Protocol Design and Analysis of a HIP-based Per-Application Mobility Management Platform

The 7-th ACM International Symposium on Mobility Management and Wireless Access

October 26-27, 2009 Tenerife, Canary Islands, Spain

László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney

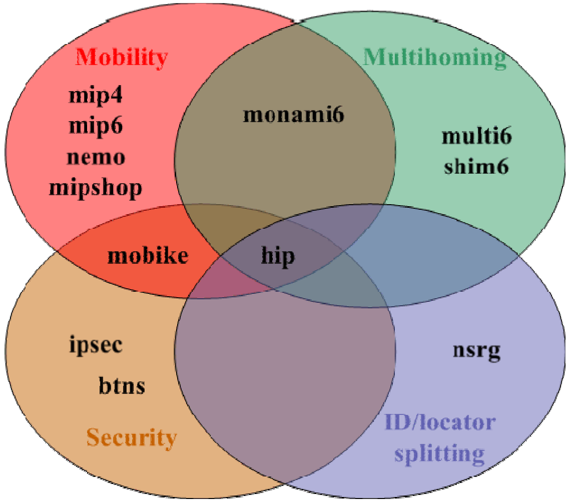
{goodzi, koci, nszabi, jeneyg}@mcl.hu

Outline

- Introduction
- Host Identity Protocol – In nutshell
  - HIP – Name space, name resolution
  - HIP – Header, Base Exchange
  - HIP – Basic Mobility and Multihoming support
- A new mobility type: per-application mobility
- The proposed HIP-based per-application mobility management platform
  - Main architecture
  - Modified HIP mechanisms
    - SA grouping and LOCATOR extension
    - Extended HIP UPDATE mechanism
    - Modified transport packet processing
- Evaluation
  - Simulation environment: HIPSim++ in INET/OMNeT++
  - Simulation topology and scenarios
  - Results
- Conclusion and future work

Introduction

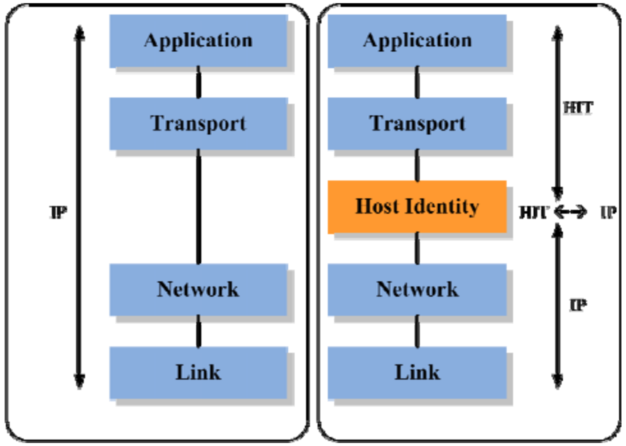
- Rapid evolution of wireless networking
  - wide-scale of different wireless access technologies
    - Bluetooth, ZigBee
    - 802.11a/b/g
    - 3G UMTS, LTE, WiMAX, etc.
  - complementary characteristics of them motivates operators to create heterogeneous access architectures
- Mobile devices
  - dynamic change of IP addresses
  - multiple interfaces, thus enabling concurrent sessions of simultaneously running applications
    - Multihoming, multiaccess
    - Managing several independent addresses
    - Managing several interfaces
- More complex scenarios
  - micromobility
  - terminal mobility, network mobility
  - session mobility, per-application mobility
- All of it only based on IP?
  - IP addresses are semantically overloaded
    - End-point Identifiers
    - Locators
  - Transport layer functions are bound to IP addresses
  - IP addresses give no information about the identity of an entity
- IETF and IRTF efforts
  - Strong need for an integrated solution!

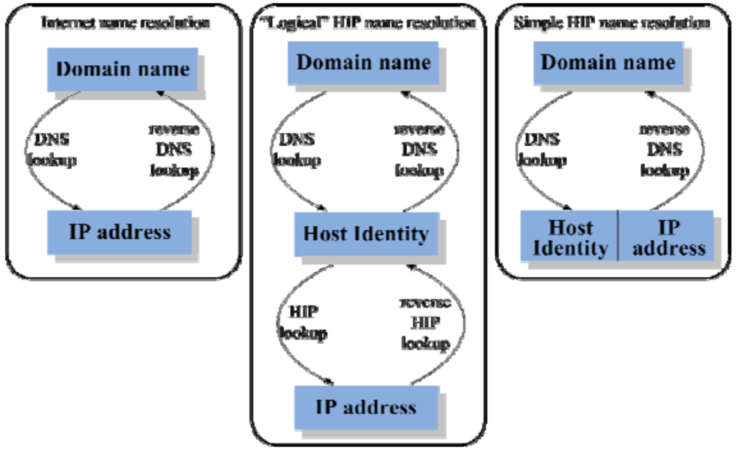


Host Identity Protocol – In nutshell

- New, cryptographical namespace
  - Assymetric key-pair
  - Secure authentication of end-points
  - Splitting identification and locator functions
- Advanced mobility and multihoming support
- Security centric, integrated, complex solution
  - Diffie-Hellman key exchange
  - End-to-end IPSec SA
- IPv4 / v6 interworking
- Modified TCP/IP architecture
  - Introducing a new layer between the network and the transport layer

