



The Evolution of Wide Area Data Services

Interop

New York

Mark Stuart Day

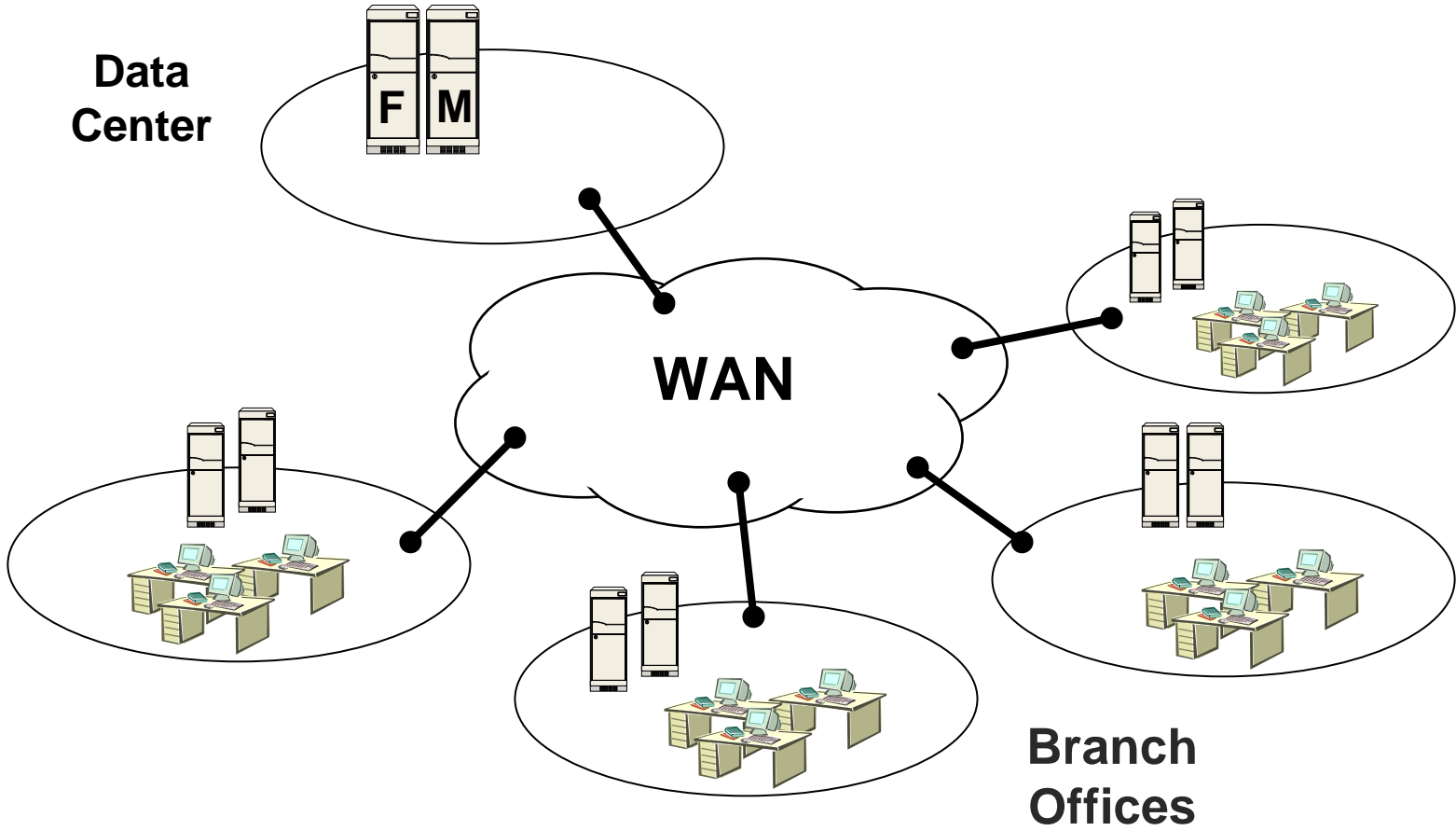
Chief Scientist



Outline

- Why centralize?
- Performance problems
- Solutions for Wide-area Data Services
 - Compression
 - Caching
 - Compression + caching
 - Universal Transport Accelerator (UTA)
 - UTA + local storage
- Summary

