



Transparent Fault-Tolerance

Mechanisms and Design Principles

Thomas C. Bressoud

Lucent Technologies
Bell Labs Innovations



Collaborators



Lucent Technologies
Bell Labs Innovations

- TFT

- Robert Cooper
- Ken Birman
- Brad Glade
- Ian Service

- FT-TCP

- Lorenzo Alvisi
- Keith Marzullo
- Dmitrii Zagorodnov

Overview



Lucent Technologies
Bell Labs Innovations

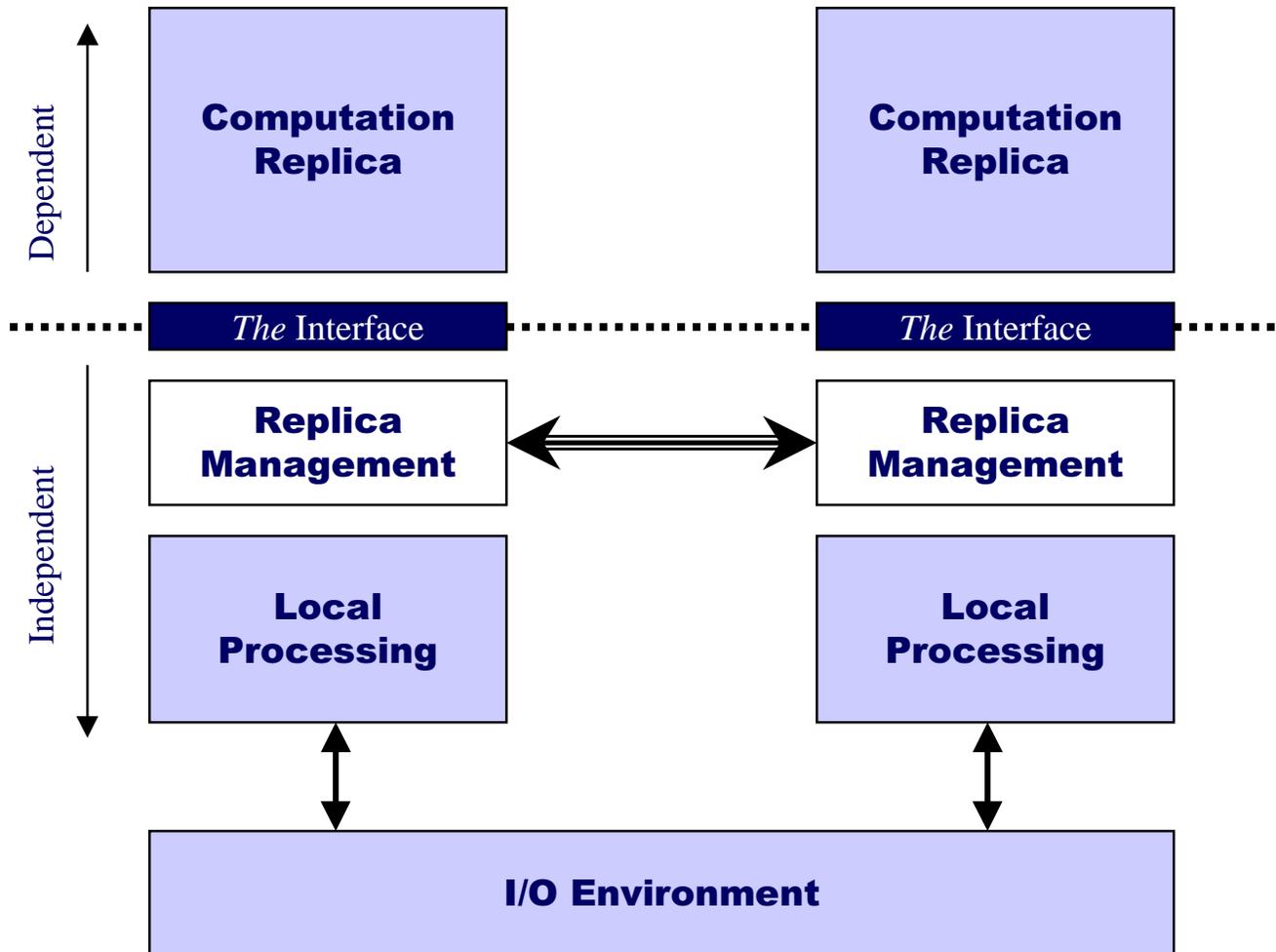
- Objective:
 - ◆ To build a fault-tolerant system whose effect on the application developer is minimized.
 - ◆ Support independent OS faults as well as HW fault coverage.

- In this talk ...
 - ◆ General approach to transparent replication
 - ◆ TFT-specific solution
 - ◆ Contending with Operating System State
 - Fault-Tolerant TCP

General Approach



Lucent Technologies
Bell Labs Innovations



Sources of Non-Determinism



Lucent Technologies
Bell Labs Innovations

- Timing
 - ◆ Identical actions from identical computation state take different time to complete on different replicas
- Control
 - ◆ Given identical computation state, the next action executed may differ between different replicas
- Data
 - ◆ An identical action from identical computation state results in a different transformation in state on different replicas