HIP – Name space, name resolution

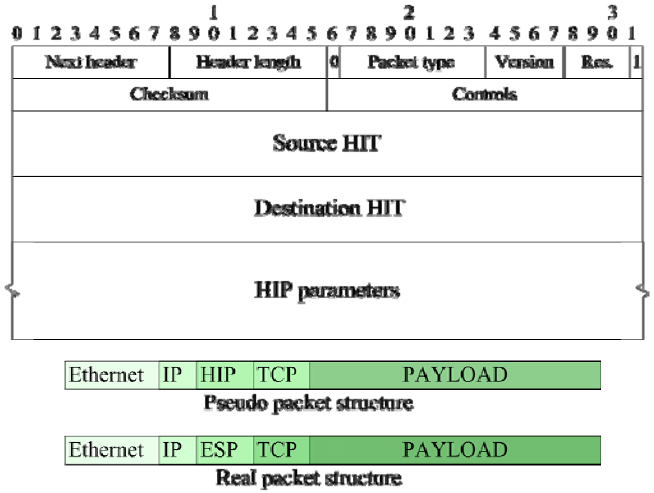


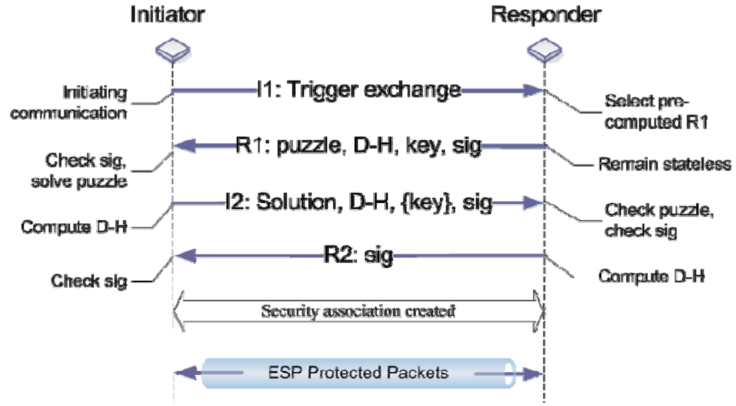


- New name space: „Host Identity” (abstraction)
- Host Identifiers (bit pattern):
  - HIT – 128 bit hashed encoding
  - LSI – 32 bit hashed encoding
- Sockets are bound to HIs
- HIs are translated to IP addresses at kernel level

- Internet name resolution
  - Domain name <–> IP address
- Simple HIP name resolution
  - No bidirectional HI <–> IP translation
  - If only HI is known, reverse DNS query is not possible
- „Logical” HIP name resolution

HIP – Header, Base Exchange



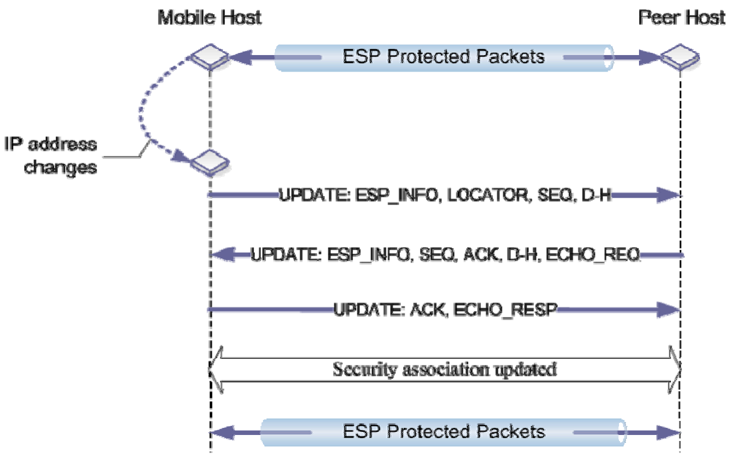


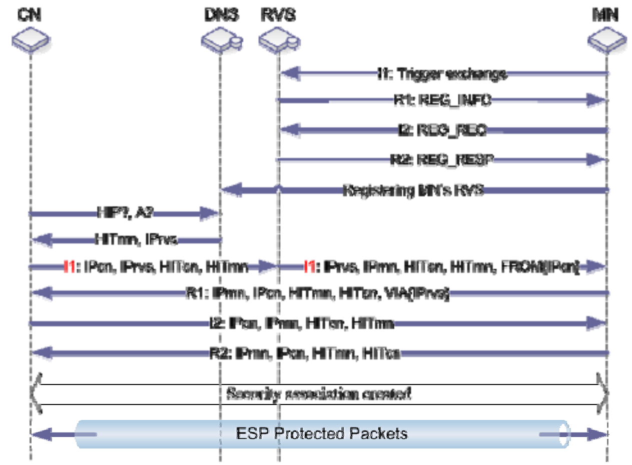
- IPSec SAs are bound to HITs
- HITs of ESP protected packets can be found by SPIs (SAs are identified by SPI)
  - HIP header doesn't appear in user data packets: small overhead

- Base Exchange
  - I1 (trigger, initialization of HIP BE)
  - R1 (sending puzzle)
  - I2 (solving puzzle)
  - R2 (finishing Diffie-Hellman)
- Carrier: IPSec ESP

MobiWac 2009 -- László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney

HIP – Mobility, Multihoming





- Update mechanism with IP address validation
- LOCATOR parameter
- Multihoming: the same update procedure will be performed for every address
- SAs are reconfigurable!

