

Design and Evaluation of Host Identity Protocol (HIP) Simulation Framework for INET/OMNeT++



The 12-th ACM International Conference on Modeling, Analysis
and Simulation of Wireless and Mobile Systems
October 26-30, 2009 Tenerife, Canary Islands, Spain

László Bokor, Szabolcs Nováczki, László Tamás Zeke, Gábor Jeney
{[goodzi](mailto:goodzi@mcl.hu), [nszabi](mailto:nszabi@mcl.hu), [koci](mailto:koci@mcl.hu), [jeneyg](mailto:jeneyg@mcl.hu)}@mcl.hu

Outline

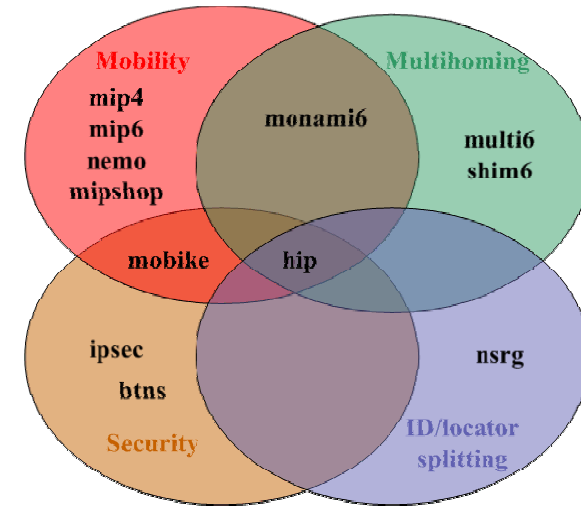


- Introduction
- Host Identity Protocol – In nutshell
 - HIP – Name space, name resolution
 - HIP – Header, Base Exchange
 - HIP – Basic Mobility and Multihoming support
- Fundamentals of INET/OMNeT++
- The HIPSIM++ Framework
 - Main modules
 - Special nodes
 - Messages
 - Modifications to INET
- Evaluation
 - HIP Testbed
 - Simulation topology and scenarios
 - Results
- Conclusion and future work

Introduction



- Mobile devices
 - Dynamic change of IP addresses
- Multihoming, multiaccess
 - Managing several independent addresses
 - Managing several interfaces
- Supporting complex scenarios
 - Micromobility
 - Network mobility (NEMO)
 - „Ubiquitous” and ad-hoc networks
- Security
 - Man-in-the-middle
 - DoS, DDoS
- 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!

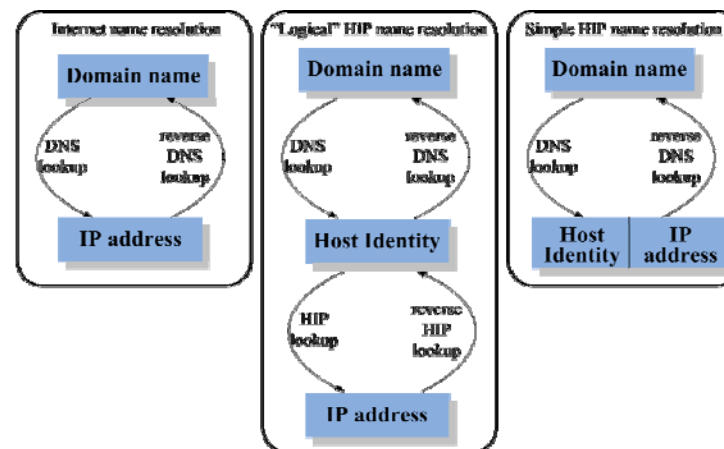
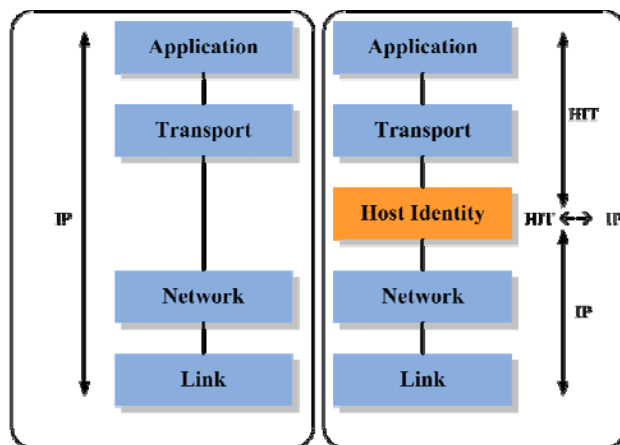


