

# ICCRG + TAPS

## On deploying new algorithms

Tommy Pauly (tpauly@apple.com)

ICCRG  
IETF 98, March 2017, Chicago

What is the Transport Services WG (TAPS),  
and why is it relevant to ICCRG?

# TAPS Charter

1. Define a set of Transport Services, minimally existing IETF protocols and **congestion control mechanisms** used between two endpoints.
2. Specify the subset of those Transport Services that end systems supporting TAPS will provide, and give guidance on choosing among available mechanisms and protocols.
3. Specify experimental support mechanisms, **explain how to select and engage an appropriate protocol** and how to discover which protocols are available.

# TAPS Charter

## Progress

- ✓ 1. Define a set of Transport Services, minimally existing IETF protocols and **congestion control mechanisms** used between two endpoints.
- ↻ 2. Specify the subset of those Transport Services that end systems supporting TAPS will provide, and give guidance on choosing among available mechanisms and protocols.
- 3. Specify experimental support mechanisms, **explain how to select and engage an appropriate protocol** and how to discover which protocols are available.

# When to use a new algorithm?

## Options

1. **Default**, for general-purpose options only
2. **Explicit opt-in**, such as socket options to enable LEDBAT on a connection
3. **Inference from Abstract Options**, such as an option to mark a connection as “background”
4. **Path awareness**, such as knowing a device is on a specific network that requires a different algorithm

# Relevant Drafts

draft-grinnemo-taps-he

*Happy Eyeballs for Transport Selection*

draft-trammell-taps-post-sockets

*Post Sockets, An Abstract Programming Interface*

draft-pauly-taps-guidelines

*Software Architecture Guidelines for Protocol Evolution*

# Protocol Ossification

A major impediment to deploying new protocols and algorithms is ossification, when only well-established protocols are effectively supported

- **Network Ossification** comes from middleboxes and firewalls that interact badly with unexpected protocols
- **Software Ossification** comes from APIs and implementations that cannot easily be upgraded to support new options

# Protocol Ossification

from software

Not all protocol inflexibility in software is bad:

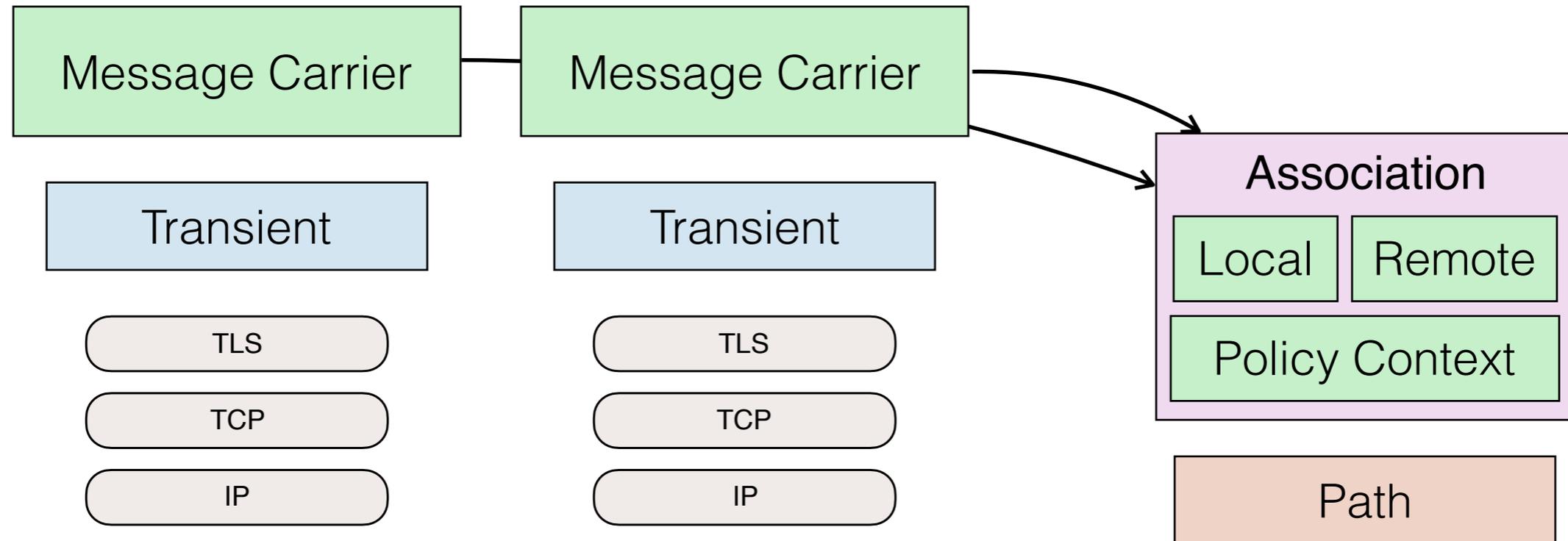
- **Specificity (voluntary inflexibility)** is when an application relies on a specific protocol in order to achieve its goals and be compatible with other devices; for example, using HTTP/TLS/TCP
- **Ossification (involuntary inflexibility)** is when an application cannot support new options that it could benefit from, but its API or implementation cannot easily support it

