Taking Advantage of Multihoming with Session Layer Striping

Ahsan Habib, *Siemens TTB*

Nicolas Christin, *Carnegie Mellon & CyLab Japan*

John Chuang, *UC Berkeley*

Disclaimer

- This is a genuine position paper
- We won’t present extended validation/simulation/experimentation results
- We mostly want to promote discussion in one area

Outline

- Background and motivation
- Strawman architecture
  - Session layer semantics
  - Connection management
  - Proposed implementation APIs
- Discussion

Problem context

- Internet access for residential users is cheap
  - ~$20-$60 for DSL in the US and Europe
  - Even cheaper in far-east Asia (Japan, South Korea)
  - Emergence of metropolitan wireless networks
    - E.g., San Francisco city
- Quality-of-service experienced by end hosts still relatively poor
- Residential multihoming (connecting to multiple ISPs simultaneously) could become attractive proposition
  - Circumvent last mile congestion
  - Benefit from diversity of peering relationships to have low overlap between different routes to destination
  - But currently, very little economic incentive to subscribe to more than one ISP!

(How) can we leverage residential multihoming?

Striping and multihoming

- Striping is resource aggregation
  - How to utilize all available network paths simultaneously
  - Technique that exploits multihoming support
- Not obvious where striping should be done
  - Link layer, network, transport, or application layer?
  - May even depend on the application!
    - For Web application network layer might work
    - For streaming and file transfer application may need explicit control
- Multipath congestion control
  - Congestion control mechanism for each path?
  - Application may decide about the transport protocol?

Design goals

- Decoupling striping from traditional network protocol stack to support multihoming
  - Avoids the overhead of rewriting most networking primitives at the application layer
  - Applications only see a single virtual "pipe," and do not need specific mechanisms
  - Multihoming support can be made independent of any specific transport protocol
  - Automated transport protocol selection on behalf of the application possible

Where should we stripe?

- Link-layer striping
  - Byte-by-byte resource aggregation → improves link utilization **but**
  - Byte-ordering must be preserved
  - IP datagrams may need to be reconstructed before crossing network boundaries → only useful for local area communications (fragmentation nightmare otherwise)
- Network layer striping
  - Multihoming can be transparent to transport layers
  - Easy to support multihoming for existing applications **but**
  - Poor transport-layer performance over heterogeneous paths
    - In particular, HOL blocking issues degrade TCP performance

Where should we stripe?

- Transport layer striping
  - Transport protocol stripes IP packets over multiple interfaces **but**
  - Need special transport protocol such as SCTP, pTCP
  - Might not suitable for all applications
- Application layer striping
  - Knows about application service expectations and can provide fine-grained performance **but**
  - Head-of-the-line blocking can reduce throughput significantly…
    - Unless application can peek at the transport-layer queues