Obligations of Replica Management



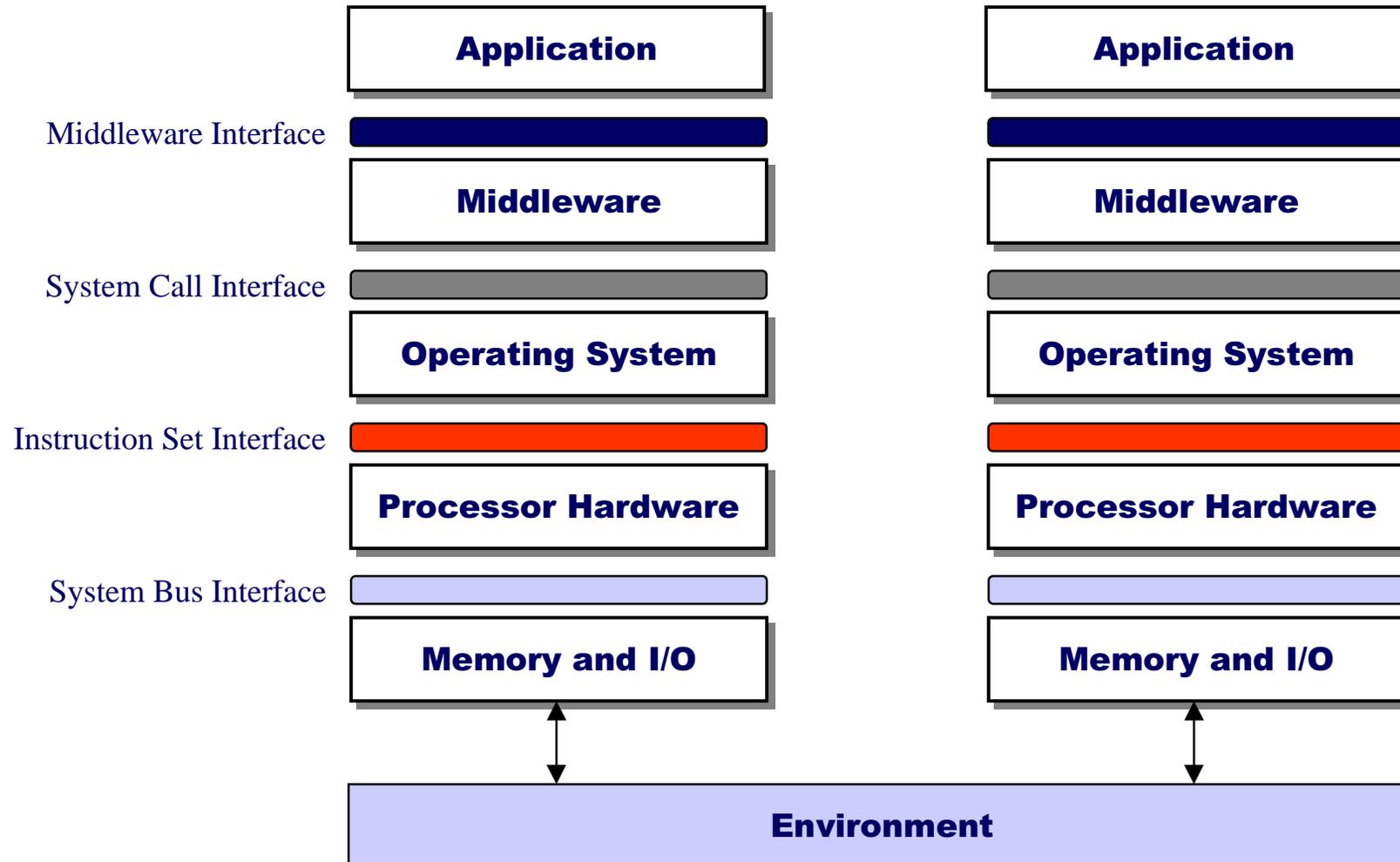
Lucent Technologies
Bell Labs Innovations

- Same initial state
- Same actions in the same relative order
- Identical transformation in state for each action
- Single correct output to the environment
- Mask effects of a failover
 - ◆ “Local Processing” State vs. Environment State

Other Approaches



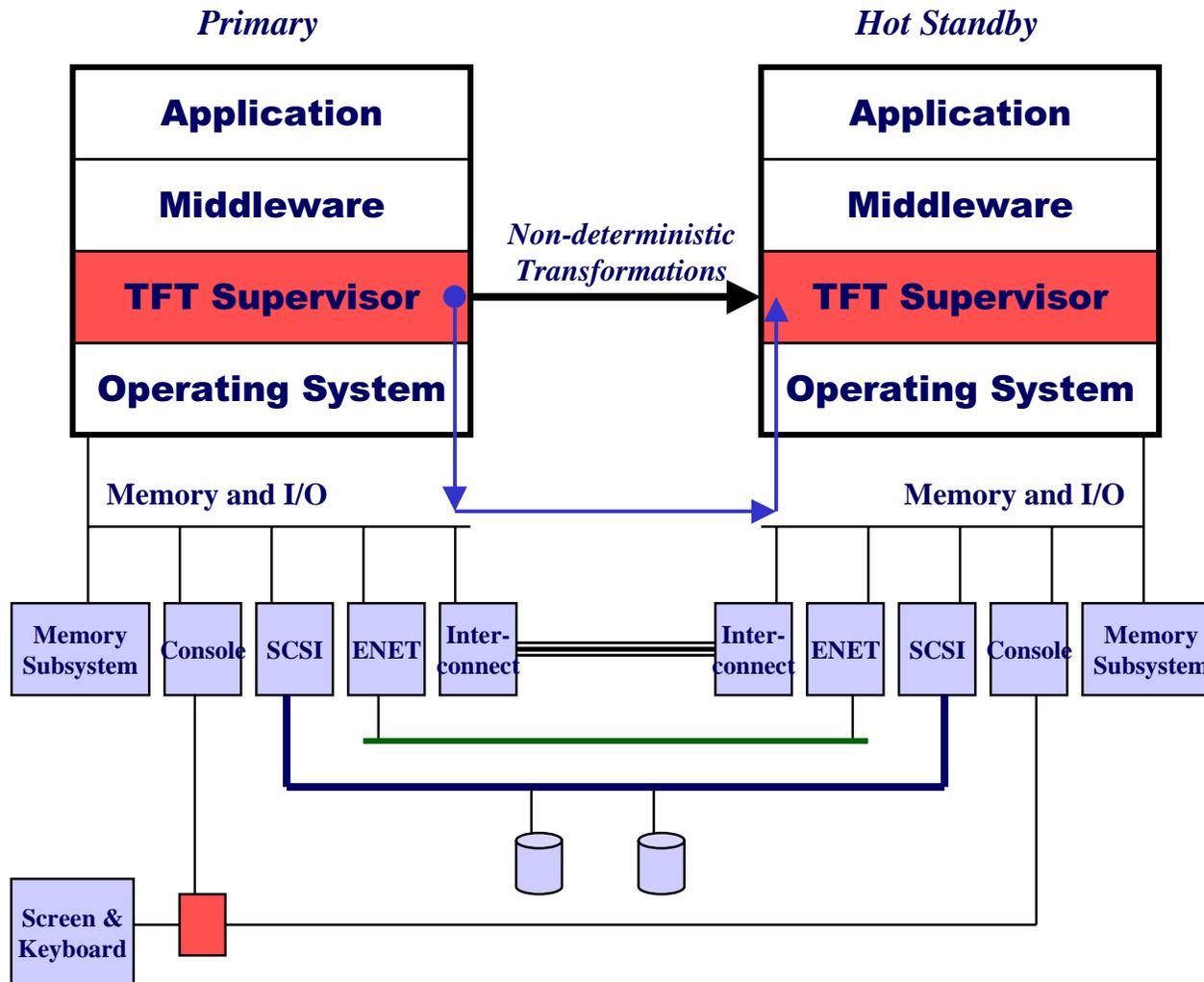
Lucent Technologies
Bell Labs Innovations



