Socket API for Multihoming Shim

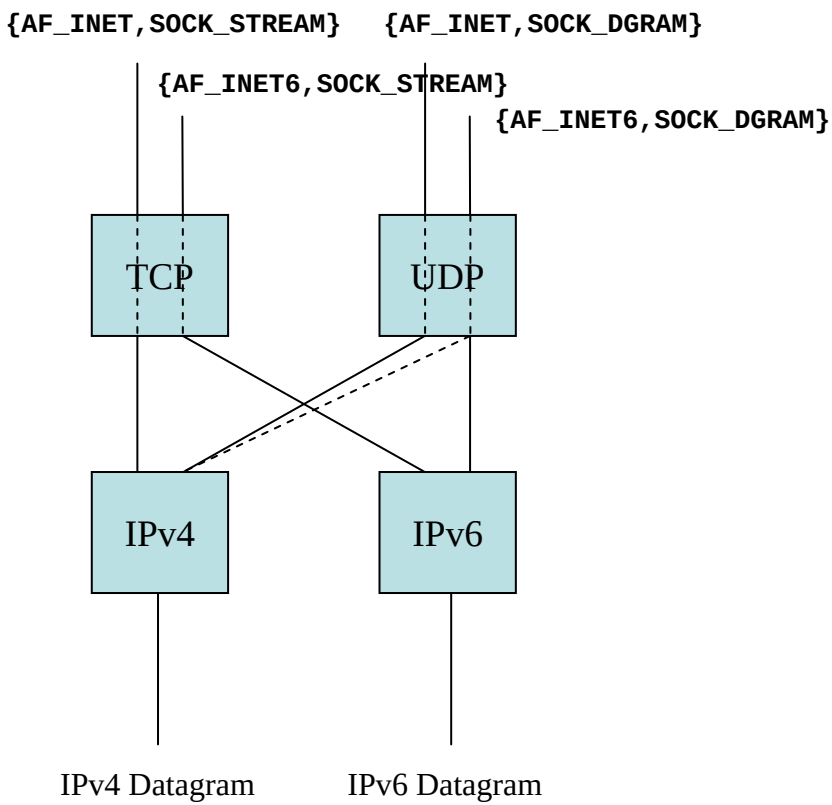
draft-sugimoto-multihoming-shim-api-00.txt

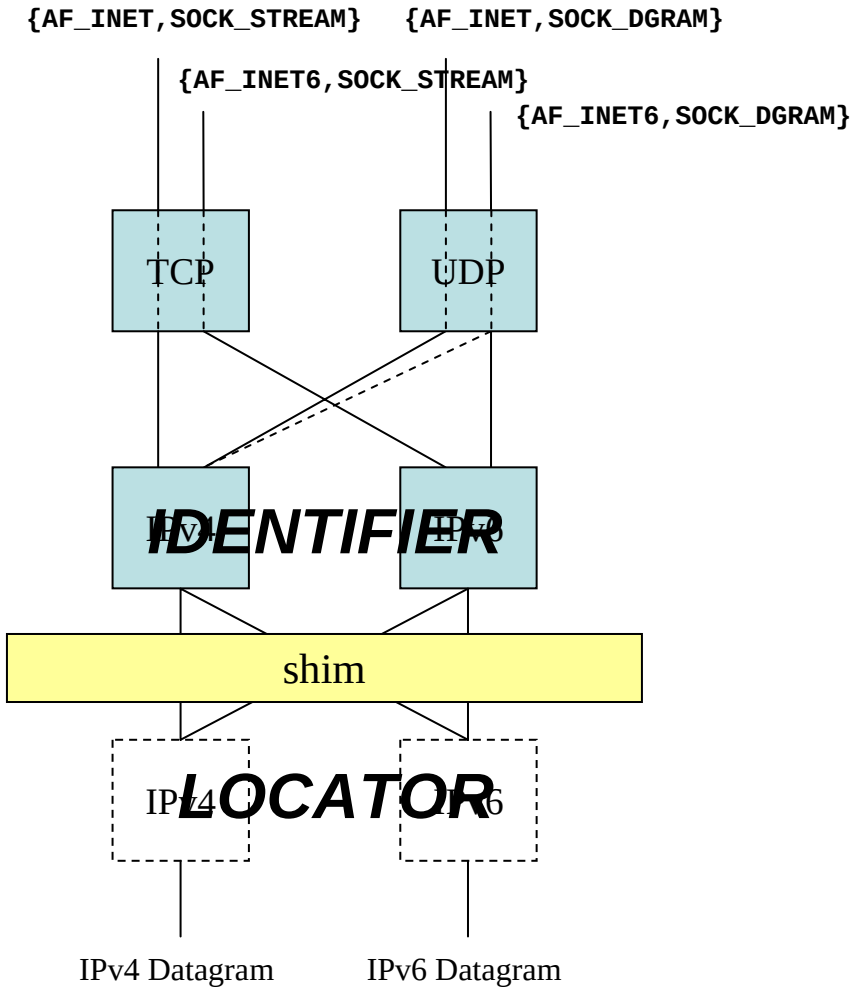
Shinta Sugimoto
Miika Komu
Marcelo Bagnulo
Kristian Slavov

