

Fault Tolerance in the Internet: Servers and Routers

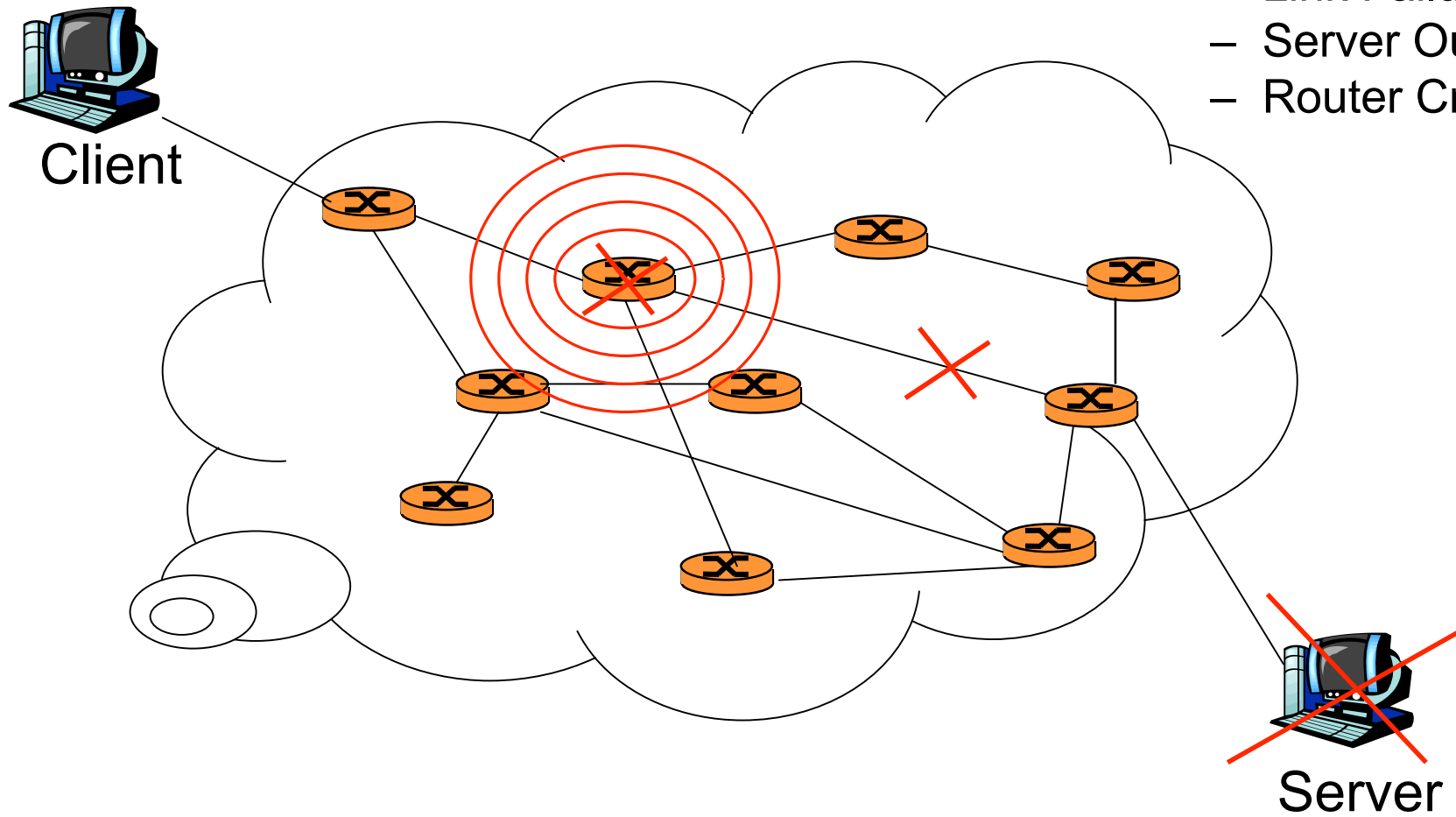
Sana Naveed Khawaja, Tariq Mahmood
Research Associates
Department of Computer Science