Session layer striping

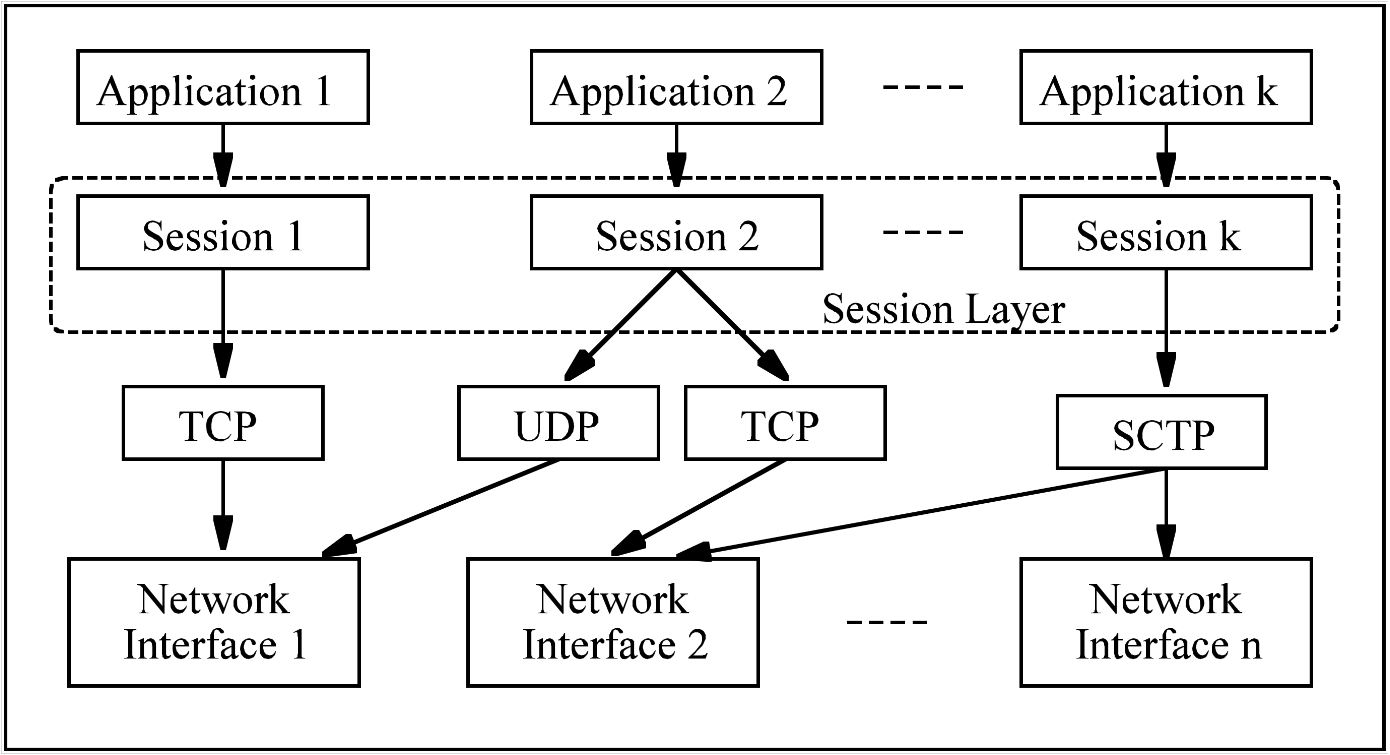
- Striping between transport and application layers makes most sense
  - ❑ Can benefit from application-layer flexibility
  - ❑ While having direct control over transport-layer flows
- Let’s resurrect the session layer for striping!
  - ❑ It was never really dead in the first place anyway

Is that such a novel idea?

- BEEP [Rose, 2001]
- MAR [Rodriguez et al., 2004]
- Congestion manager [Balakrishnan et al., 1999]
- TCP with TCB sharing [Touch, 1997]
- SCTP [Stewart et al., 2002, Iyengar et al., 2004]
- …

- Not designed for general multihoming framework
  - I.e., do not support arbitrary transport connections over arbitrary number of channels

Strawman architecture



- Applications informs session-layer about their QoS needs
- Session layer determines necessary transport protocols and striping mechanism to meet the requirements

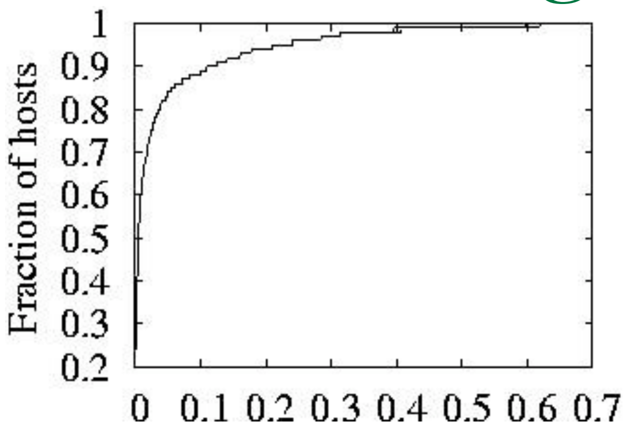
Session layer semantic objectives

- At least reliability semantics of single-homed connections
  - Lossless delivery
  - In-order delivery
  - No guarantees on loss or ordering
- Application performance improvement metrics
  - Throughput maximization
  - Latency, jitter, or loss minimization
- Fairness
  - Not necessarily an objective, but can be required by the network!
    - TCP friendliness enforcement
  - One may want to distribute traffic fairly over multiple stripes
    - E.g., Congestion manager, TCP block sharing over multiple connections