- DNS was designed for quick resolution and not for quick spreading of updates
- HIP Rendezvous Server (RVS) solves this problem
- RVS relays the packet to the actual IP address of the mobile node

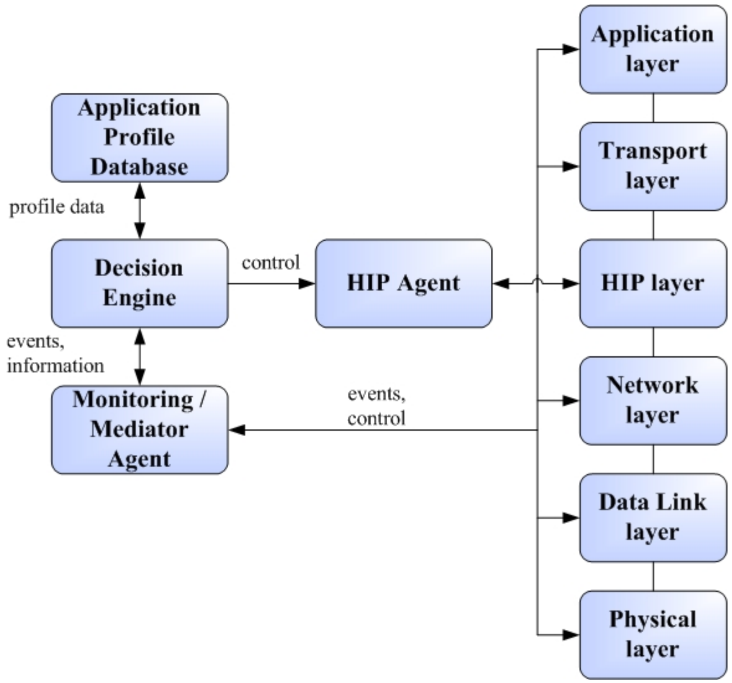
Per-application mobility

- per-application mobility is quite a new mobility concept which refers to scenarios when a host is multihomed with more than one wireless interfaces and wants to control mobility decisions in an application-wise manner
- application mobility (process migration) vs. per-application mobility
  - application mobility moves an already running application from a source host to a destination host to provide load share management, fault tolerance and data access locality
  - per-application mobility focuses on multihomed hosts with multiple wireless interfaces: switches traffic of each application running on the host independently between available interfaces (i.e. networks)
    - Original idea: Moonjeong Chang et al.[1]: IP-level solution inheriting all the difficulties derived from the semantically overloaded IP addresses
- Per-application mobility requires advanced multihoming and multiaccess and that's why HIP comes into picture!

[1] *Moonjeong Chang, Meejeong Lee, Hyunjeong Lee: "Per-Application Mobility Management with Cross-Layer Based Performance Enhancement," IEEE WCNC 2008., March 31 2008-April 3 2008 pp: 2822–2827*

HIP-based per-application mobility management platform

- The main goal of our HIP-based per-application mobility management platform is to adapt the paradigm of per-application mobility to HIP at protocol-level
- *Application Profile Database*: stores and maintains the requirements and user preferences of different applications
- *Monitoring/Mediator Agent*: gathers measurement data and other relevant information from every layer (interface types, link qualities, IPv6 addresses, HITs, application port numbers, mobility events, etc.)
- *Decision engine*: decides what kind of HIP responses (i.e. mobility and multihoming mechanisms) are to be initiated and instructs the HIP Agent
- *HIP Agent*: translates the layer-independent instructions to HIP functions (e.g. HO is to be performed -> HIP Update sequence)



Operation of HIP Agent: HIP mechanisms to be modified

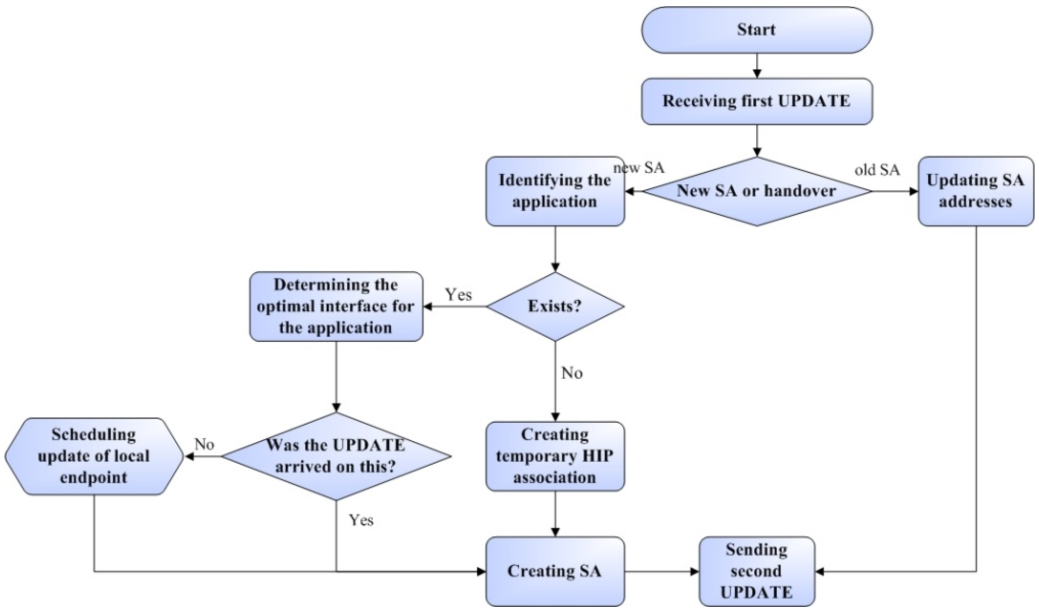
- Multihomed Security Associations are the protocol entities which can make a HIP system to be able to handle per-application mobility in an effective way
- In consequence, a certain SA grouping scheme and a modified UPDATE mechanism are the keys to the HIP-based per-application mobility management framework
- Our approach is end-to-end based and uses spanned SAs between communicating peers: standards of HIP DNS and RVS extensions can remain intact and only SA handling, UPDATE procedure and packet processing mechanisms are needed to be modified