Motivation/Background

- Target: SHIM6 and HIP
- Focus: common multihoming features:
  - Locator Management
  - Failure Detection and Path Exploration
- Motivation: Let application have more control on multihoming features of the shim sub-layer.
- Online discussions:
  - Mailing list: multimobsec-api@ietf.org
  - Issue tracker: http://hip4inter.net/cgi-bin/roundup.cgi/hip-shim6/index

System model





Implications to existing socket API

- Basic assumption is that the shim sub-layer should let existing socket API deal with *identifier* rather than *locator*.
  - IP_RECVDSTADDR, IPV6_RECVDSTADDR
  - IP_PKNTINFO, IPV6_PKTINFO
- Naming of the socket:
  - We assume that getsockname() and getpeername() return *identifier* assigned to local/remote node for the specified socket.

Shim specific extensions for socket API

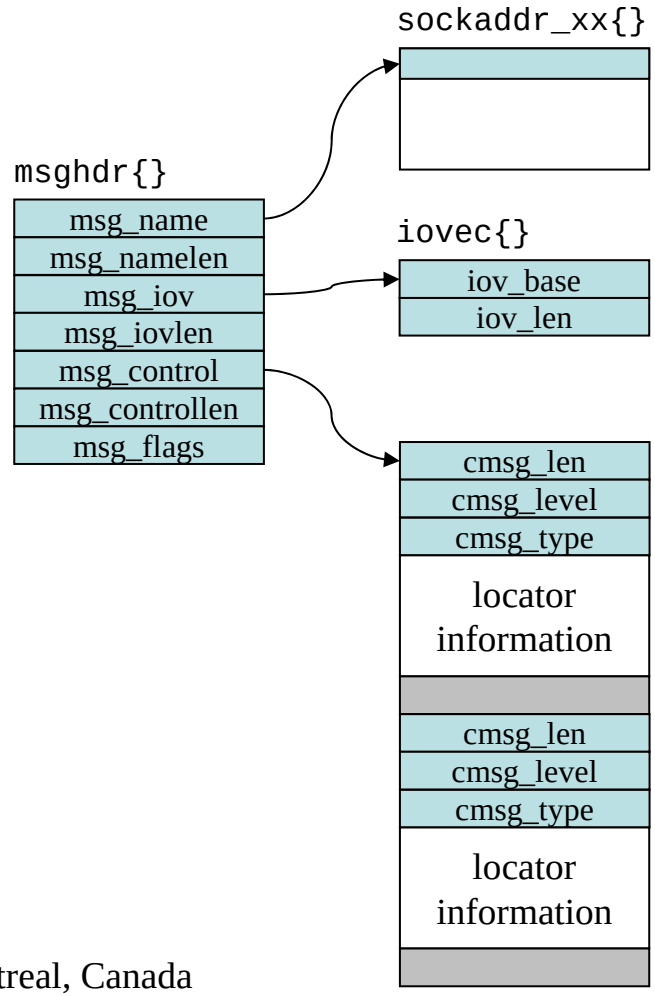
- Shim specific socket options:
  - Can be used by getsockopt() and setsockopt()
  - By definition, these socket options are ‘sticky.’
- Shim specific ancillary data:
  - Can be handled by sendmsg() and recvmsg()
  - The ancillary data may contain one or more locator information.

