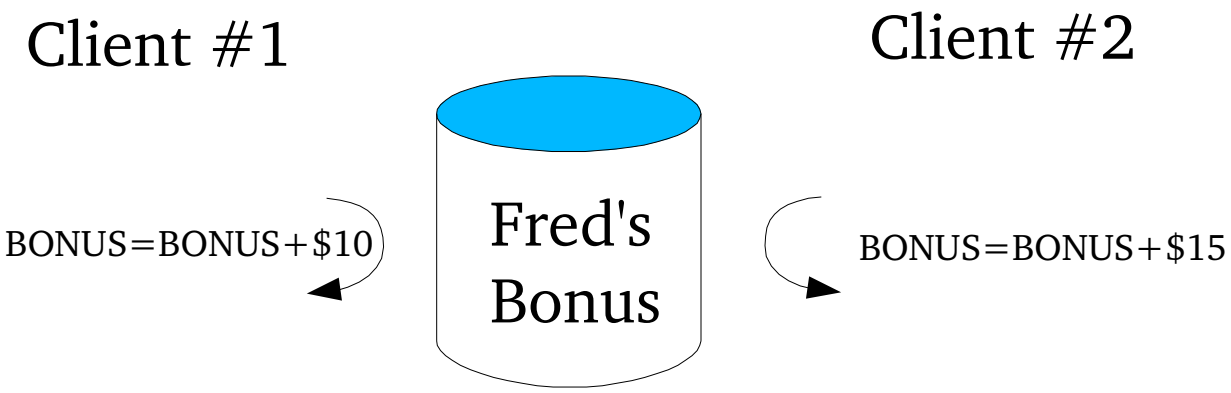
Replication and Transaction Management in a Temporal Database

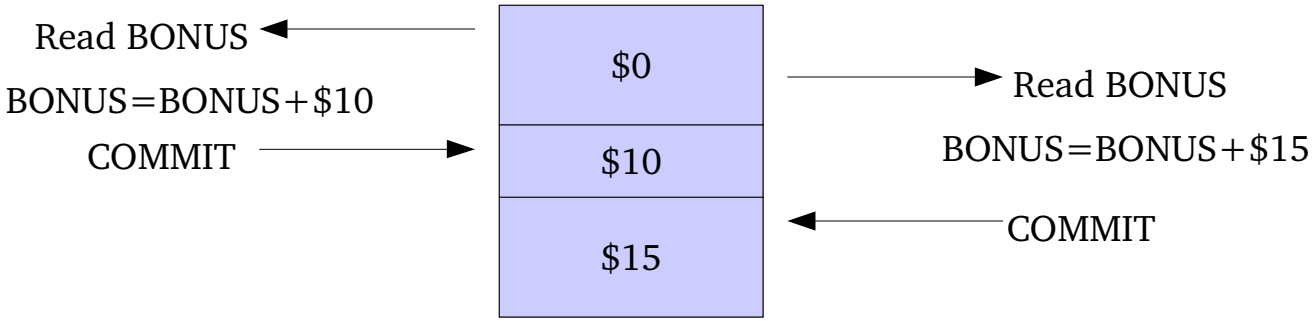
Matthew Dillon
Backplane, Inc.

Transactional Consistency



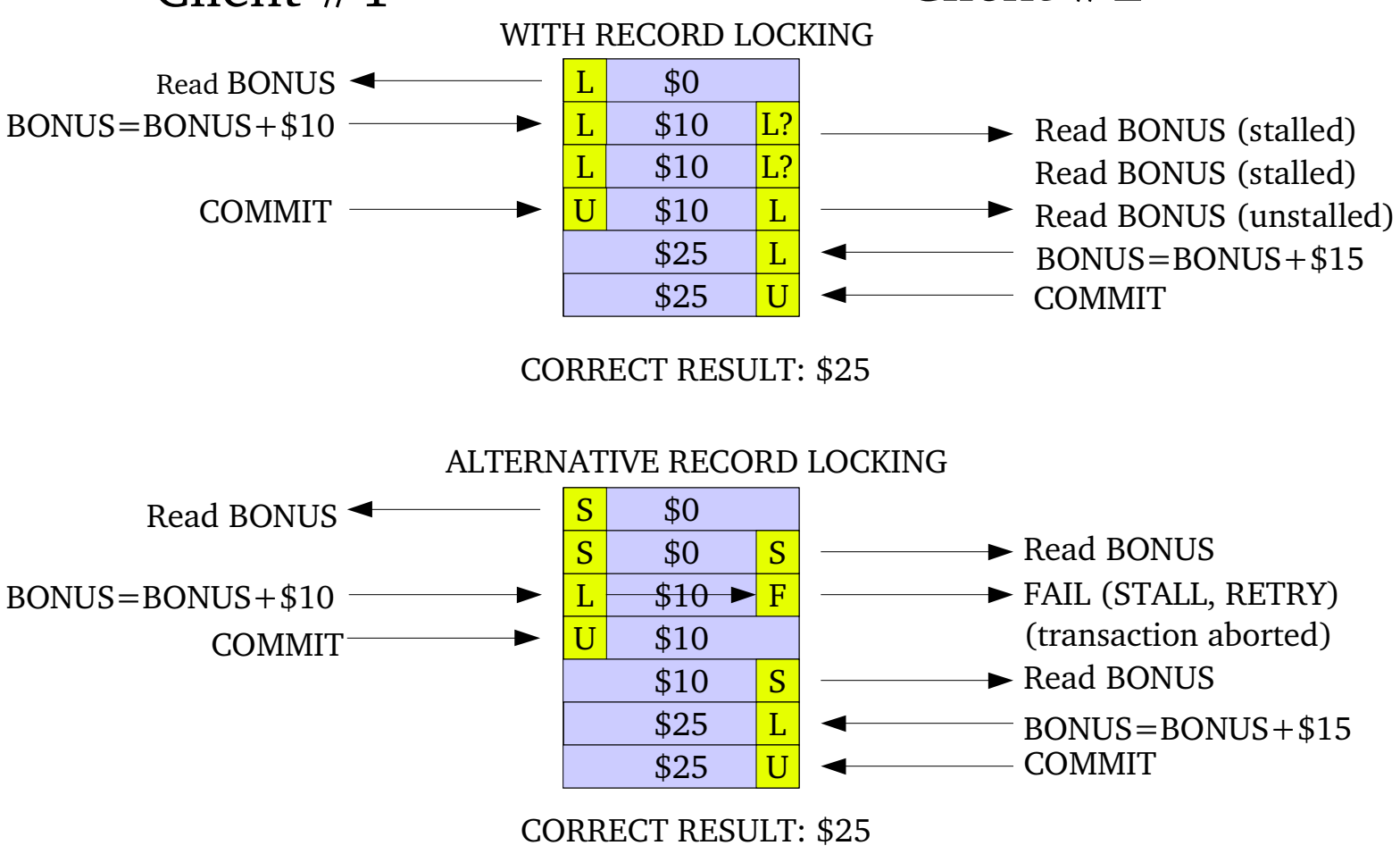
CORRECT RESULT: $25

INCONSISTENT/INCOHERENT (MY BONUS GOT LOST!)



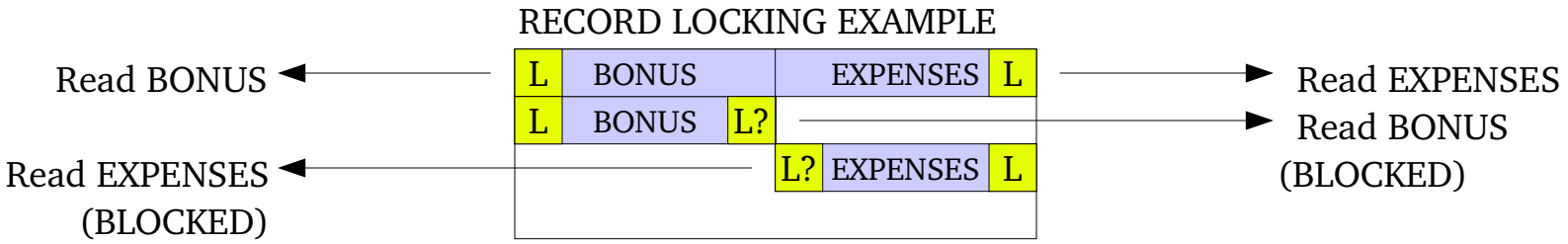
INCORRECT RESULT: $15

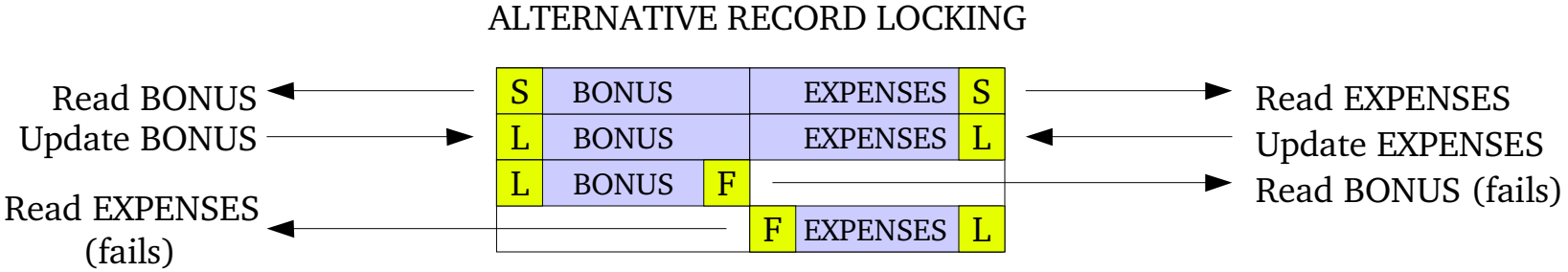
Enforcing Consistency with Record Locking