Host Identity Protocol – In nutshell



- New, cryptographical namespace
 - Assymmetric 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

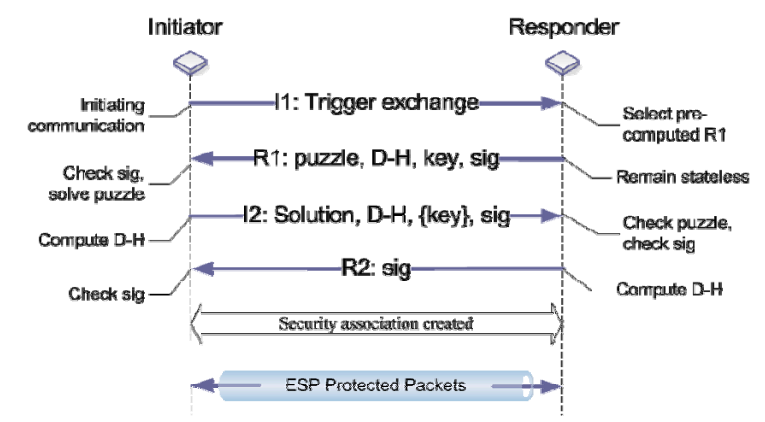
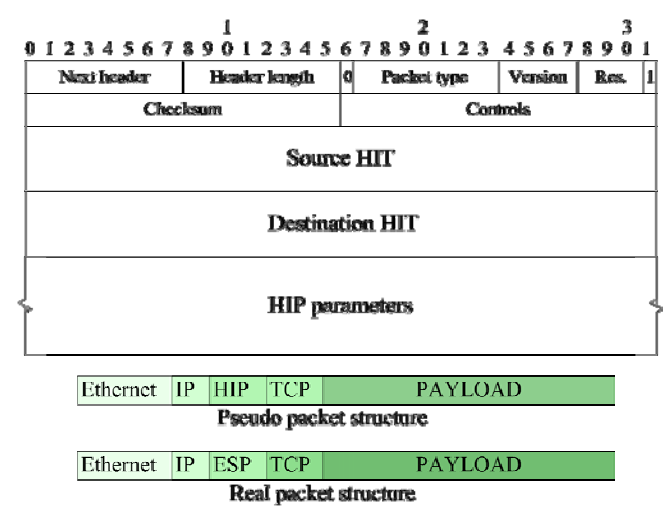
HIP – Name space, name resolution



- 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 \leftrightarrow IP address
 - Simple HIP name resolution
 - No bidirectional HI \leftrightarrow 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

Fundamentals of OMNeT++



- OMNeT++ is a discrete event simulation environment for modeling
 - communication networks
 - queueing networks
 - hardware architectures, etc.
- Component-based, modular structure
 - a model consists of modules communicating with message passing
 - active modules are named as *simple modules* (atomic elements written in C++, using the OMNeT++ simulation class library), while the modules composed from simple modules are the *compound modules*
 - modules are communicating with messages (`.msg` definitions)
- Topology description of simulation models is defined by the NED (NEtwork Description) language
 - defines modules and their interconnection
 - `.ned` description file consists of
 - simple module declarations (i.e. description of the module's interfaces),
 - compound module definitions (i.e. declaration of the module's external interfaces and definition of submodules and their interconnection)
 - network definitions (i.e. compound modules that are self-containing simulation models)
- Initial parameters of simulation runs are specified in `.ini` files, independently both from the C++ and the NED codes
- Wide scale of existing simulation models, helpful community
- OMNeT++ aspires to be the optimal solution between open-source, research-oriented simulators and high-priced commercial softwares

