Transport Option Selection

• Paths Characteristics

• Application Needs
  • Optimize for bandwidth, latency, or cost
  • Traffic characteristics
  • Tolerance towards packet/data/connection loss

• Transport Protocol Stack
  • On-Path-Elements!?
Socket Intents 101

With Socket Intents, applications MAY express their communication preferences in order to take advantage of the available transfer diversity.

Socket Intents are applications’ hints for Transport Option Selection

• Intuitive
• Generic
• Protocol Independent
• Best Effort
Initial Socket Intents Types

Application programmers opening a communication channel typically know how this channel will be used.

- Traffic Category
- Object Size sent/ received
- Duration
- Burstiness
- Stream Bitrate sent/ received
- Timeliness
- Application Resilience
- Cost Preferences

➢ Inputs to automatic Transport Option Selection.
Socket Intents Integration

- *Socket Intents can be used on a per Flow, Association, Stream, or Object level.*
- Socket Intents are API independent
  - BSD Socket based implementation + connection caching
  - Post-Socket implementation feasible
Comparing Transport Options

• Path Properties
  Bandwidth, RTT, loss probability

• Application Needs
  Socket Intents, QoS/IntServ

• Protocol Stack?
  IPv4/TCP(new Reno,Nagel)/TLS(1.3),
  IPv6/ESP(…)/UDP/QUIC

• On-Path-Elements?
  middle-boxes / MPTCP proxies / PMIP

➢ We need a framework to compare protocol stacks
Comparing Protocol Stacks

Decompose Protocol Stacks into

• Communication Units
• Mechanisms

This perspective allows us to compare mechanism like distributing requests of an application among different paths or using MPTCP despite their different nature and layer of implementation.
Communication Unit Hierarchy

• Object (Message)
  • HTTP-Request/Response-Header/Body for HTTP/2
  • XML message in XMPP

• Stream
  • Stream in QUIC or SCTP
  • TCP connection used for XMPP

• Association / Flow
  • QUIC / TCP connection carrying HTTP/2 frames.
  • IP/TCP or IP/UDP with same 5-tuple.

• Association Set / Flow Set
  • RTP stream and corresponding RTCP control messages.
Mechanisms in Multi-Path Systems

• Destination Selection
  • DNS Resolution

• Path Selection
  • Set SRC for outgoing connection
  • Establish an MPTCP subflow
  • Choosing a Subflow for a Segment

• Chunking
  • Objects of a Website
  • TCP Segments

• Scheduling
  • Use path “X” as long as available
  • Assign segment to subflow as long as it fits the congestion window (MPTCP)

Transport Protocol Stack Instance Selection means composing a System from Protocol Instances implementing these Mechanisms.
Socket Intents for BSD Sockets

• Prototype implementation as wrapper library for BSD sockets
• Socket Intents represented as socket options
• Three API Variants
  • Classic API with muacc_context argument
  • Classic API with all automation in getaddrinfo
  • Socketconnect API

https://github.com/fg-inet/socket-intents/tree/release-0.6
In our current implementation, the policy is a piece of code which can in principle execute arbitrary instructions.
API Implementation Experiences & Lessons Learned

• Name resolution is quite separate from socket handling
  • Need to be closely related for automation

• File Descriptors Considered Harmful
  • Semantic of FD operations is protocol specific
  • Protocol stack instances selection does not fit classic API calls
  • Much undefined behavior with automation results
Open Questions

• How to proceed with Socket Intents?
• Relationship to other WG work?
  • I-D.pauly-taps-guidelines
  • I-D.trammell-post-sockets
  • I-D.gjessing-taps-minset
Open Questions

• Proceed with this document within the WG?
• Any wishes what to elaborate on?

_We consider this draft primarily as background to I-D.tiesel-taps-socketintents and input to other WG work_
Open Questions

• Impact on Terminology other drafts
  …in particular, I enjoy the versatility of the term “flow”.

• Challenges of transport option selection
  • Mechanisms fit into multiple abstraction levels
  • Protocol Stack Instances might duplicate functionality
  • Prefer features at application level vs at transport level
Open Questions

• How to use all those Inputs in a Policy Framework?
• How to represent a Policy?