



Education

Pre-Deployment Testing & Validating

Applications Infrastructure for Distributed Storage

Kevin Przybocki
Anue Systems, Inc.

Today's Enterprise Storage Environment

- The squeeze:
 - ◆ New/multiple applications
 - ◆ Legacy infrastructure
 - ◆ Greater complexity
 - ◆ Tight schedules
- Time to deployment is critical
- User expectations for transaction response time
- Business and regulatory requirements for data integrity & security
- The bottom line: Know what your solution will deliver before you deploy



➤ Bandwidth

- ◆ Congestion/Bottlenecks
- ◆ QoS prioritization
- ◆ TCP Window size

➤ Delay

- ◆ Distance
- ◆ Hops
- ◆ Acknowledgements
- ◆ Buffer Credits

➤ Impairment

- ◆ Packet loss
- ◆ Bit errors
- ◆ Jitter



Consequences of Impairments

- Increased risk of data being lost or unavailable
- Increased spending on equipment and remedial actions
- Greater operating cost
- Expanded development/QA schedules
- Longer backup times
- Decreased productivity
- Increased tech support costs
- Reduced user satisfaction
- Legal ramifications/Sarbox compliance

Your Solution and Impairment

- Packets arrive in sequence and then
 - ◆ The wrong packet arrives
 - ◆ The packet arrives with corrupt data
 - ◆ The packet arrives too late
 - ◆ The packet never arrives
- What would these impairments do to your backup and recovery process?
- How would they affect the:
 - ◆ State machine
 - ◆ Transaction response time
 - ◆ Data integrity
 - ◆ Network resiliency



Real World Testing of Storage Apps

➤ Realistic conditions are essential when testing:

- ◆ Data Availability
- ◆ QoS schemes
- ◆ RTO & RPO
- ◆ Transaction response
- ◆ Performance thresholds
- ◆ Compliance with SLAs
- ◆ Compliance with regulatory requirements

