

# Native HIP API

Miika Komu <[miika@iki.fi](mailto:miika@iki.fi)>

Helsinki Institute for Information Technology

Julien Laganier <[ju@sun.com](mailto:ju@sun.com)>

SUN Microsystems / LIP (INRIA/CNRS/UCBL/ENSL)

<http://hipl.hiit.fi/hipl/hip-native-api-snapshot-20040708.pdf>

# Goals 1/3

- support variable sized HIs in the sockets API
- apps can gain a better control of the HIP layer
  - application specified identifiers
  - HIP socket options
- the API requires modifications in the app code
  - the application becomes HIP aware
- compatibility with the sockets API
  - extend where reuse is not possible

## Goals 2/3

- hide HI/HIT/locator representation and management from apps
  - assumption: applications trust the system
  - ease the transition to IPv6
  - manual configuration is still possible
  - enables process migration (delegation)

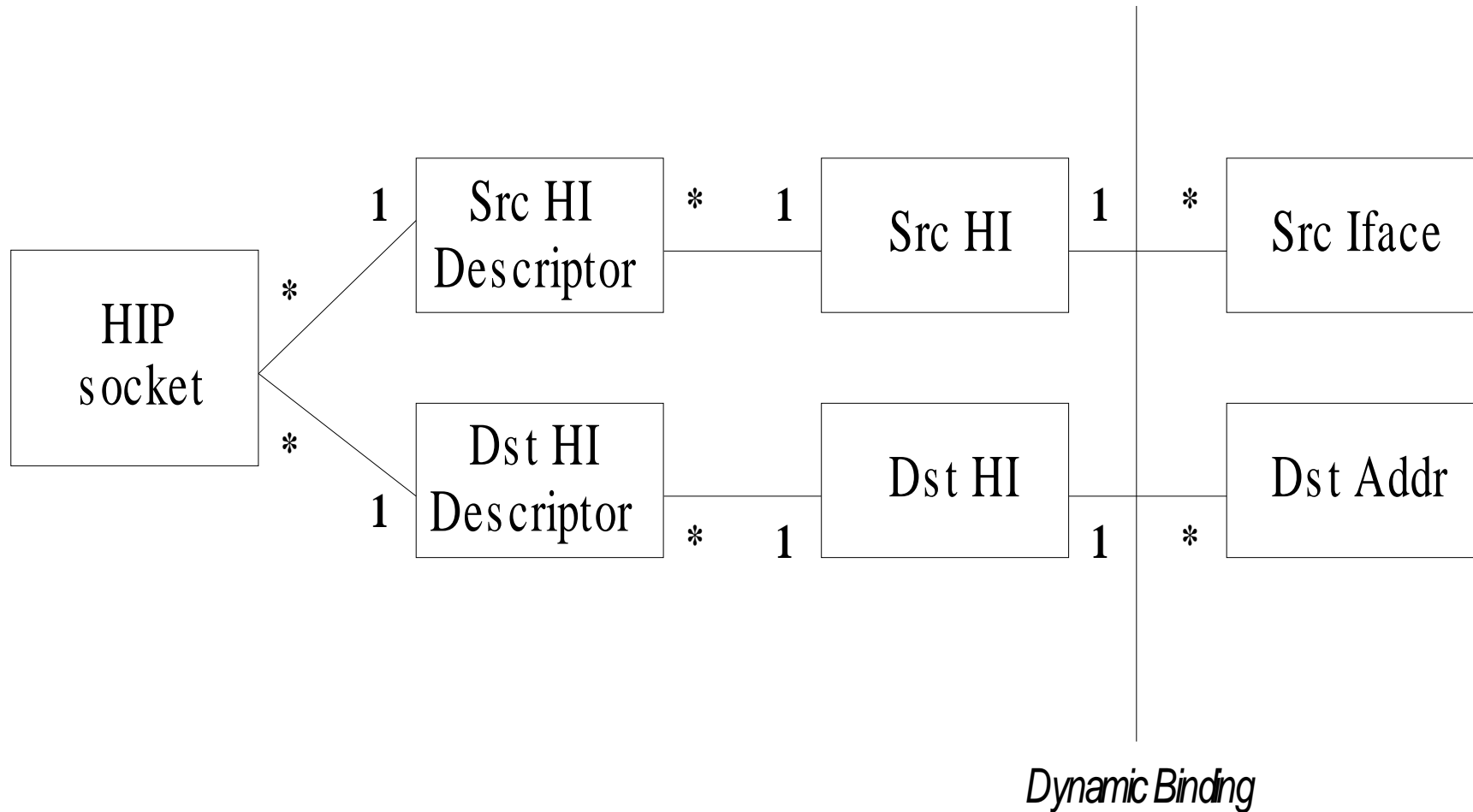
# Goals 3/3

- src locator is a network interface instead of an IP address
  - e.g. bind() to interface rather than IP address
- opportunistic HIP can be used when no identifiers for the peer are found
  - fallback to plain TCP/IP also possible

# Layering

Application Layer	Application			
Socket Layer	IPv4 API	IPv6 API	HIP API	
Transport Layer	TCP		UDP	
HIP Layer	HIP			
Network Layer	IPv4		IPv6	
Link Layer	Ethernet			

# Socket Bindings



# Endpoint identity descriptor

- forward compatible
  - the size and format of HIT can change
  - host identifier mobility = delegation of HIs
  - the “price” for this is the extra translation step
- collision free AID
  - HIT is not 100% collision free
- modular
  - each layer (app, transport, network) has their own identifiers