Shim specific socket options

Option name	Description
SHIM_ASSOCIATED	See if the socket is associated with any shim context
SHIM_DONTSHIM	Don’t apply any ID/Locator adaptation
SHIM_HOT_STANDBY	Request a hot-standby connection
SHIM_LOC_LOCAL_PREF	Preferred locator of the local node
SHIM_LOC_PEER_PREF	Preferred locator of the peer node
SHIM_LOCLIST_LOCAL	Locator list of local node
SHIM_LOCLIST_PEER	Locator list of peer node
SHIM_APP_TIMEOUT	Inform the shim layer of application timeout
SHIM_FEEDBACK_POSITIVE	Positive feedback from ULP to shim
SHIM_FEEDBACK_NEGATIVE	Negative feedback from ULP to shim
SHIM_DEFERED_CONTEXT_SETUP	Specify type of context setup
SHIM_PATHEXPLORE	Specify how aggressive should the path exploration be performed

Shim specific ancillary data

- Introduce shim specific ancillary data which can be used by sendmsg() and recvmsg() calls.
- By using the shim specific ancillary data:
  - Application can get locator information of received IP packet.
  - Application can specify locator information for outgoing IP packet.



Shim ancillary data

cmsg_type	sendmsg	recvmsg	Descriptions
SHIM_LOC_LOCAL_RECV		o	Get locator of local node from received IP packet
SHIM_LOC_PEER_RECV		o	Get locator of peer node from received IP packet
SHIM_LOC_LOCAL_SEND	o		Specify locator of local node for outgoing IP packet
SHIM_LOC_PEER_SEND	o		Specify locator of peer node for outgoing IP packet
SHIM_IF_RECV		o	Get physical interface from which the IP packet was received
SHIM_IF_SEND	o		Specify physical interface for outgoing IP packet

Issue #1: errno value

- Should we specify errno values for shim specific errors ?
  - A case where there is no associated context found (ENOENT or ENOSHIMCONTEXT ?)
  - A case where invalid locator was specified by application (EINVAL or EINVALIDLOCATOR ?)

Issue #2: Protocol conversion problem

- Identifier/Locator adaptation may lead to protocol conversion in a case where the address family of the identifier and locator are different.
- How should the shim layer behave in such case ?

Issue #3: Ambiguity of ‘applying’ shim to a given communication

- In case of SHIM6, default way of setting up a context is deferred setup.
- In case of HIP, context should be setup prior to the establishment of the communication.

Issue #4: Placeholder for locator information

- What kind of data structure should we use to store locator information ?
  - Locator could be either IPv4 or IPv6 address.
  - Required changes should be minimized.
- Case-1: Locator information to be handled by getsockopt() and setsockopt()
  - Could be one or more locator information.
  - Probably defining a specific data structure would be the right choice.
- Case-2: Locator information to be included in ancillary data for sendmsg() and recvmsg()
  - A control message (cmsg{}) may contain a single locator information.
  - sockaddr{} would do the job.

Issue #5: Level of shim specific socket options

- Which level should the shim specific socket options be defined ?
  - IPPROTO_IP, IPPROTO_IPV6 ?
  - SOL_SOCKET ?
  - SOL_SHIM (a new level for shim) ?
- Advantages of defining SOL_SHIM:
  - Shim sub-layer is inherently independent from any IP protocols.
  - We would like to avoid defining a set of shim socket options for both IPv4 and IPv6.
- Drawbacks of SOL_SHIM:
  - Requires introducing a new level to socket API framework.

Issue #6: Negative feedback from upper layer protocol

- What kind of information should be provided along with negative feedback from upper layer protocol ?

Summary

- Working on a set of extensions to socket API for common multihoming feature of SHIM6 and HIP.
- Focusing on locator management and failure detection & path exploration. Requirements have been sorted out and initial solutions have been proposed. More work needs to be done.
- APIs that are specific for SHIM6 should be defined separately.
- We would appreciate your comments/feedback !

Thank you.

Questions ?