OMNeT++



Fundamentals of INET for OMNeT++



- In the area of communication networks currently the most powerful and widespread simulation model set for OMNeT++ is the INET Framework
- INET Framework contains detailed and accurate models for:
 - IPv4/IPv6, TCP/UDP/SCTP, MPLS, RSVP, LDP, several applications (like telnet, video streaming)
 - link-layer models (e.g. PPP, Ethernet and 802.11b/g)
 - routing protocols (OSPF, RIP), etc.
- INET uses the same concept as OMNeT++:
 - models consist of modules communicating by message passing (.msg)
 - protocols are usually represented by simple modules in which external interfaces (gates/connectors and parameters) are described in a .ned file, and the implementation is prepared as a C++ class with the same name
- INET modules:
 - protocol implementations (e.g. IPv4, IPv6, UDP, TCP, SCTP)
 - autoconfigurators for whole network topologies (e.g. *FlatNetworkConfigurator6*)
 - data storage and manipulation modules (e.g. *InterfaceTable*, *RoutingTable6*)
 - managers for inter-module communication (e.g. *NotificationBoard*)
 - radio channel managers (e.g. *ChannelControl*)
 - implementations of host mobility (e.g. *RandomWPMobility*), etc.
- Wide scale of pre-assembled hosts can be found in the /Nodes directory of INET (e.g. *StandardHost6*, *Router6*, *WirelessAP*, etc.)
- It is also possible to assemble new network entities from the existing modules, and to implement your own modules and to use them as building blocks of new simulations

OMNeT++



HIP for INET: HIPSim++



- Wide scale of INET extensions:
 - OverSim is designed to model overlay networks and P2P protocols
 - AODV-UU, DSR-UU and OLSR protocol models
 - xMIPv6 (Extensible Mobile IPv6) is a model of IPv6 mobility protocols, etc.
- Our contribution: HIPSim++ aims to be an extensible and precise simulation model for the Host Identity Protocol working on the top of the 20081128 version of INET/OMNeT++
- HIP framework uses the IPv6 networking stack of INET such fulfilling the requirements of global HIP communication based on the 128bit HITs
- The introduced 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.
- The main design goal of HIPSim++ was to accurately simulate core HIP instruments focusing on the advanced mobility and multihoming capabilities and wireless behavior of the protocol.
- A 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

Main Modules of HIPSim++



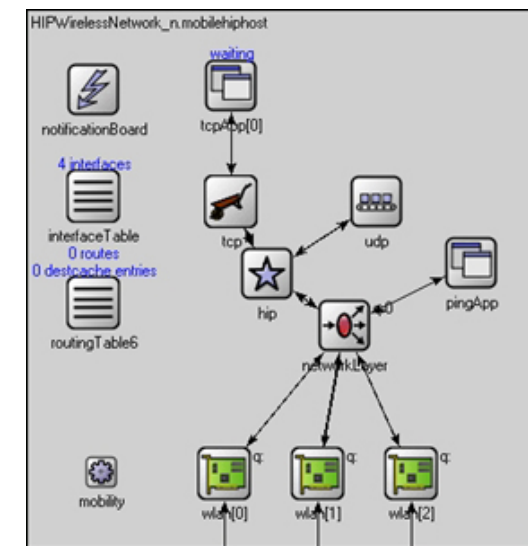
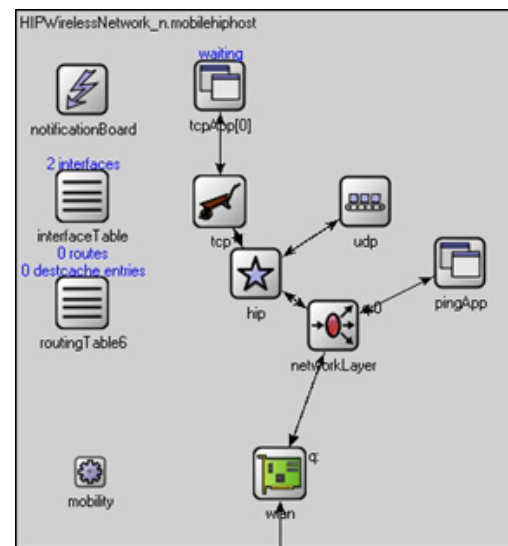
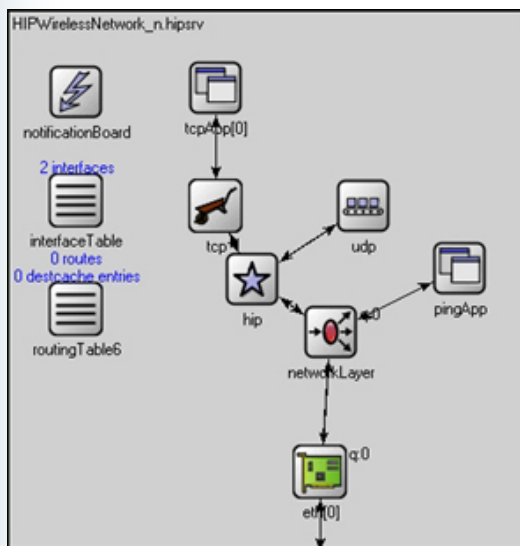
- **HIP and HIPSM modules**
 - HIP is the core module of our HIPSim++
 - creates a daemon instance of the HIP State Machine (HIPSM) for every new HIP session
 - handles HIP Base Exchange, RVS registration and HIP mobility functions, manages changes occurring in the states and addresses of host interfaces
 - one such daemon instance cares of one SA, which will be identified by the local SPI
 - HIPSM daemons are registered by destination and source HITs (and SPIs) in the HIP module
 - 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