Lahore University of Management Sciences

Motivation

- Link Failures
- Server Outages
- Router Crashes



Fault Tolerance

Ability to consistently deliver service:

- Even in the presence of any unexpected errors
- Recover from and restore normal operation after any failure

Operational Requirements:

- Scalability
- Inter-operability
- No loss of data
- Transparency

Fault Tolerant Configurations

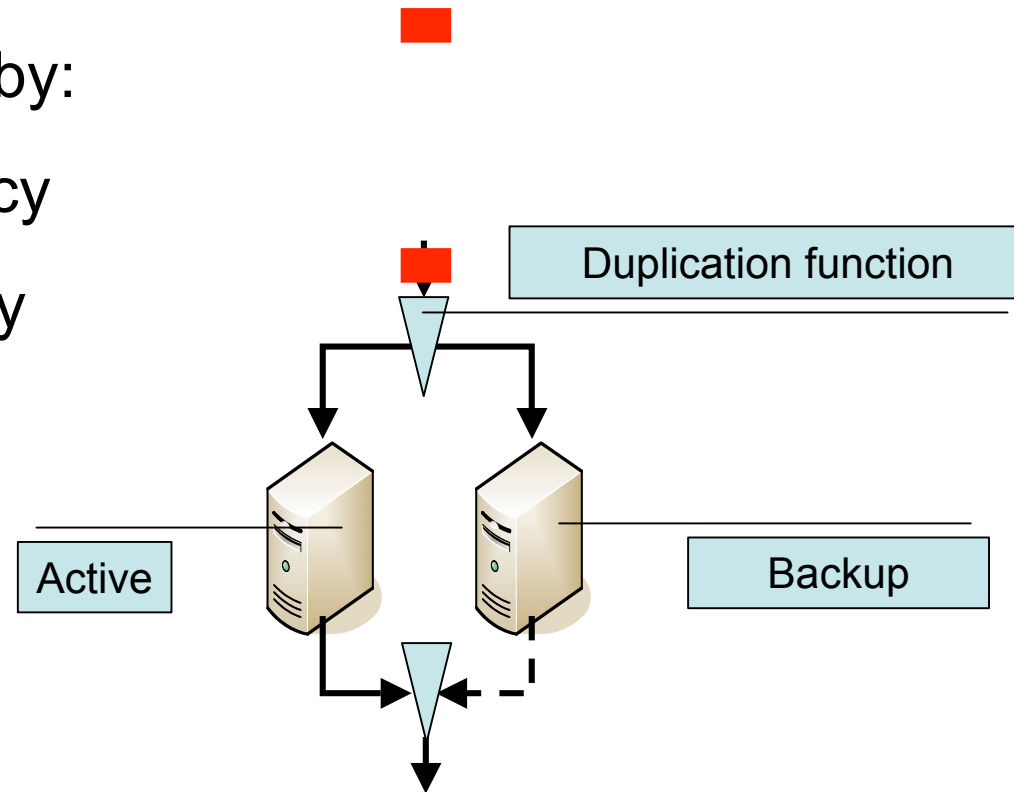
A fault tolerant system can be implemented using one of the following three fault tolerant models or configurations:

- Hot Standby
- Cold Standby
- Warm Standby

Hot Standby Configuration

Fault tolerance ensured by:

- hardware redundancy
- software redundancy



Complete Synchronization between the active and the backup

Merits/Demerits of Hot Standby

Efficient and quick response time of the backup module

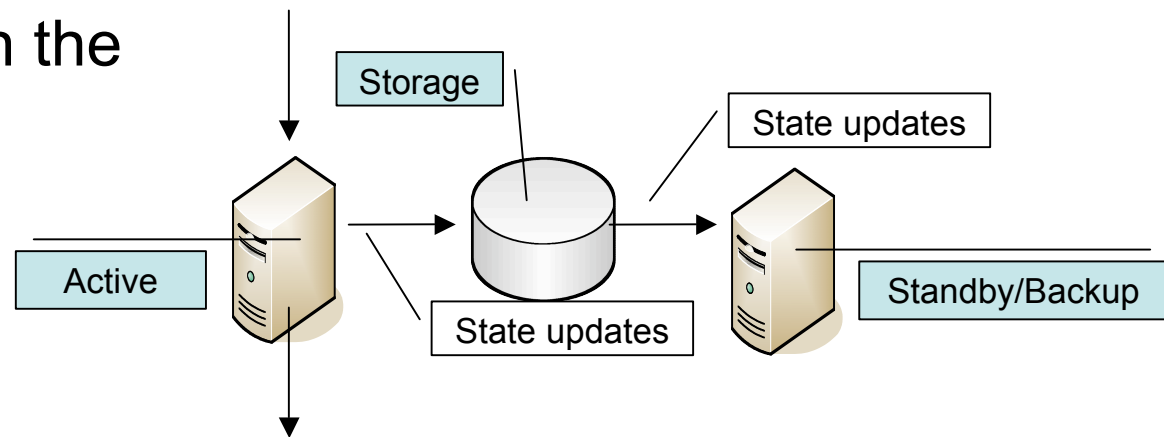
The strict synchronization requirement creates an overhead

Wastage of resources: the backup also has to be online with the primary.

Cold Standby Configuration

The active processes the incoming traffic updating the internal state changes to a reliable storage medium.

When the active crashes, the backup is updated with the state information in the repository.



Merits/Demerits of Cold Standby

Completely asynchronous solution

The backup does not have to be online and could be doing something else as long as the active is up

Relatively long recovery time resulting in a noticeable time window

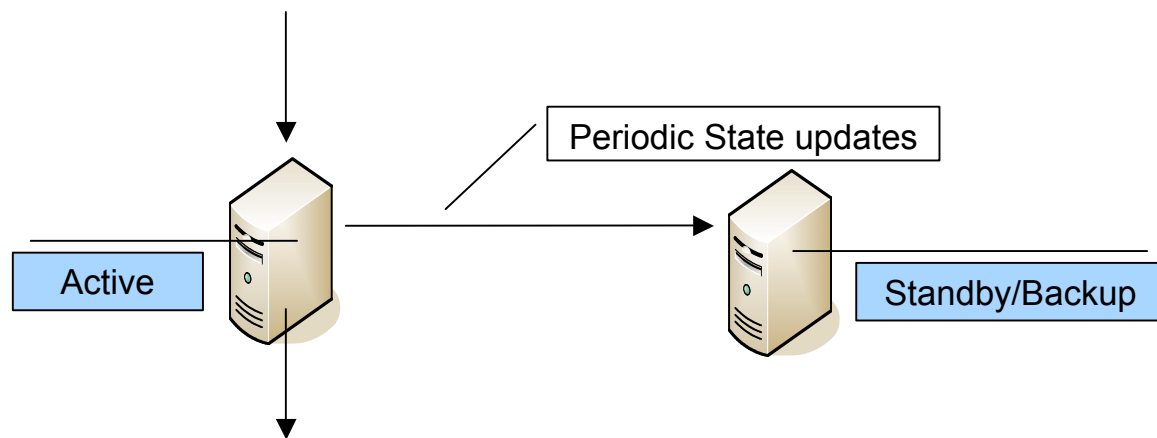
May not be completely transparent to the client

May not be suitable for routers where a delay will result in the neighboring routers becoming aware of the fault

Extra non-volatile storage requirements

Warm Standby Configuration

- The backup is not in sync with the active (as in the hot backup scheme) and it is also not completely ignorant of what the active is doing (as in the case of cold backup)
- Check-pointing