Operation of HIP Agent: SA grouping and LOCATOR extension

- We propose to create and manage exclusive SAs for every running application between two HIP hosts and distinguish them with a quintet of *Source HIT*, *Destination HIT*, *Source port*, *Destination port* and *Transport type* (SA grouping quintet)
- Such a quintet could be assigned even to smaller entities (e.g. sockets) but in that case scalability easily becomes a serious issue because of the signaling overhead of SA management and related HIP messages
- Since we can not presume on that applications transact bi-directional sessions, there is a need to unambiguously represent which SA belongs to which application
  - we propose to use the above defined quintet in LOCATOR parameter for marking applications and SAs
  - HIP signaling messages already contain Source and Destination HITs, thus existing HIP LOCATOR parameter must be extended only with fields of Source and Destination port and Transport type
  - these extensions can be easily derived from the existing *type 1* LOCATORs: a new 16 bit Source port, a 16 bit Destination port and a 4 bit Transport type descriptor field are to be introduced

Operation of HIP Agent: Extended HIP UPDATE mechanism

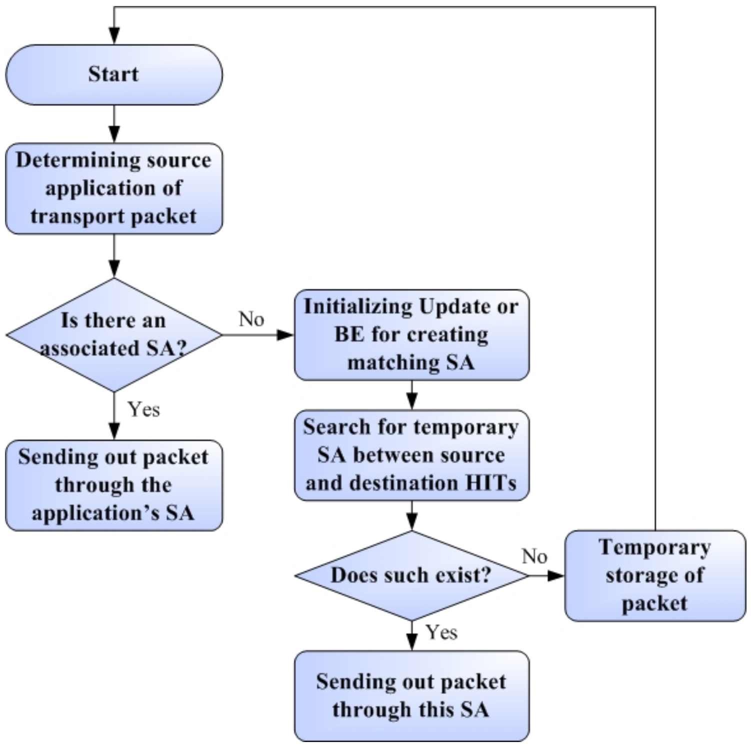
- HIP UPDATE procedure is initiated in two distinct cases
  - a new application is to be introduced in an existing HIP association (requires setup of a new SA)
  - an already running application is to be handed over (i.e. a new interface was chosen for the application and handover is to be performed)
- The UPDATE sequence itself and its main parameters are not changed:
  - only the new LOCATOR type appears
  - and the processing mechanism of incoming UPDATE messages changes



MobiWac 2009 -- László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney

Operation of HIP Agent: Modified transport packet processing

- In our framework incoming data packets are treated completely according to the current HIP standards: SPI values are used to identify the appropriate SA
- However, in case of packets arriving from the transport layer (originated by the applications) the standard way of HIP operation changes:
  - the fitting SA has to be selected for the outbound communication
- Every transport packet has to be inspected and the originating application and its corresponding SA has to be determined
- If no associated SA exists, then HIP BE or UPDATE procedures are initiated
  - packets will be temporarily stored or sent out through an existing – but not yet optimized – interface, as long as the BE/UPDATE finishes and the optimized path (i.e. SA) gets ready



