SIP over an Identifier/Locator Splitted Next Generation Internet Architecture

Conference Paper · March 2008

DOI: 10.1109/ICACT.2008.4493839 · Source: IEEE Xplore

4 authors, including:

Christian Esteve Rothenberg
University of Campinas
105 PUBLICATIONS **3,340** CITATIONS
SEE PROFILE

Walter Wong
University of Campinas
12 PUBLICATIONS **98** CITATIONS
SEE PROFILE

Fabio L. Verdi
Universidade Federal de São Carlos
58 PUBLICATIONS **368** CITATIONS
SEE PROFILE

SIP OVER AN IDENTIFIER/LOCATOR SPLITTED NEXT GENERATION INTERNET ARCHITECTURE

Christian Esteve Rothenberg, Walter Wong, Fábio L. Verdi, Maurício F. Magalhães
Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
State University of Campinas (UNICAMP), Brazil
Email: {chesteve, wong, verdi, mauricio}@dca.fee.unicamp.br

***Abstract*—Research around the tenets of a next generation Internet architecture has resulted in numerous future Internet proposals, both evolutionary and clean slate. One promising approach is the identifier/locator split, which opens a new paradigm of network communications by using static node identifiers uncoupled from the actual network location. In this work, we validate our instantiation of an id/loc splitted next generation Internet architecture in respect of legacy application support. Our prototype implementation demonstrates that existing SIP services can benefit from the inherent capabilities of the proposed architecture in terms of transparent mobility and security support.**

***Keywords* — Identifier, node, identity, locator, id/loc, next generation, Internetworking, SIP, mobility, security, IPv6**

1. Introduction

Back in the late 70s, it was stated that Inter-networking [1] required three basic network functions clearly defined and related to each other: a) *naming* (How to refer to an entity?), b) *addressing* (How to refer to a route to an entity?) and c) *routing* (How to deliver packets to the entity?). The approach taken by the Internet Protocol (IP) suite uses IP addresses for both naming and addressing; combining thus two basic roles of networking [2]:

1. End-point Identifier: Name of interface on host
2. Network Locator: Name of topological location

The adoption and evolution of the IP protocol stack is a success story that fulfilled by far the networking needs for which it was designed (simple, resilient, scalable, mainly static). However, today's use of the Internet exposes limitations including seamless mobility, security, multi-homing, reduced address space, and so on [3]. Many of these shortcomings can be rooted back to the IP semantic overload problem (as per Saltzer in RFC 1498) of using IP as identifier at the transport layer and as topological locator at the network layer.

Apart from the "patching" attempts (e.g. NAT, Mobile IP, IPSEC), a series of clean slate architectural proposals have arisen recently. These include FARA, TurfNet, Plutarch, DONA, i3 [4], ROFL [5] and the Node Identity Internetworking Architecture (NodeID) [6]. These future Internet architectures often share design principles (e.g., id/loc separation) but approach the problem from different perspectives (e.g., new name and/or address spaces, flat- and content-based routing, network vs. host intelligence).

Basically, the goal of the identifier / locator (id/loc) split approach [7] is to separate the roles of IP addresses. However, this approach does not come without a cost. Binding the new identifiers to the actual IP addresses becomes a challenge, as a consequence research efforts are required in the field of routing on flat (topology free) identifiers. Recently, the IETF has started architectural discussions on id/loc separation [8] and even the ITU-T is working on a draft recommendation [9] on id/loc separation requirements.

Inspired by several proposals in the literature including FARA, i3 and specially the NodeID architecture (first presented in [2] and recent IETF I-D [10]), we proposed and successfully implemented an operational framework [11] that enables the validation of novel networking concepts based on node identifiers uncoupled from their network locators.

Now that our framework is maturing [11][12], it is important to validate the claim of seamless support for existing application and to move beyond the abstract benefits of id/loc separation. For these purposes, we have chosen the Session Initiation Protocol (SIP) [13], a widely adopted multimedia session control protocol demanding security, mobility support and operation over heterogeneous IP networks. We explore how SIP can operate with and benefit from the inherent capabilities of an id/loc separated Next Generation Internet (NGI) architecture.

