Resilience Strategy

User-Controlled Adaptive Applications

• Introduction

• Resilience strategy
  - network survivability and resilience
    1. maintain survivable connectivity when possible
    2. survivable communication even when not connected
    3. resilient network mechanisms
    4. technologies to enhance resilience and survivability
  - end-to-end resilience and disruption tolerance
  - disruption-tolerant user-controlled adaptive applications

• Summary
## Application Flow Characteristics

<table>
<thead>
<tr>
<th>Application flow</th>
<th>Characteristic</th>
<th>Individual bandwidth</th>
<th>Start/transient delay</th>
<th>Steady-state delay</th>
<th>Latency budget</th>
<th>Loss tolerance</th>
<th>Adaptability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed computing</td>
<td></td>
<td>low–high</td>
<td></td>
<td>real-time</td>
<td>1μs–10ms</td>
<td>none</td>
<td>low</td>
</tr>
<tr>
<td>Process control</td>
<td></td>
<td>low</td>
<td></td>
<td>real-time</td>
<td>1μs–10ms</td>
<td>none</td>
<td>low</td>
</tr>
<tr>
<td>Haptics</td>
<td></td>
<td>very low</td>
<td></td>
<td>real-time</td>
<td>10 ms</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Live interactive voice</td>
<td></td>
<td>low</td>
<td>interactive</td>
<td>real-time</td>
<td>30 ms</td>
<td>very low</td>
<td>limited</td>
</tr>
<tr>
<td>Live interactive video</td>
<td></td>
<td>med</td>
<td>interactive</td>
<td>real-time</td>
<td>300 ms</td>
<td>low</td>
<td>moderate</td>
</tr>
<tr>
<td>Stored streaming video</td>
<td></td>
<td>mod</td>
<td>interactive</td>
<td></td>
<td>1–10 s</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Stored interactive video</td>
<td></td>
<td>mod</td>
<td>interactive</td>
<td>interactive</td>
<td>100 ms</td>
<td>low</td>
<td>moderate</td>
</tr>
<tr>
<td>Web browsing</td>
<td></td>
<td>med–high</td>
<td>interactive</td>
<td></td>
<td>100 ms – 1 s</td>
<td>none</td>
<td>moderate</td>
</tr>
<tr>
<td>Information push</td>
<td></td>
<td>low–med</td>
<td>push</td>
<td></td>
<td>1 min – 1 d</td>
<td>moderate</td>
<td>high</td>
</tr>
<tr>
<td>Telemetry</td>
<td></td>
<td>low–med</td>
<td></td>
<td>varies</td>
<td>varies</td>
<td>none</td>
<td>limited</td>
</tr>
<tr>
<td>Remote Backup</td>
<td></td>
<td>high</td>
<td>push</td>
<td>deadline</td>
<td>1 hour</td>
<td>none</td>
<td>high</td>
</tr>
<tr>
<td>Email</td>
<td></td>
<td>low</td>
<td>push</td>
<td>best effort</td>
<td>1 min – 1 hr</td>
<td>very low</td>
<td>high</td>
</tr>
</tbody>
</table>
# Application Categories

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Information Access</th>
<th>Telepresence</th>
<th>Distributed Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Relationship</td>
<td>client/server</td>
<td>peer-to-peer</td>
<td>varies</td>
</tr>
<tr>
<td>Bandwidth symmetry</td>
<td>asymmetric</td>
<td>symmetric</td>
<td>symmetric</td>
</tr>
<tr>
<td>Transfer granularity</td>
<td>large</td>
<td>continuous</td>
<td>varies</td>
</tr>
<tr>
<td>E2E Synchronisation</td>
<td>none</td>
<td>real time</td>
<td>varies</td>
</tr>
</tbody>
</table>
Application Categories

Information Access

- Client accessing information from a server
- Asymmetric bandwidth
- Response time is important metric
  - $100 \text{ ms} \leq T_r \leq 1 \text{ s}$ target
- Significant bandwidth requirement
  - individual and aggregate
  - bandwidth challenged for interactive response bounds
Application Categories