The TFT Solution



Lucent Technologies
Bell Labs Innovations



Actions of the User/Kernel Interface



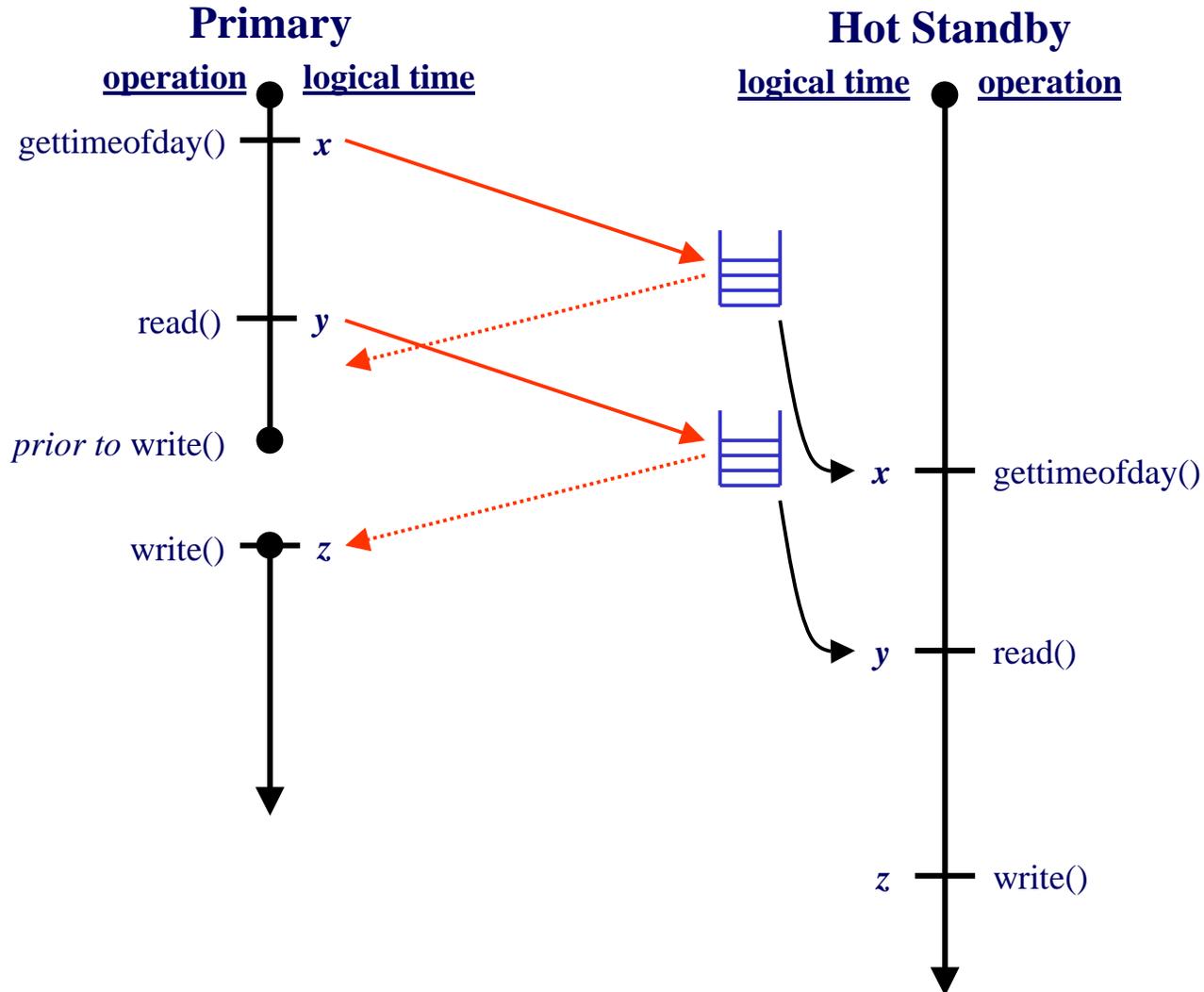
Lucent Technologies
Bell Labs Innovations

- Operations
 - ◆ Non-privileged Instructions
 - ◆ System Call
 - Deterministic
 - Non-deterministic
- Exceptions
 - ◆ Signals
 - ◆ Asynchronous Notifications