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*)

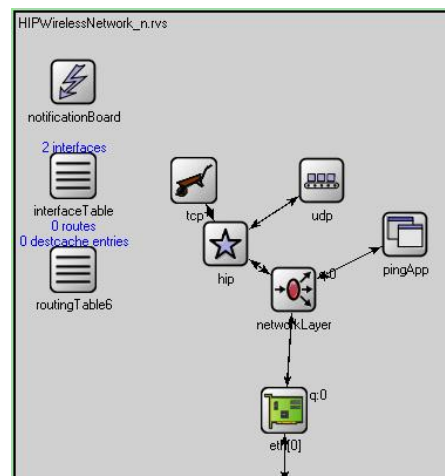


Special Nodes in HIPSim++ II.



■ 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

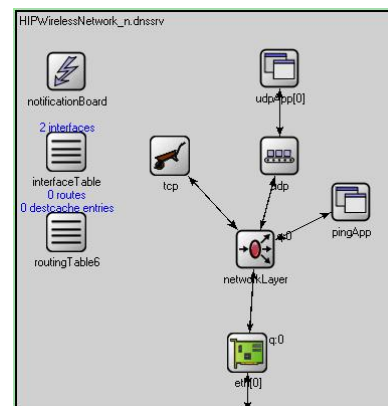


Special Nodes in HIPSim++ III.



■ 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]
 - As proper implementation of all the cryptographic mechanisms in HIP is outside of the scope of our researches, we use only simplified Encapsulating Security Payload Header mechanisms for distinguish HIP data packets based on SPIs
 - 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
 - Every *ESPHeaderMessage* has a special object (called *IPv6EncapsulatingSecurityPayloadHeader*) per header to carry the SPI value as parameter
 - This object is derived from the *IPv6ExtensionHeader* class of INET
- DNS messages
 - The basic HIP namespace resolution functions are implemented using a simple query/response message pair called *DnsQuery* and *DnsResponse*