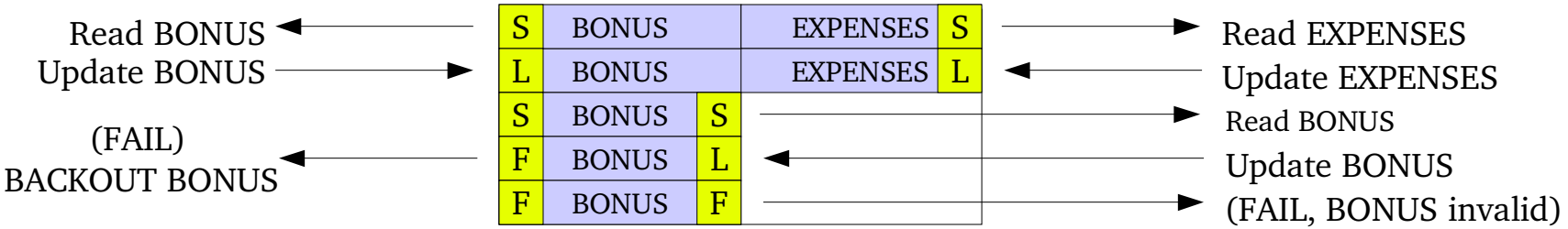
Deadlock/LiveLock Issues With Record Locking

Client #1 Client #2



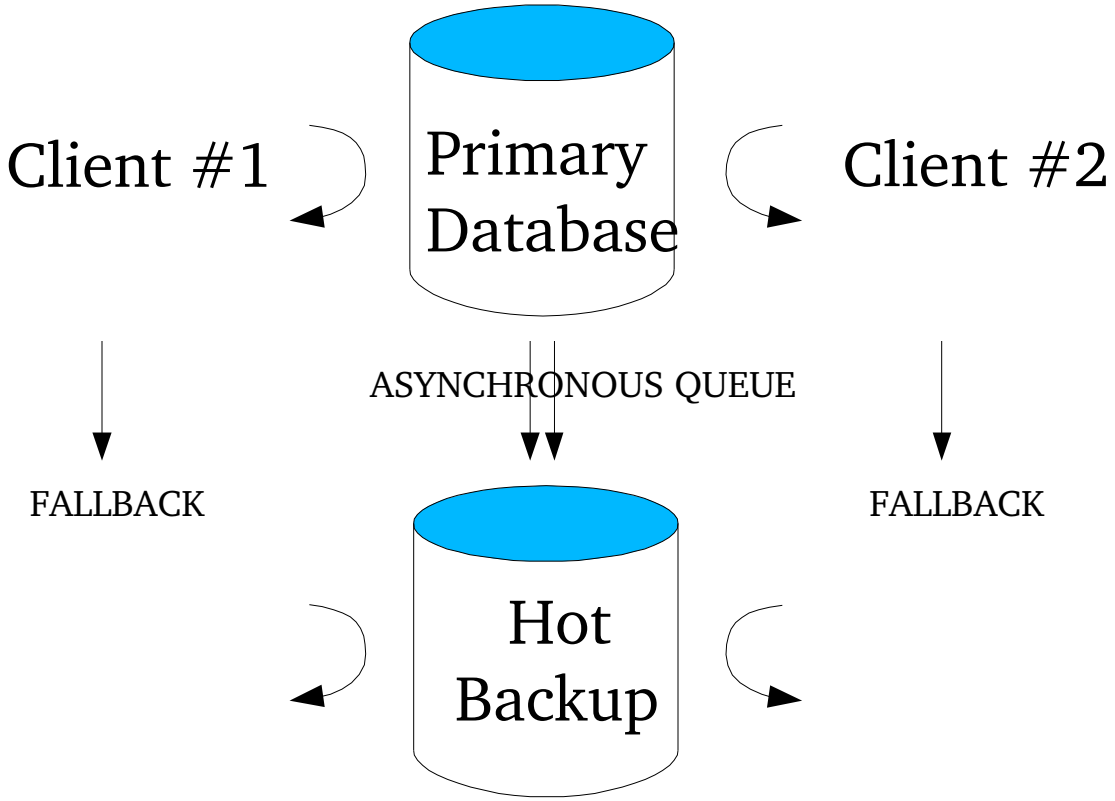


Possible Livelock. If we block instead of fail, guarenteed deadlock



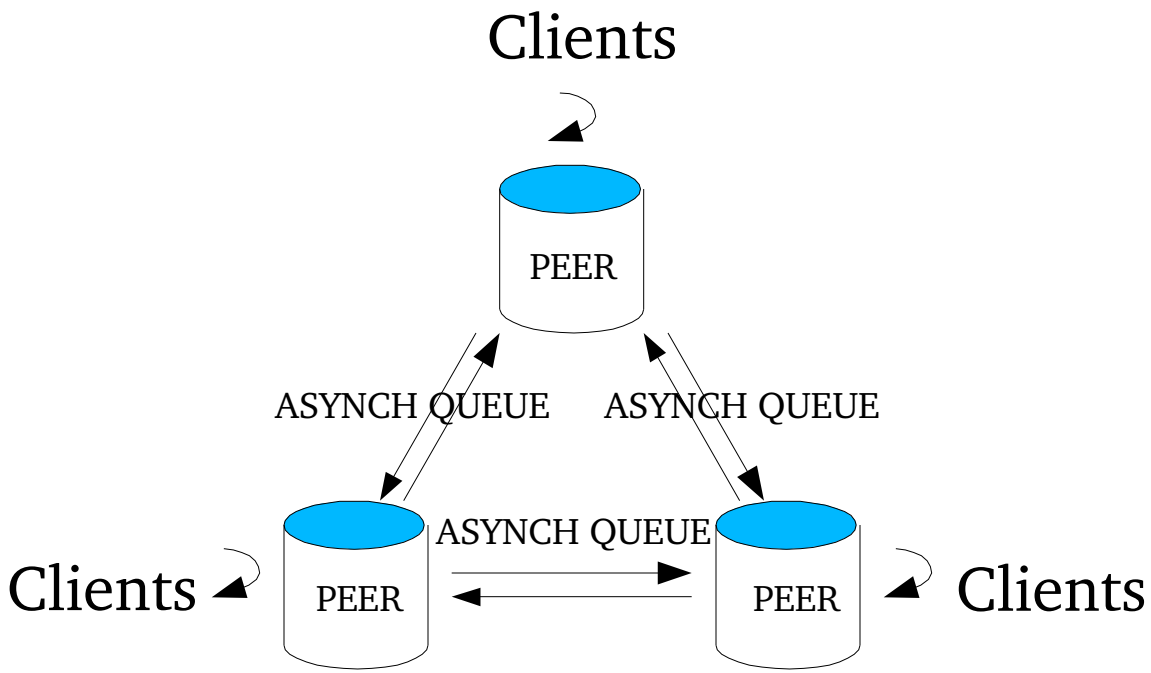
If we allow client #2 to use uncommitted data from client #1, and client #1 later fails, client #2 must then fail leading to a possible livelock.