Non-Deterministic System Calls



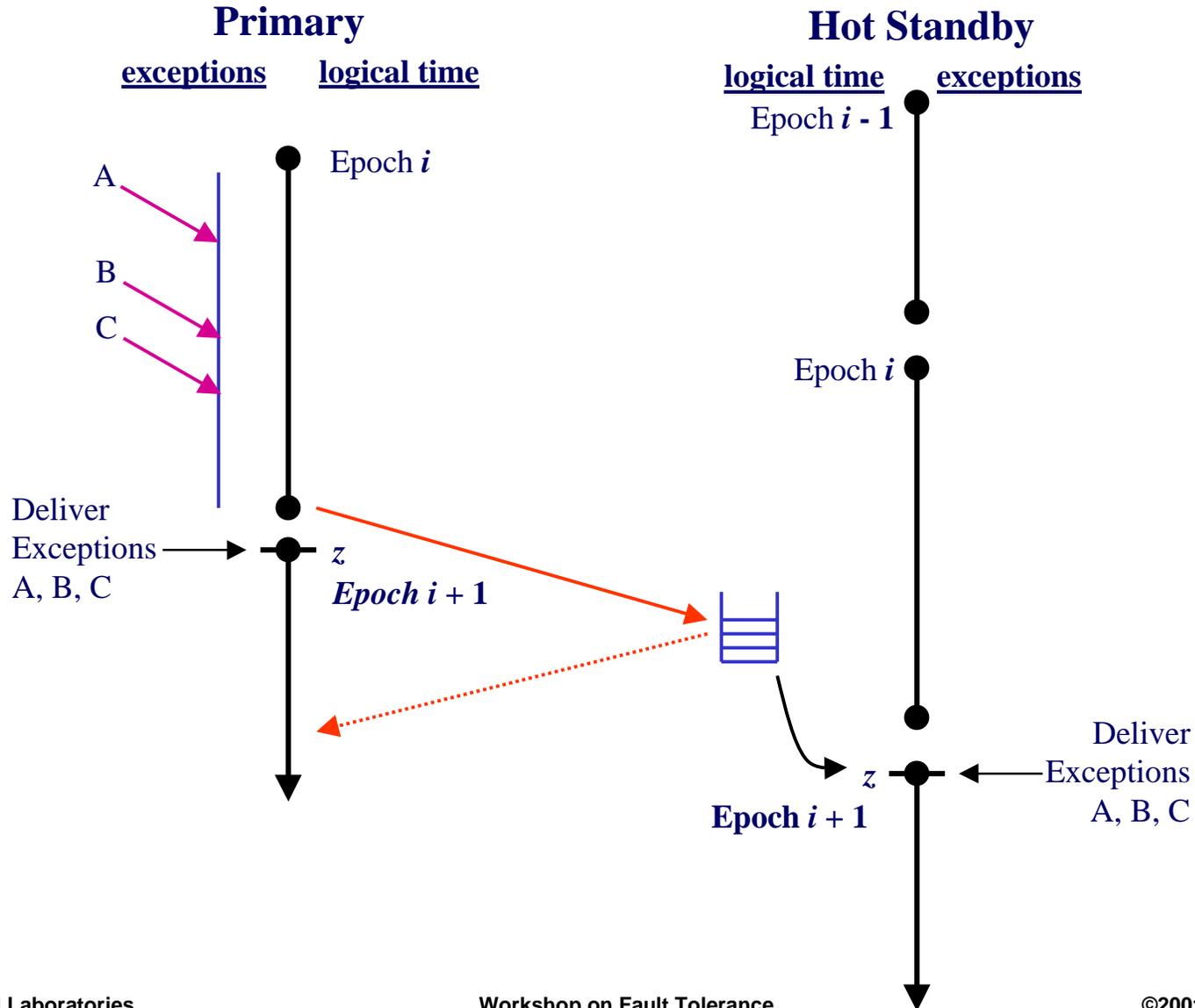
Lucent Technologies
Bell Labs Innovations



Exceptions



Lucent Technologies
Bell Labs Innovations



Interaction with the Environment



Lucent Technologies
Bell Labs Innovations

- Failure-free operation
 - ◆ Primary performs all interaction with the environment, notifying backup of I/O completions
 - ◆ Backup maintains a list of outstanding I/O operations
- Failure recovery based on operation characteristics
 - ◆ Idempotent
 - ◆ Testable
 - ◆ Other

The “State” of the Operating System



Lucent Technologies
Bell Labs Innovations

- Types of application-related OS state:
 - ◆ Names of OS abstractions: process ids, file handles, etc.
 - ◆ Pure caching of information from the environment or about the application
 - ◆ State encapsulating interaction with the environment
 - TCP/IP protocol state
- OS state leakage problem
 - ◆ The state of the OS can leak into the application through the User/Kernel interface – another form of data non-determinism
 - ◆ The stat of the OS can leak out to the environment through communication channels

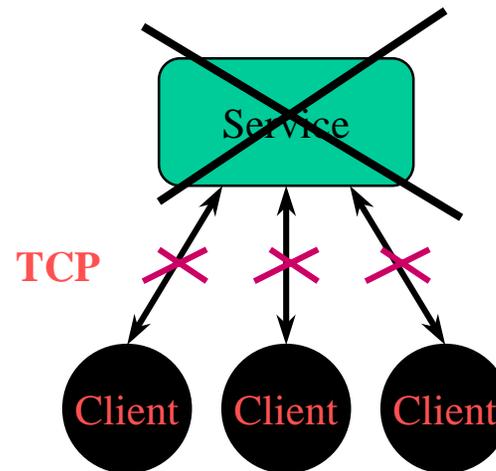
Fault-Tolerant TCP



Lucent Technologies
Bell Labs Innovations

■ Fault Scenario

- Service fails
- TCP connections break
- Service recovers
- *New* TCP connections are established