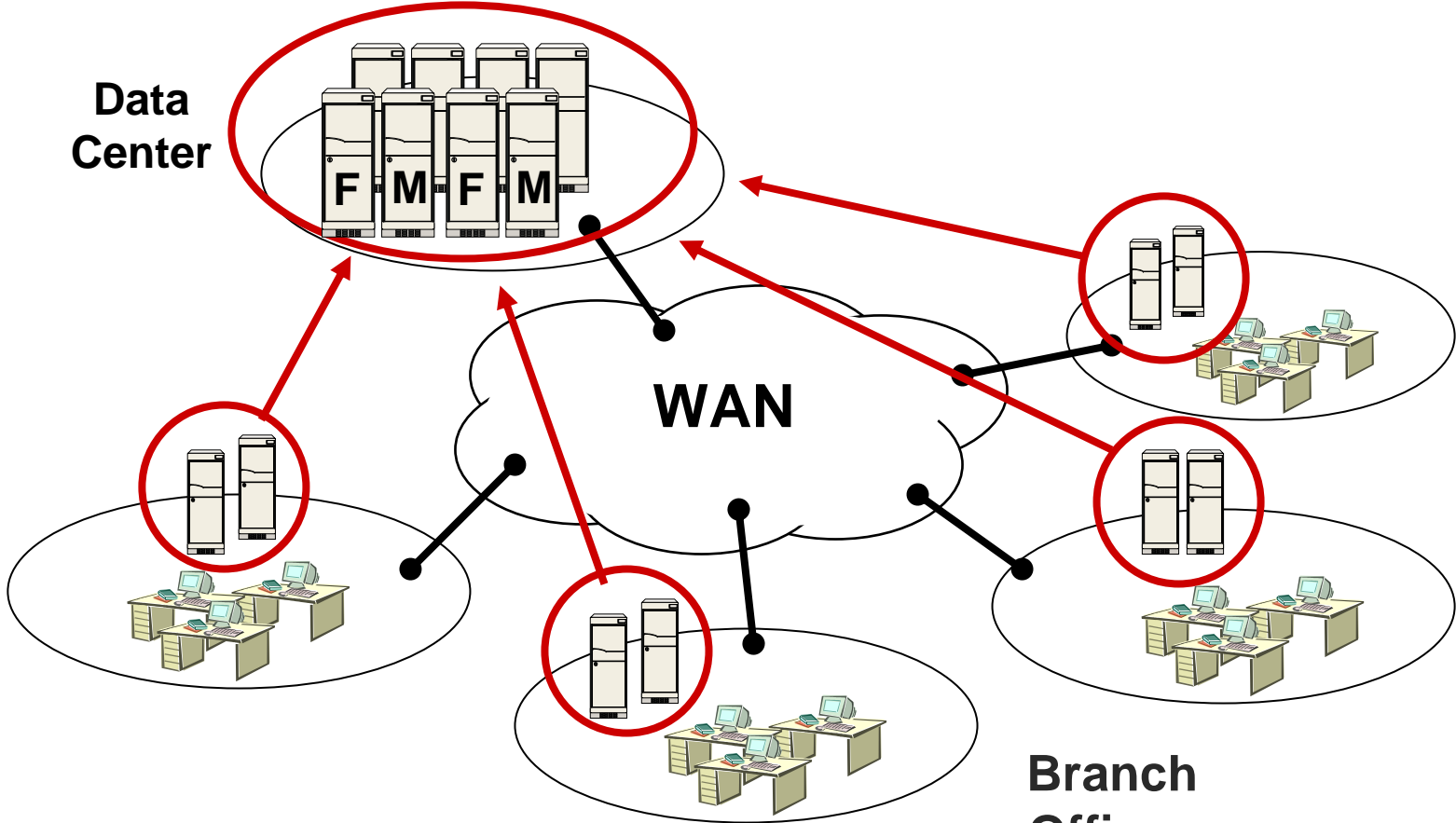
Today's distributed systems...



