Internet Area

IPv6 Multi-Addressing, Locators and Paths

Objective

- To facilitate an Internet Area discussion in the next 45 (or so) minutes on IPv6, Multi-Addressing and Path Maintenance approaches

- Goals:
  - Raise awareness of the concepts
  - Summarize current activities
  - Flag open issues
  - Consider further activity

Background

- Conventionally, IP addresses are
  - Endpoint identifiers
  - Routing objects
  - Key value for Forwarding Lookup

    (but you knew this already)

Background

- Challenges to the IP Address Model
  - Mobility and nomadism
  - Multi-homed endpoints
  - Scoped address realms
  - Routing Complexity and Scaling
  - VOIP and Peer-to-Peer applications
  - NATs, ALGs, and firewalls
  - Unwanted traffic, session hijacking and disruption

百花齊放，百家爭鳴 *

- Our current direction appears to be developing solutions in diverse permutations of this split identity / locator space simultaneously:
  - Multi-Party Applications
  - Application Agents
  - Rendezvous protocols
  - DNS Incremental Updates and DNSSEC
  - DNS Indirection and Referral
  - SCTP, HIP at the transport-layer
  - Mobile IPv6
  - Mobile IPv4
  - Multi6
  - And probably many more!

* *Let a hundred flowers bloom: let a hundred schools of thought contend*
Mao Zedong, 1956

Background

- Generic approach: decouple the semantics of identity and location:
  - Associate multiple locations to a single identity
- Consequent “binding state”: mapping an identity into a viable locator
  - in a packet header for the sender
  - reverse mapping for the receiver
- Using the IP layer as the point where this binding state is maintained
- Once a binding state is established
  - transport and above uses identifiers
  - IP and below uses locators

Background

- A number of current IETF activities are looking at aspects of decoupling identity and location at the IP layer:
  - IKEv2 + MOBIKE (+ BTNS)
  - MIP4 + MIP6 + combinations (MIPSHOP, MOBOPTS)
  - NEMO
  - SHIM6
  - HIP

Functional Components

- From a functional perspective, the approaches appear to have similar structural components:
  - Discovery of locator functionality between end-hosts
  - Identity / Locator mapping state Setup
  - State Update (locator set change)
  - Path Maintenance

We already have multiple **Discovery** and **Setup** protocols …

- Different security assumptions behind each approach
  - IKEv2 (+MOBIKE), MIP6, SHIM6, HIP, …
- Different functionality requirements
- Different domains of intended applicability

- There appears to be limited capacity and/or benefit in attempting to unify these approaches

Could we have a single **locator** / **path** Update and Maintenance module?

- Is it possible to use a single common locator update protocol as a plug-in to the signalling protocol?

- Is it possible to use a single common path property discovery / maintenance mechanism as a plug-in to the signalling protocol?

Issues – Transport Requirements

- Who cares about locator switch events (and why)?

- Various different transport session requirements:
  - TCP
    - avoid session resets
    - optimise path performance
  - UDP streamers
    - avoid stream disruption
    - Prefer rapid failover to pre-configured path
    - match path performance to media requirements
  - UDP transactions
    - avoid excessive transaction overhead

Issues - Locator / Path Maintenance

- Path integrity monitoring: Upper Level Signalling vs IP Level Monitoring
  - Indirect: Use Transport Session referred signals
    - Transport session timeout generates a locator switch signal
      - Locator pair testing?
      - Interpretation of signals? (Firewalls and filters for specific transport ports?)
  - Direct: Use pseudo-transport session
    - Probe and response within the shim layer
      - Complete pair-wise locator maintenance
      - On failure locator testing

Issues - Identity Equivalences

- Locator State Maintenance
  - What is an identity state equivalence set?
    - Per Host pair

      For some generic form of associating multiple IDs with a single endpoint
    - Per ID pair

      The ID pair forms a unique lookup key to the mapping state
    - Per session class

      The ID pair plus a session “type” value forms the state lookup key
    - Per transport session

      The ID Pair plus the session identifier forms the state lookup key
  - What is required to identify an incoming packet in terms of selecting the correct mapping state?

Issues - Path Maintenance

- Passive: await locator switch signal and then select a “new” pair and test
  - Maintain timed cache of ‘bad’ pairs
  - Test new candidate locator pair
    - Testing may generate n**2 probes
    - Testing of new pairs requires extended timeouts
    - Parallel vs serial test procedures
- Active: Actively maintain and probe all locator pairs asynchronously
  - Rapid failover – high overhead
- Active ++ : maintain path characteristics per locator pair
  - Path matching failover options – higher overhead

So - is it possible…

- To construct the identifier / locator mapping module in such a way that it can be modular?
- That the signals in / out of the module can be defined in a functionally complete manner?
- That the module can support multiple setup and signalling protocols?
  - Sharing the mechanisms and probe information but
  - Probably not sharing the (complete) state
- That the module's internal operation can be opaque to the calling interface?