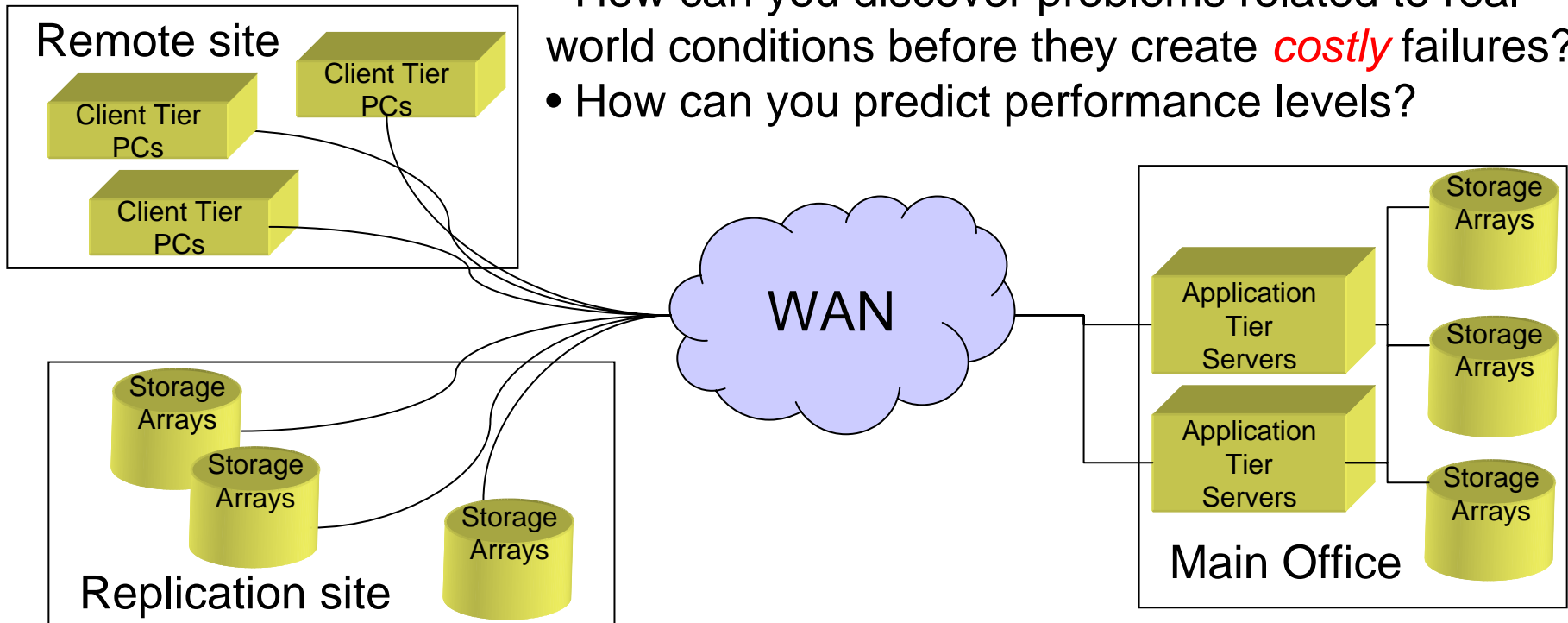


Testing Distributed Storage Applications

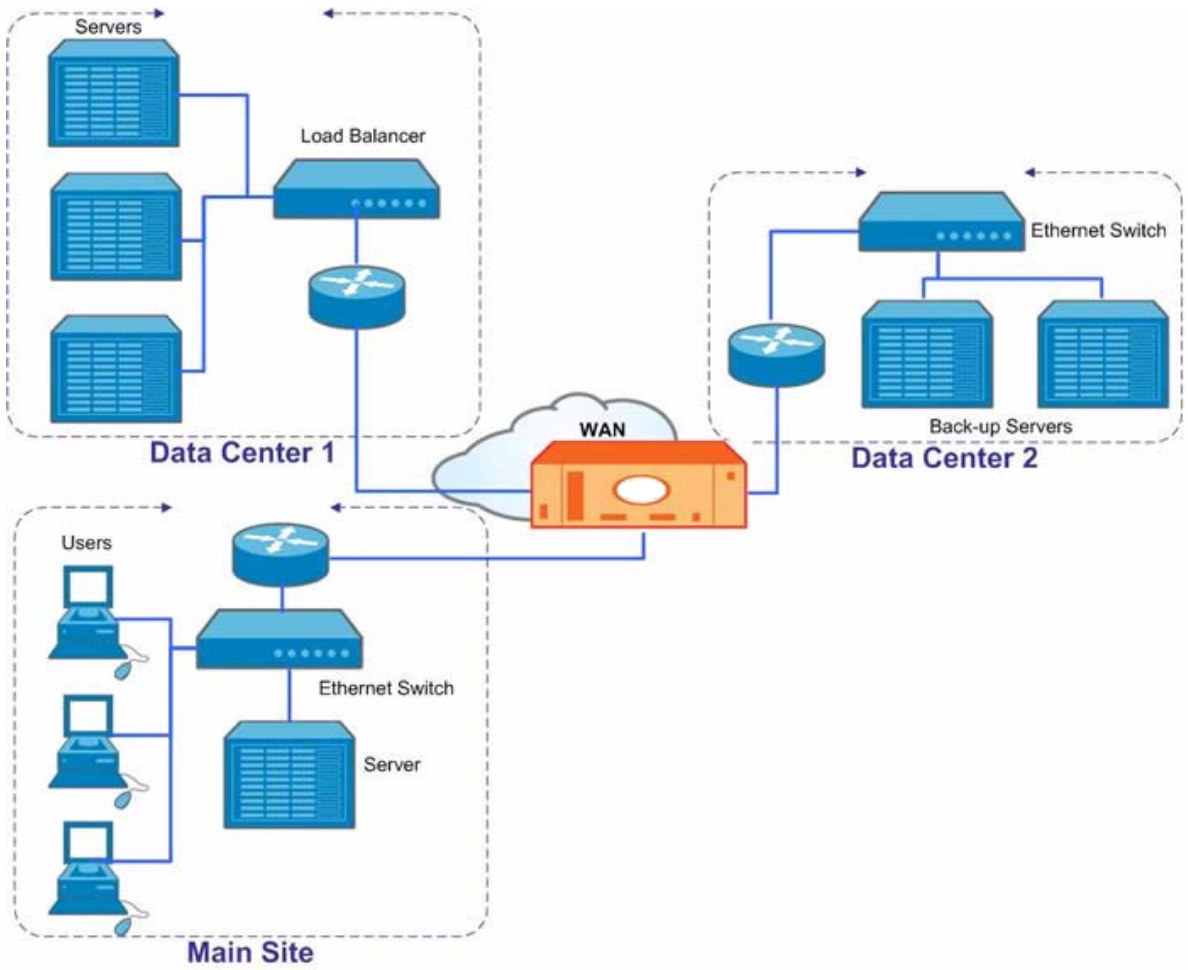
Conformance – Do they operate to spec? **Yes**
Interoperability – Do they operate with each other? **Yes**
System – Do they operate as a system? **Yes**

Great,
but . . .