Fallback Replication



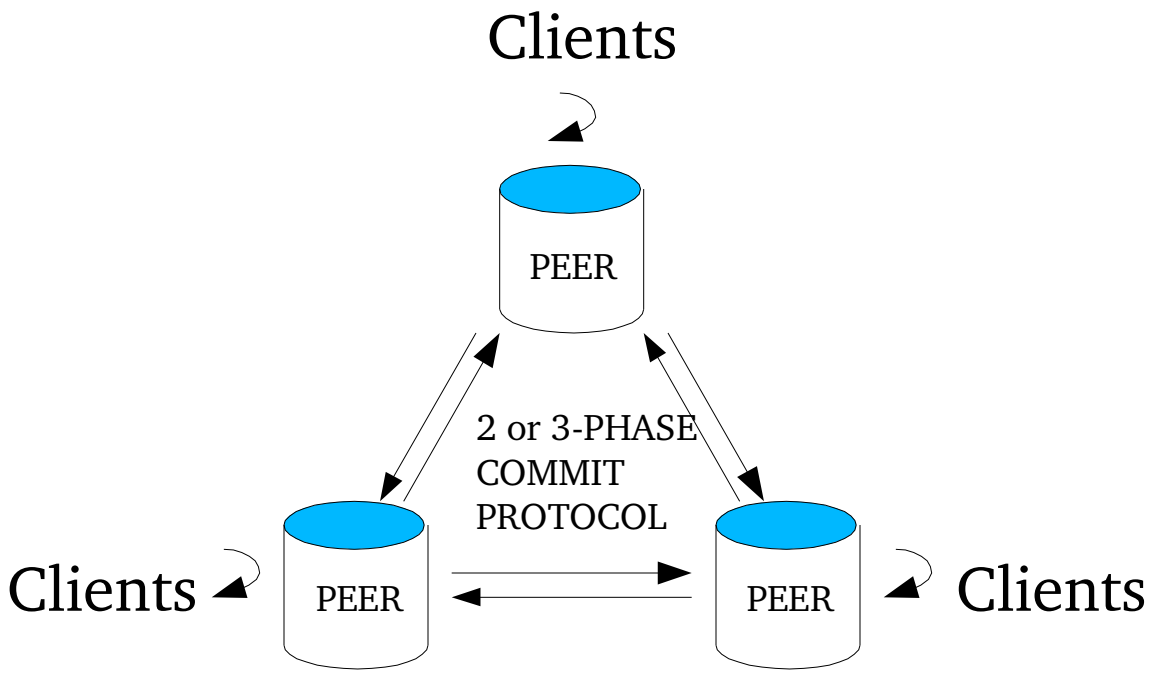
- FALLBACK TO HOT BACKUP ON FAILURE
- HOT BACKUP MAY NOT BE SO HOT
- COMMIT PERFORMANCE SAME AS SINGLE-DB CASE
- FALLFORWARD IS MORE PROBLEMATIC

Asynchronous non-Coherent Replication



- TRANSACTIONS CAN BE DISTRIBUTED
- BOTH READ-ONLY AND MODIFYING TRANSACTIONS SCALE
- COMMIT TO SINGLE PEER, REPLICATE TO OTHERS. COMMIT DOES NOT GUARENTEE CORRECTNESS
- CONFLICT RESOLUTION MUST OCCUR AFTER THE FACT

Fully Synchronous Coherent Replication



- TRANSACTIONS CAN BE DISTRIBUTED
- READ-ONLY TRANSACTIONS SCALE
- MODIFYING TRANSACTIONS MUST TALK TO ALL PEERS
- POTENTIALLY SERIOUS LOCKING & TIMEOUT ISSUES IF A PEER FAILS
- MANY IMPLEMENTATIONS REQUIRE A DESIGNATED 'MASTER' NODE

Temporal Database Table Structure

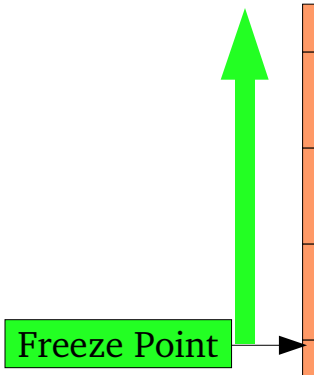
		TRANSID	KEY	VALUE	
INSERT	I	0830	BONUS	$0	INSERT BONUS = $0
UPDATE	D	0831	BONUS	$0	UPDATE BONUS=BONUS+$10
	I	0831	BONUS	$10	
UPDATE	D	0832	BONUS	$10	UPDATE BONUS=BONUS+$15
	I	0832	BONUS	$25	
DELETE	D	0833	BONUS	$25	DELETE BONUS

Freeze Point →

- LOCKLESS TRANSACTIONS ARE POSSIBLE
- HISTORICAL AS-OF QUERIES ARE POSSIBLE (INCLUDING META-DAT)A)
- EASIER TO BACKUP, RESTORE, AND ARCHIVE
- EASIER TO RECOVER CORRUPTED DATABASE
- APPEND-ONLY FILE STRUCTURE POSSIBLE
- LOCKLESS TRANSACTIONS CAN EXTEND TO REPLICATED PEERS USING TWO-PHASE COMMIT
- INCREMENTAL REPLICATION WITHOUT QUEUES POSSIBLE

- UNLESS VACUUMED, TABLE FILES GROW WITH EACH INSERT, UPDATE, OR DELETION
- LOTS OF 'DELETED' RECORDS CAN CLUTTER INDEXES AND TABLE DATA
- AN UPDATE APPENDS TWO RECORDS INSTEAD OF ONE
- A DELETE APPENDS ONE RECORD INSTEAD OF FLAGGING AN EXISTING RECORD

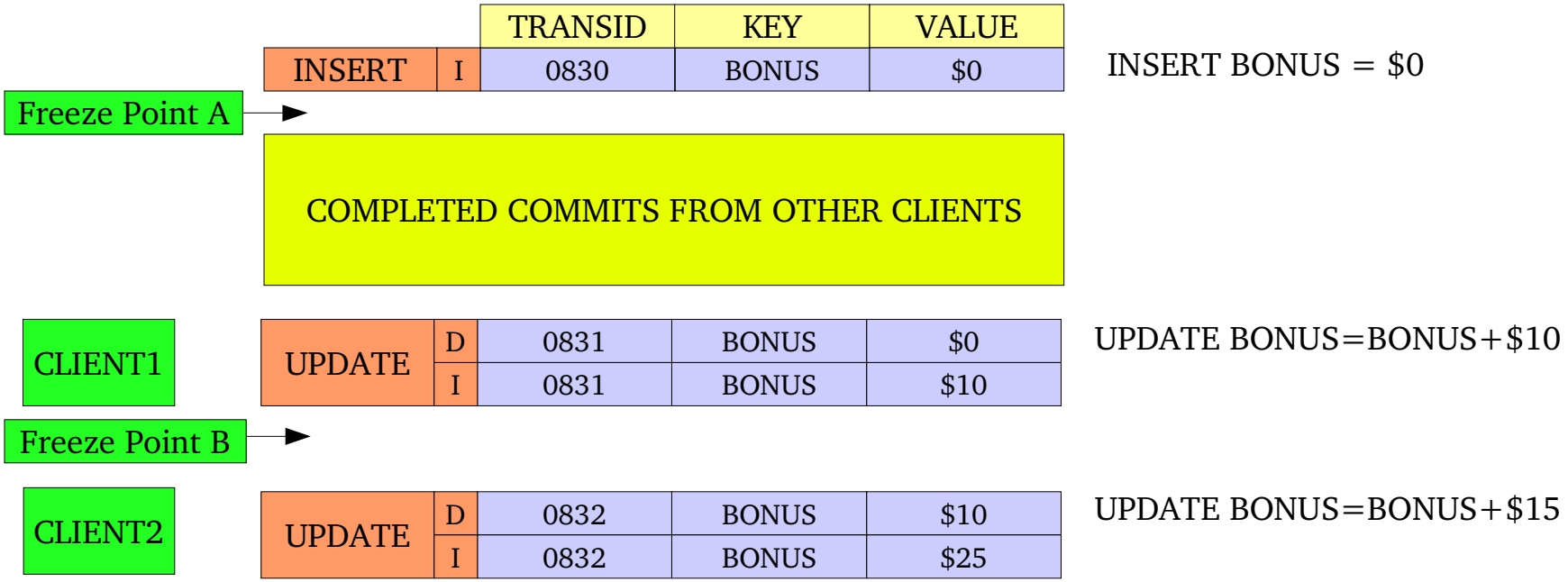
Reverse Scan Optimization



		TRANSID	KEY	VALUE	
INSERT	I	0830	BONUS	$0	INSERT BONUS = $0
UPDATE	D	0831	BONUS	$0	UPDATE BONUS=BONUS+$10
	I	0831	BONUS	$10	
UPDATE	D	0832	BONUS	$10	UPDATE BONUS=BONUS+$15
	I	0832	BONUS	$25	
UPDATE	D	0832	BONUS	$25	UPDATE BONUS=BONUS+$15
	I	0832	BONUS	$40	
UPDATE	D	0832	BONUS	$40	UPDATE BONUS=BONUS+$15
	I	0832	BONUS	$55	
DELETE	D	0833	BONUS	$55	DELETE BONUS