# Protocol Ossification

from software

- Specifying a specific algorithm or option in API means that if this option is deprecated or replaced, it may be hard to transition the application
- We should be careful in how we recommend exposing explicit options
- Goal should be providing the system with the right information to choose the best option

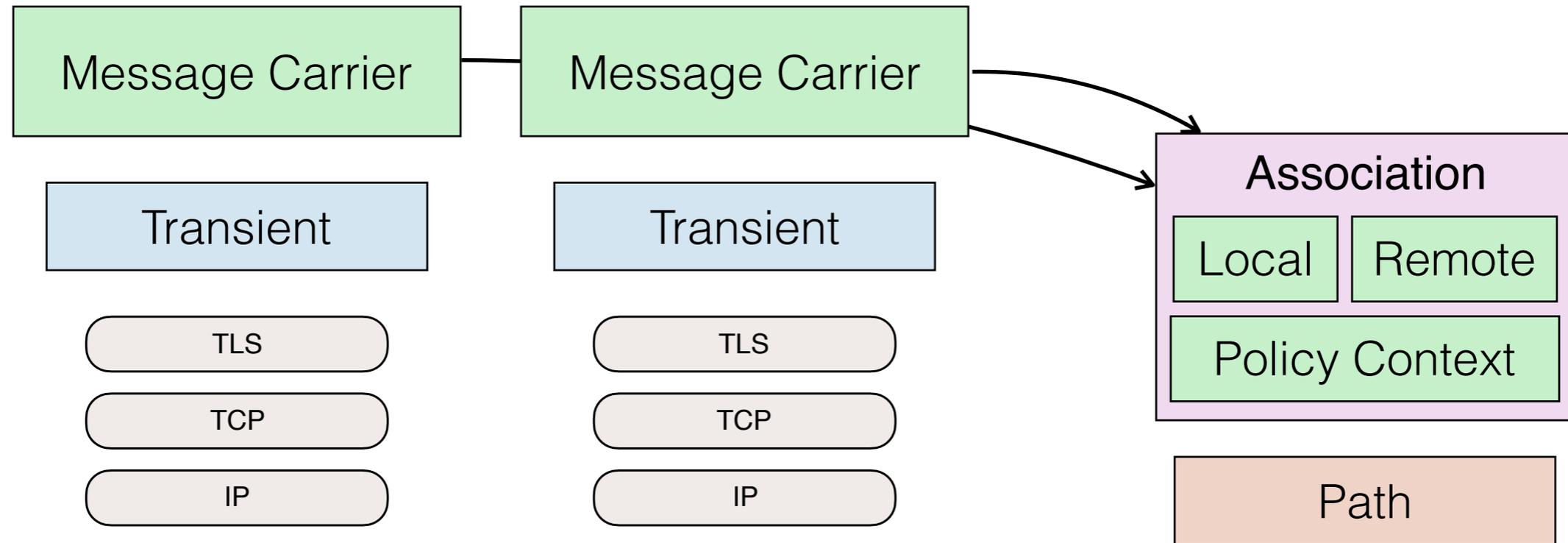
# Post Sockets Architecture



Sockets provide a single transport stack instance

Post-Sockets has an association that can let instances share state, and each “Carrier” may attempt/race different transport stack instances

# Post Sockets Architecture



The **Association** and **Path** objects can hold information about which CC algorithms should be used based on historical use, and characteristics of the network

# Open Questions

- What are the right API semantics for CC algorithm options?
- What can we learn about our network/path to help us choose the right algorithms?
- How can systems try new algorithms safely, by racing or employing fallback heuristics?

# Contribute

Come to TAPS!

Tuesday 16:40-18:40, Zurich E/F

Send out ideas to ICCRG and/or TAPS lists