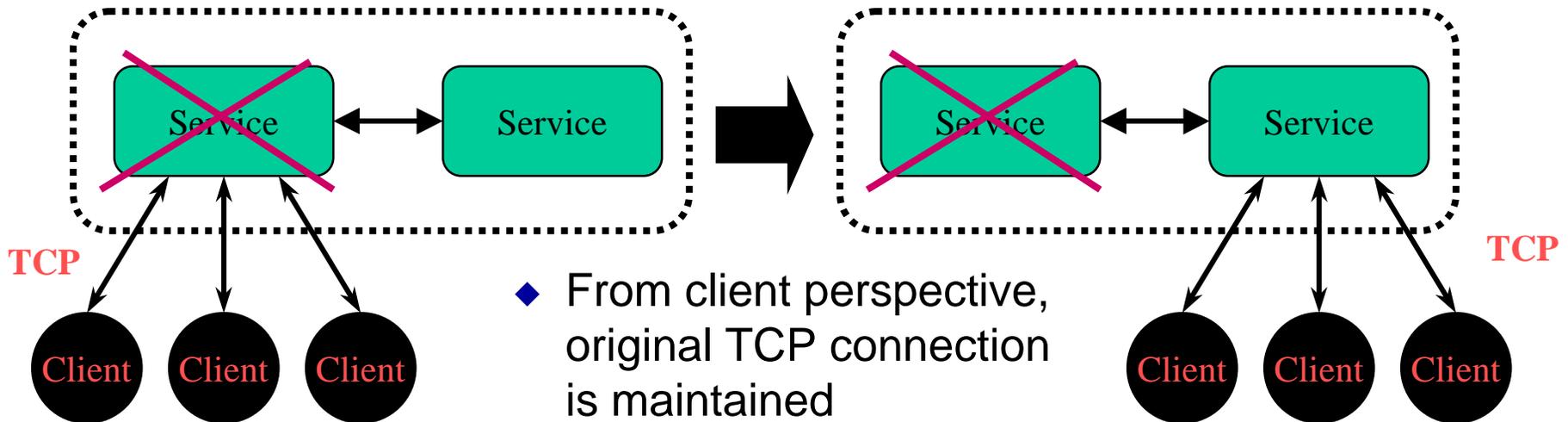
Service Examples:

Web Server
Network Storage
Computational Services
3-Tier Architectures
H.323 Control
BGP Peering Sessions

Fault-Tolerant TCP



Lucent Technologies
Bell Labs Innovations

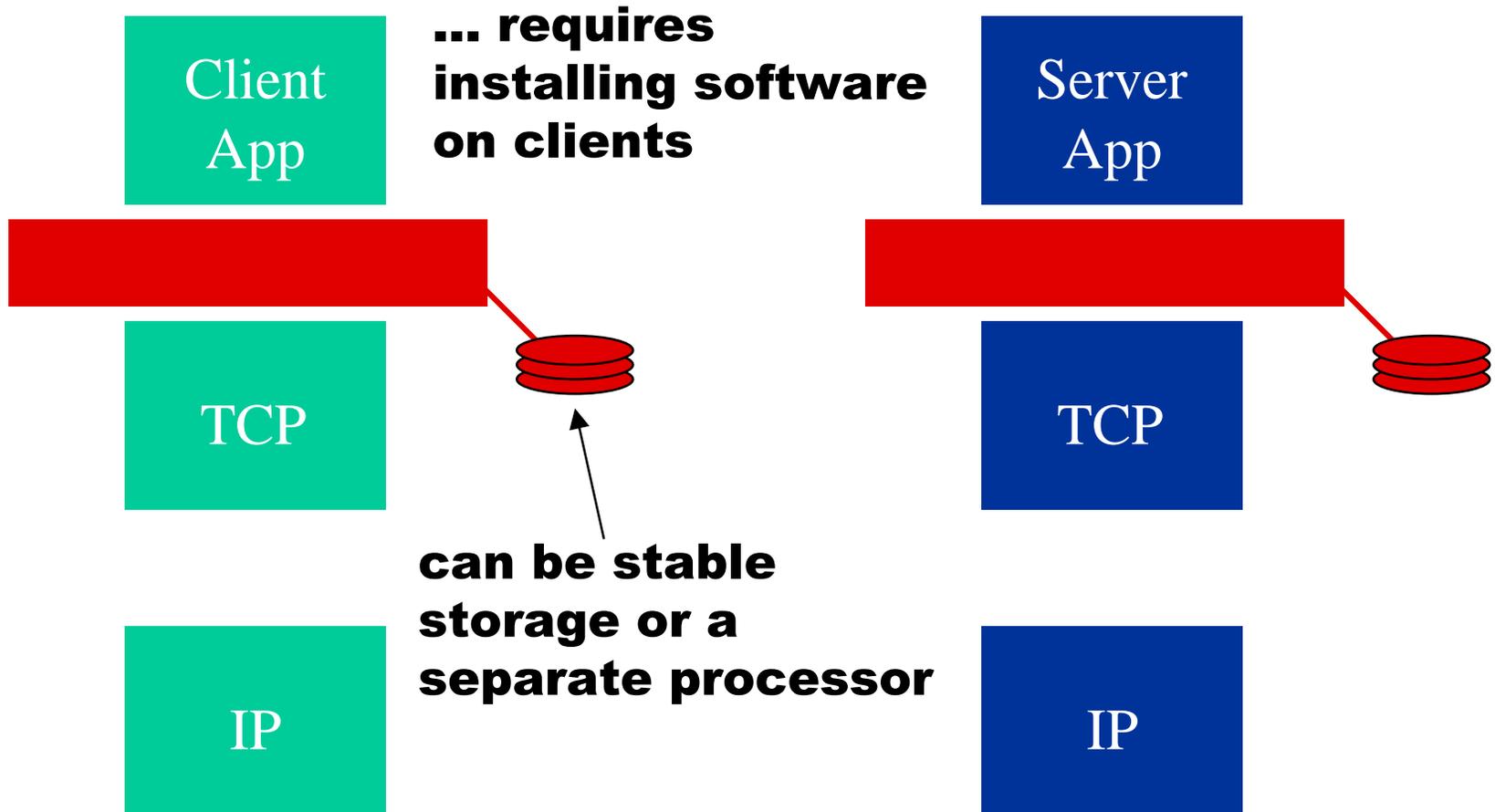


- ◆ From client perspective, original TCP connection is maintained
- ◆ **Transparency** -- Client is not modified in any way
- ◆ FT-TCP works *with* service state recovery to obtain consistent state on recovering service