- How will they work outside a sterile lab in a real network?
- How can you discover problems related to real world conditions before they create *costly* failures?
- How can you predict performance levels?



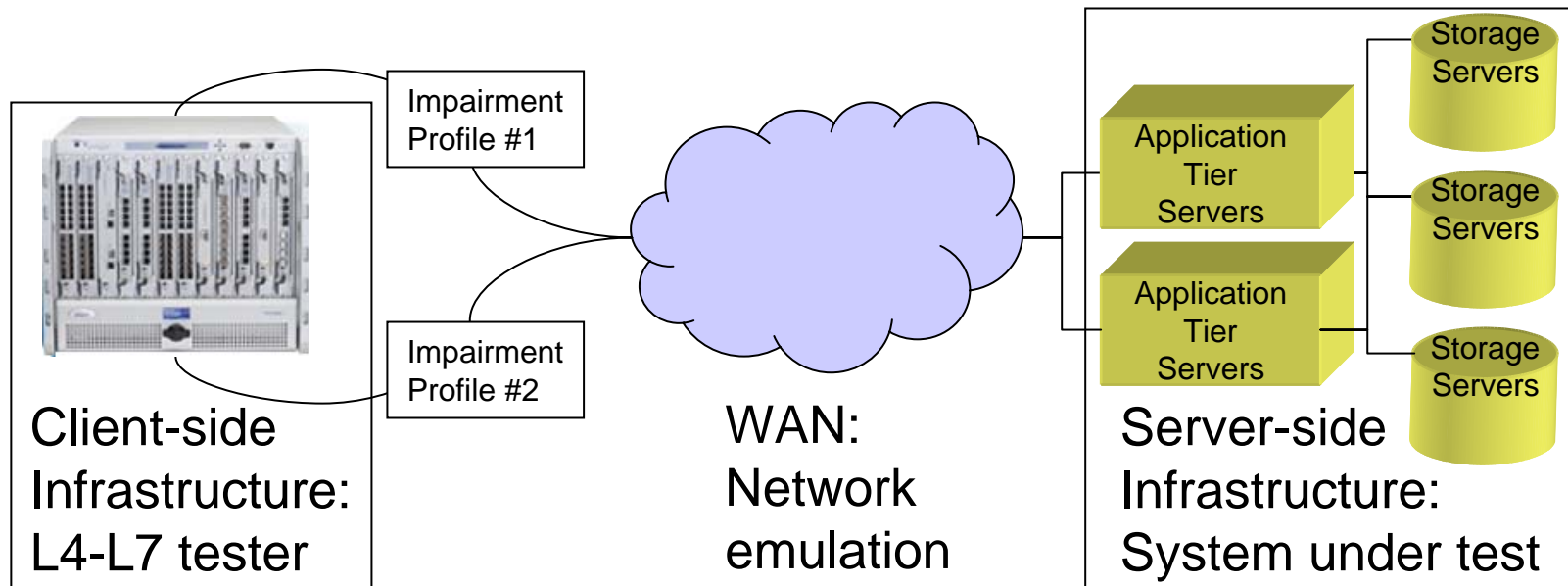
Emulating your Storage Environment Education **SNIA**



