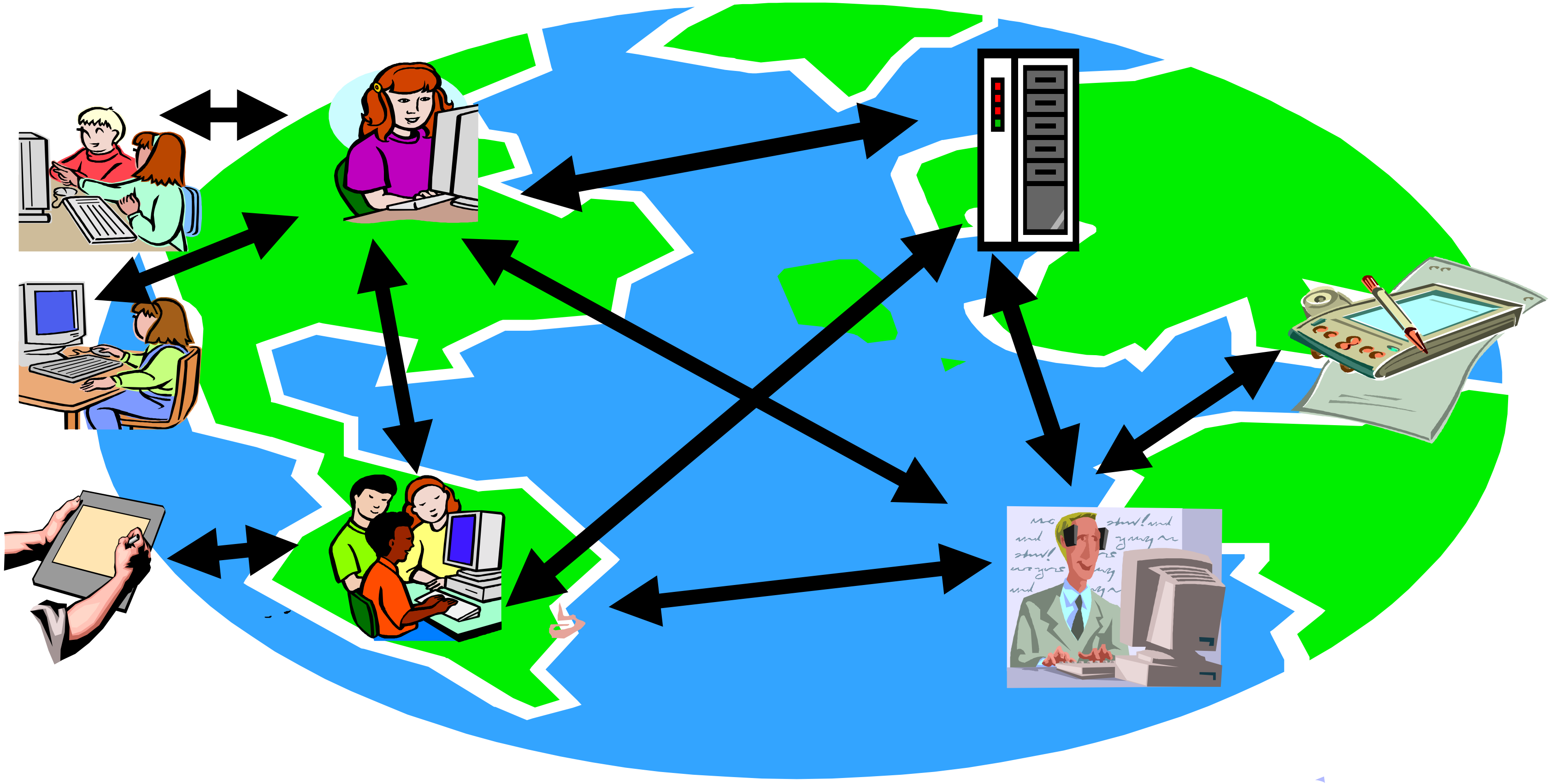
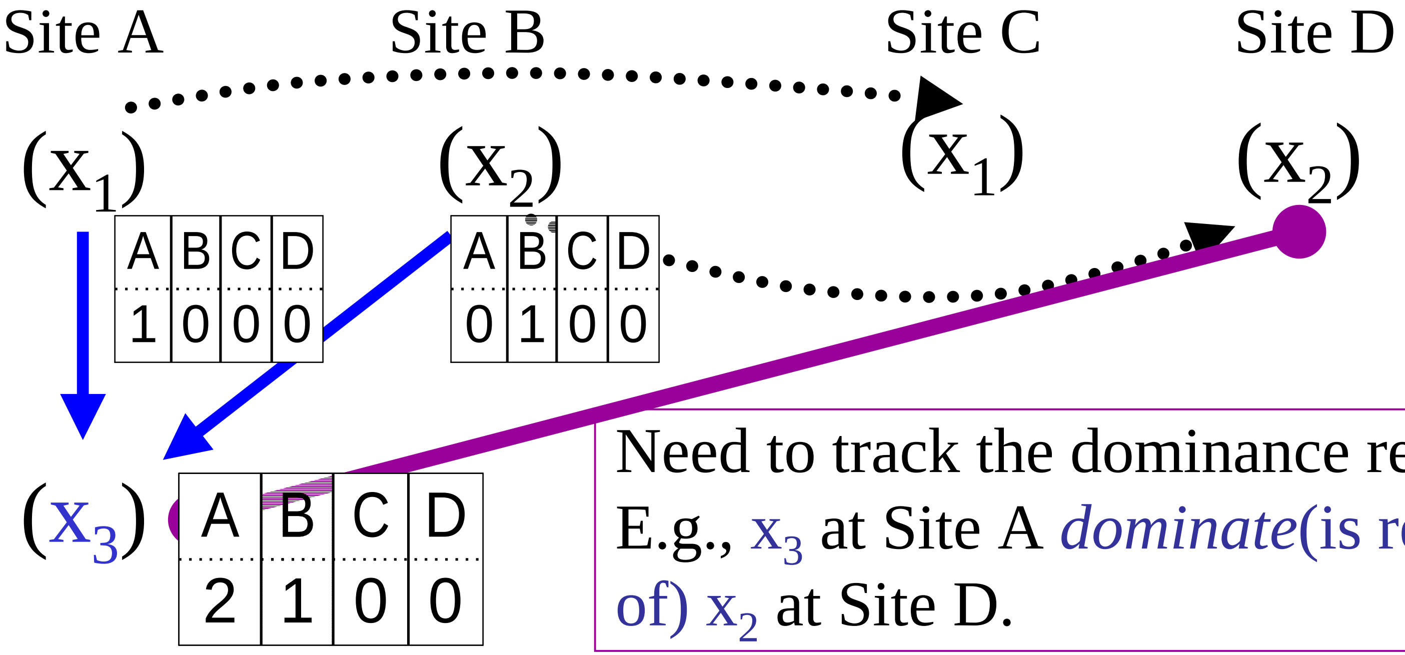
Can Replicas Converge Across Partitioned Networks ?