The remainder of this work is as follows. Section 2 provides required background information on our next generation Internet architecture framework and SIP. In Section 3 we evaluate SIP over our proposed architecture in terms of transparency, security, mobility and performance. Section 4 includes final remarks and future work. Finally, Section 5 concludes this work.

2. Fundamental concepts

A. Prototype of a Next Generation Internet Architecture

Our NGI architecture framework [3] implements generic functionalities such as identifier/locator separation, name resolution, flat routing and legacy application support, which enable the low cost (time, code size, modular implementation) instantiation of prototype NGI architectures.

Our prototype borrows from NodeID [2] the idea of augmenting the Internet by introducing an identity layer to globally identify end-hosts over the Internet, enabling desired NGI features such as mobility, multi-homing and embedded security. As in the Host Identity Protocol (HIP)[14], end-points use flat self-certifying node identifiers (NIDs) derived from a public/private cryptographic key pair. At this point it is important to recall what [14] states: An *Identity* refers to the abstract entity that is identified and an *Identifier*,

on the other hand, refers to the concrete bit pattern that is used in the identification process. Figure 1 depicts how these identifiers become the new waist of the envisioned Internet hourglass model.

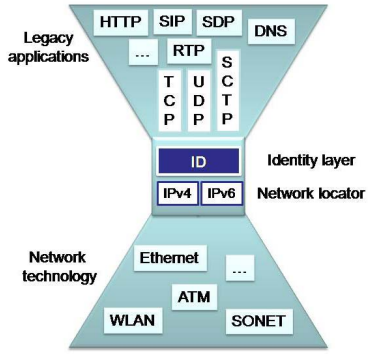

Figure 1. Hourglass-model of the proposed future Internet architecture

The new host identity space lies between the hostname and network address spaces. Instead of mapping human-readable host names directly into network addresses, as in the Domain Name System (DNS), the architecture adopts two common spaces, a shared name space and a common identity space, thus enabling inter-domain communications. In this way, the host name is mapped into network topology independent host identities. A second mapping dynamically translates host identities into host addresses that are suitable for network layer data forwarding. The architecture manages the global name (FQDN is assumed as per [15]) and identity spaces, whereas the address space is local to each individual autonomous network (domain).

As defined in the Node ID architecture, the current view of the Internet is simplified by considering the existence of a core Internet at the root, and non-core domains attached to the edge of the core domain. Non-core domains are considered dynamic and attach to other domains arranging into a tree-like structure.

Our NGI architectural proposal (see Fig. 2) introduces the novel concept of Domain Identifier (DID), a global and unique identifier for each domain. The DIDs are propagated towards the core to be registered in a distributed Locator Service, instantiated for example by a Distributed Hash Table (DHT). A DID Router (DR) is responsible for inter-domain communications and network protocol translation.

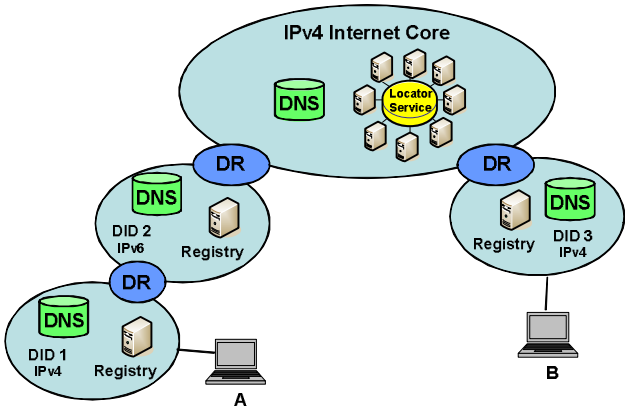

Figure 2. Reference model of our instantiated next generation Internet.

A Registry Service instantiated by a Rendezvous Server (RVS) is available in each domain. The RVS performs node registration and provides local mapping of the 128-bit node identifier to domain network technology locator information (e.g. IPv4, IPv6). End-to-end routing is based on the pair of flat identifiers destination DID and NID. Further details on our NID/DID routing approach (e.g., DR operations, scalability issues) are not relevant for the purposes of this work and will be presented in an upcoming publication.

B. Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) [13] is a client/server protocol used for the initiation and management of multimedia sessions between users. Commonly referred to as SIP client application, a User Agent (UA) is a logical entity that integrates UA server (UAS) and UA client (UAC) functionality. Intermediate SIP elements are known as Proxy Servers (outbound/inbound) responsible for routing (proxying) SIP signaling messages. Each SIP domain is served by at least one Proxy Server that commonly embeds the functionality of a Registrar and a Redirect Server. The Location Server (LS) stores and provides location information about users and is typically queried by Proxy and Redirect Servers to locate callees. Thus, the SIP overlay composed by the SIP proxy servers provides a user location service for UAs and enables rich communication services. As shown in Fig. 3, control signaling (SIP) and data (RTP stream) paths are uncoupled and rely on DNS resolutions and intermediate SIP elements for successful session establishment.

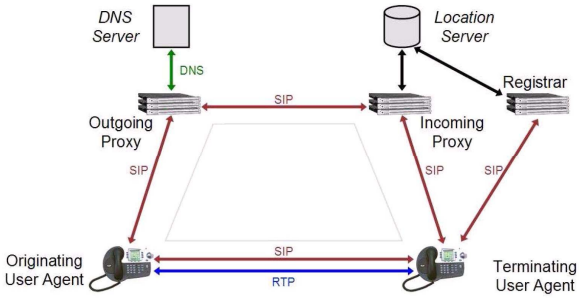

Figure 3. Typical SIP configuration known as "SIP trapezoid".

A UA registers at its domain serving Registrar that stores the current location information in the LS, and in this way becomes globally available. Afterwards a UA can contact another UA by exchanging signaling information through its dedicated outgoing Proxy Server which carries out the routing function. Address resolution of the next routing hop is done by contacting the DNS Server [16]. The next routing hop can be an intermediate Proxy Server, the corresponding inbound Proxy Server or a PSTN Gateway. As part of the SIP signaling request, the Session Description Protocol (SDP) [17] is used to negotiate a number of characteristics of the desired session (e.g. codec type, contact information, ports, etc.).

SIP already uses the concept of id/loc separation, however at a different layer. A SIP Unified Resource Identifier (URI) is an application layer identifier for SIP entities. In the most general form, a SIP URI (Section 19.1 of RFC 3261[13]) looks like: *sip:user:password@host:port;uri-parameters?headers.* However, the SIP specification [13] only mandates the URI to

contain sufficient information to initiate and maintain communication sessions with the requested resource (e.g. UA, Proxy Server). Only the *host* part of a SIP URI is mandatory and is specified to contain a fully qualified domain name (FQDN) form or a numeric IPv4 or IPv6 address. The so-called Address of Record (AoR) e.g. *sip:alice@atlanta.com* is a SIP user identifier that the Registrar binds to the host address where the SIP user can be contacted (Contact URI). End-to-end connectivity is thus tied to the specific SIP application and depends on DNS resolutions.

Resolving a NID to a network locator (IP address) is actually a very similar problem to resolving a SIP URI to a Contact URI. Both mappings are located in rendezvous-like registry services, centralized per domain (LS in the SIP-domain and RVS in the DID-domain) or distributed in case of the DNS (SRV lookups [16]) and the DR resolution (Locator Service) happening at the core.

IETF work on providing a distributed SIP service is underway in the P2PSIP WG. Interestingly, current discussions regard the use of HIP as an id/loc separation approach that provides static end-host identifiers on which the SIP services may run in a distributed P2P fashion. Further related work on the combined operation of SIP and HIP was discussed in [18][19].

3. Evaluation

We set up the typical SIP "trapezoid" configuration and evaluate its operation over our framework in terms of transparency, security, mobility and overall performance.

Figure 4 illustrates the SIP scenario under evaluation. After registration, Alice first sends an INVITE message to Bob through its domain outbound Proxy P1 (Step 1). P1 queries the DNS (Step 2) to find the next SIP hop to reach Bob and then forwards the message (Step 3) to the inbound Proxy P2 of Bob's domain. After triggering the Registrar information, the INVITE message is delivered to Bob UA2 location (Step 4). Upon SIP/SDP negotiation, end nodes can exchange data directly to each other (Step 5).