- IF DATA IS KNOWN TO BE UNIQUE, ONLY ONE RECORD NEEDS TO BE SCANNED
- DELETIONS CAN BE PAIRED WITH INSERTS MORE EFFICIENTLY WITH A REVERSE SCAN

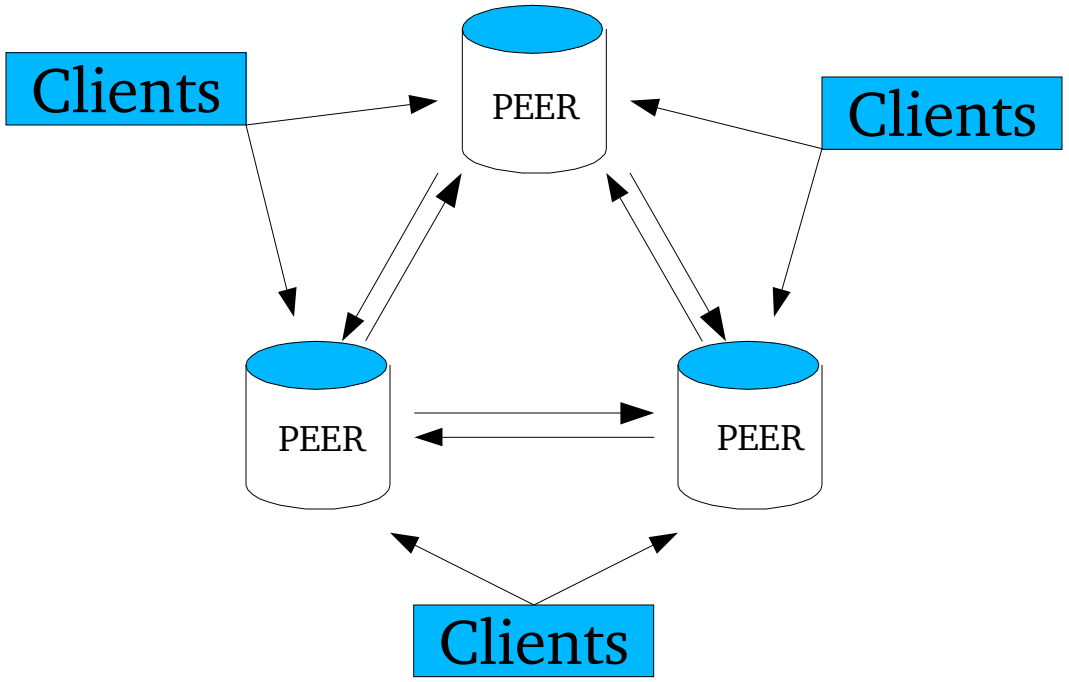
Lockless Transactions



- SIMULTANEOUS TRANSACTIONS BY CLIENTS 1 AND 2
- CLIENT QUERIES RELATIVE TO FREEZE POINT A
- MODIFICATIONS MADE TO TEMPORARY TABLES
- CLIENT COMMIT-PHASE-1 COPIES TEMPORARY TABLE TO DATA SPACE
- CLIENT COMMIT-PHASE-1 RERUNS QUERIES WITH FREEZE POINT TEMPORARILY MOVED
- ANY DATA ACCESSED BETWEEN FREEZE POINT A AND DATA COPY INDICATES CONFLICT
- (NON-REPLICATED) AT LEAST ONE CLIENT ALWAYS GUARANTEED TO SUCCEED
- COMMIT-PHASE-2 SETS NEW ADDITION TO DATA SPACE IN STONE
- MULTIPLE CLIENTS CAN COMPLETE COMMIT-PHASE-2 OUT OF ORDER

- QUERIES ARE RUN TWICE

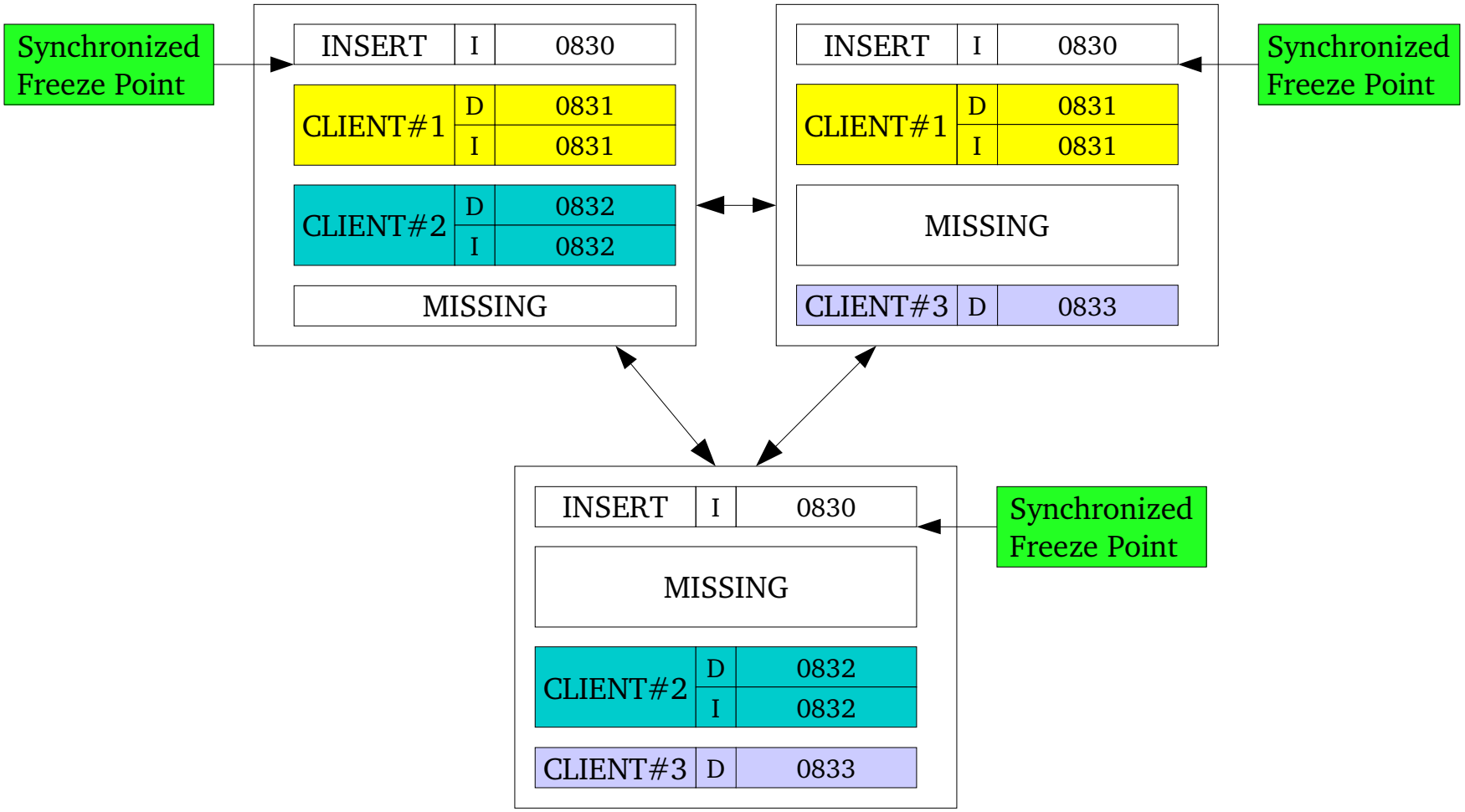
Quorum Based Commit



- TRANSACTIONS ARE RELATIVE TO A FREEZE POINT.
- SELECTED QUORUM MUST BE SYNCHRONIZED TO THE SPECIFIED FREEZE POINT
- COMMIT ONLY NEEDS TO OCCUR ON A QUORUM OF PEERS
- SYNCHRONIZED FREEZE POINT IS NOT UPDATED BY COMMIT
- REMAINING PEERS AND SNAPSHOTS GET UPDATED VIA REPLICATION
- SYNCHRONIZED FREEZE POINT IS UPDATED VIA REPLICATION