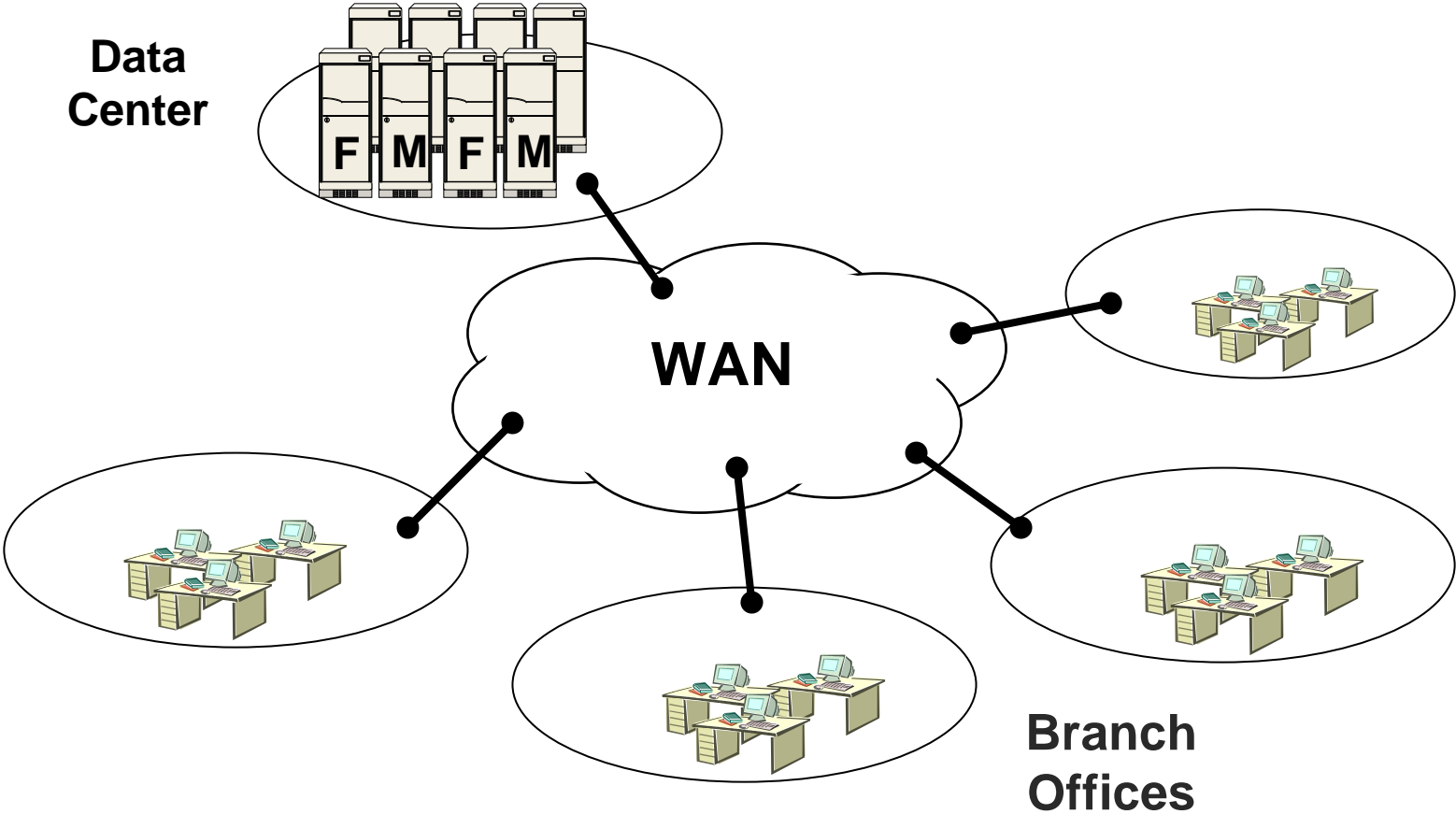
...are causing significant problems

- Cost (IT staff, equipment)
- Manageability
 - Backups
 - Virus protection
 - Software updates
- Compliance issues
 - Was your backup *really* done?
 - *How* was it done?
 - Do you really know where your data is?

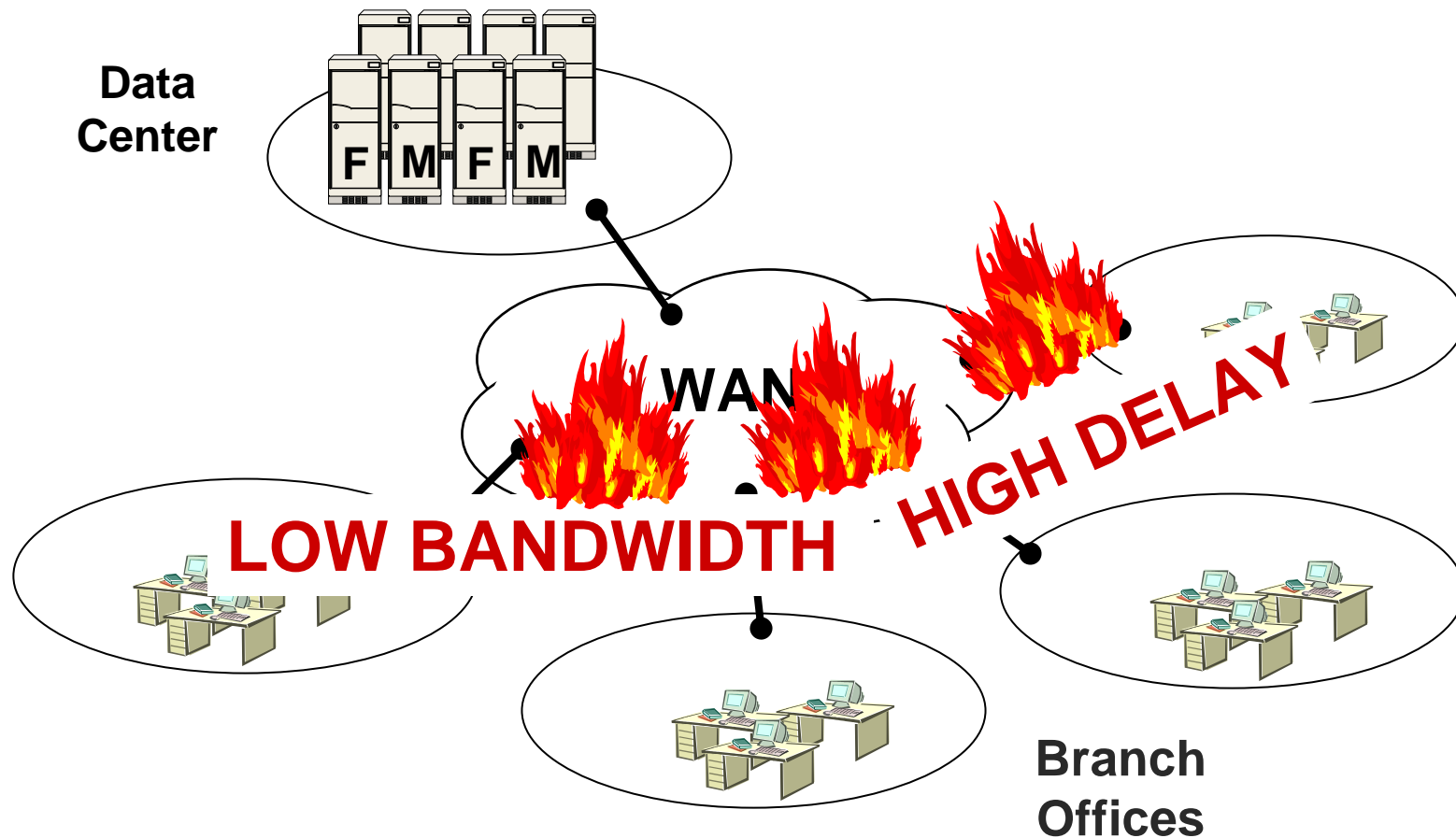
The new (old) paradigm



Serverless Branches: Great!