HIP for INET/OMNeT++: HIPSim++

- Basis of our HIP-based per-application mobility simulations:
  - OMNeT++ (discrete event simulation environment)
  - INETwithMIPv6 (powerful and widespread simulation model set for OMNeT++)
  - HIPSim++[1] our extension for the 20081128 version of INETwithMIPv6
- The simulated HIP layer registers HIT-IP bonds for every communication session, and when packets from the transport layer arrive, destination and source HITs are replaced by destination and source IP addresses. Higher layers know only about HITs and Port numbers.
- HIPSim++ accurately simulates core HIP instruments focusing on the advanced mobility and multihoming capabilities and wireless behavior of the protocol
- Full implementation of IPSec and relating algorithms is not part of our simulation model, mapping of all the security algorithms is out of scope of our current efforts:
  - HIPSim++ possess only skeleton implementation of Diffie–Hellman mechanisms, RSA engine, cryptographic hash functions, puzzles and other parts of the mathematical apparatus

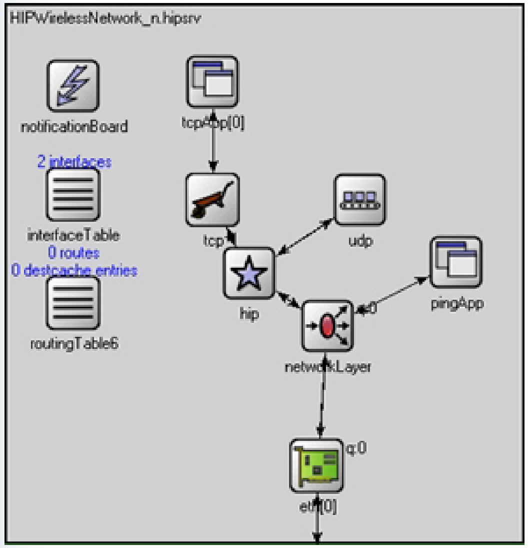
[1] HIPSim++: A Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++, Official homepage: http://www.ict-optimix.eu/index.php/HIPSim

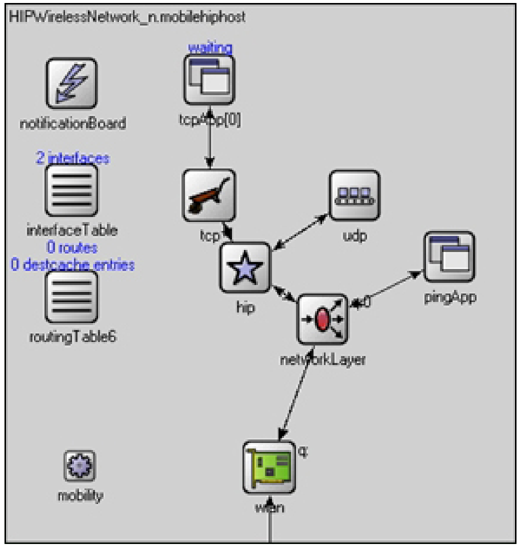
Main Modules of HIPSim++

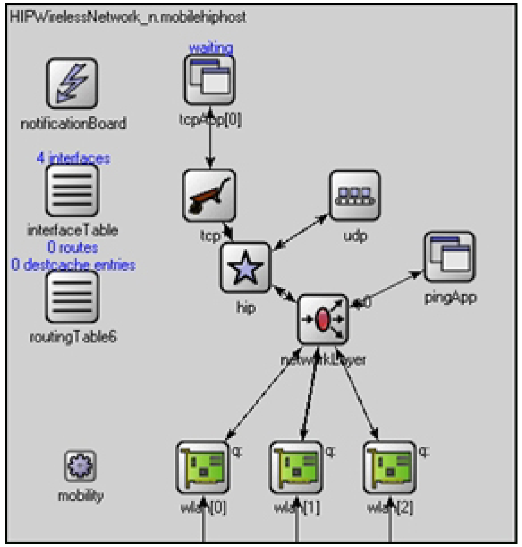
- HIP and HIPSM modules
  - The core module which creates a daemon instance of the HIP State Machine (HIPSM) for every new HIP session
  - handles HIP Base Exchange, RVS registration and HIP mobility functions, manages changes occurring in the states and addresses of host interfaces
  - one such daemon instance cares of one SA
  - HITs have to be provided by the applications (or rather the transport layer), therefore HIP-capable DNS extensions are also integrated into HIPSim++
- RvsHIP module
  - RvsHIP is based on the HIP module and extends it with the RVS functions
  - handles incoming registration messages according to the HIP standards and by forwarding I1 messages to the appropriate HIP responder chosen from the registered ones.
- DnsBase module
  - DnsBase module is an UDP application realizing DNS functionality for name resolution of HIP hosts
  - implements the new Resource Record (DNS HIP RR) defined in RFC 5205
  - resolves domain names to HITs and IP addresses and in case of mobile HIP hosts also provides RVS information
- PerappDecisionEngine module
  - implements the main functions of our per-application mobility scheme
  - marks application sockets with SA grouping quintets
  - calculates the optimal interface for a given application profile based on the actual conditions
  - implements the Monitoring/Mediator Agent (based on the INET's *NotificationBoard* messages)
    - socket information from the transport layer
    - IPv6 prefix information from the networking layer
    - link information from the link layer
  - Implements APDB functions (based on an `.xml` database)

Special Nodes in HIPSim++ I.