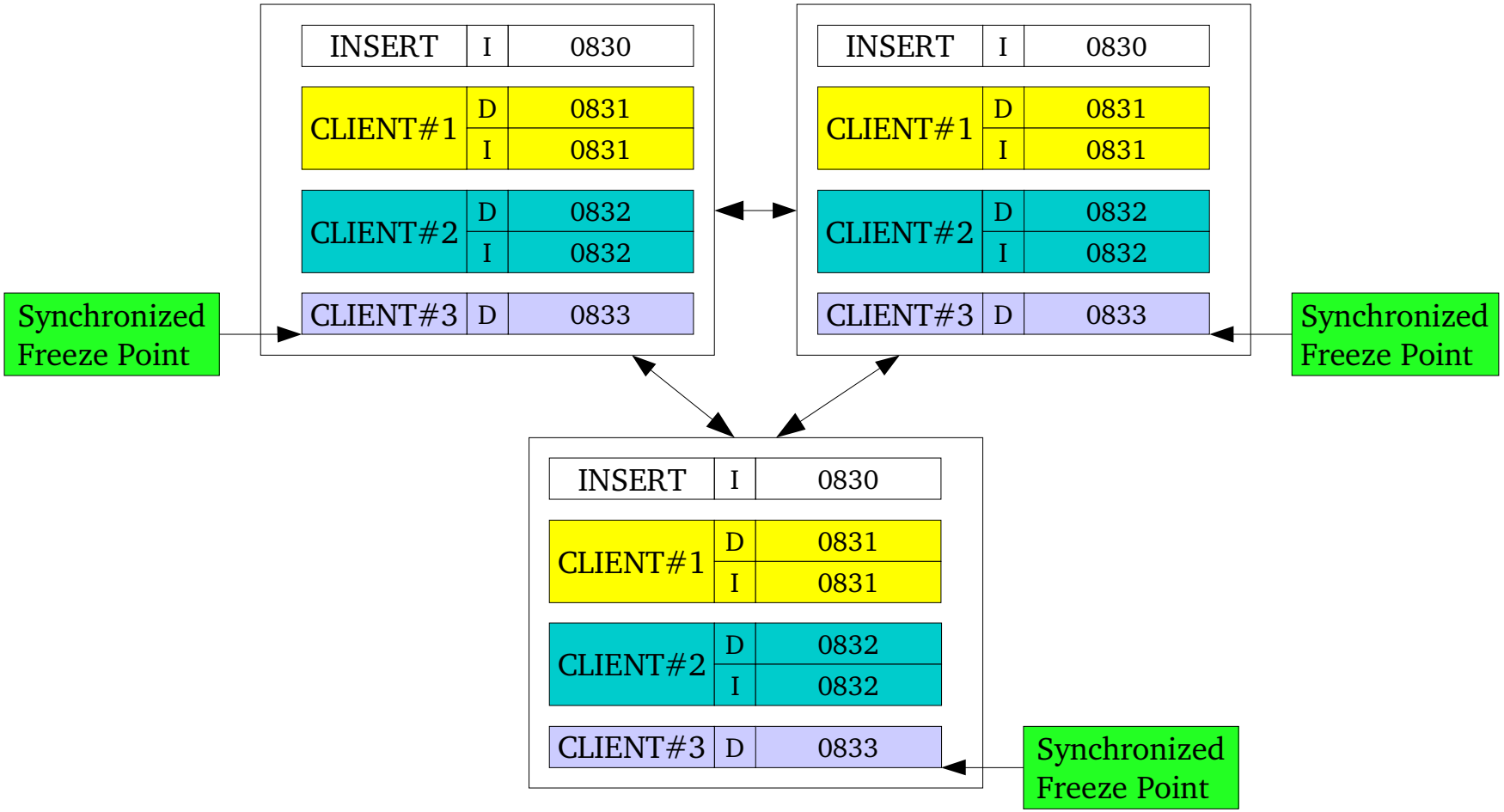
- PARTITIONING A PROBLEM IN WAN TOPOLOGIES

Quorum Based Replication (BEFORE)



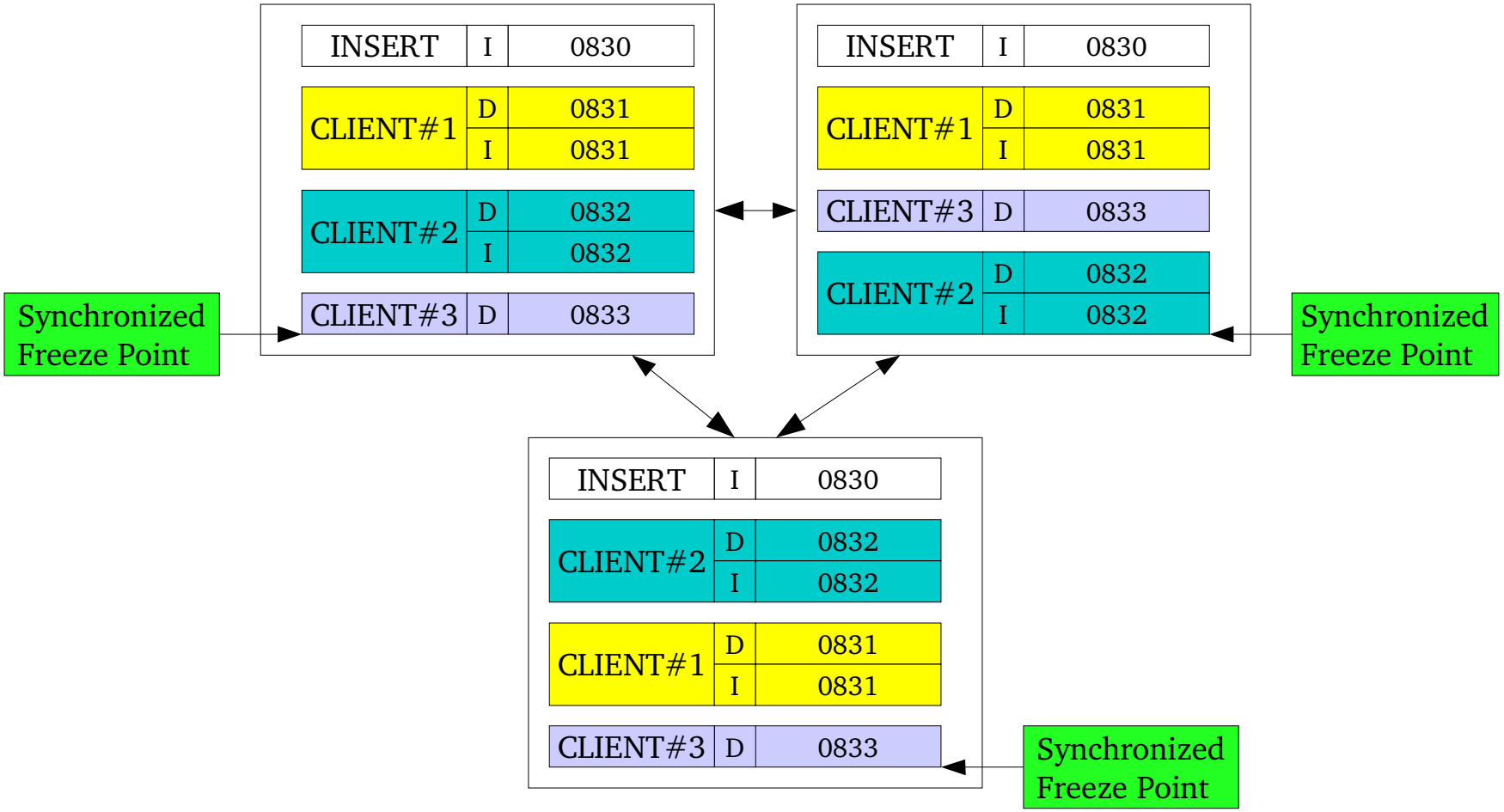
NOTE: CLIENTS MADE NON-CONFLICTING COMMITS

Quorum Based Replication (AFTER)