New problem: application performance



Poor Remote Application Performance has Multiple Causes

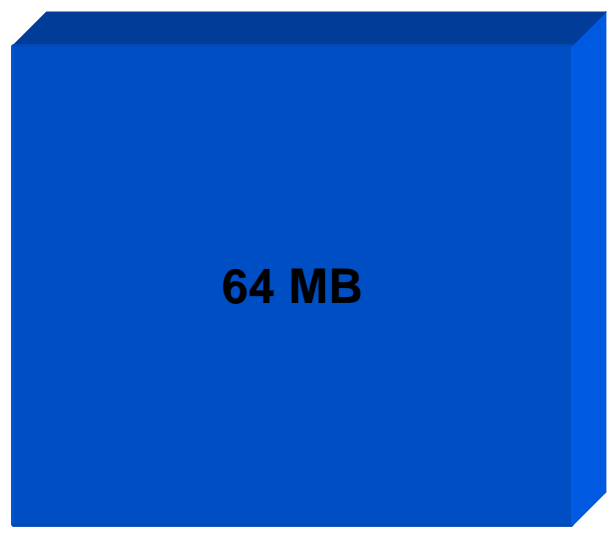
Application Protocol Chattiness and Latency

TCP Chattiness and Latency

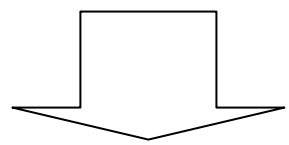
Limited WAN Bandwidth

TCP Chattiness ("window effect")

Send large file across the WAN (via WAN Friendly Protocol like FTP, HTTP, other)



With unlimited bandwidth and cross country latency the data transfer would take **60 seconds** due to the TCP based round trips