Modifications to INET

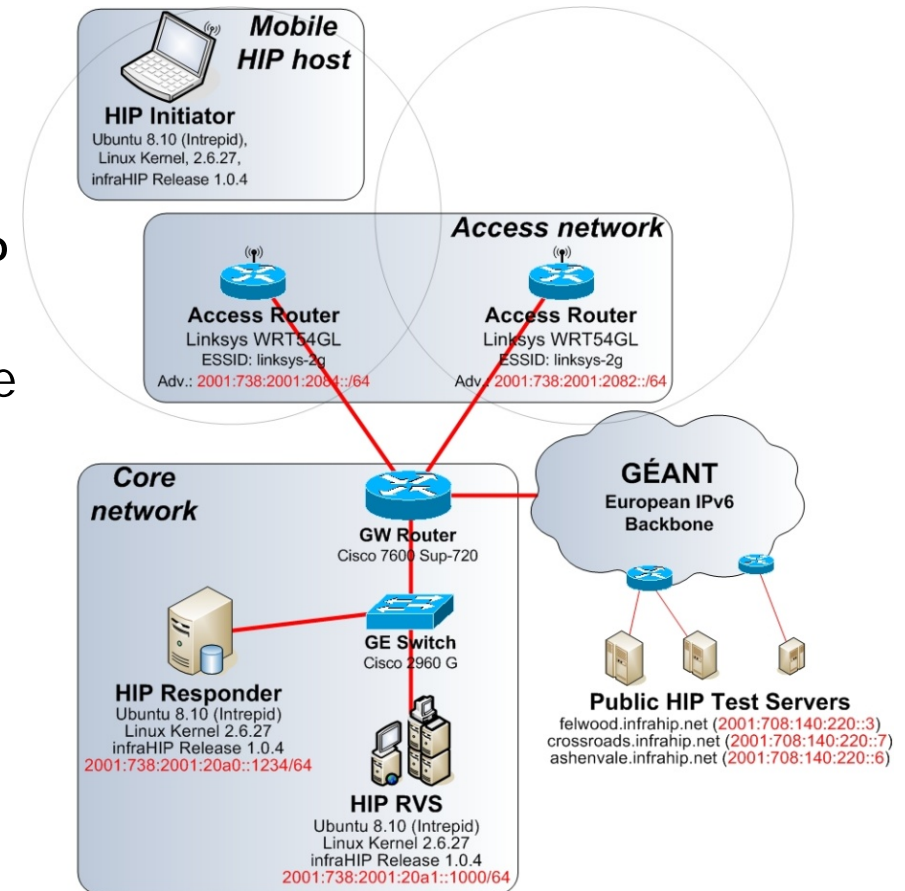


- In order to integrate HIPSIM++ with the INET framework (INETwithMIPv6 version 20081128), several extensions and modifications have had to be introduced in the existing modules/classes of INET
- However these modifications are not transparent to the current INET building blocks, it has to be emphasized that the revision we made is basically a set of additional supplements and bugfixes without breaking or changing the original functionality
- A short list of the changes and their nature:
 - */Network/IPv6/IPv6ExtensionHeaders.msg*: insertion of HIP header and parameter structure; ESP header extended with SPI.
 - */Network/IPv6/IPv6Datagram.cc*: integration of mechanisms for HIP header management.
 - */Transport/UDP/UDP.cc*: correction of a bug preventing proper UDP communication over IPv6/HIP.
 - */Network/IPv6/IPv6.cc*: correction of a bug causing memory leaks during packet transmission towards upper layers.
 - */Network/ICMPv6/IPv6NeighbourDiscovery.cc*: introduction a new *NotificationBoard* message designed to inform the HIP layer about address changes after finishing Duplicate Address Detection procedures of IPv6.
 - */NetworkInterfaces/Ieee80211/Mac/Ieee80211Mac.cc/h*: introduction of a simple Radio module identifier for *NotificationBoard* messages.
 - */NetworkInterfaces/Ieee80211/Mgmt/Ieee80211MgmtSTA.cc/h*: extension of *NotificationBoard* messages with a new object for proper identification.
 - */NetworkInterfaces/Ieee80211/Mgmt/Ieee80211AgentSTA.cc/h*: extension of *NotificationBoard* messages with a new object for proper identification; introduction of a new function for updating *InterfaceTable* information; introduction of new *NotificationBoard* messages for distinguishing new and old WLAN Access Points.
 - */NetworkInterfaces/Radio/AbstractRadio.cc/h*: introduction of a new function for Radio module identification and ID setup.
 - */World/ChannelControl.cc*: extension with support of multiple radio channels.
 - */World/ChannelAccess.cc*: extension with support of multiple radio channels.
 - */Network/IPv6/Contact/InterfaceEntry.cc/h*: extension with a toolset for managing connection states.

HIP Testbed



- In order to evaluate our HIPSim++ simulation framework, we designed and implemented a real-life HIP testbed based on the InfraHIP implementation:
 - InfraHIP is the most complete and standard compliant HIP implementation
 - Linux kernel 2.6.27
 - wide-scale features including RVS and even DHT and HI³ support
- Full mobility and multihoming support