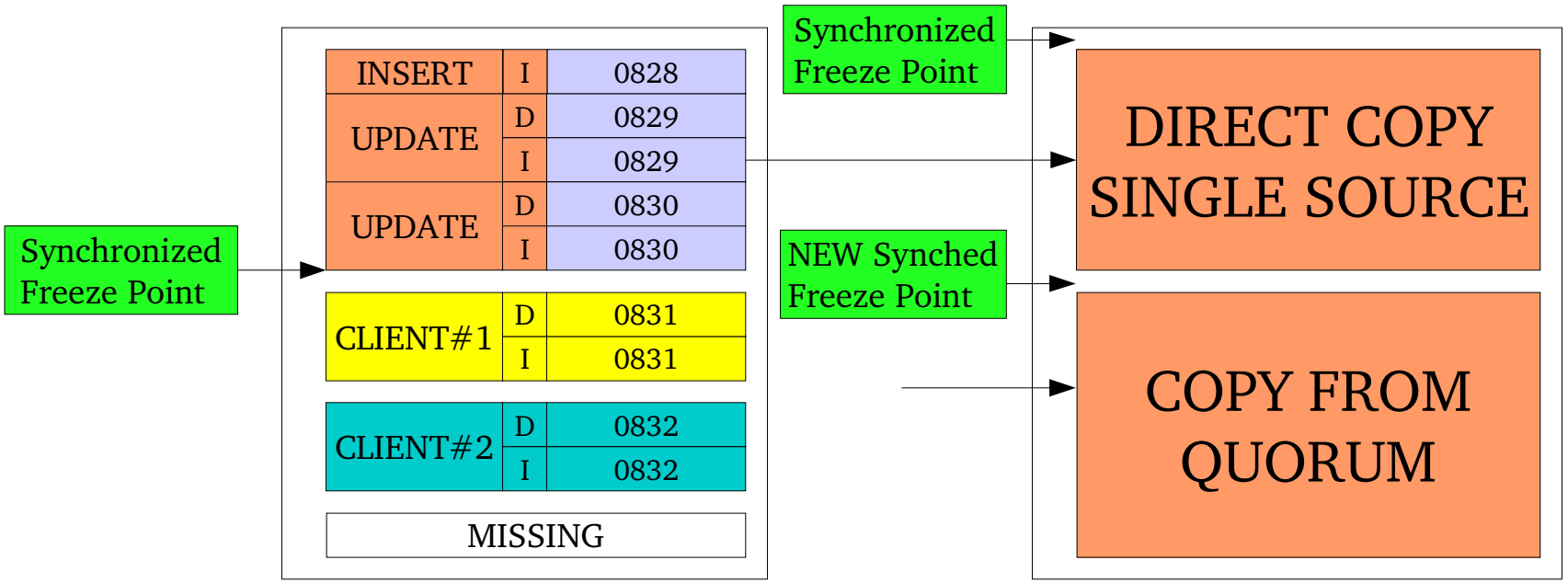
- SYNCHRONIZED FREEZE POINT UPDATED WITH QUORUM

Out of Order Quorum-Based Replication (AFTER)



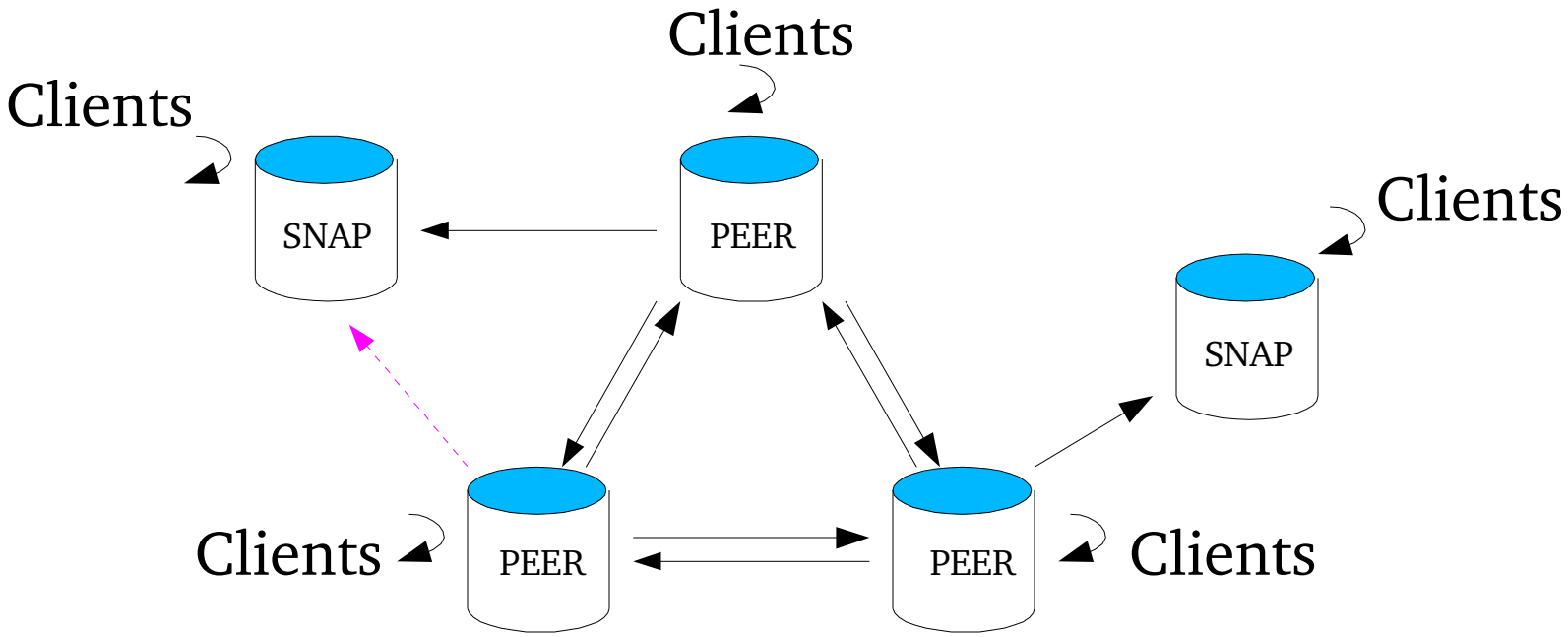
- ONLY NON-CONFLICTING TRANSACTIONS WILL REPLICATE OUT OF ORDER
- GENERALLY HARMLESS
- NO EFFECT ON REVERSE SCAN OPTIMIZATION
- TRANSACTION ID'S NOT MONOTONIC (SOLUTION: INDEX TRANSACTION ID'S)

Optimizing the Replication



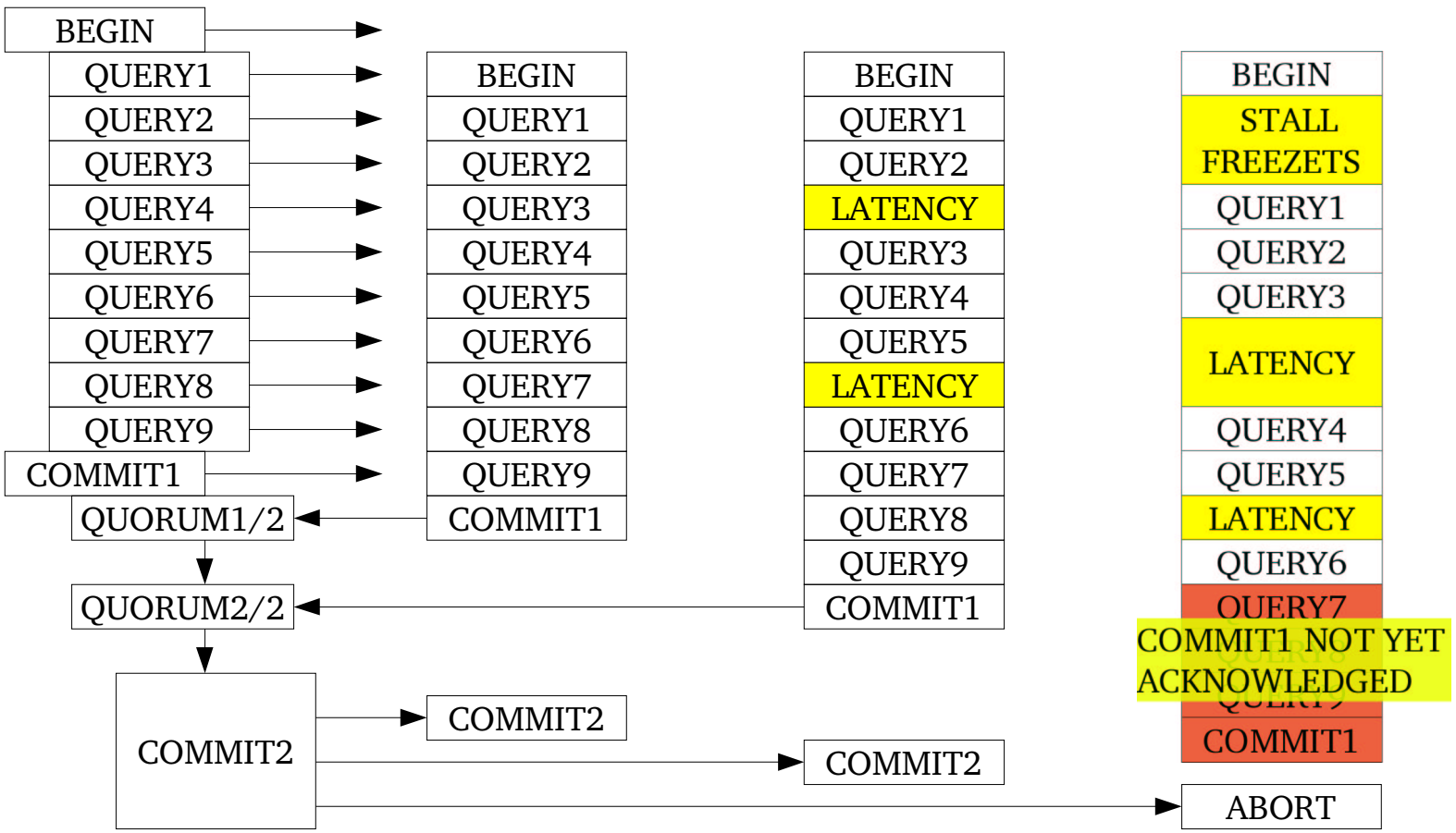
- DIRECT COPY FROM A SINGLE SOURCE UP TO THE EXISTING SYNCHRONIZATION POINT
- USE THE CLOSEST SOURCE, OR THE MOST COMPLETE SOURCE?
- QUORUM BASED REPLICATION FOR THE REMAINDER

