Moving Beyond Sockets

Architecture and Observations

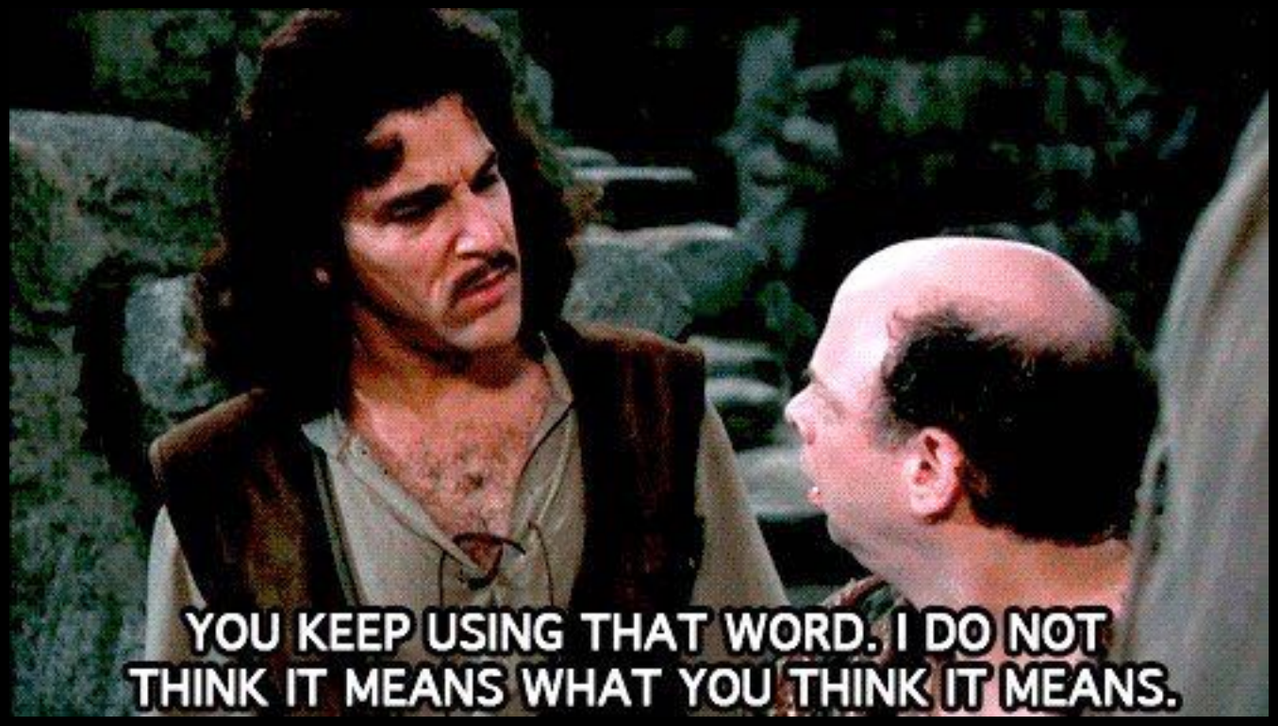
Tommy Pauly (tpauly@apple.com)

TAPS
IETF 97, November 2016, Seoul

Context

- TAPS is about providing easier ways to use various transport protocols, and fallback between them gracefully
- The Post-Sockets proposal aims to define the abstraction to allow asynchronous, multipath, multistream, secure, message-based networking
- We've been developing networking infrastructure at Apple to allow us to deploy robust implementations of Happy Eyeballs, interface fallback, MPTCP, etc. We'd like to see this converge with the TAPS and Post-Sockets efforts

Definitions



Endpoint: *An identifier for a network service, such as an IP address + port, hostname + port, or Bonjour service name. Often have both local and remote.*

www.example.com:443

myserver._http._tcp.local

192.0.2.1:80
@en0

2001:DB8::1.443

Path: *A view of network properties (a provisioning domain) that can be used to communicate to an endpoint. Route lookup++.*

Remote Endpoint: www.example.com:443

Parameters:

Prohibit interface type

Result:

Interface Type

Quality

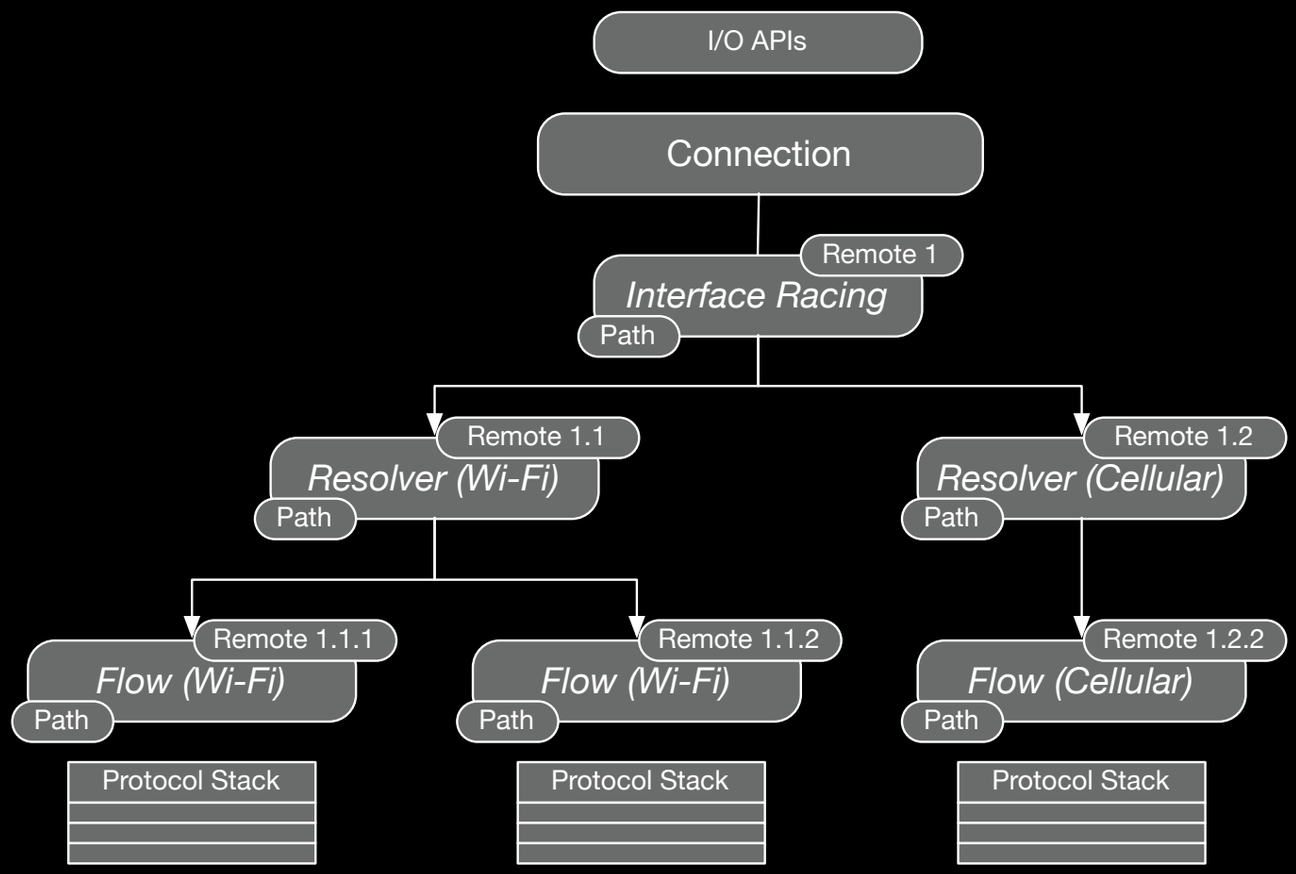
Interface

MTU

Cost

Connection: *A flow of data between two endpoints, using one or more paths. Created with a set of parameters about client preferences.*