One Approach



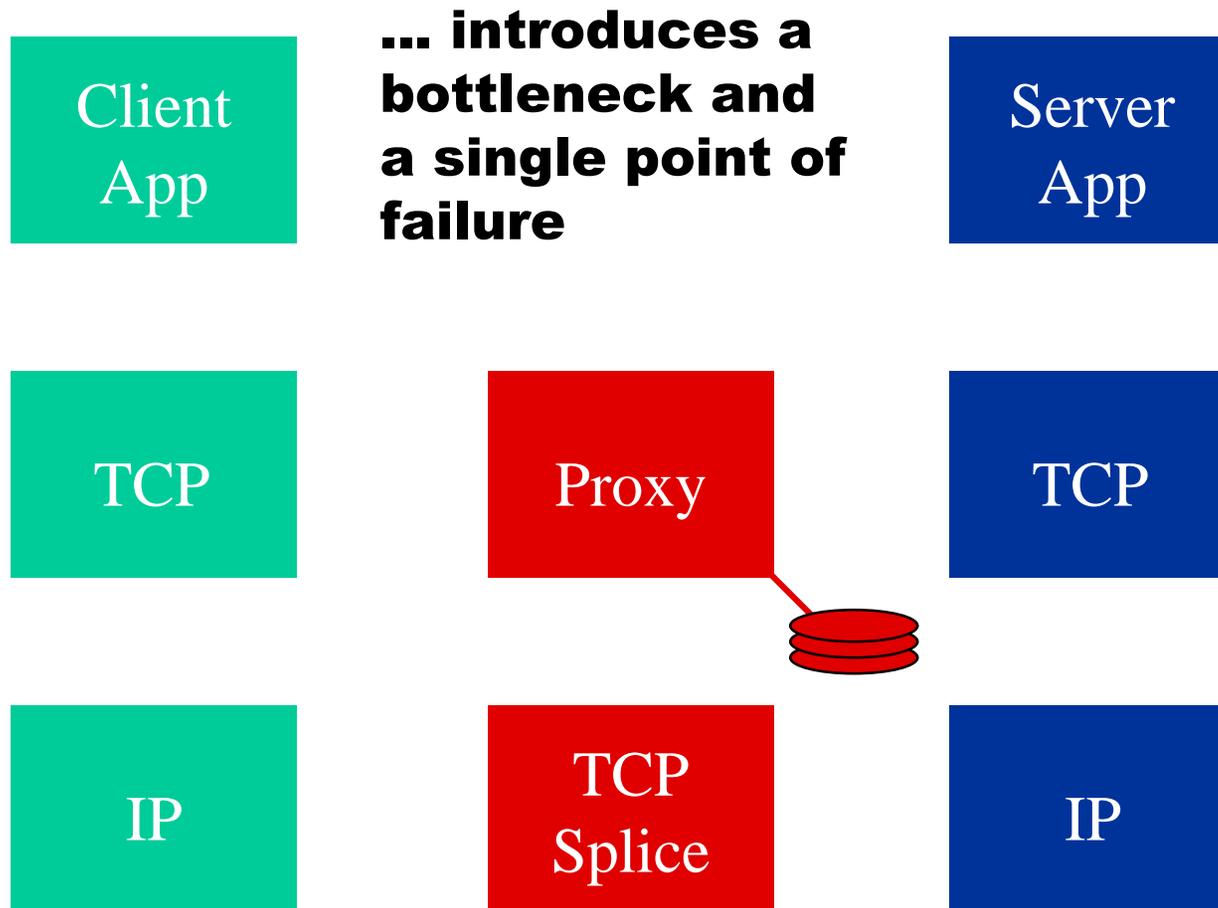
Lucent Technologies
Bell Labs Innovations



Another Approach



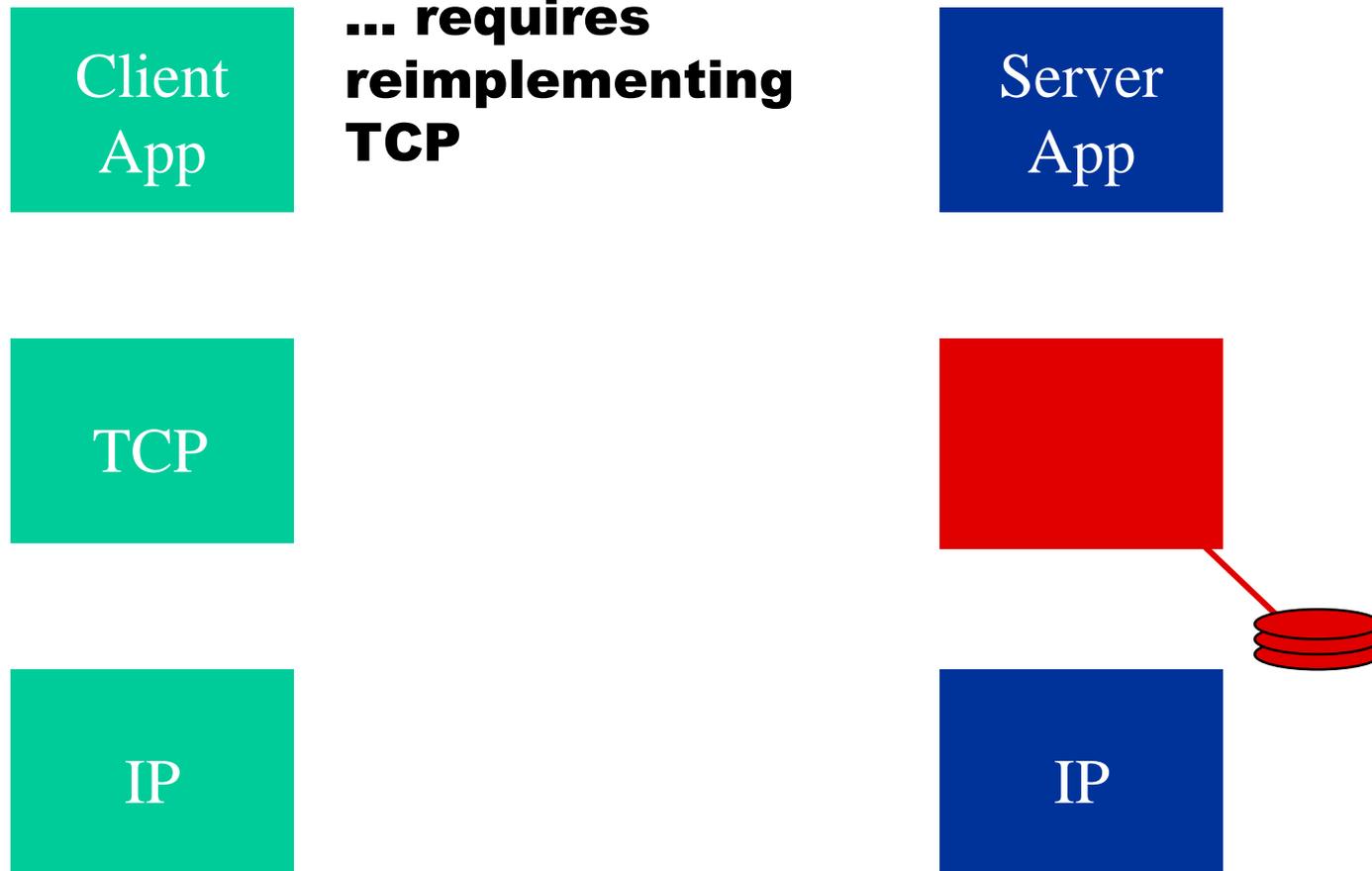
Lucent Technologies
Bell Labs Innovations