Replicating a Historical Database



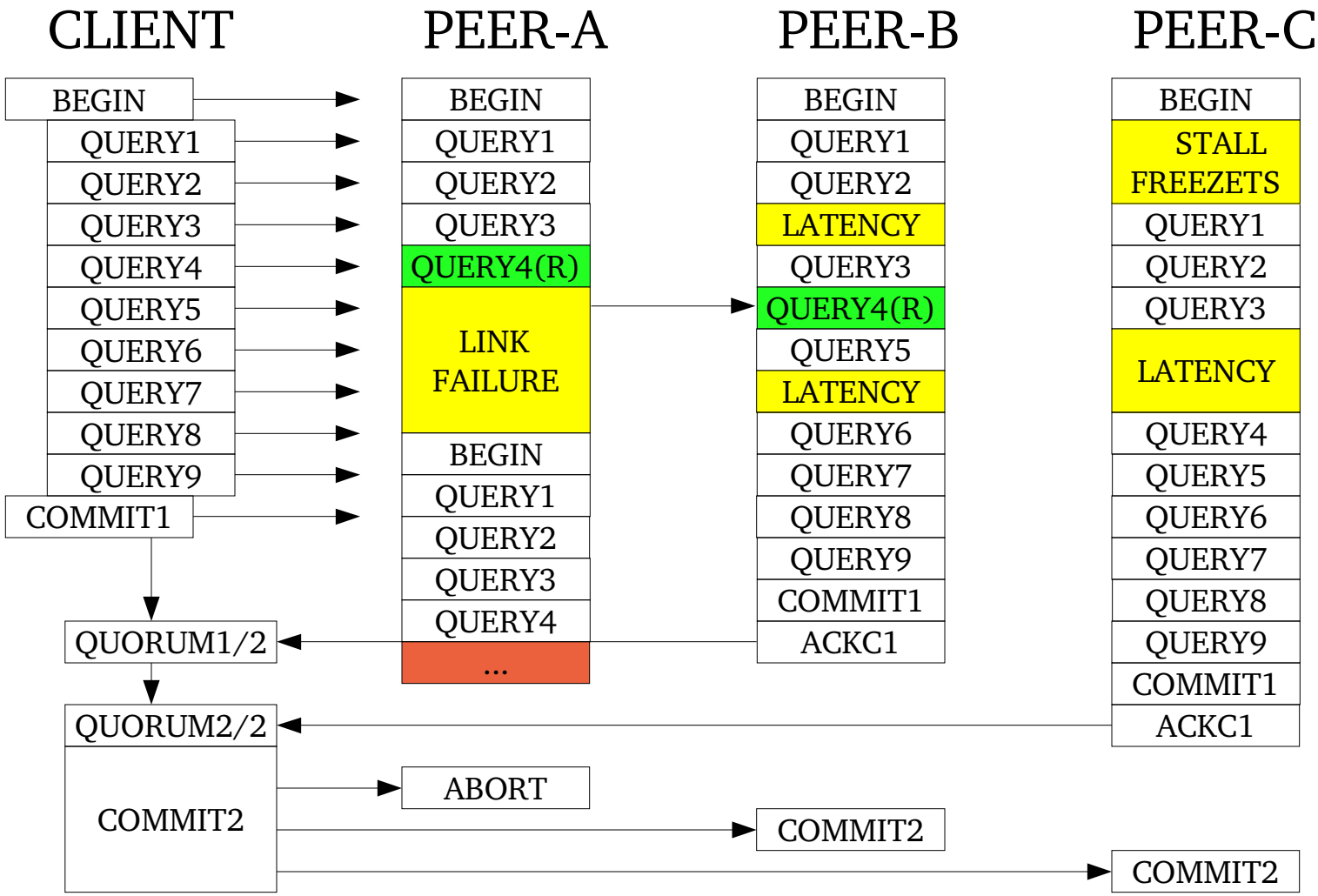
- CLIENTS MAY TALK TO A SINGLE NODE
- DATABASE SERVICES AT NODE ABSTRACT-OUT QUERY/COMMIT PROTOCOL
- REPLICATION PROCESS IS INDEPENDENT FROM QUERY/COMMIT PROCESS
- SPANNING TREE PROTOCOL REDUCES THE EFFECT OF LINK FAILURES
- PARTICIPATION DEPENDS ON FREEZE POINT / SYNCHRONIZATION OF NODES
- FIRST-RESPONDING-QUORUM MINIMIZES LATENCY
- AUTOMATIC QUERY RESTART IF LINK FAILURE INTERRUPTS RESPONDENT
- NATIVE REPLICATION POSSIBLE (NO QUEUEING)
- FULL-ON BACKUP LINKS POSSIBLE
- HIGH-LATENCY SNAPSHOT/BACKUP LINKS POSSIBLE

Query Management in a Replicated Environment



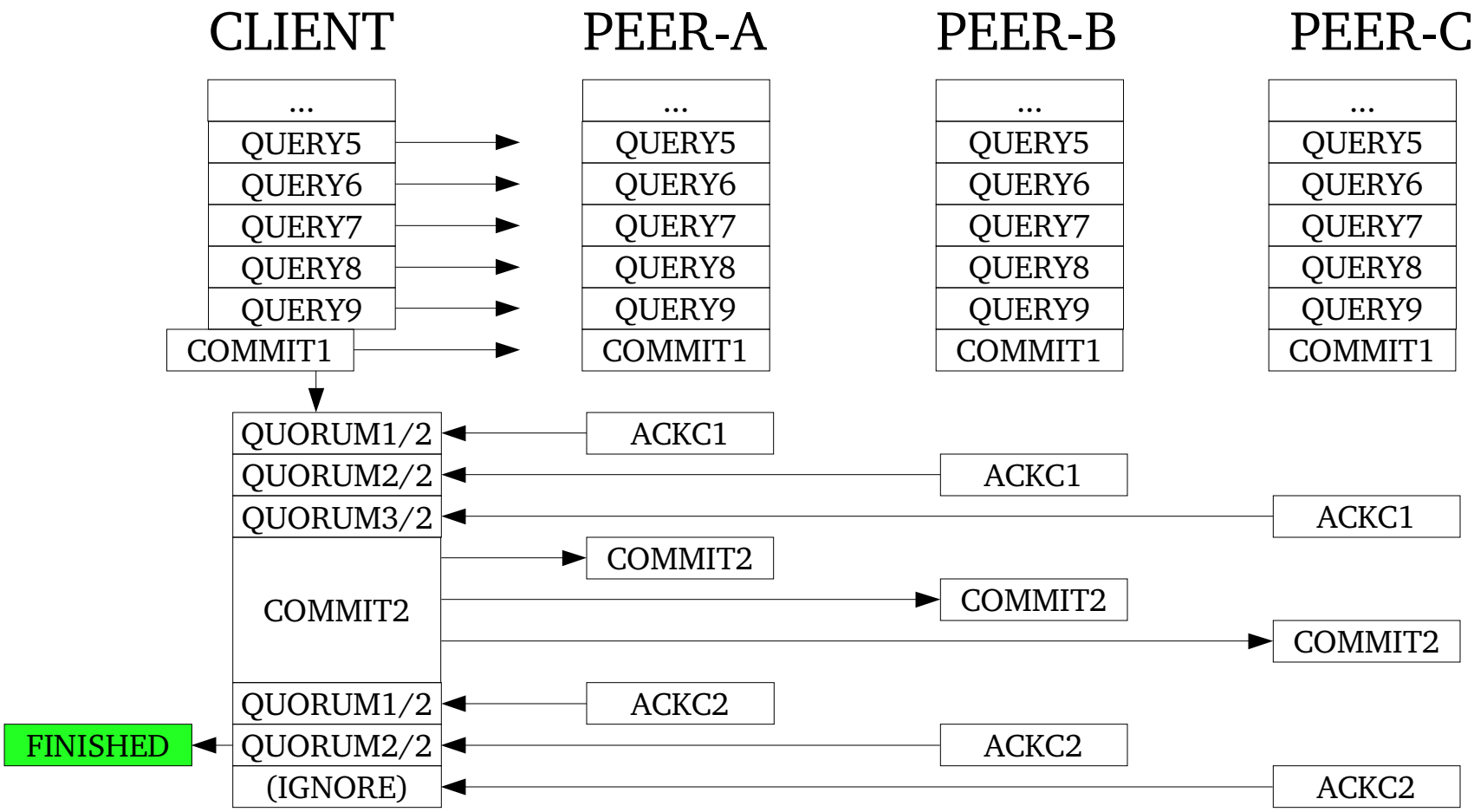
- MUST ABORT PEERS WHICH HAVE NOT YET ACKED COMMIT-1 AFTER SENDING COMMIT-2
- ONLY ONE PEER NEEDS TO RETURN QUERY RESULTS, BUT ALL MUST RECORD QUERIES
- CLIENT CAN CONTINUE BASED ON FIRST RESPONSE, COMMIT-2 WHEN QUORUM REACHED
- SOME PEERS CAN RETURN FAILURE LEGALLY AS LONG AS QUORUM RETURNS SUCCESS