I/O APIs may have Stream, Datagram, or Message semantics



Observations

1. Attempt multiple paths
2. Build paths dynamically
3. Use parallel protocol stacks

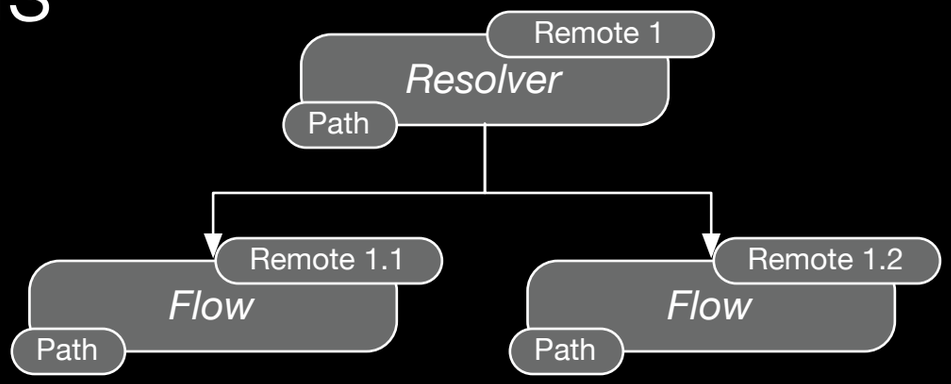
Observation 1

Attempt Multiple Paths

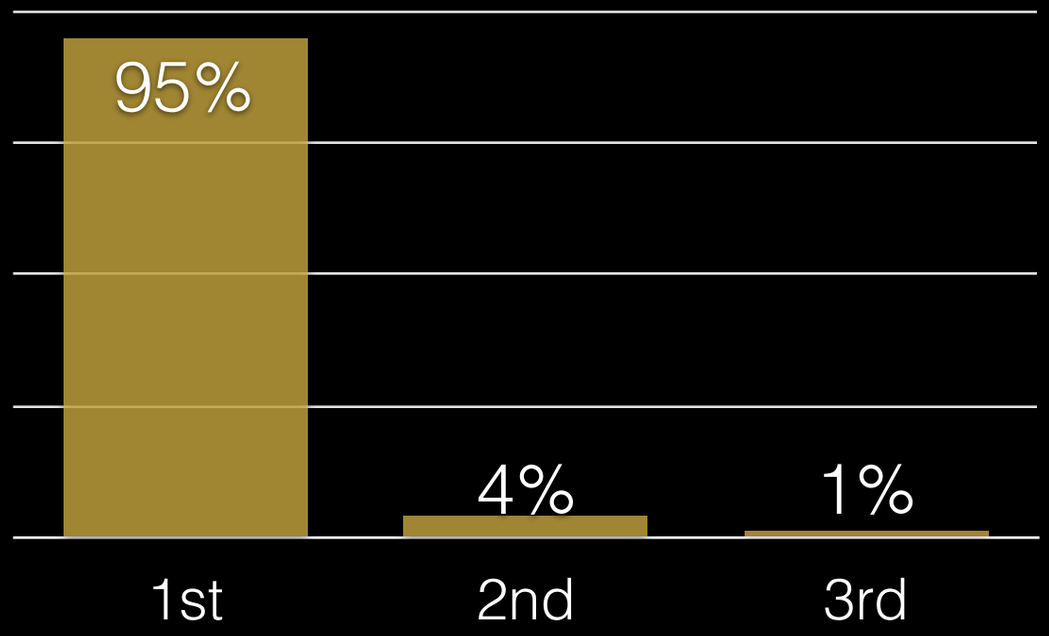
- Try multiple IP Addresses per hostname
- Try multiple hostnames per bonjour service
- Try different interfaces and/or local addresses
- Try different protocol stacks (transports, proxy protocols, etc)

Racing Resolved Addresses

- We receive multiple addresses for 47% of all connections



- Of those, 5% of connections end up needing to try something other than the first address