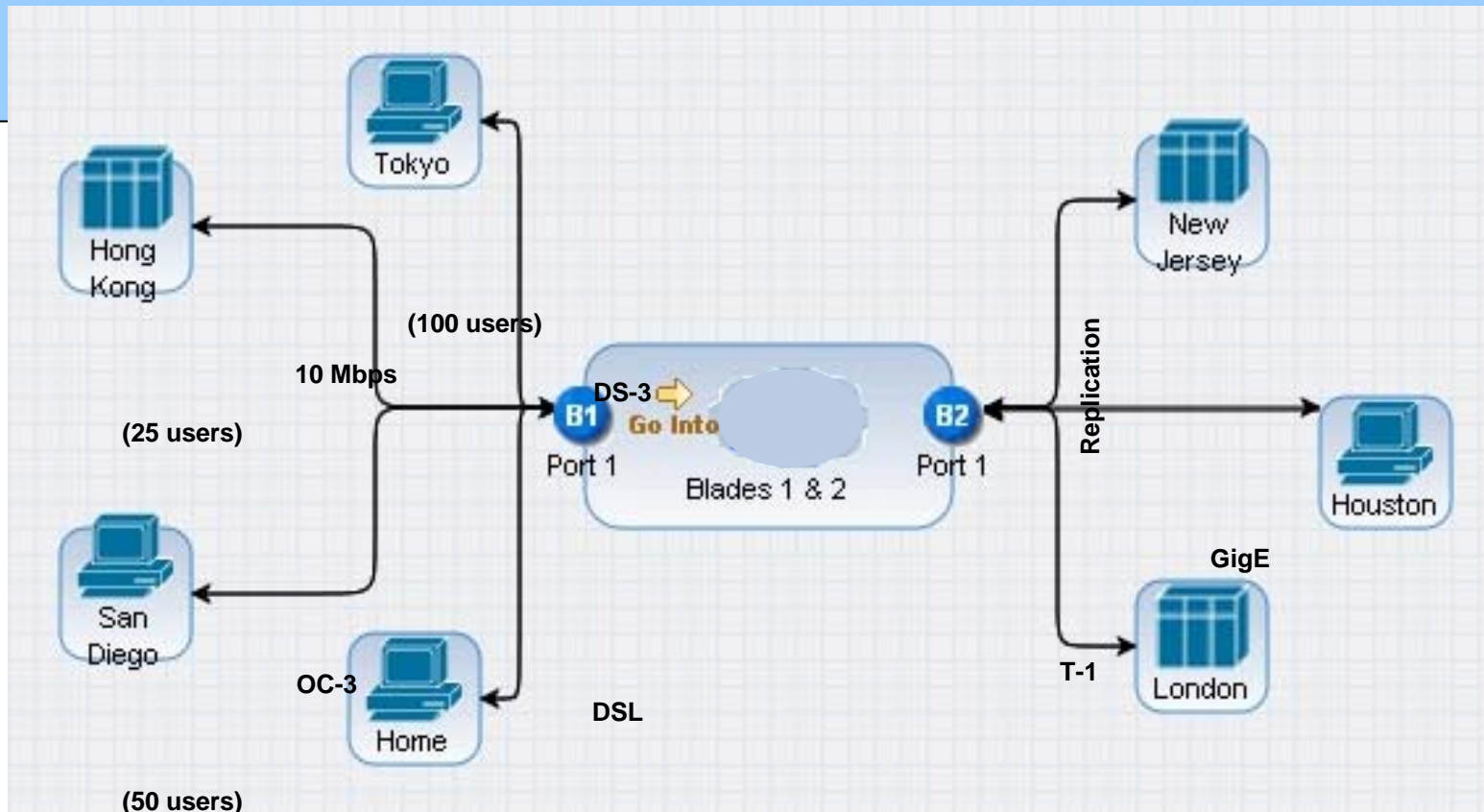
Six Steps to Predicting App Performance

Phase	Step
Baseline	Create traffic profiles
	Establish performance metrics
	Establish failure thresholds
Network Emulation	Create network profiles
	Establish performance metrics
	Establish failure thresholds

- Client-side infrastructure: L4-L7 tester
- WAN: Network emulation
- Server-side infrastructure: Storage Servers



Testing with Network Emulation



Service Level Objectives (SLO)

- Define the performance targets for
 - ◆ Distance Backup and Disaster Recovery/Business Continuity
 - ◆ Storage Security, Replication and Integration
 - ◆ ILM
 - ◆ Express Targets as Transaction Response Time (TRT), Recovery Time Objective (RTO), Recovery Point Objective (RPO)
- Established in cross-functional meetings with Storage IT and user departments
- Determine Delay on the Network
 - ◆ Compliance may require remote backup at pre-specified distances
 - ◆ Include network induced data delays (caching, congestion, fiber cuts, switching delays, etc.)
- Used in testing to determine pass/fail results

Phase 1: Establish the Baseline

Phase	Step
Baseline	Create traffic profiles
	Establish performance metrics
	Establish failure thresholds
Network Emulation	Create network profiles
	Establish performance metrics
	Establish failure thresholds

Step 1: Create Traffic Profiles

- ▶ Periodic user behavior
 - ◆ Time of day (e.g. morning log-in)
 - ◆ # of Users
 - ◆ End of week, month, quarter
- ▶ Event-driven behavior
 - ◆ Backups
 - ◆ Disaster Recovery Scenarios