Information Access: Server Push

- Reduce response time by pushing data to user
  - server knows what user wants
  - data already present when user requests
  - reduces peak bandwidth
Application Categories

Telepresence

Data streams with embedded synchronisation

- Peer-to-peer exchange of virtual presence
  - Example: video conferencing
- Relatively symmetric bandwidth
- Bandwidth requirements
  - Teleconferencing
    - Aggregate bandwidth enhanced (operates at low speed)
  - Applications with specific requirements
    - Bandwidth enabled
      - Example: telemedicine
Application Categories
Distributed Computing and Net Storage

data exchange and synchronisation

- Distribution beyond a room (>LAN) of:
  - computations
  - storage connection to CPU

- Arbitrary exchange of control, data, state

- Application-dependent partitioning critical
Application Categories

Composed Applications

• Complex applications consist of multiple components
• Composition of
  – information access
  – telepresence
  – distributed computing
• Example: distance learning
  – information access for class and reference materials
  – telepresence for student/teacher interaction
Disruption Tolerant Applications

Adaptive with Knobs and Dials

- Applications should adapt to and mask disruptions
  - *dials* provide feedback from end-to-end paths (& network)
  - *knobs* influence transport and network behaviour
- Delay *translucency* (not transparency)
Application Adaptation

Compression

- Compression to reduce delay
  - total delay includes
    - transmission delay
    - compression/decompression
  - benefit tradeoff between
    - path bandwidth
    - processing rate and cycles
Disruption Tolerant Applications
Communication Association

• Recall: disruption tolerance goals for *applications*
  – information access by the user or application
  – end-to-end communication association

• Applications should *adapt* to path conditions
  – adaptive within modes
    • e.g. frame rate and resolution based on available bandwidth
  – adaptive between modes
    • e.g. video → audio → chat → messaging → email

• Driven by user preferences
  – e.g. tradeoff between frame rate and resolution
Disruption Tolerant Applications

Information Access

- Recall: disruption tolerance goals for *applications*
  - information access by the user or application
  - end-to-end communication association
- User experience highly dependent on response time
  - *interactive* information access
    - subsecond target response time
    - 100 ms ideal response time

![Image of latency and utility graph]
Disruption Tolerant Applications

Example: WVM Latency-Aware Web Browser

• Distributed information access
  - access information from remote locations
  - Web provides most common infrastructure
    • web browser as client
    • HTTP as protocol

• WVM (Web VADE MECUM)
  - knobs and dials
  - user behaviour emulation
  - prototype: Mozilla on Linux using IBM Research WBI toolkit
Disruption Tolerant Applications
Example: WVM Motivation

- Earth
- Mars
- Jupiter
- Server
- Demand Cache
- Server Push
- User
- Cache
- \( d_{est} \)

8 – 40 min
1 – 1.5 hr
1 – 1.5 hr

Sterbenz
Disruption Tolerant Applications

Example: WVM Knobs and Dials

- **Dials**
  - past response time $t_r$ history*
  - weak link connectivity (instantaneous rate $r_i$ to lighthouse)*
  - object size from server metadata
  - average end-to-end delay $l_i + l_s$ via probes to server
  - cached freshness* * implemented in prototype

- **Knobs**: application and user
Disruption Tolerant Applications
Example: WVM Dials - GUI

• Link color gives high level \{fast, old, slow, unknown\}

• Status bar indicates global and per link details
  - past response time
  - dynamic adjustment of weak link to lighthouse
Disruption Tolerant Applications

Example: WVM User Influence Knobs

- **Fetch action**
  - left click: default
    - get cached if available
    - profile based action
  - right click gives options:
    - fetch definitive
      refresh window when definitive copy arrives
    - nonblocking fetch
      definitive copy in new window when available

- **View menu selection**
  - allows display of unmodified page (un-munge HTML)
Disruption Tolerant Applications