A Third Approach



Lucent Technologies
Bell Labs Innovations



Our Approach



Lucent Technologies
Bell Labs Innovations

Client
App

TCP

IP

+ No change to client software.

+ No change to server TCP.

+ No proxy.

- As with other approaches, possible impact on TCP timing measurements.

Server
App

North side wrap

TCP

South side wrap

IP



Recovery Review I



Lucent Technologies
Bell Labs Innovations

The wraps implement a *recovery unit*:

- ◆ records incoming data and choices made by nondeterministic actions;
- ◆ periodically checkpoints application's state;
- ◆ replays incoming data, discards outgoing data and forces same choice of nondeterministic actions upon recovery.

The current implementation:

ignores checkpointing;

treats *read socket calls* as the only nondeterministic action.

Recovery Review II



Lucent Technologies
Bell Labs Innovations

- A *pessimistic receiver-based* approach logs all data before allowing data to leave recovery unit.
 - ◆ The most obvious approach when one wishes to have all new code at the server.
- We use a *hybrid sender/receiver-based* approach.
 - ◆ The client's TCP already implements part of this protocol.

TCP Review II



Lucent Technologies
Bell Labs Innovations

- The initial sequence numbers for both sides are assigned when the connection is established.

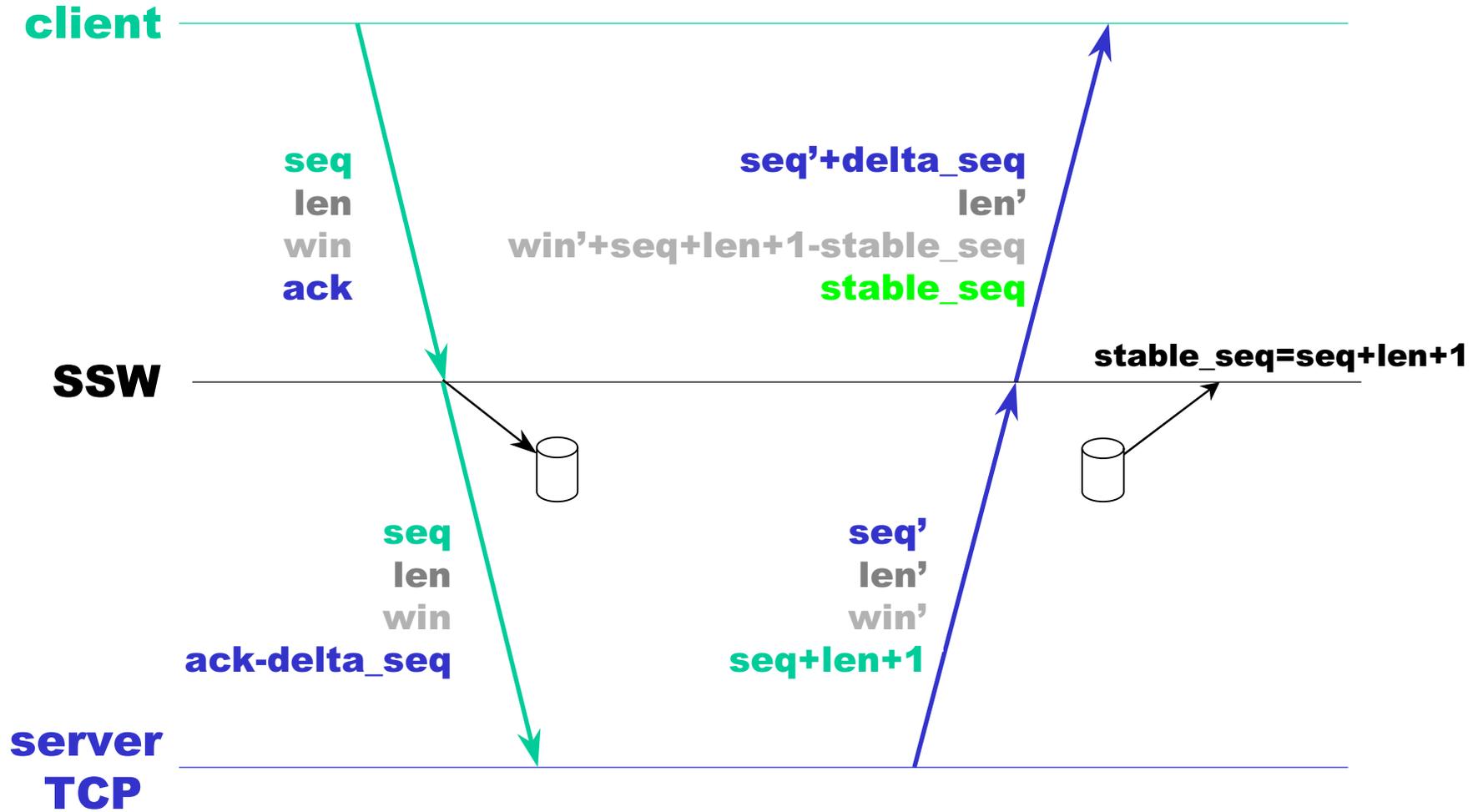
- TCP State of interest
 - ◆ ISN
 - ◆ WIN
 - ◆ ACK
 - ◆ RTT