Step 2: Establish Performance Metrics

- One or more test runs per traffic profile
- TRT metrics recorded for each test run (min/max/avg)
- Results compared to SLO for pass/fail



Step 3: Establish the Failure Threshold

- Ramp up traffic in steps
- Identify the point that the SLO threshold is crossed
- Establishes the maximum transaction load the system can handle in an ideal network
- Used to identify scalability limits and plan for growth

Phase 2: Emulate the Network Environment

Phase	Step
Baseline	Create traffic profiles
	Establish performance metrics
	Establish failure thresholds
Network Emulation	Create network profiles
	Establish performance metrics
	Establish failure thresholds

Step 4: Create Network Profiles

- ◆ Identify the conditions for each remote site:
 - ◆ Delay
 - ◆ Bandwidth
 - ◆ Packet jitter
 - ◆ Packet loss
 - ◆ Packet reorder
 - ◆ Packet error

Step 5: Establish Performance Metrics

- Conduct one or more test runs per traffic profile per network profile
- Record TRT metrics recorded for each test run (min/max/avg)
- Measure recovery time and recovery points after failure occurs

Profiles		Results
Traffic	Network	TRT
User A	Site 1	
User A	Site 2	
User A	Site 3	
User B	Site 1	
User B	Site 2	
User B	Site 3	

Step 6: Establish Failure Thresholds

➤ Traffic load threshold

- ◆ For each network profile
 - › Ramp up traffic
 - › Identify the point that the SLO is violated
 - › Establishes the maximum transaction load that remote site can support

➤ Impairment threshold

- ◆ For relevant impairments (delay, loss, etc)
 - › Ramp up impairment
 - › Identify the point that the SLO is violated
 - › Establishes the level of service

Network Emulation and Applications SNIA

➤ Testing should emulate:

- ◆ Line Bit Errors
- ◆ LOS / LOF Sync
- ◆ Static Delay
- ◆ Jitter Delay
- ◆ Packet Drop - Burst, Random, Static
- ◆ Packet Re-order
- ◆ Duplication
- ◆ Data Corruption
- ◆ Higher Layer Bit Errors
- ◆ Modification
- ◆ CRC Corruption

➤ Should filter/classify traffic on...

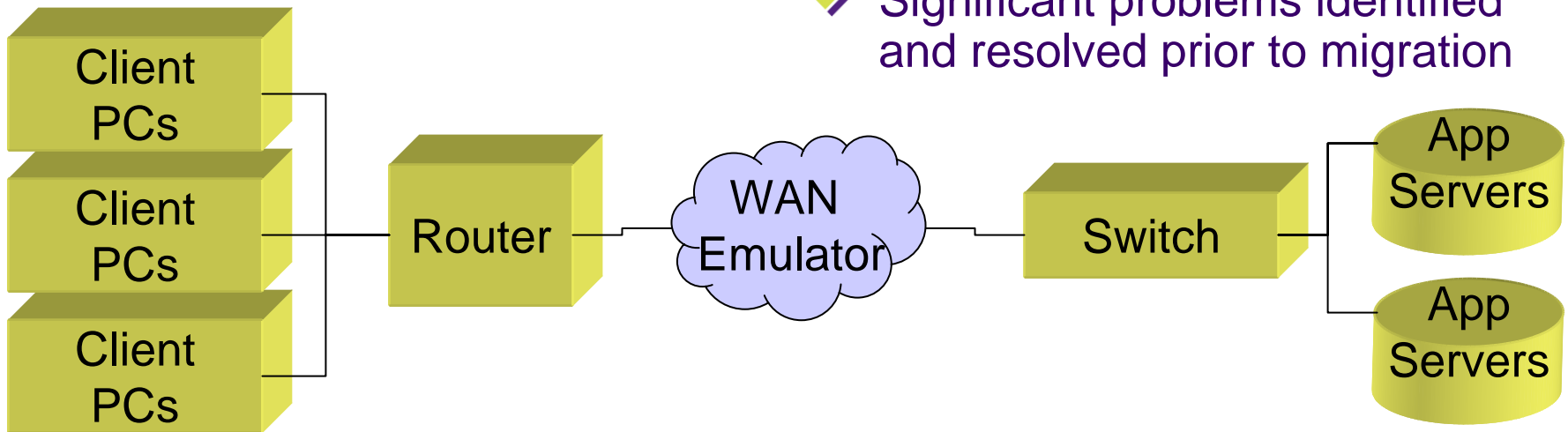
- ◆ MAC or IP Source Address
- ◆ MAC or IP Destination Address
- ◆ VLAN
- ◆ MPLS
- ◆ IP ToS
- ◆ TCP or UDP Source or Destination Port Numbers
- ◆ DiffServ
- ◆ Other Protocols (Layer 3-7)
- ◆ Dynamic Pattern Matching