InfraHIP

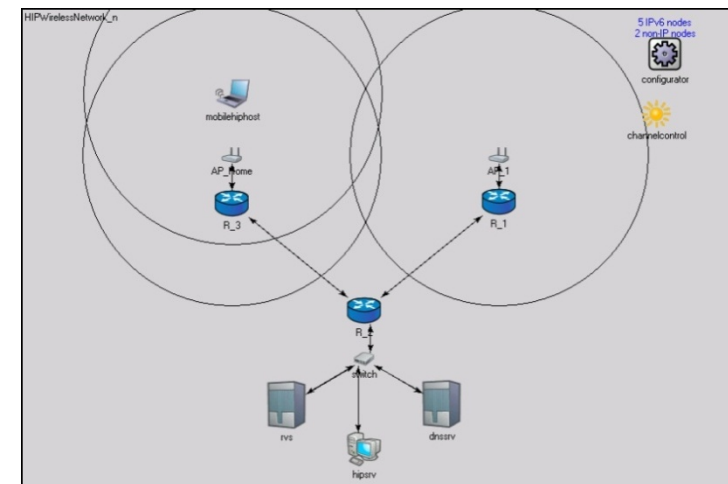
Infrastructure for HIP

Simulation topology and scenarios



- The network topology built for HIPSim++ evaluation can be considered as an exact copy of our real-life HIP testbed architecture
- There are only two differences:
 - the lack of DNS server in the testbed (here the `/etc/hosts` file is used for name-HIT-IP resolution). Note, that it has no relevance in the comparison: DNS procedures are initiated only before connection establishment (i.e. Base Exchange).
 - the testbed comprises IPv6 connection to the GÉANT network while the simulation topology lacks of such feature. This is also irrelevant, because GÉANT connection was not used during the evaluation.

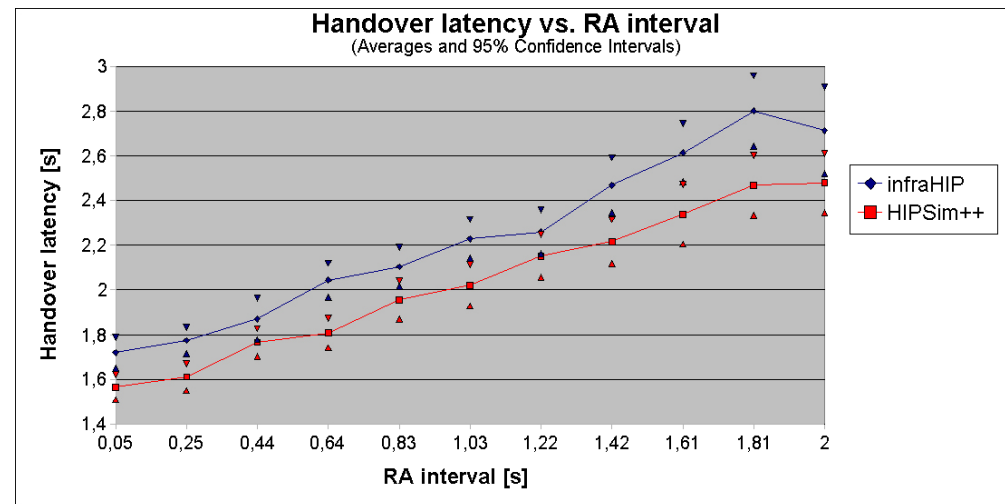
- In focus: mobility management
- Three main scenarios:
 - Handover latency vs. RA interval
 - UDP packetloss vs. offered datarate
 - TCP throughput vs. handover freq.



Handover latency vs. RA interval



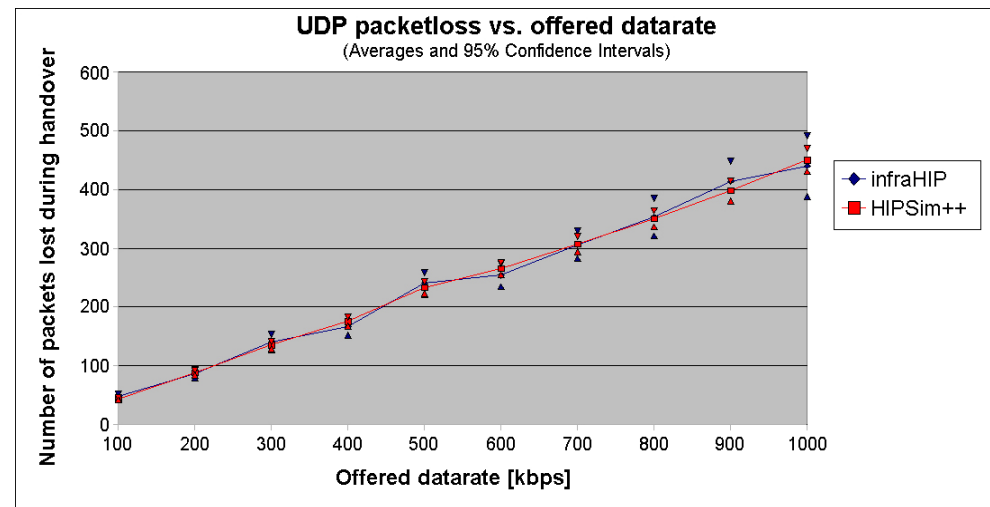
- HO latency: the time elapsed between losing the connection at the old AP and the mobile sending out the third UPDATE packet while connected to the new AP
 - Configuration of new IPv6 address by means of stateless auto-configuration using RAs sent by the routers
 - Handling the IP address change by the HIP layer
- 11 series with different average RA intervals
- Both real-life and simulated networks were set up to trigger 100 handovers in every series
- The maximum difference is around 0.3 sec (at 1.81s)
- A 95% confidence interval gives a view on the statistical properties of the results