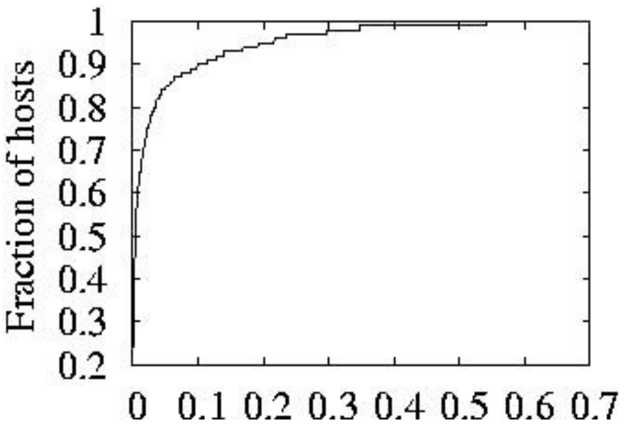
Conflicting semantics

- What happens if one objective contradicts another one? E.g.,
  - I want to minimize **both** loss and delays, but…
  - ISP1 always seem to provide lower losses than ISP2
  - But ISP2 always provide smaller latencies than ISP1
- Define discordance ratio D_{m_1,m_2} between two metrics (m_1,m_2)
  - Probability (averaged over time) that it is impossible to optimize for both metrics simultaneously
  - E.g., $D_{latency,loss}$ = 0.1 ⇔ 10% of the time, the interface with the lower latency has higher loss rates