Divide traffic and send 64KB at a time across the WAN

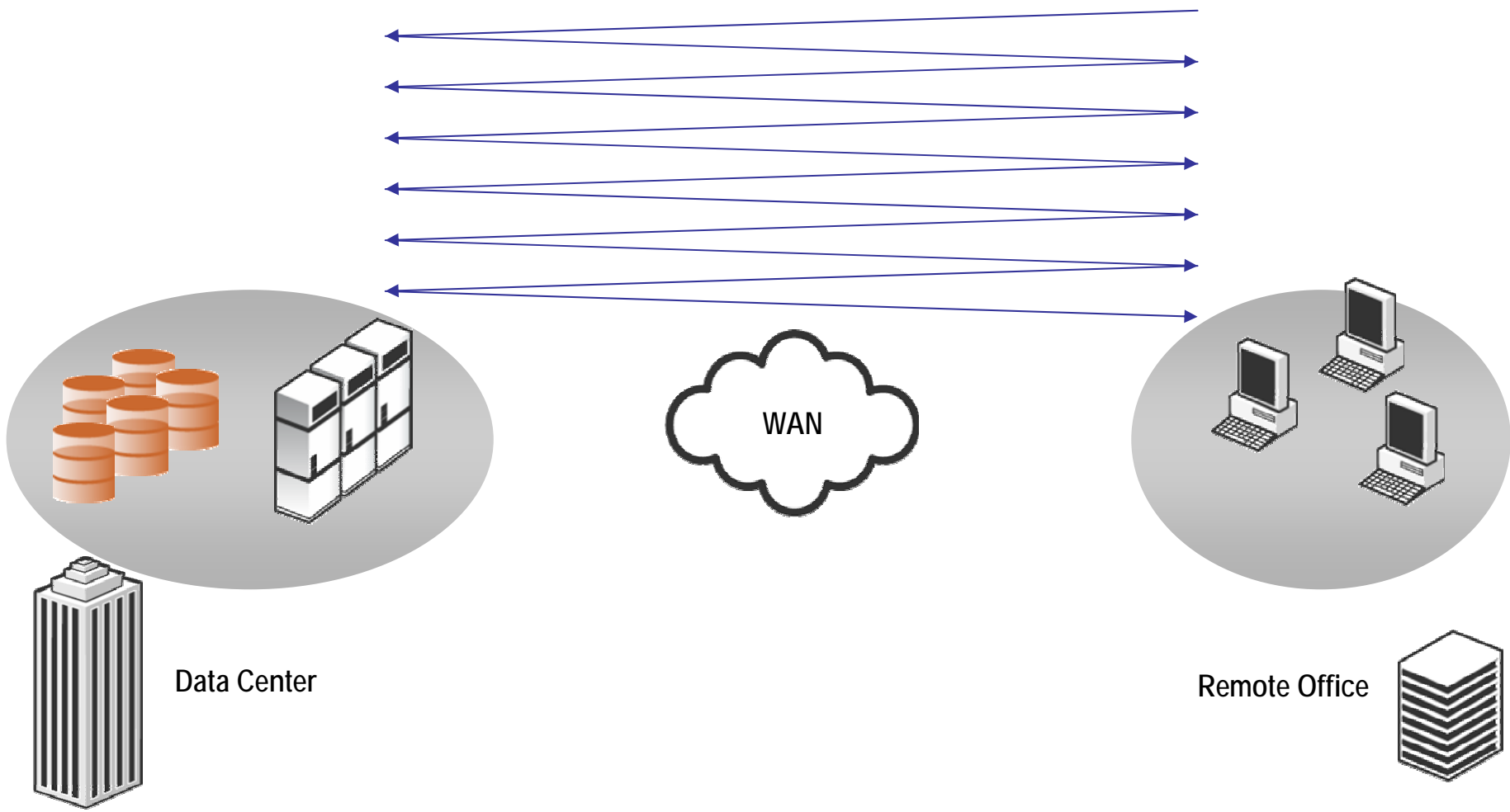


TCP chattiness (“window effect”)

- TCP: No more than one window of data in flight at once
 - Wait for ACK before sending next window
- Max TCP throughput: Window/RTT
 - For common configurations, 4Mb/s or less!
- Not fixable by adding bandwidth or packet compression

Application Chattiness

Typical Application Chattiness



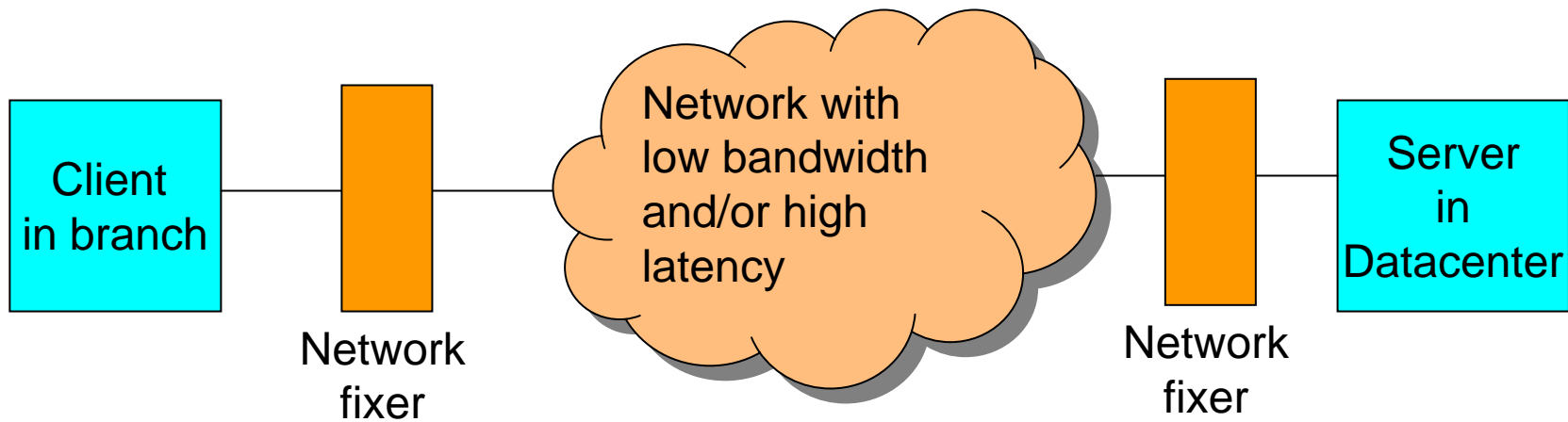
Application chattiness

- Widespread examples:
 - CIFS (Microsoft Windows file sharing)
 - MAPI (Microsoft Exchange email)
- Not really designed for wide-area usage
- Work OK in LAN – frequent small requests (“chatty”)
- Usage in high-latency networks leads to large delays