# PF\_HIP socket family

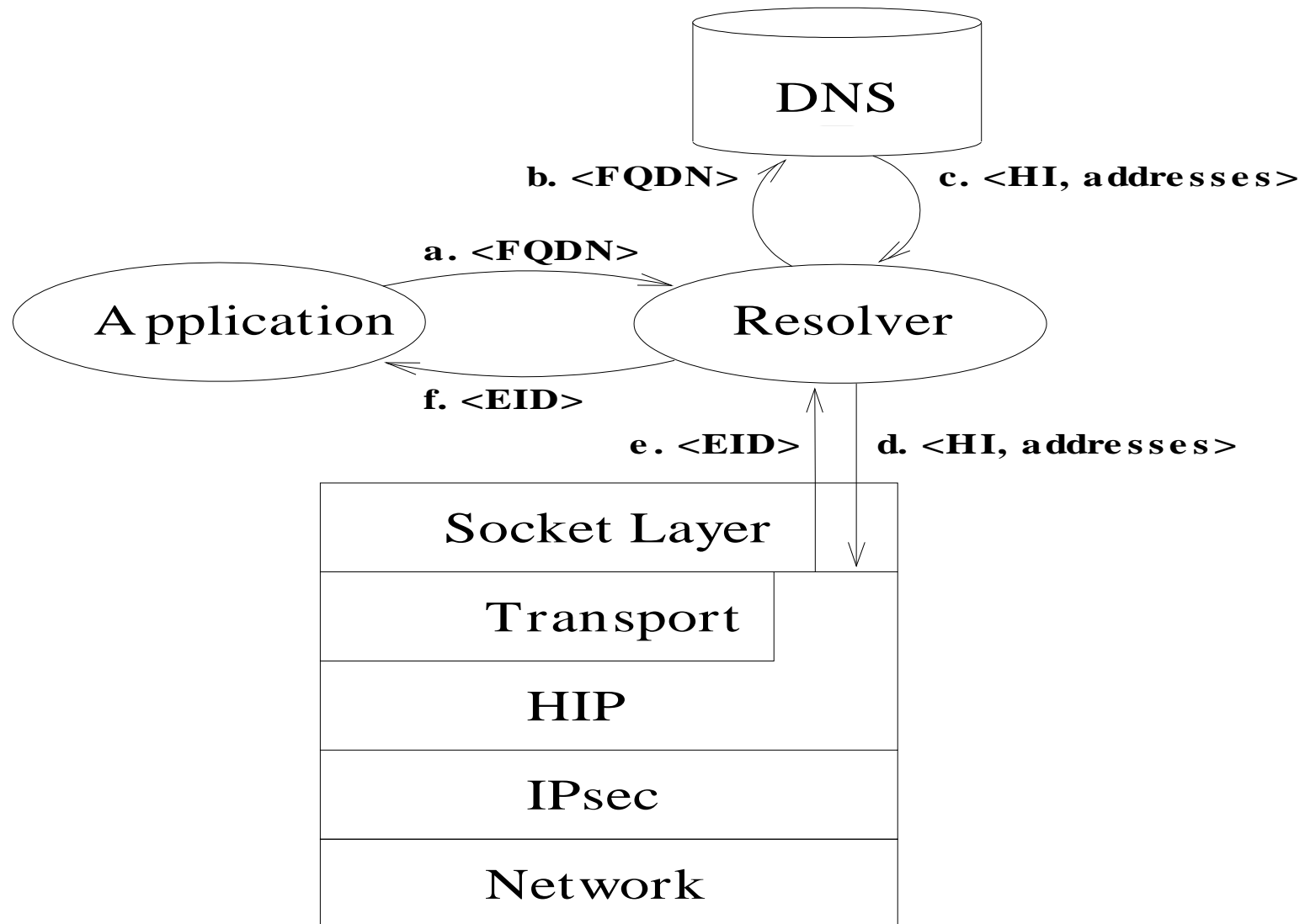
- results a cleaner implementation
  - no hooks required in the IP socket handlers
  - garbage collection in networking stack is less prone to cause problems
- HIP socket options
  - HIP SA attributes
  - QoS parameters



# Resolver 1/2

- outputs endpoint identity descriptors, which can be used directly in bind, connect, etc
- provides HI-to-IP mappings to the HIP module
- the HI-to-IP binding is secure
- detects the HIP capability of a host
  - fall back to IP addresses possible

# Resolver 2/2



# Benefits

- applications can utilize the HIP layer better
- clean interfaces
  - PF\_HIP socket family isolates HIP socket handler from the PF\_INET and PF\_INET6 socket handlers
- EID guarantees forward compatibility

# Drawbacks

- applications need to be changed
- referrals need to be queried via separate function call
  - obtain locators in addition to identifiers
- the endpoint descriptor adds another layer of indirection
  - but explicit identifiers can be returned from different function calls

# Evaluation

- resolver library implemented
- HIP socket handler was implemented on the Linux kernel (2.4 and 2.6)
- application specified HIs implemented
- telnet v6 was successfully ported to native API
- native HIP API for Java was implemented by Jaakko Kangasharju

# Conclusion

- Native API provides secure HI-to-IP bindings
- HIP socket options and application specified identities are important elements of the API
- Is the generality of descriptors better than explicit identifier handling?

# Questions?

- Contact Miika Komu <[miika@iki.fi](mailto:miika@iki.fi)>
- Source code available in  
<http://hipl.hiit.fi/hipl/>
- See the full documentation in  
<http://hipl.hiit.fi/hipl/hip-native-api-snapshot-20040708.pdf>