An Architecture for Transport Services

draft-ietf-taps-arch-01

Tommy Pauly
TAPS
IETF 102, July 2018, Montreal
Status Update

Architecture draft uploaded as WG document *(draft-ietf-taps-arch-01)*

Added section for **Protocol Stack Equivalence**

Added section for **Message Framing and Parsing**
Protocol Stack Equivalence

A Protocol Stack is defined as a set of application, transport, and Internet protocols; along with protocol-specific options

Multiple Candidate Protocol Stacks can be raced during connection establishment

Equivalence is defined as a requirement for being able to race or swap between two Candidate Stacks
Protocol Stack Equivalence

1. Same API surface

Both stacks must offer the same interface to the application for connection establishment and data transmission.
Protocol Stack Equivalence

2. Same transport services

Both stacks must offer the same transport services, as required by the application.

Reliable Transmission Required

UDP
IP
≈

TCP
IP
TLV Frame

Unreliable Transmission Allowed

UDP
IP
≈

TCP
IP
TLV Frame

UDP
IP
≠

TCP
IP
Protocol Stack Equivalence

3. Same security properties

Security protocols must be identical to ensure equivalence.

H2 Stream
TLS 1.3
TCP
IP

TLS 1.3
TCP
IP

TLS 1.3
TCP
IP

TLS 1.2
TCP
IP
Message Framing and Parsing

Defines focus for data transfer around Messages

Protocols that natively support messages or datagrams translate naturally

Stream protocols only preserve one message boundary (the end of the stream)

Message framing is pushed down into Protocol Stacks to allow boundaries to be defined within streams
Open Issues
Document Restructure

Needs expanded motivation section in introduction

Messages are introduced late in the document, and should be described up front
Open Issues
Privacy and Security

Needs sections on Privacy and Security Considerations

Storage and caching policy can leak information between connections