Observation 2

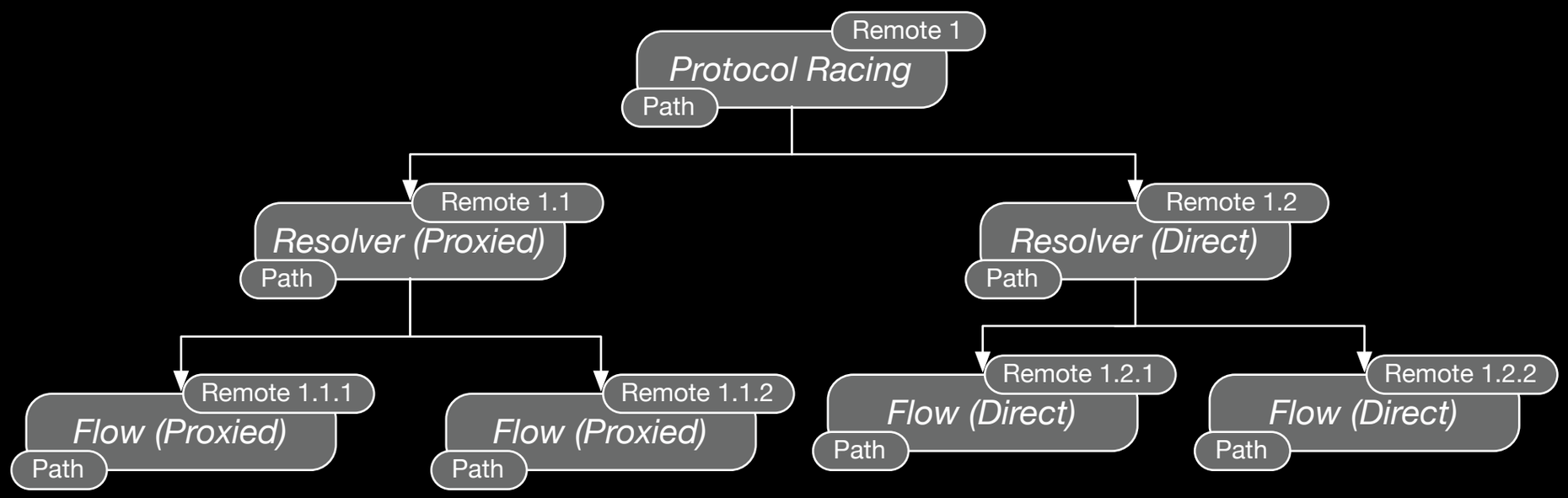
Build Paths Dynamically

- Client parameters
- System settings
- Network conditions and state

Racing Protocols

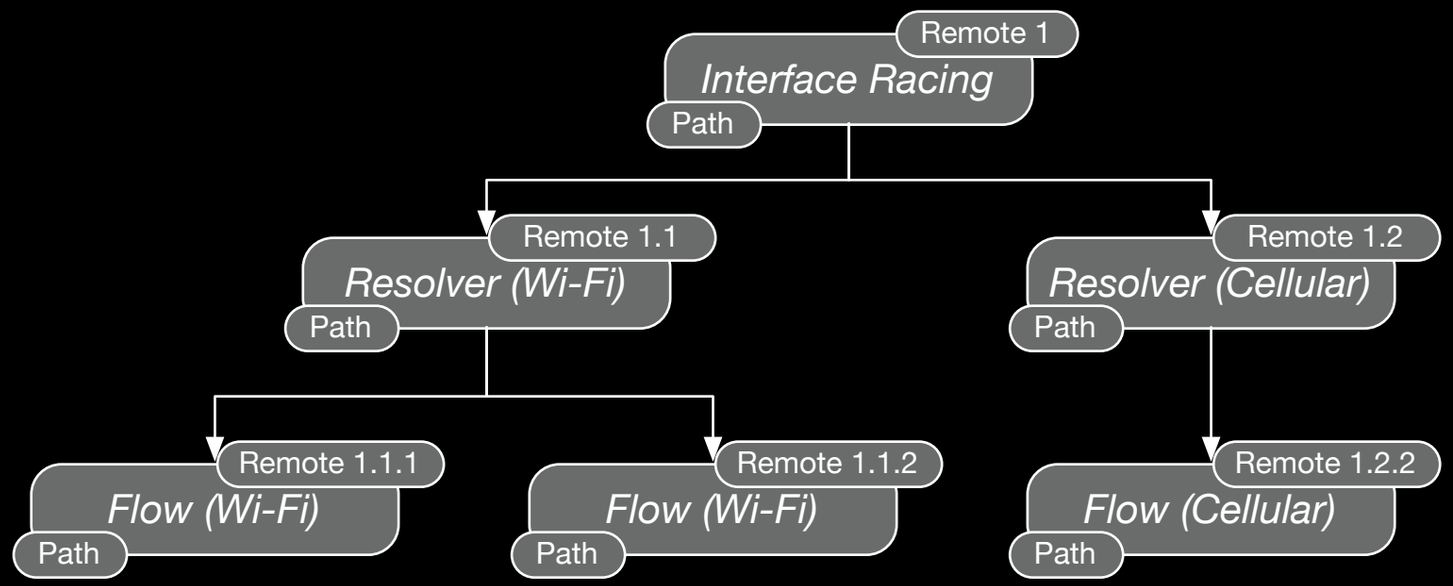
- Around 7% of connections are eligible for modifying their protocol stack to use a proxy or other modifier

  5% by client opt-in, 2% by system configuration



Racing Interfaces

- When we determine that attempts should be made over multiple interfaces, 20% of connections use the non-primary interface



Observation 3

Use Parallel Protocol Stacks

- TCP attempts in parallel
- Application-level handshakes (such as TLS, proxies, or HTTP) can occur in parallel
- Required to enable fast-open or 0-RTT protocols

Parallel Protocol Stacks

- Each flow attempt can contain a separate instance of each protocol, allowing racing to occur independently (and use Fast Open, etc)