Brent Byunghoon Kang

Computer Science Division, UC Berkeley

Pair-wise Reconciliation in Optimistic Replication



A	B	C	D
1	0	0	0

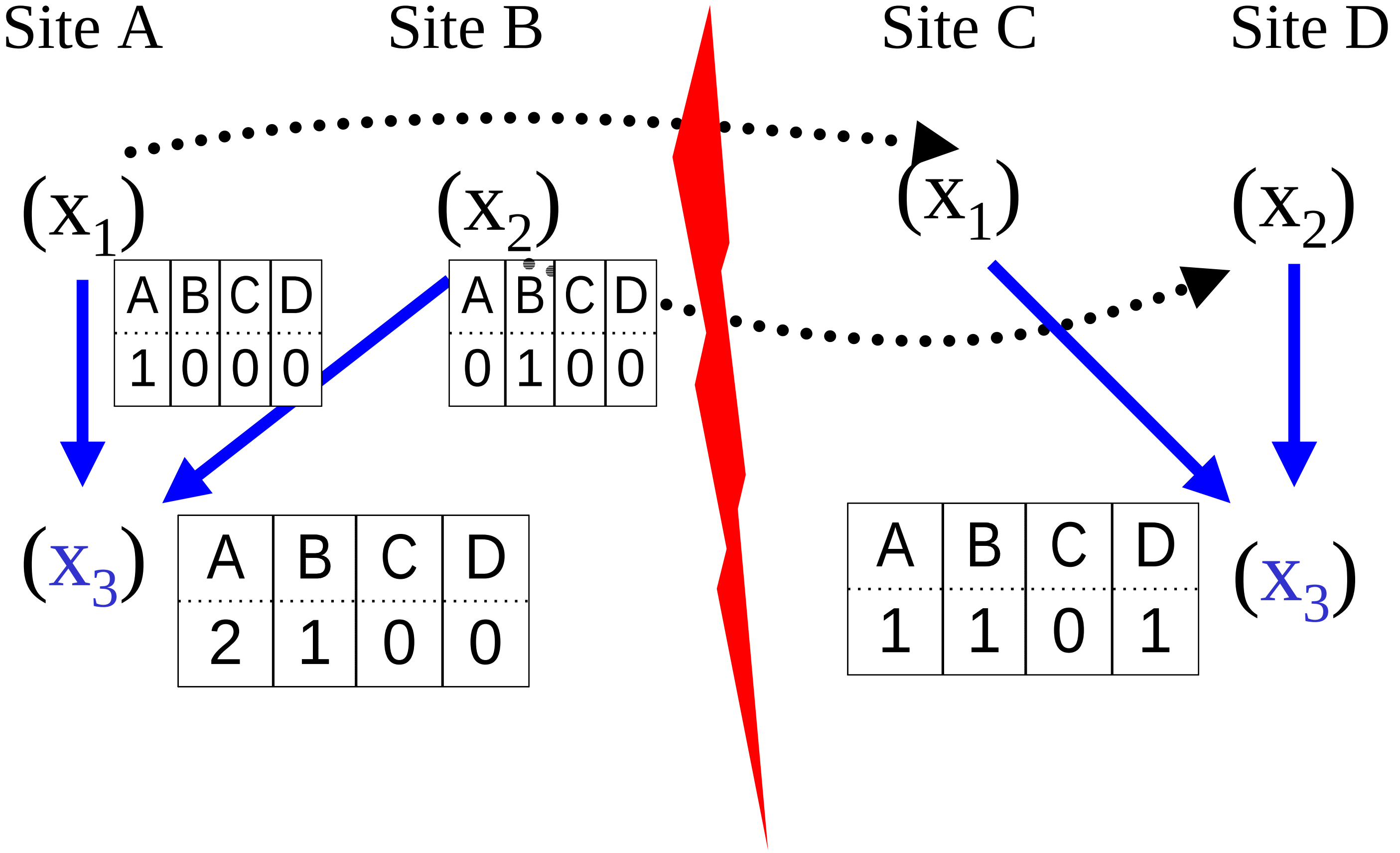
A	B	C	D
0	1	0	0

A	B	C	D
2	1	0	0

Need to track the dominance relation. E.g., x_3 at Site A *dominate*(is revision of) x_2 at Site D.

- Replica updated in any place; Later, converges to a consistent state by reconciling independently accrued updates.
- Typically through random pair-wise update exchanges for high availability. (i.e., anti-entropy)

Cannot Converge across Partitioned Networks



Site A

Site B

Site C

Site D

(x_1)

A	B	C	D
1	0	0	0

(x_2)

A	B	C	D
0	1	0	0

(x_1)

(x_2)

(x_3)

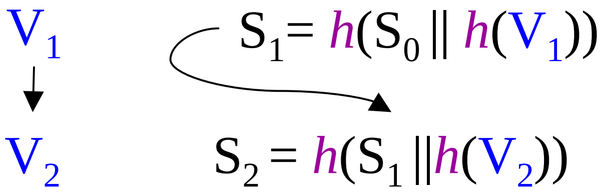
A	B	C	D
2	1	0	0

A	B	C	D
1	1	0	1

(x_3)

Summary Hash History (SHH)

- *Summary hash* = hash of version history graph
  - *hash* (predecessor’s summary hash + content hash)
  - Collision resistant hash function: SHA-1
  - E.g.,