Test Case #1: Global IT Service Provider

Application performance after server migration

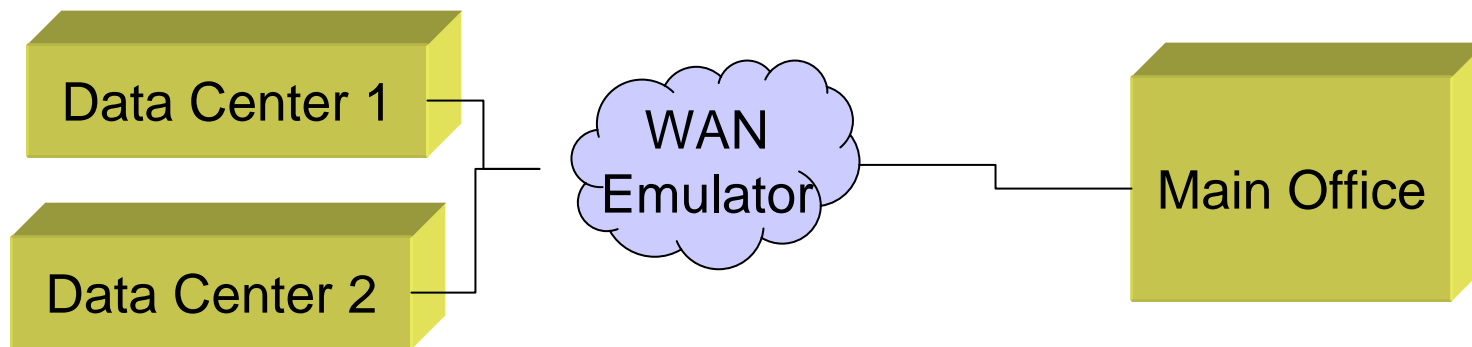
- Mainframe relocation from TX to CT with thousands of users
- Dynamically increased delay and decreased bandwidth

- Target delay 80ms. Results:
 - ◆ At 5ms: performance declined
 - ◆ At 40ms: application failed
- Reduced risk by avoiding SLA penalties of \$20k-\$40k per hour when service is down
- Significant problems identified and resolved prior to migration