- HIP Initiator and HIP Responder (HIP hosts)
  - Hosts implementing HIP Initiator and/or Responder functions (i.e. HIP hosts) are derived from the INET’s existing *StandardHost6* compound module by inserting the HIP layer between the transport and the network layers
  - Wired HIP Initiator/Responder (*HipHosts6*)
  - Wireless HIP Initiator/Responder (*WirelessHipHosts6*)
  - Wireless HIP Initiator/Responder with multiple interfaces (*WirelessMultihomeHipHosts6*)



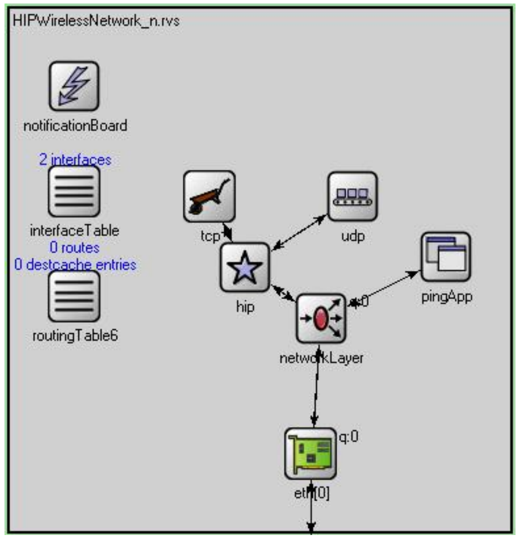




Special Nodes in HIPSim++ II.

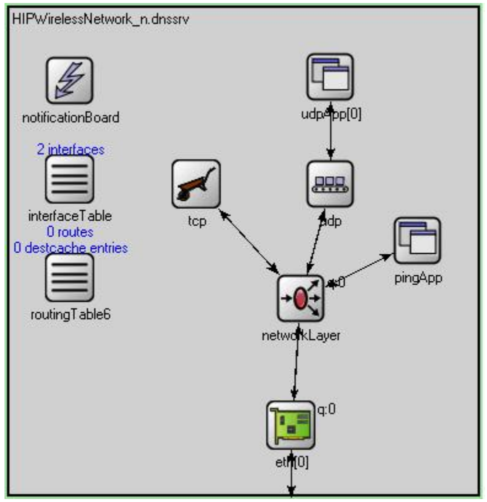
HIP Rendezvous Server

- RVS nodes implementing HIP rendezvous functions (*RvsHost6*) in our simulation framework are also derived from the *StandardHost6* compound module by interposing a modified HIP layer module prepared to handle RVS tasks
- *RvsHost6* node forwards I1 messages originated by (wired or wireless) HIP Initiators to the appropriate (wired or wireless) HIP Responder signed in the RVS
- potential Responders must register themselves in the RVS and in place of their own IP address, Responders must use their RVS’s IP address in the Domain Name System
- HIP nodes must continuously inform their RVSs about events of locator changes



Special Nodes in HIPSim++ III.

- DNS Server
  - DNS Server node is responsible to provide name resolution for HIP hosts by implementing the basic functions described in RFC 5205
  - DNS Server node is based on the *StandardHost6* compound INET module too, but extends it with the DNS server application
  - The DNS database is an `.xml` file containing resource records of every node in the simulation topology
  - DNS queries are handled by the Host Identity Layer:
    - the first transport packet initiates the query process based on the destination HIT (and the pre-set DNS IP address)
    - Basic Exchange starts right after the response provides with the locator belonging to the destination HIT



HIPSim++ Messages

- HIP signaling messages
  - In accordance of RFC 5201, different HIP messages start with a fixed header
  - The HIP header is logically an IPv6 extension header such in HIPSim++ all HIP messages are implemented as additions to the INET's *Ipv6ExtensionHeader*
  - All the already standardized HIP message types and parameters are defined, including also the LOCATOR parameter which is realized as an array of *HIPLocator* structure
- HIP data messages
  - In HIPSim++ we currently use the Encapsulated Security Payload (ESP) based mechanism for transmission of user data packets [RFC 5202]
  - Every HIP data message travels in ESP: packets coming from the transport layer will be encapsulated in an *ESPHeaderMessage* labeled with the appropriate SPI value
- DNS messages
  - The basic HIP namespace resolution functions are implemented using a simple query/response message pair called *DnsQuery* and *DnsResponse*
- Per-application mobility control messages
  - inboard per-application mobility signaling
    - PERAPP_NEWAPP indicates the appearance of a new application
    - PERAPP_CHANGEDAPP signals changes in socket descriptors of running applications
    - PERAPP_DELETEAPP indicates exiting of a running application
    - PERAPP_UPDATE is sent if source address or interface of a running application changes
    - PERAPP_LAST_IF_DOWN message stands for the event when the last usable interface was disappeared