Temporary Link Failures / Query Restarts Prior to Commit



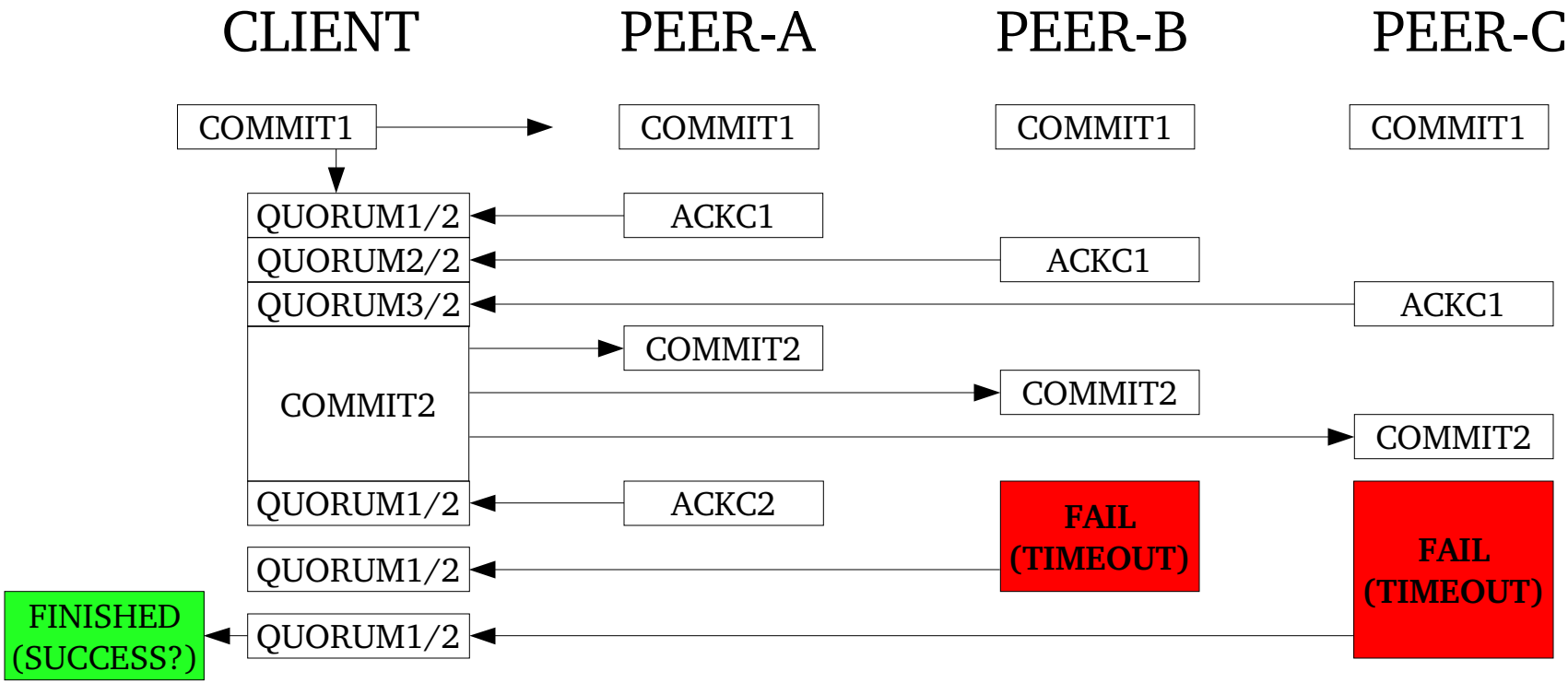
- PARTIAL RESULTS MAY HAVE TO BE THROWN OUT IF REAQUIRING FROM A NEW PEER
- NO HICUPS, NEXT LOWEST-LATENCY PEER CAN COMPLETE THE TRANSACTION
- NO LOCKING OVERHEAD WITHIN TRANSACTION BODY – RESTARTS EASY

Commit Phase – A Quorum is only a Minimum



- FINAL COMPLETION TO CLIENT CAN OCCUR AFTER QUORUM'S WORTH OF COMMIT-2 ACKS
- SENDING COMMIT-2 TO MORE THEN A QUORUM IMPROVES ROBUSTNESS AND SAFETY
- ASYNCHRONOUS BACKGROUND REPLICATION IS THE KEY TO FLEXIBILITY
- REMEMBER, MUST THROW AWAY ACKC1's AFTER FIRST COMMIT-2 SENT

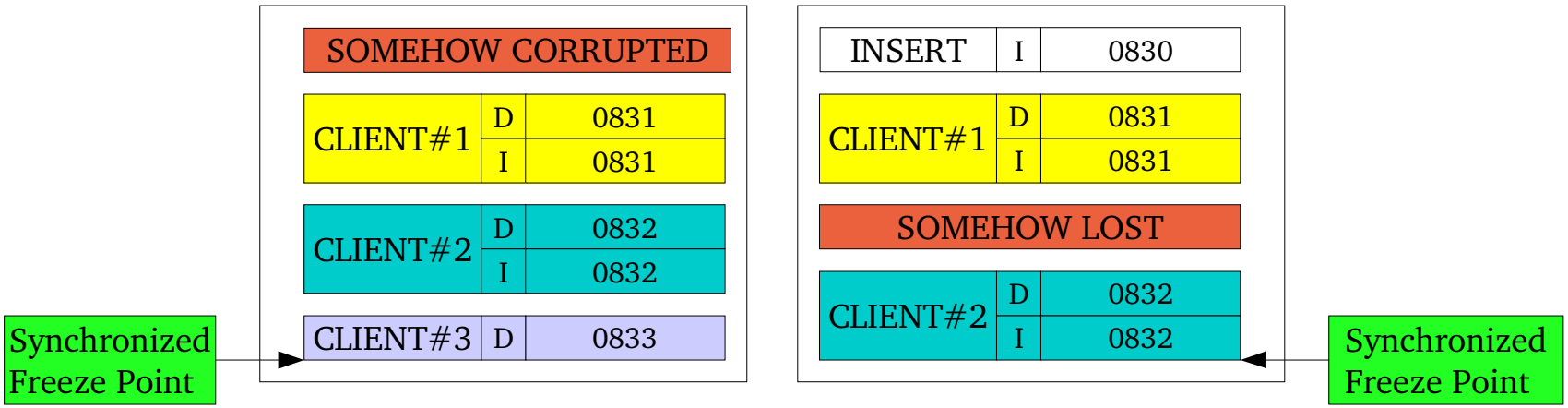
Failures During the Commit Phase



- DATA MAY NOT HAVE BEEN COMMITTED TO A QUORUM EVEN THOUGH WE MEANT TO
- WE CANNOT RESTART THE TRANSACTION ON B AND C DUE TO ASYNCHRONOUS REPLICATION
- QUORUM (B AND C) MAY MOVE THEIR SYNCHRONIZATION POINT PAST THE TRANSACTION
- IF SYNCH POINT MOVED ON B AND C, REPLICATION FROM PEER-A MIGHT NOT OCCUR

- B AND C KNOW THAT A TRANSACTION WAS IN PROGRESS, STILL HAVE THE COMMIT-1 DATA
- B AND C DO NOT KNOW THE TRANSACTION ID IF THEY DID NOT GET THE COMMIT-2
- REPLICATION FAILURE OR RECOVERY CONFLICT CAN STILL BE DETECTED AFTER THE FACT

Detecting Data Corruption



- CRC RANGE OF (SORTED) TRANSACTION ID'S AND CHECK AGAINST ALL OTHER COPIES
- UNIQUE DATA NOT UNIQUE
- TABLE CONSTRAINTS FAIL
- INDEXES MISS SOME OF RECORDS
- LOG DOES NOT MATCH DATA
- (RUN TIME) ORDERED QUERY RESULTS DO NOT AGREE