V_1

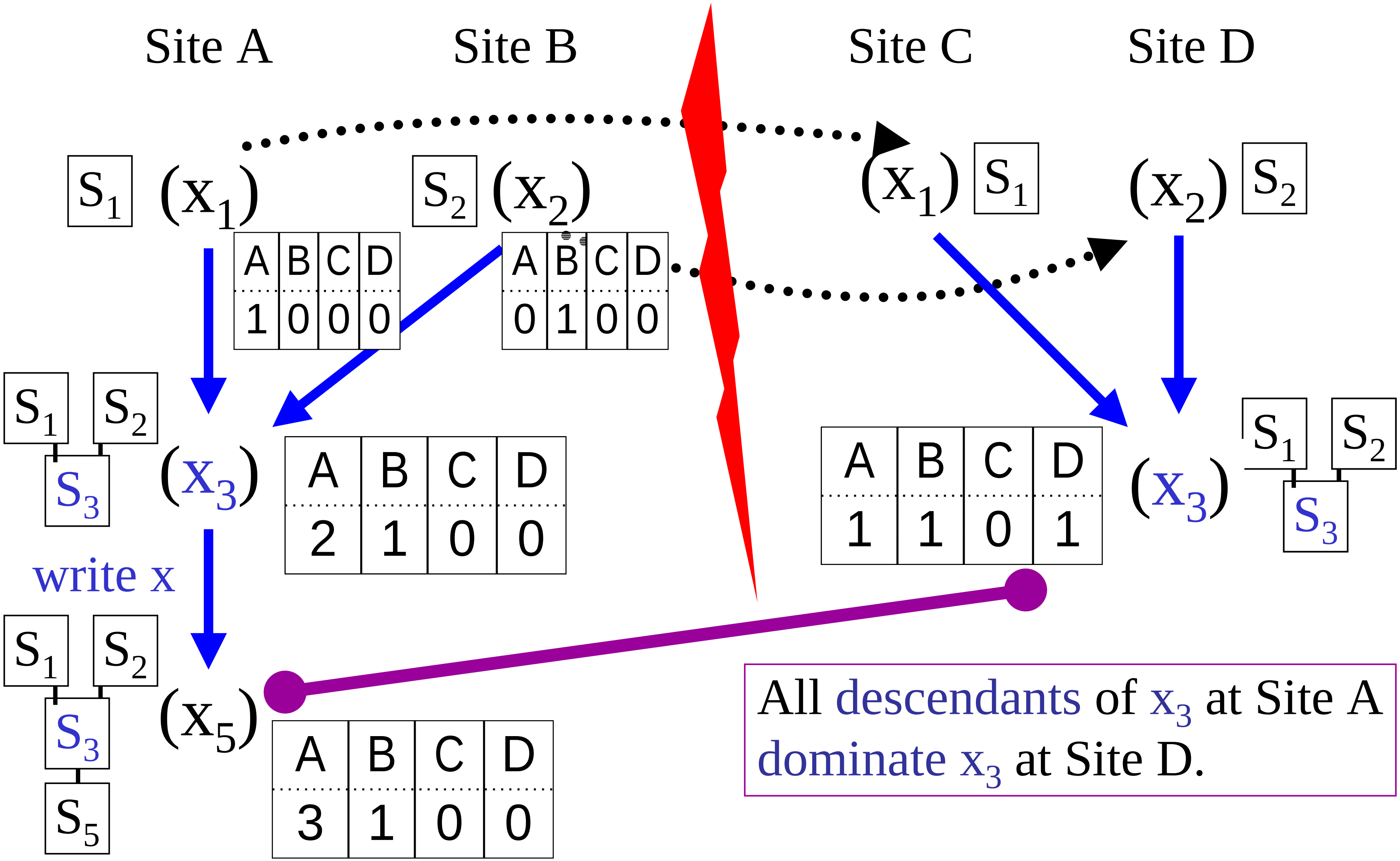
V_2

$$S_1 = h(S_0 \| h(V_1))$$

$$S_2 = h(S_1 \| h(V_2))$$

- *Summary Hash History* (SHH) uses summary hash as version ID.
  - During Reconciliation, sites exchange SHHs.
  - From an SHH, sites can securely reconstruct the full version history graph
    - From which, each can decide which version is later or both versions are concurrent (conflict).