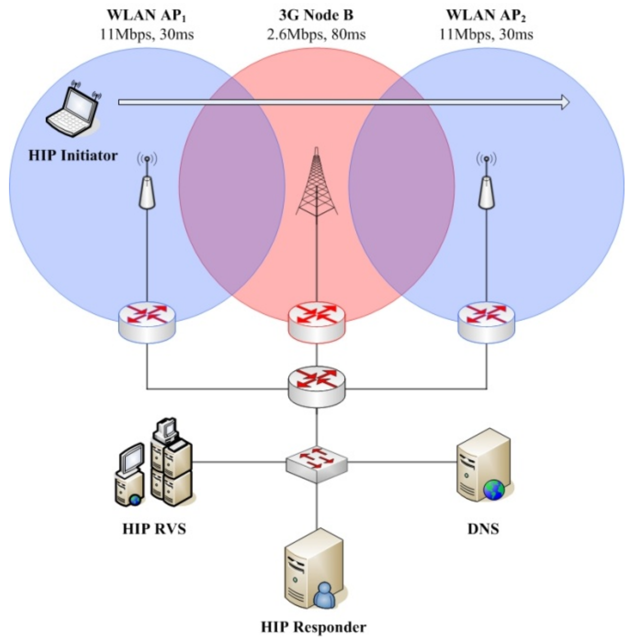
MobiWac 2009 -- László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney

Simulation topology and scenarios

- Goal: evaluation of the proposed HIP-based per-application mobility management platform against the standard HIP per-host multihoming scheme
  - HIP-based per-application mobility without active measurements of available networks and actual application performances (“Perapp”)

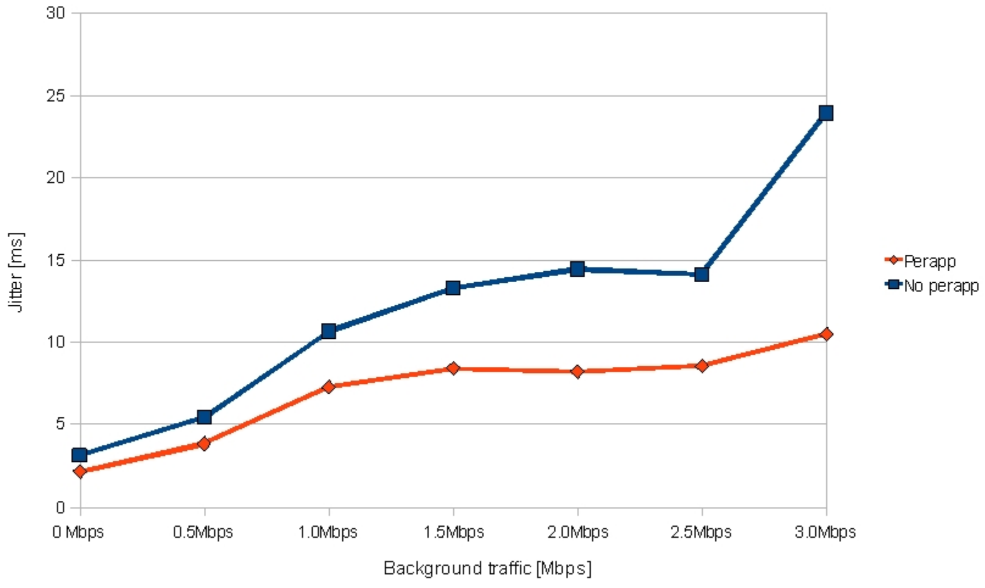
vs.

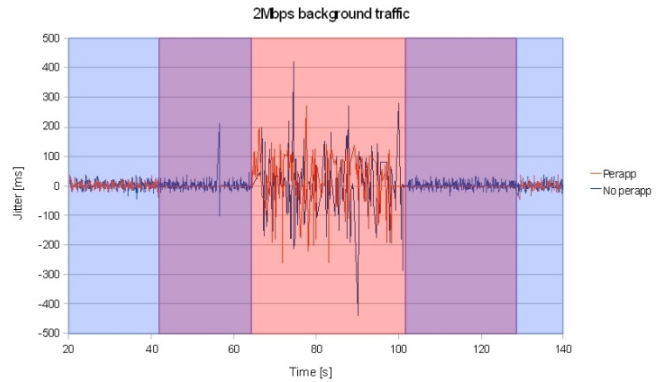
  - only one SA is built-up between endpoints and only one interface is chosen for every application based on static priority (“No perapp”)
- Inputs for per-application HO decision: pre-defined QoS parameters of applications / access networks (i.e. only static profiles) and dynamic connection discovery
- Two applications with different QoS characteristics (i.e. QoS profiles) and two different decision policies:
  - FTP (downlink Reno)
    - aims to maximize the TCP throughput.
    - the policy sets the FTP application to always use the connection with the highest bandwidth
  - VoIP (15 Kbyte/sec CBR, packet size=80 byte)
    - aims to minimize the handover latency, packet loss and end-to-end packet delay variation
    - the policy sets the VoIP application to use the network with the lowest number of possible handovers and simultaneously communicating applications



Jitter of VoIP vs. Background traffic

- 120 sec long measurement interval
- WLAN, WLAN+3G, 3G, WLAN+3G, WLAN sequence during movement
- points stand for the average Jitter value for completed packet sequences of 25 complete simulation runs
- if per-application mobility management is employed, Jitter values are smaller thanks to the optimally chosen and dedicated UMTS access in the overlapping network areas
- the advantage expands with increasing background traffic