Merits/Demerits of Warm Standby

The backup can be kept offline until the primary fails

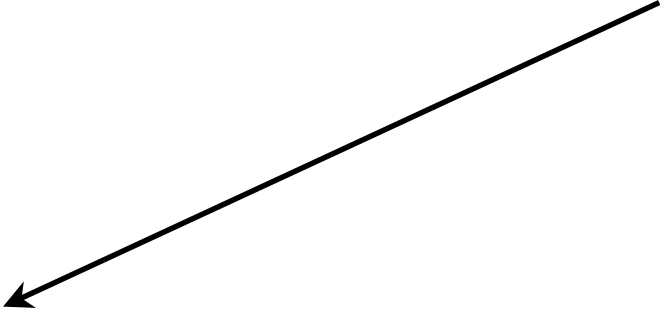
Suitable for applications where state change is not very frequent (e.g. routers)

Approximate restoration: when the active crashes, the backup takes over and starts providing the service from the last check point

The transaction between the last check point and the crash (no matter how small) is lost

Case Study FTTCP: Challenges

- Scalability
 - Inter-operability
 - No loss of data
- Client Transparency

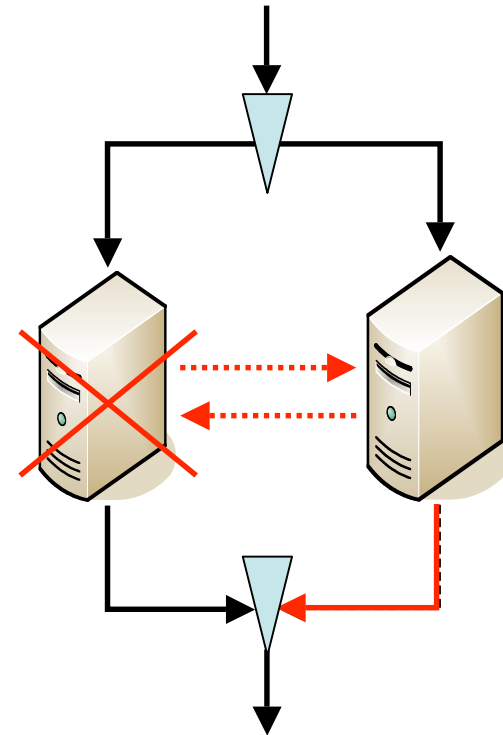
- 
- TCP session replication and synchronization
 - Successful determination of application failures
 - Switching of roles within a certain time window
 - No change in existing protocols at the client end (ideally)

Case Study: FTTCP

- For cold and warm backup, we need to investigate the minimal and optimal (TCP and application level) state information that is sufficient to provide fault tolerance for TCP connections.
- In case of the hot backup, this is automatically taken care of.
- We developed FTTCP based on the hot-standby configuration.

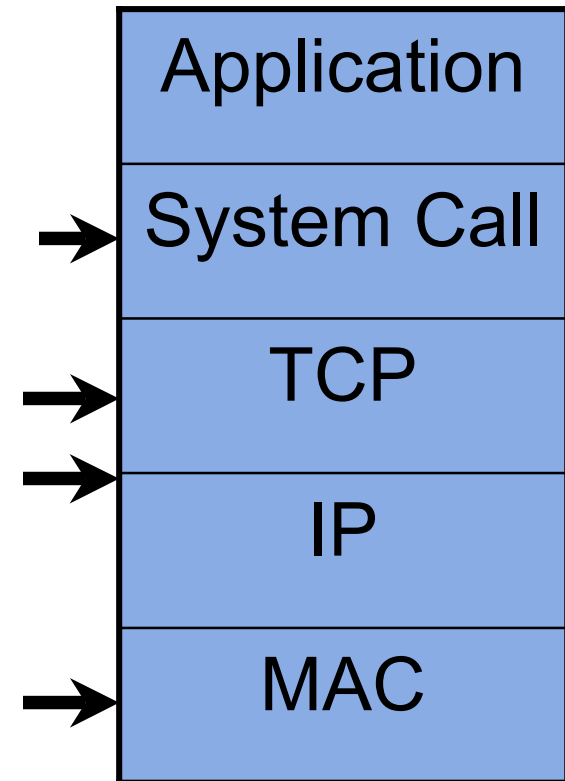
FTTCP Modules

- Lock-Step Synchronization
 - Session Duplication
 - Data duplication
 - Non-determinism
 - Commit Protocol
- Failure detection
 - Process Health Monitoring
 - Heartbeats
- Switchover
 - IP Spoofing
 - Connection Takeover