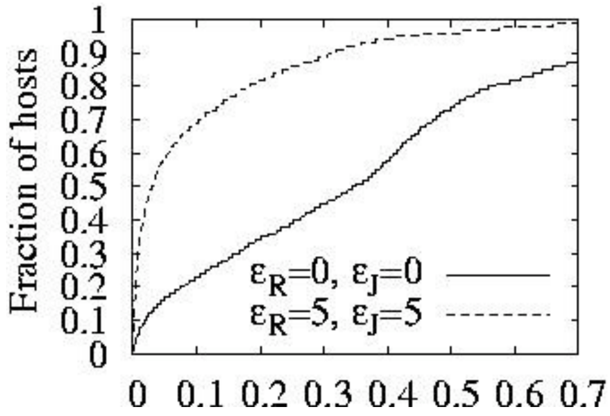
Conflicting semantics



$D_{latency,loss}$



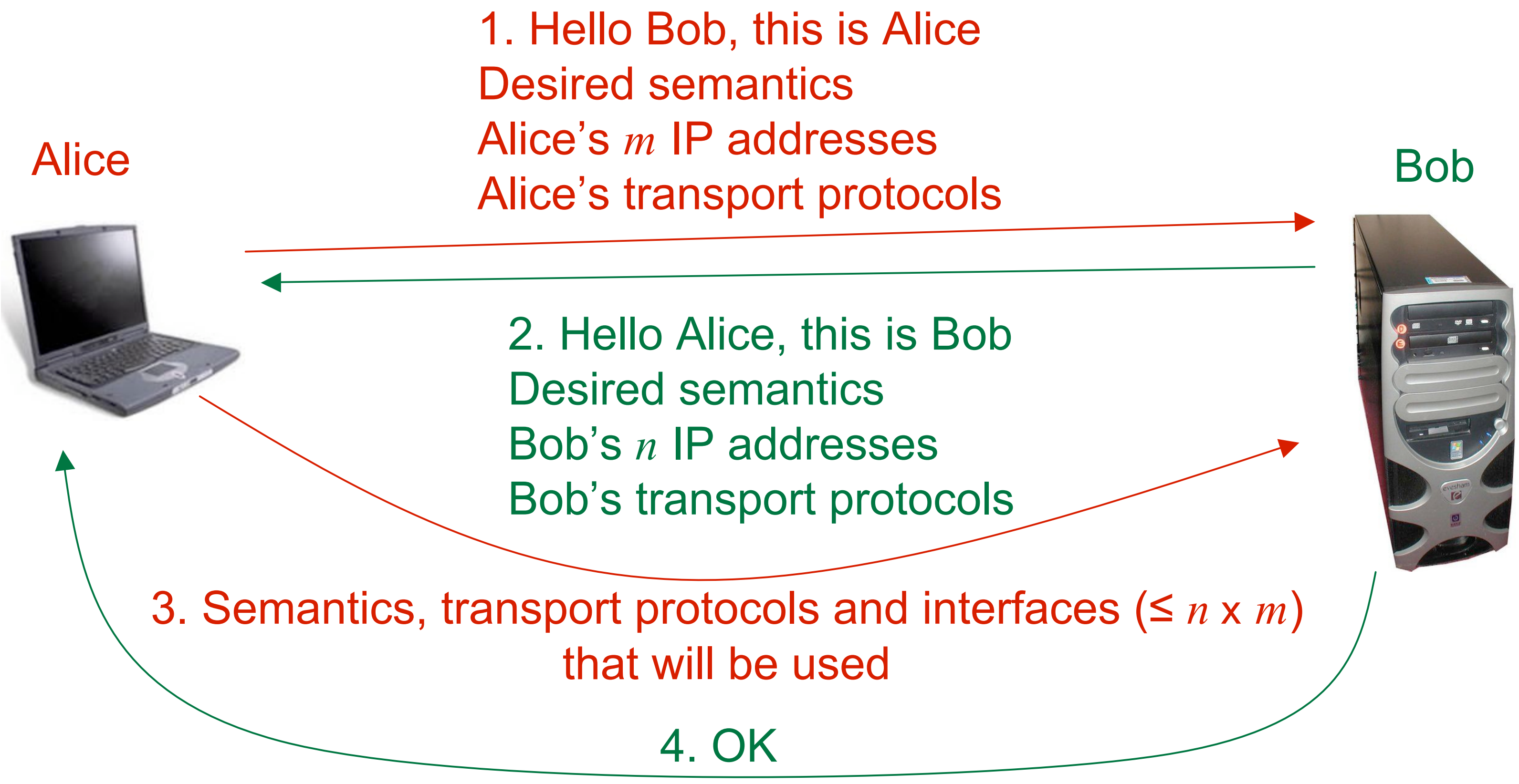
$D_{jitter,loss}$



$D_{latency,jitter}$

- Ran a quick experiment from a two-homed host to 100,000 hosts to get a rough idea of the situation
  - **Limited** experiment
  - We don’t claim results generalize
- Conflict in achieving objectives on several metrics seems to occur rarely
  - Static priority order may be enough

Connection establishment



Alice

Bob

1. Hello Bob, this is Alice
Desired semantics
Alice’s m IP addresses
Alice’s transport protocols

2. Hello Alice, this is Bob
Desired semantics
Bob’s n IP addresses
Bob’s transport protocols

3. Semantics, transport protocols and interfaces (≤ n x m) that will be used

4. OK

Connection established over reliable transport protocol

Connection management

- Path evaluation
  - Network layer metrics are evaluated for performance
  - Active measurements?
  - Short-lived connection can use scoreboard of recently used paths across all sessions
- Connection management
  - Managing all transport connections
  - Preserving order of packets before giving to session layer
- Data delivery
  - Depends on the performance guarantees supported
  - Tons of QoS literature on the subject can inform us
  - Weighted deficit round robin algorithm?

Implementation

- User space vs. kernel space
  - User space?
    - Easier to deploy
  - Kernel space? (e.g., kernel daemon)
    - Allows to easily obtain transport layer state variables for performance optimization
- API specification
  - BSD-type socket interface
  - Any application can bypass these APIs and use standard socket interfaces

APIs

Function	Parameters	Purpose
`session_socket`	Desired semantics	Create comm. endpoint
`session_bind`	Session descriptor, Port number	Listen to a local port
`session_connect`	Session descriptor, remote address and port	Establish session with remote host
`session_read`	Session descriptor, blocking flag	Request data from session layer
`session_write`	Session descriptor, data chunk, blocking flag	Provide data to session layer
`session_close`	Session descriptor	Terminate session

Summary

- Multihoming becomes a single virtual pipe to all applications
- Decoupling of striping primitives and traditional network stacks → independent of transport protocol
- Simple primitives for applications to use multiple interfaces
- Useful to describe service requirements of different applications

Open problems

- How do we securely exchange session information?
  - Diffie-Hellman type of exchange
    - Similar to TLS
  - But, we need (yet another?) PKI…
  - Zero-knowledge exchanges?
- Specific instances of performance optimization algorithms that can be used within this framework?
  - See MMCN'06
- Proof-of-concept implementation
- Still falls short of complete application transparency
  - Should we/can we build another piece to intercept regular socket calls?

Discussion/Questions?