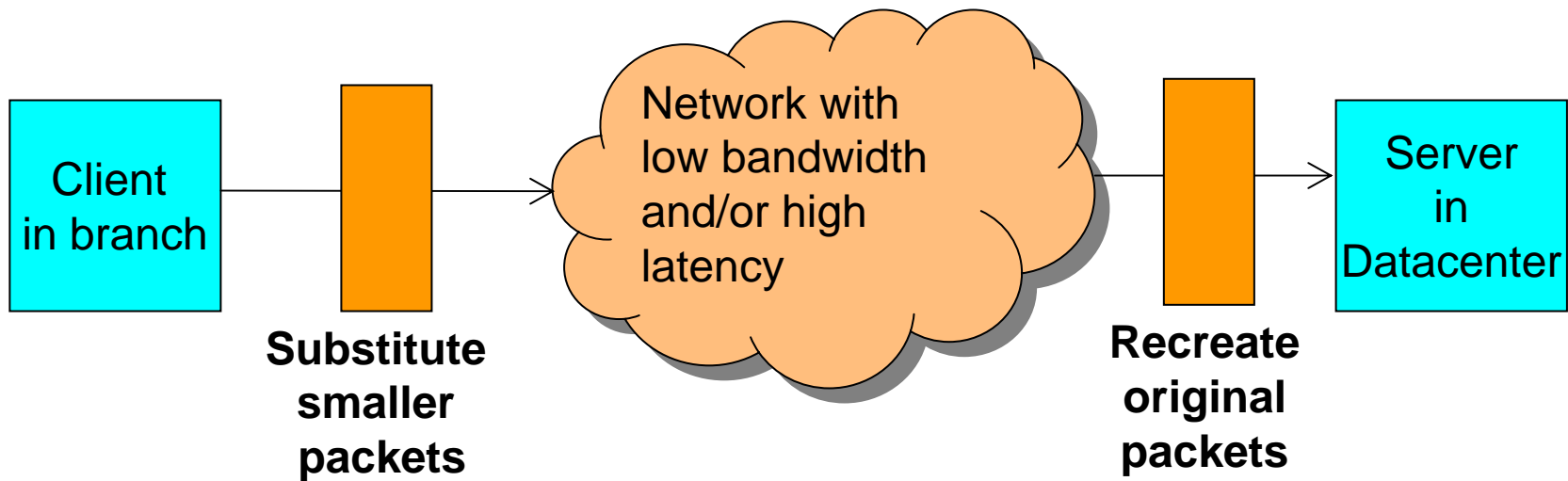
Solutions

- Compression
- Caching
- Compression + Caching
- Universal Transport Accelerator (UTA)
- UTA + local storage

Wide-Area Data Services: Conceptual solution



Compression: Block diagram

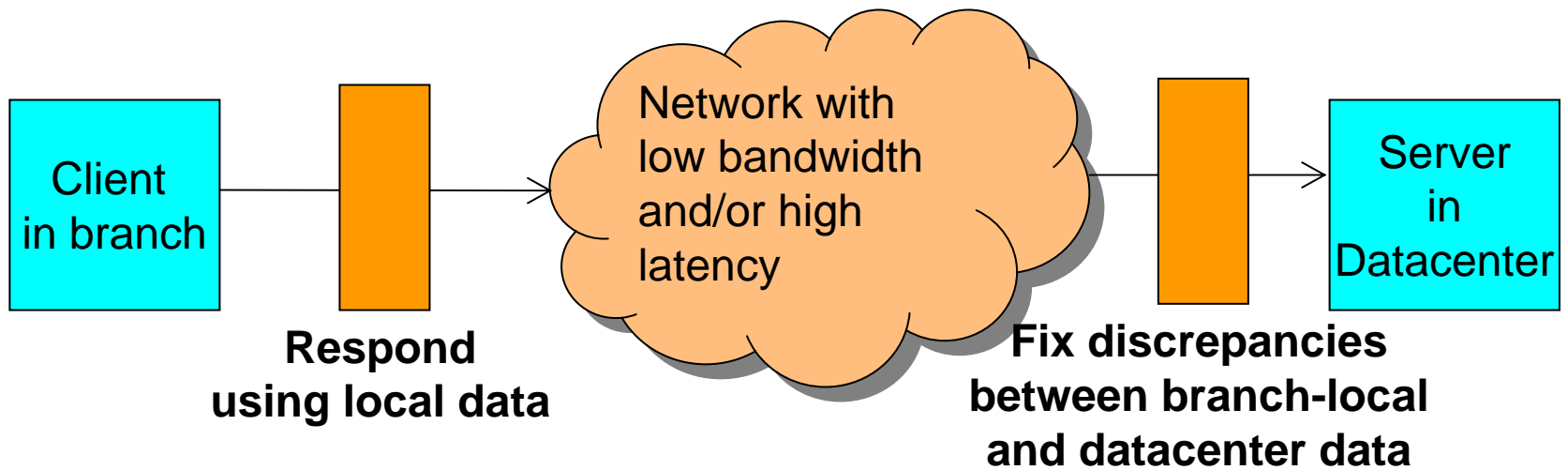


Note: Process is actually symmetric, identical behavior in opposite direction

Compression

- + Improves bandwidth usage
- + General-purpose
- TCP chattiness unaffected
- Application chattiness unaffected
- Memory-based implementations easily overrun