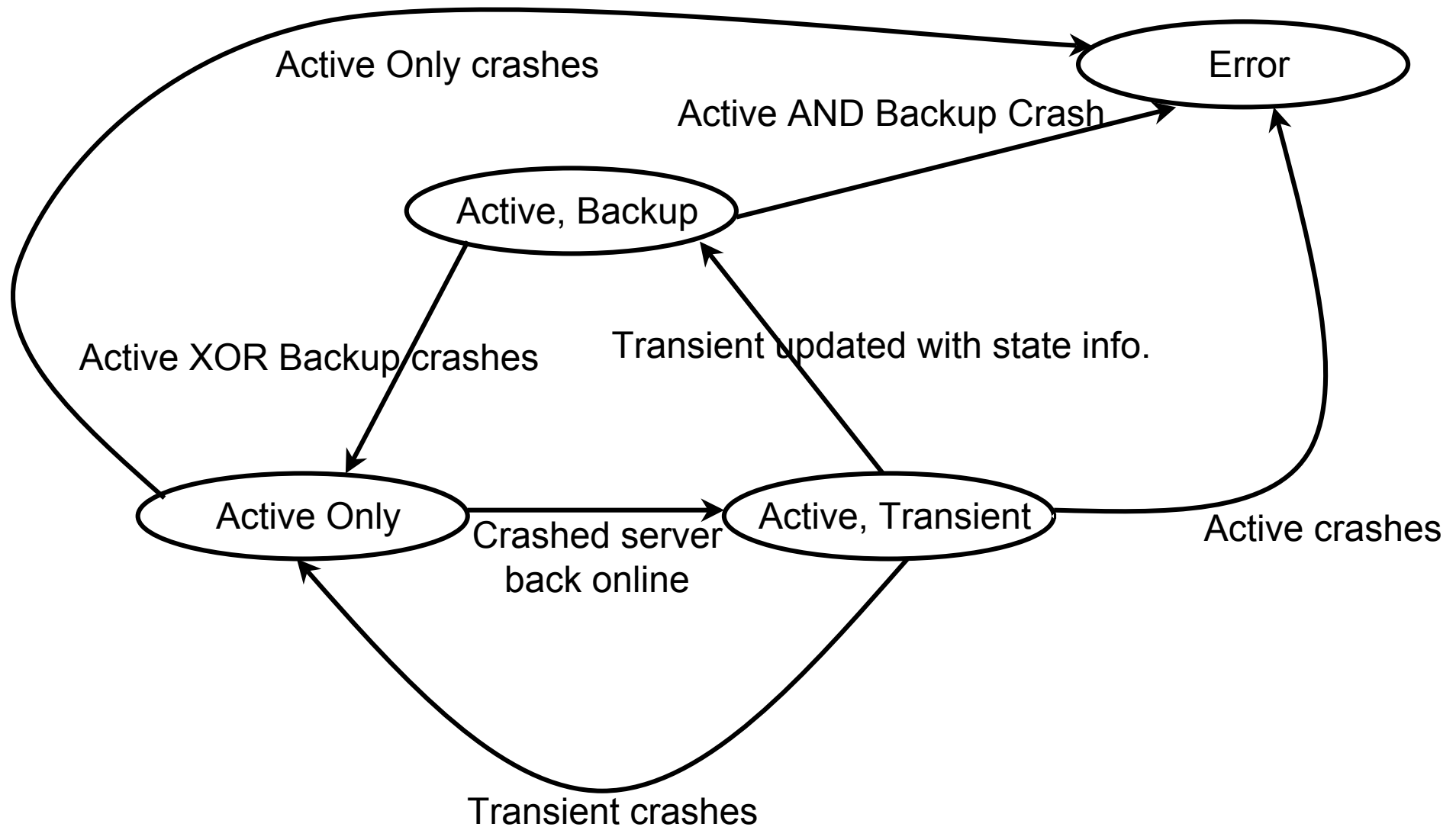


Implementation (Linux Kernel 2.4)

