Native HIP API

http://www.iki.fi/miika/docs/draft-mkomu-hip-native-api-00-pre1.txt
http://infrahip.hiit.fi/hipl/hip-native-api-final.pdf

Miika Komu <miika@iki.fi>
Helsinki Institute for Information Technology
InfraHIP

Goals

- Apps can control HIP layer better
  - Application specified identifiers
  - HIP layer security attributes
- Hide representation of HI/HITs with Endpoint Descriptors
  - Process migration
- Detect HIP capability
  - Fallback to plain TCP/IP

Status of the Draft

- The draft is just the original proposal converted to a draft format to get an editable document
  - all comments are welcome
- Feedback
  - Manual configuration of HIs and IP addresses is possible without DNS interaction.
  - Binding to interfaces instead of IP addresses
    - You bind to an HI, not an IP address. It is also possible to select a specific IP address.
  - Acronyms/constants: EID -> ED, EF_HI

HIP Socket Handler Implementation

- socket(PF_HIP, SOCK_STREAM, 0);
- Primary tasks
  - (De)allocates Endpoint Desciptors
  - Translates Endpoint Descriptors to HITs
- Implemented as a wrapper to IPv6 socket handler
  - Creates an IPv6 socket
  - Replaces IPv6 addresses with HITs in the socket

Fallback to TCP/IP

- What if a host does not support HIP?
- Localhost HIP capability can be detected by creating a HIP socket
- Peer capability can be detected from the DNS or trying base exchange
    - If the peer has no HI/HIT in DNS, the resolver can return IP addresses
    - Opportunistic mode: it is possible that the socket handler falls back to TCP/IP after a timeout

Open Questions / Next Steps

- Service Identifiers?
- The endpoint resolver name
  - getaddrinfo extension: porting is easier
  - getendpointinfo: future extensions easier?
- Should the resolver be socket based?
- Should we specify only semantics (and not the syntax) at the IRTF/IETF?