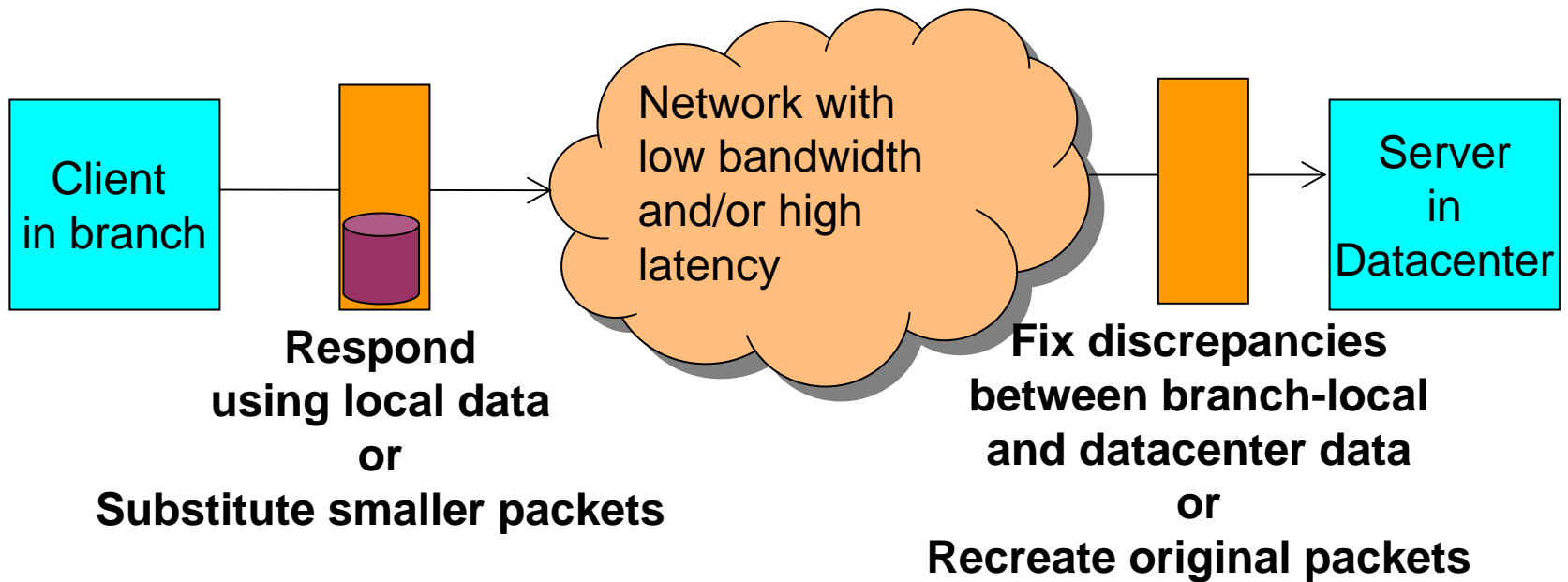
Caching: Block diagram



Caching

- + Reduce bandwidth usage
- + Reduce TCP chattiness
- + Reduce application chattiness
- + Disk-based implementations accommodate more traffic
- Only address specific protocol(s)
- Improvement only on cache hit
- Consistency/freshness issues

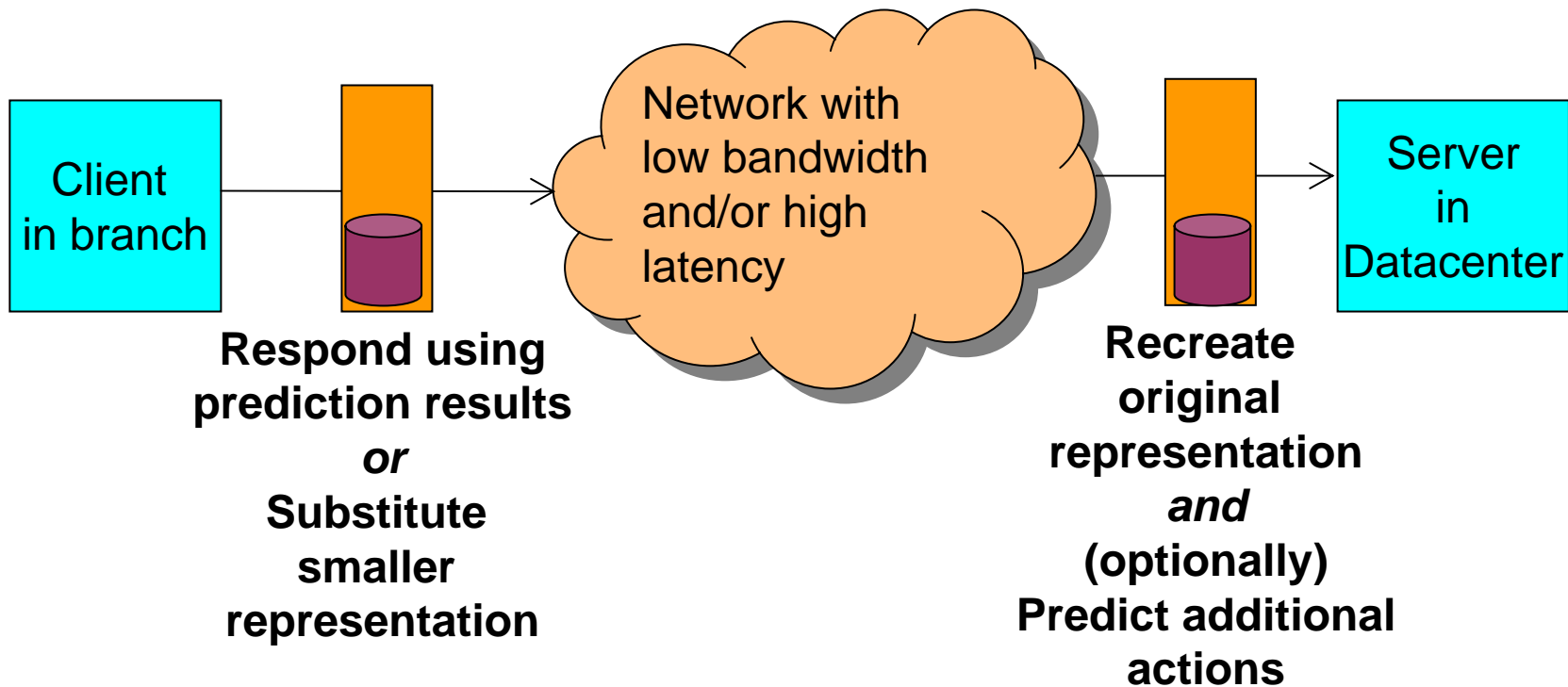
Compression+Caching: Block diagram



Compression + Caching

- + Reduce bandwidth usage
- + Reduce TCP chattiness
- + Reduce application chattiness
- + Disk-based implementations accommodate more traffic
- Chattiness improvements only on cache hit, only for cached protocol
- Consistency/freshness issues for cached protocol
- Memory implementation of non-cached protocols easily overrun