Convergence across Partitioned Networks with SHH



A	B	C	D
1	0	0	0

A	B	C	D
0	1	0	0

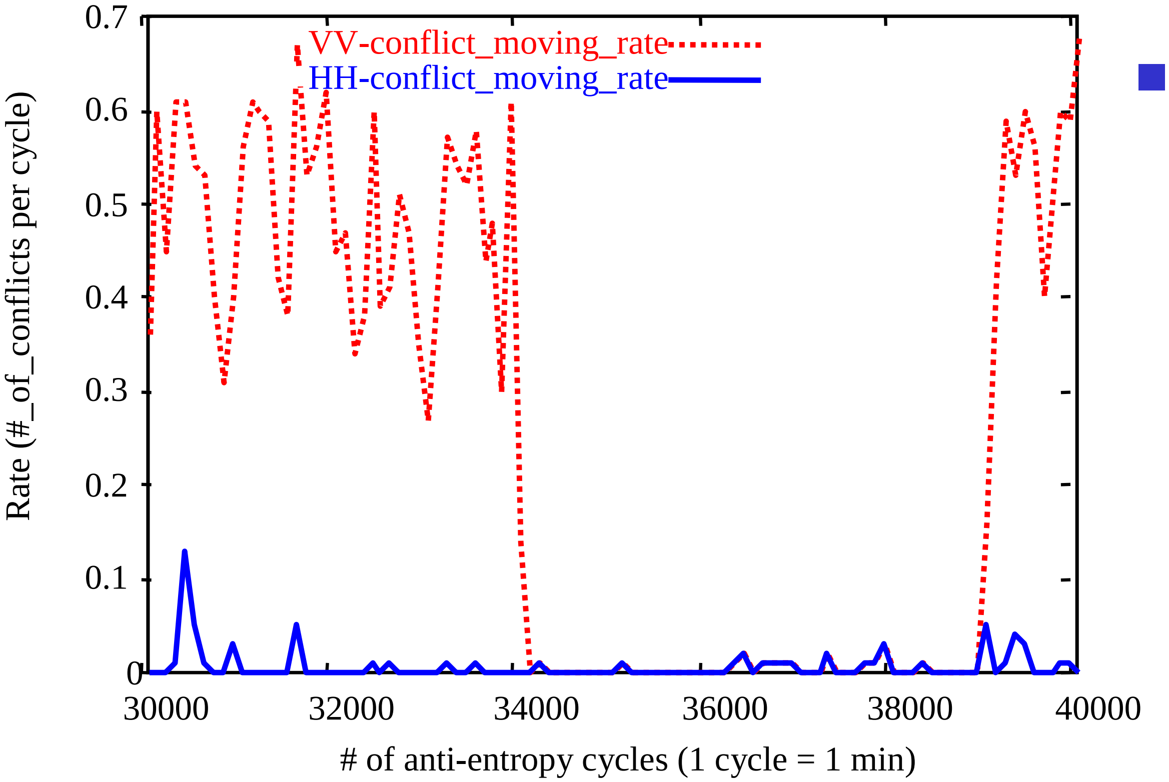
A	B	C	D
2	1	0	0

A	B	C	D
1	1	0	1

A	B	C	D
3	1	0	0

All descendants of x_3 at Site A dominate x_3 at Site D.

Coincidental Equalities with Vast Cumulative Effects



- Simulator using CVS trace data
  - 2281 CVS commit events for 12/1999-4/2002;
  - Total 64 number of users;
  - Inter-commit time: average: 237.8 min median: 34.6 min.