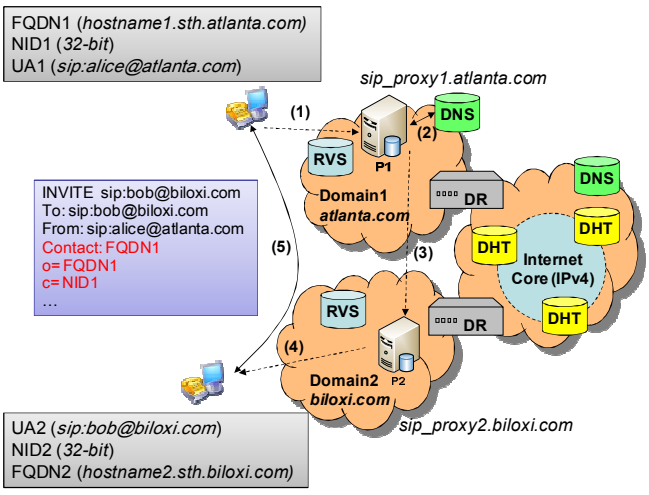

Figure 4. SIP session establishment over the ID/locator uncoupled next generation Internet architecture instantiation

Our NGI prototype is a linux user space implementation and all prototype modules were implemented using C language. SIP proxy servers used the Openser implementation [25] and several SIP UAs were tested (e.g., X-lite, Kphone, Ekiga, PJSUA, Sofia).

A. Transparency

Unmodified SIP applications are transparently supported through legacy name resolution proxies and legacy packets interceptors provided by our framework [3]. The former intercepts SIP application DNS queries and resolves the requested node and domain identifiers. The *NID Mapper* module creates and saves the NID association between the communicating nodes. SIP implementations typically call the *gethostbyname()* function to resolve their own host address and the address of the next hop SIP entity. A 128-bit (or 32-bit in IPv4 case) cryptographic hash of the node identity, namely the node identifier (NID), is returned to the SIP application. The received NID is transparently used in the SIP/SDP signaling (e.g. *Contact* header, *c*= field) and socket binding as if it were a typical host IP address.

Both SIP [13] and SDP [17] specifications recommend the use of FQDN as network connection information whenever available (for both IPv4 and IPv6). However, typical SIP client implementations do not check whether the host FQDN is a globally available reference to the system address. UAs commonly insert the host IP address and relay on NAT transversal solutions to ensure end-to-end connectivity. Introducing the identity layer with unique NIDs of global scope allows all nodes to have a meaningful FQDN [6][15].

We used the SIPp traffic generator [25] as one communicating UA to create several SIP signaling scenarios containing only FQDNs as network connection information (*Contact* header, *c*= and *o*= fields) of SIP/SDP. As expected, SIP registration and session establishment operations using only FQDNs run normally. We observed that Proxy Servers and UAs correctly performed the correspondent FQDN resolutions that transparently triggered the NID connection establishment by our framework.

Thinking in an "all-NID" scenario, NID/DID descriptors for the extension-friendly SIP and SDP parameters could be defined and used instead of the host FQDN.

Figure 5 shows the resulting mapping and relationship between a SIP URI, the crypto-enabled node identity, the node identifier, the FQDN and the locator information.

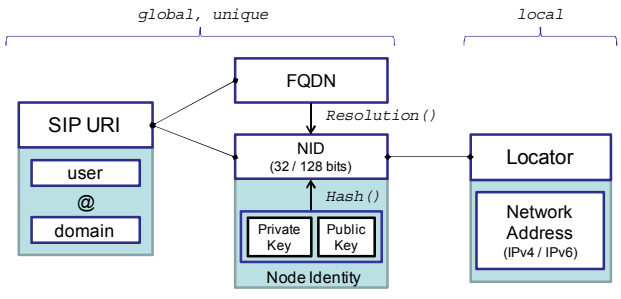

Figure 5. Relationship between SIP URIs, NIDs, FQDNs and locators.