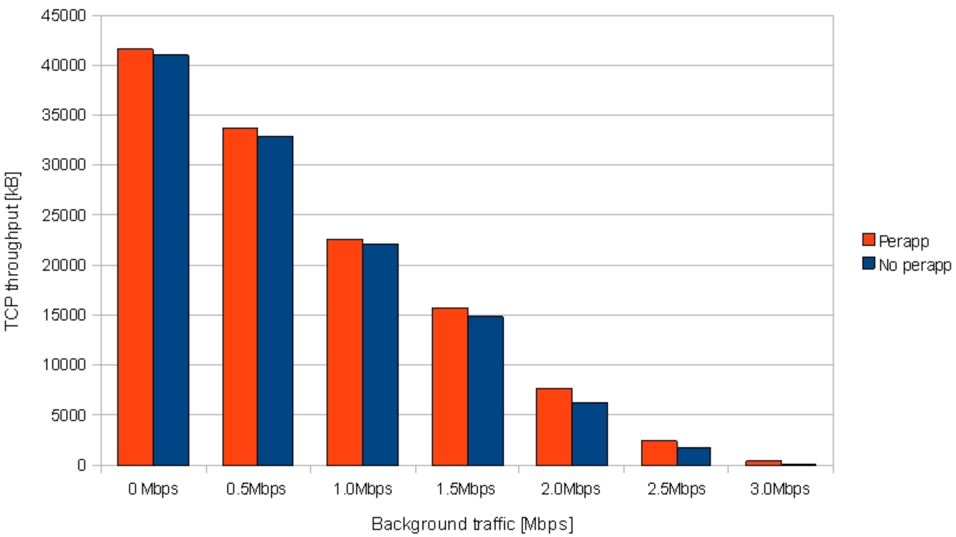




MobiWac 2009 -- László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney

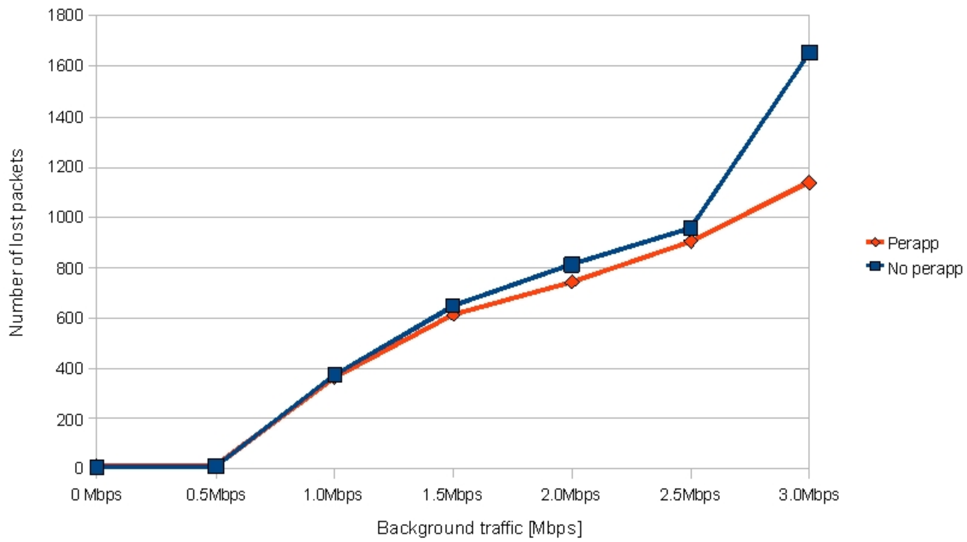
Cumulated TCP throughput of FTP vs. Background traffic

- 120 sec long measurement interval
- WLAN, WLAN+3G, 3G, WLAN+3G, WLAN sequence during movement
- columns stand for the average cumulated FTP throughput value of 25 complete simulation runs
- the throughput decreases as the background traffic increases (congestion and the lost available bandwidth)
- measured throughput is always higher when per-application mobility management is used



Cumulated UDP packet loss of VoIP vs. Background traffic

- 120 sec long measurement interval
- WLAN, WLAN+3G, 3G, WLAN+3G, WLAN sequence during movement
- points stand for the average cumulated UDP packet loss value of 25 complete simulation runs
- in cases of low background traffic no significant advantage of per-application mobility can be pointed out
- when high background traffic is injected in the network, the method shows its power



Conclusion and future work

- We designed a HIP-based per-application mobility management platform aiming to be a comprehensive handover decision and execution architecture founded on the promising Host Identity Protocol and the cross-layer building blocks closely incorporating with it
- We modeled HIP and the proposed per-application handover mechanisms in the INET/OMNeT++ simulation environment (HIPSim++)
- We assessed our HIP-based per-application mobility management platform against standard per-host HIP multihoming performance in means of:
  - Jitter
  - TCP throughput
  - UDP packet loss
- Results show that our HIP-based application-wise mobility system overcomes standard HIP mobility solutions in most cases
- As a part of our future activities we will further extend our scheme and simulation model with
  - active network measurements
  - active measurements of actual application performances
  - complex profiles and policies
  - advanced cross-layer extensions (e.g. active measurements based interface selection and on-the-fly intervention into protocol layers' operation)
  - extensive comparisons against pure IP-based solutions

Thank you!

Questions?

For more information please visit our website: **http://www.ict-optimix.eu**

László Bokor, László Tamás Zeke, Szabolcs Nováczki, Gábor Jeney
{goodzi, koci, nszabi, jeneyg}@mcl.hu