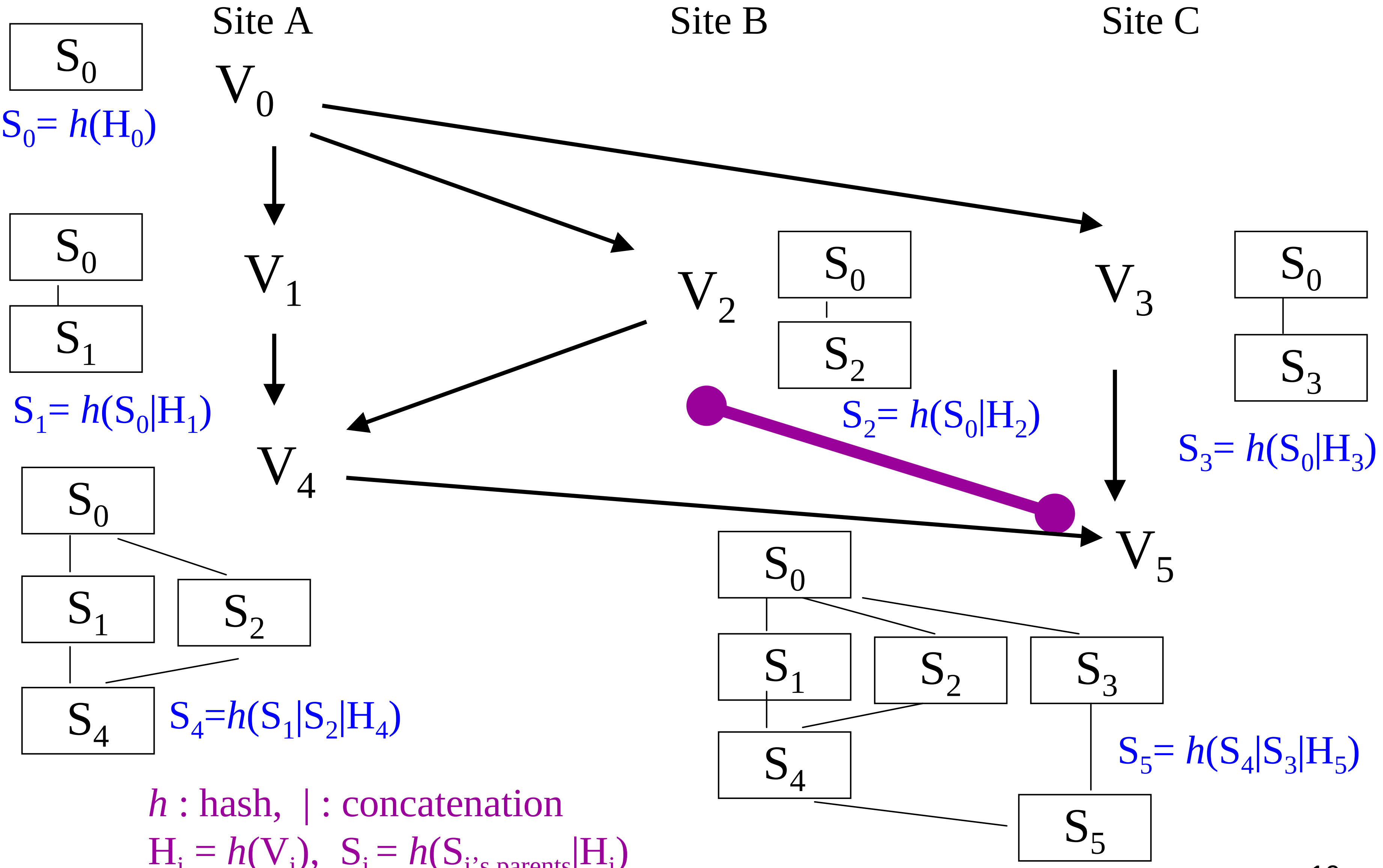
- Every 60sec, two randomly chosen sites perform reconciliation.
- Concurrent versions are merged deterministically.
- Track the result of dominance determination for each reconciliation.

- So, the answer is “Yes”, replicas can converge across partitioned networks with SHH.
- More information
  - http://www.cs.berkeley.edu/~hoon/
  - hoon@cs.berkeley.edu

Benefit: Fastest Convergence

- SHH converges faster by producing no false conflicts
  - Capturing coincidental equality
  - Are prevalent due to deterministic merging
- Convergence even in the network partition !!!
  - Each partition can merge into a version
  - SHH captures if these versions are the same or not, instantly without communication.

SHH Reconciliations



$S_0 = h(H_0)$

$S_1 = h(S_0|H_1)$

$S_2 = h(S_0|H_2)$

$S_3 = h(S_0|H_3)$

$S_4 = h(S_1|S_2|H_4)$

$S_5 = h(S_4|S_3|H_5)$

h : hash, | : concatenation

$H_i = h(V_i)$, $S_i = h(S_{i\text{'s parents}}|H_i)$

Background: Optimistic Replication

- Widely used in distributed systems
  - To achieve increased availability during
    - Network-Partition or Server Failure
  - Useful for collaboration across administrative domain
    - Difficult to set up a shared central server.
  - Bayou, Coda, Ficus, Pangaea, and Usenet.
- Replica updated in any place
  - Later, converges to a consistent state by reconciling independently accrued updates.
  - Typically through random pair-wise update exchanges for high availability. (i.e., anti-entropy)