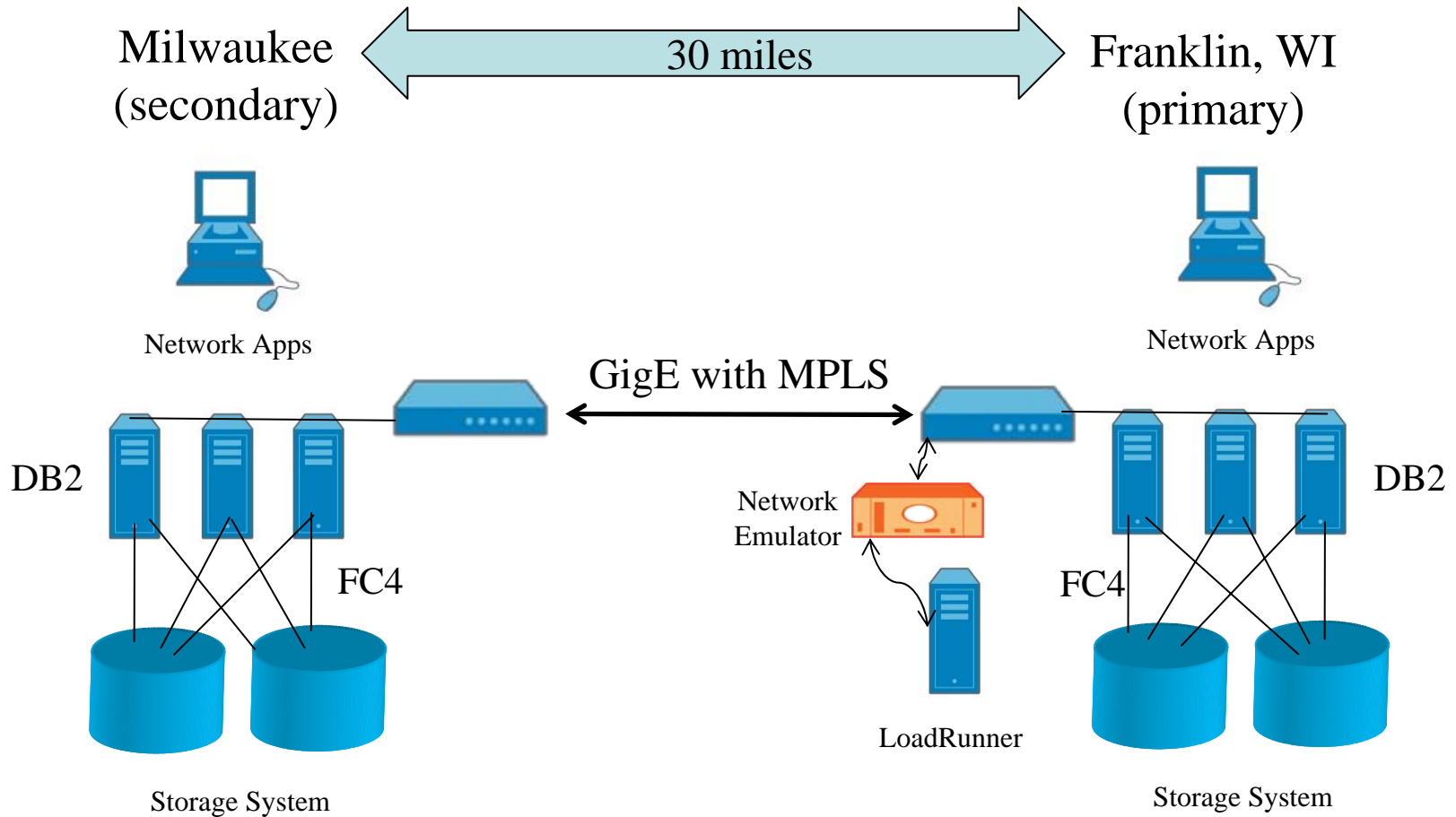


Test Case #2: Consumer Products Company

- Data replication & backup over 300 km between two data centers
- GigE and Fibre Channel
- Synchronous replication between main office and two data centers
- Testing failure recovery scenarios
 - ◆ Looking at application performance
 - ◆ Testing response time to failures
- Ensure regulatory compliance
- Helped define required SLAs



Test Case #3: Motor Vehicle Manufacturer



“During testing of our long distance backup plan, network emulation was an indispensable component of our proof of concept (POC) test environment. We used emulation for storage router POC testing in order to determine how our equipment and backup applications would perform over a wide area network. Performing distance simulation saved us time and money because it was used in lieu of actually sending our data 900 miles during the POC. By introducing real world network conditions such as latency and bit errors in our lab, we gained an accurate understanding of how our remote storage solution would perform once it was deployed.”

Project Manager II
Cincinnati OH

With Network Emulation You Can

- *Evaluate* the performance of existing or emerging Storage Solutions
- *Characterize* breaking points of new infrastructure or applications
- *Validate* DR/BC solutions before deployment
- *Discover and define* minimum required SLAs
- Enable real-world testing with *accuracy* and *repeatability*
- Save time and money by *increasing efficiency* of testing
- *Reduce risk* while building confidence, quality and a competitive edge

- ◆ There are many networking challenges involved when implementing distributed storage
 - ◆ Storage protocols are latency sensitive and require “hand shakes” between locations.
 - ◆ Slow throughput, retransmissions & buffer credit starvation can affect performance distributed storage applications
- ◆ To assure a robust solution you must do real-world testing with network emulation

***“If a packet hits a pocket on a socket on a port,
And the bus is interrupted as a very last resort,
And the address of the memory makes your disk drive abort,
Then the socket packet pocket has an error to report!”***

With Apologies to Theodore S. Geisel



- ▶ Please send any questions or comments on this presentation to the SNIA:
trackvirtualization@snia.org

Many thanks to the following individuals
for their contributions to this tutorial.

SNIA Education Committee

Frank Bunn
Curt Kolovson
Ben Kuo
John Logan
Gene Nagle
Rob Peglar
Abbott Schindler
Nik Simpson
Wolfgang Singer
David Thiel