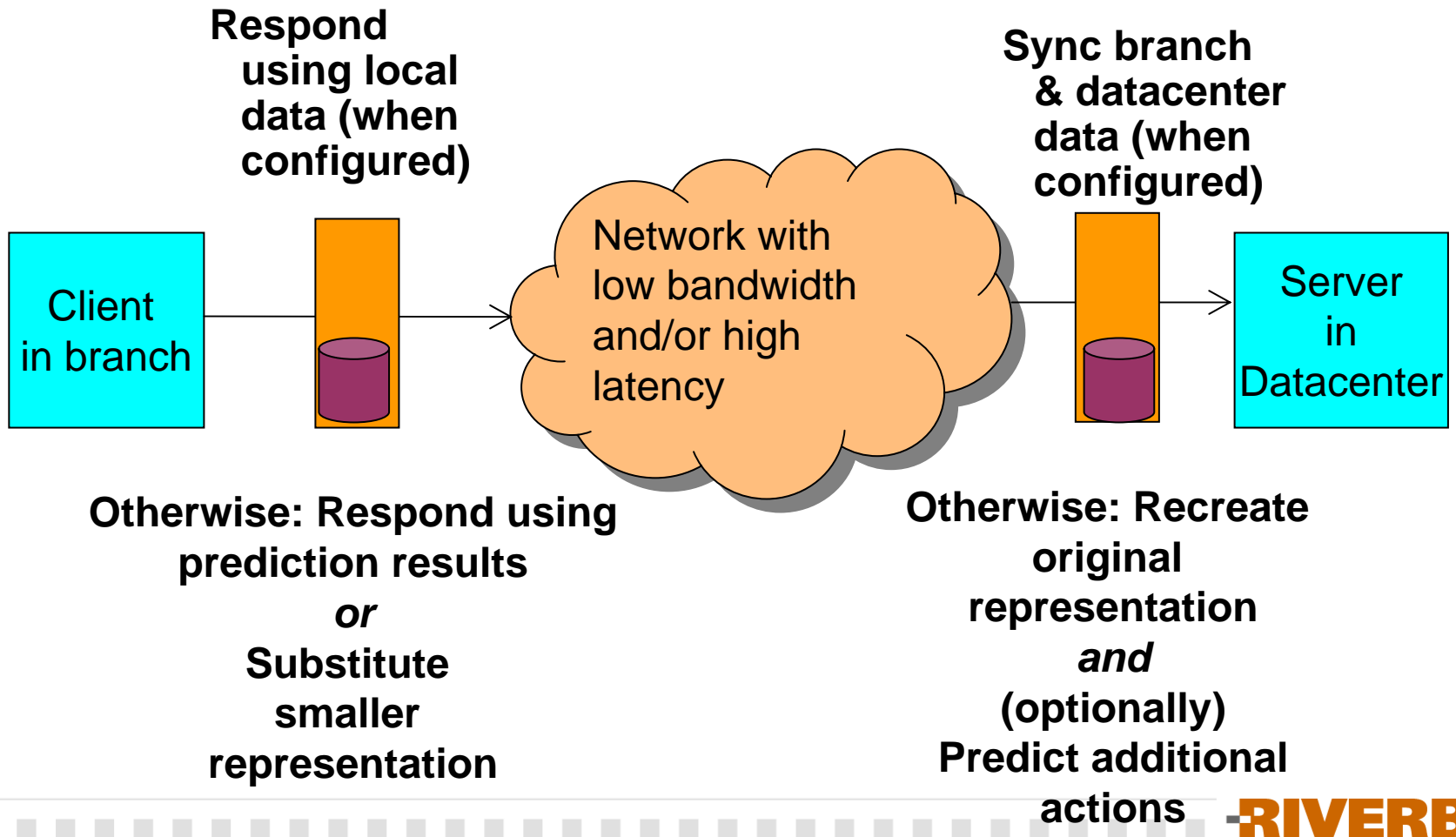
Universal Transport Accelerator (UTA): Block diagram



Universal Transport Accelerator (UTA)

- + Reduce bandwidth usage
- + Reduce TCP chattiness
- + Reduce application chattiness
- + Disk-based implementations accommodate more traffic
- + No consistency/freshness issues
- Application chattiness improvements only for predictions
- Always contact server

Universal Transport Accelerator (UTA) + local storage: Block diagram



Universal Transport Accelerator (UTA) + local storage

- + Reduce bandwidth usage
- + Reduce TCP chattiness
- + Reduce application chattiness
- + Disk-based implementations accommodate more traffic
- + No consistency/freshness issues (except as configured)
- Application chattiness improvements only for predictions + configured
- Always contact server (except for configured)

Summary

- Compression, caching both problematic for supporting Wide-area Data services
 - Combining compression and caching: better but not good enough
- Universal Transport Accelerator (UTA) rearranges compression/caching functionality and adds prediction
- UTA has better properties than caching or compression
- UTA + local storage gives additional interesting properties