- Lock-Step Synchronization
 - Session duplication: TCP sequence numbers, window size
 - Data duplication: packet duplication at the MAC layer
 - Non-determinism: lock step synchronization of system calls
 - Commit protocol: control (IP) packets
- Failure detection
 - Process health monitoring at the local machine
 - Heartbeats to monitor the health of the other server
- Switchover
 - IP Spoofing: incoming and outgoing
 - Connection takeover: backup packets are no longer suppressed

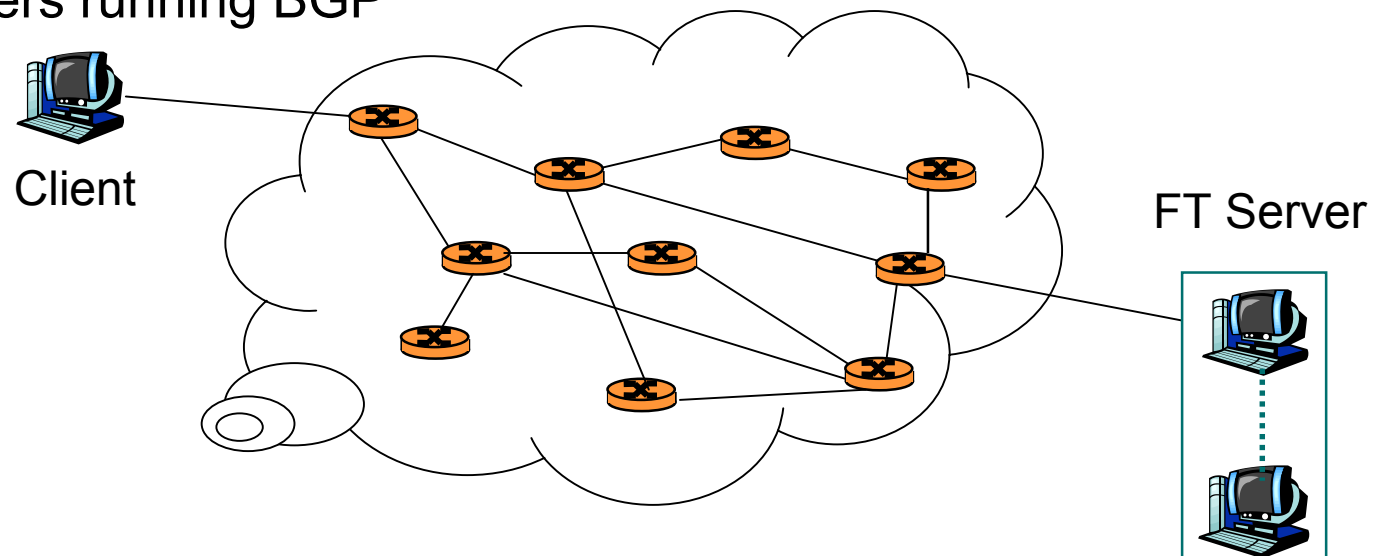


Application Modes



Applications

- Fault tolerant servers
 - TFTP, echo server (for testing)
 - Web servers
 - Email servers
- Fault tolerant routing
 - Soft routers running BGP



Issues

- Health monitoring
 - Deadlock detection
- Multithreading, signal handling
 - Non-deterministic scheduling
- Inherent non-determinism
 - Entails modification of applications
 - Generic solution to fault tolerance is still an open challenge



Questions?

Sana Naveed Khawaja, Tariq Mahmood

Research Associate

LUMS