UDP packetloss vs. offered datarate



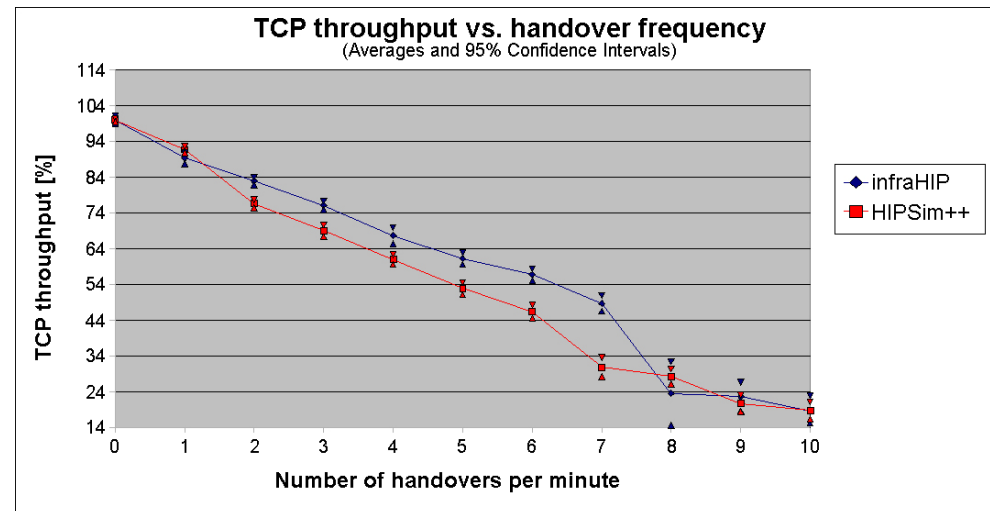
- Number of lost UDP packets during a handover in a HIP system
- 10 series with different data rates offered by the HIP responder to the mobile
- A point on the plot represents the average UDP packet loss of 100 handovers both in the real-life and the simulated cases
- A 95% confidence interval gives a view on the statistical properties of the results



TCP throughput vs. handover freq.



- The TCP throughput measurements were taken between the HIP initiator and responder at different handover frequencies
- Every point represents the average throughput of a one hour long period applying the same value for the number of handovers per every minute of that hour
- 11 series with different number of handovers per minute (0-10)
- Results are expressed as a percentage of the throughput of the no-handover scenario (the first result on the left)



Conclusion and future work



- We designed and implemented a Host Identity Protocol simulation model (called HIPSim++) integrated into the INET/OMNeT++ simulation environment
- We built and configured a real-life HIP testing environment based on InfraHIP, and compared the outcomes of our HIPSim++ model with the reference results obtained from the testbed
- Results show apparent accuracy of HIPSim++ in terms of handover metrics:
 - Handover latency
 - UDP packet loss
 - TCP throughput
- This accuracy has been provided by modeling HIP messages, nodes and mechanisms according to the actual recommendations of current IETF RFCs, and by re-using the existing detailed IPv6, mobility, channel, etc. models of the INET Framework
- As a part of our future activities we will further extend HIPSim++ with
 - HIP signaling delegation and service discovery mechanisms;
 - advanced HIP multihoming solutions;
 - special HIP-based mobility protocols (micro-mobility, network mobility, per-application mobility, etc.);
 - INET/OMNeT++ 4.x support.

Thank you!

Questions?



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

László Bokor, Szabolcs Nováczki, László Tamás Zeke, Gábor Jeney
{[goodzi](mailto:goodzi@mcl.hu), [nszabi](mailto:nszabi@mcl.hu), [koci](mailto:koci@mcl.hu), [jeneyg](mailto:jeneyg@mcl.hu)}@mcl.hu