- TCP implementations are complex and have been tuned to give good performance.
 - ◆ Slow start
 - ◆ Nagle
 - ◆ When to generate an empty segment (ack)?

Failure-free operation



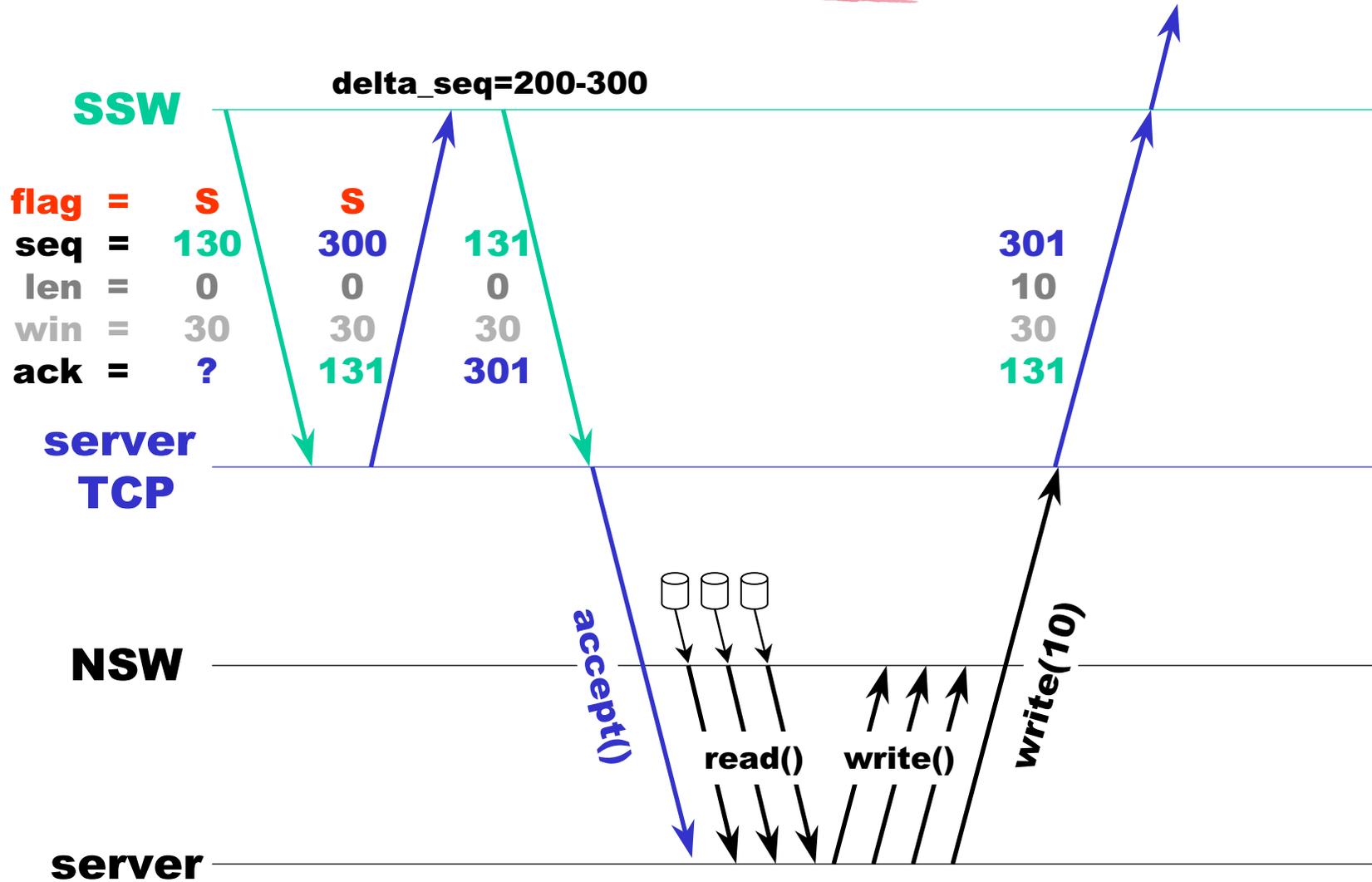
Lucent Technologies
Bell Labs Innovations



Recovery Process



Lucent Technologies
Bell Labs Innovations



Conclusions



Lucent Technologies
Bell Labs Innovations

- Transparency is achievable to differing degrees
 - ◆ Obstacles:
 - Enforcing Determinism
 - Completeness of the Replication Management Interface
- Transparency solutions are forced to make pessimistic assumptions, which may affect performance
- Promise on the horizon in object oriented systems
 - ◆ Client and Service interfaces are well-defined
 - ◆ Mechanisms available to “virtualize” object references
- Fault tolerant TCP has applications beyond TFT embedded use