Example: WVM Prototype Information Flow

- **Request**: User profile
- **HTTP Get**: RTT, bw, connectivity
- **Http Request Editor**: Set $ parms
- **User Profile**: Update
- **Store Statistics**: Hit, hit info + metadata
- **Response Editor**: Pass content re-color URLs
- **Local $**: (miss ∨ old) ∧ fetch
- **Web**: Net prot stack
- **Response Editor**: RTT, bw, connectivity
Disruption Tolerant Applications

Example: WVM Proxy States

- Strongly Connected:
  - serve from network and cache
  - aggressive preload and refresh
  - profile (all states)

- Weakly Connected:
  - serve from cache and network
  - controlled refresh and preload

- Disconnected:
  - serve only from cache

*states based on CODA*
Disruption Tolerant Applications
Example: WVM Profile and User Emulation

• All user requests pass through WVM
• WVM builds a user-specific **WVM-ProfileGraph**
  - tracks and represents user Web access pattern
  - nodes are URLs accessed; directed edges for traversal
• URL priority limits cache life and refresh bandwidth
  - nodes and edges fade using a **decay formula**
• Autonomic user emulation
  - graph is automatically traversed when fully connected
  - graph slowly/selectively traversed when weakly connected
  - cache kept fresh even when user away from client
  - cache is as fresh as possible after state change \{weak\|disc\}
  - coarse-grained learning possible (e.g. daily schedule)
Disruption Tolerant Applications

Example: WVM Profile Graph
Resilience, Survivability, DT

Summary

• Introduction

• Resilience strategy
  - network survivability and resilience
    1. maintain survivable connectivity when possible
    2. survivable communication even when not connected
    3. resilient network mechanisms
    4. technologies to enhance resilience and survivability
  - end-to-end resilience and disruption tolerance
  - disruption-tolerant user-controlled adaptive applications

• Summary
Survivability & Disruption Tolerance

Summary

• Attack problem at all levels:
  - physical, MAC, and link robustness and agility
  - network survivability
  - end-to-end survivability and disruption tolerance
  - disruption-tolerant user-controlled adaptive applications

• Beyond fault tolerance and crypto

• Design for survivability and disruption tolerance
  - expect challenging communication channel environment
  - expect and exploit mobility
  - expect, adapt, and mask high latency with user influence
  - interlayer awareness and control (knobs and dials)
  - intelligent resource & constraint tradeoffs (P, M, B, E, L)
Survivability & Disruption Tolerance

Primary References

Available from http://ww.sterbenz.org/sumowin

**SUMOWN/DTN**


**ETEN**


**WVM**


**Long latency**

Acknowledgements: Technical Contributors

• Lancaster University
  - Martin Dunmore
  - Hayat Kara
  - Laurent Mathy
  - Andreas Mauthe
  - Ben McCarthy
  - Chris Edwards
  - Idris Rai
  - Manolis Sifalakis
  - Steven Simpson
  - Linlin Xie

• UMass – Amherst
  - Jim Kurose

• BBN
  - Mark Allman*
  - Alden W. Jackson
  - Rajesh Krishnan
  - Ram Ramanathan
  - Tushar Saxena*
  - Martha Steenstrup*
  - Fabrice Tchakountio

* former affiliation

some of this work performed at BBN
Survivability & Disruption Tolerance

Acknowledgements: Funding Agencies

- Telekom Austria
- EC
  - 6NET
  - ANA Consortium
    - Telekom Austria AG
    - Universität Basel
    - ETH Zürich
    - Fraunhofer Fokus – Berlin
    - Lancaster University
    - Université de Liège
    - NEC Europe – Heidelberg
    - Universitetet i Oslo
    - Université Pierre et Marie Curie – Paris
  - E-NEXT (EC Net. Ctr. Excellence)
- DARPA
  - Doug Maughan* (SUMOWIN)
  - Rob Ruth* (GloMo)
  - Jean Sholtz* (WiaB)
- NASA
  - William Ivancic
    * former affiliation
- Cisco
End of Foils