Legacy packets interceptor and handling mechanisms were implemented using the *Iptables* tool to capture legacy packets

sent to a virtual interface whose address is the NID. The implemented *Filter* module captures SIP and RTP packets (Fig. 6a) and amends the ID Header (IDH) containing source and destination NID and DID information (Fig. 6b). Packets are then encapsulated and sent to the destination host over an IP+UDP tunnel (Fig. 6c). At the destination, the payload is correctly delivered to the legacy application. VPN-like implementations follow similar packet capture and encapsulation principles. SIP and RTP traffic are transparently exchanged between the communicating nodes across the domain(s) following the DID routing procedures.

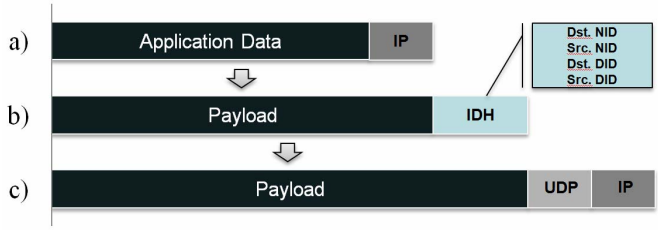

Figure 6. Legacy SIP and RTP traffic is transparently delivered to the end-host application.

B. Security

Besides the initial security considerations contained in the SIP standard, many extensions have been and are being developed within the IETF community to secure the SIP protocol and the RTP traffic, including IPsec, TLS, HTTP Digest authentication, S/MIME and SRTP (see Table 1).

Table 1. IETF work to secure SIP-based communications.

RFC	Title*	Date
3310	HTTP Digest Authentication using AKA	09/2002
3325	Private Extensions to SIP for Asserted Identity	11/2002
3329	Security Mechanism Agreement for SIP	01/2003
3702	AAA Requirements for SIP	02/2004
3853	S/MIME AES Requirement for SIP	07/2004
3893	SIP Authenticated Identity Body (AIB) Format	09/2004
4189	Requirements for End-to-Middle Security for SIP	10/2005
4453	Requirements for Consent-Based Communications in SIP	04/2006
4567	Key Management Extensions for SDP and RTSP	07/2006
4568	SDP Security Descriptions for Media Streams	07/2006
4474	Enhancements for Authenticated Identity Management	08/2006
4572	Connection-Oriented Media Transport over TLS in SDP	07/2006
4961	Connected Identity in SIP	06/2007

* Slightly adapted to fit the column size

SIP security threats [20] include flooding and vulnerabilities at the signaling-application level. For example, the SIP/SDP offer/answer model allows DoS attacks by simply including the victim's IP address as media destination in the SDP offer.

Our framework's security model works on the identity layer, providing authentication, data integrity and confidentiality. We employ cryptographic end-hosts identifiers as proposed by HIP [14]. These identifiers are generated from the public key of a pair of asymmetric keys provided by a Public Key Infrastructure (PKI). End-hosts may self-claim their identities, thus embedding security in the communication.

Secure DNS insertion or modification is provided by the adoption of the *DNSSec* extension. Moreover, nodes must establish a security association with the RVS prior to any insertion or modification. This association is negotiated during the bootstrap process or whenever a node arrives at a new domain. Communicating nodes exchange their certificates with each other to verify their authenticity and exchange the Diffie-Hellman parameters required for a symmetric session key establishment that provides end-to-end authentication and confidentiality.

The single secure channel established at the identity layer of the communicating nodes (UA1-P1, P1-P2, P2-UA2, UA1-UA2) is more efficient than the security model provided, for example by TLS that requires a separated secure channel for each TCP flow. Furthermore, the "NID pipe" can be used for all traffic between the communicating nodes. As in HIP, in order to prevent DoS attacks, the node initiating the connection must first solve a computationally expensive puzzle generated by the destination.

Running SIP over our architecture results in both the SIP signaling and the data being transparently secured end-to-end natively by the architecture. As a result, SIP and RTP are released from these heavy duties.

C. Mobility

SIP implements mobility management at the application layer [21] and supports means for *personal, service, session* and *terminal* mobility. *Personal* mobility is natively provided by the fact that users are addressed by their SIP URI independently of the network location and terminal choice. Also implicitly supported is the so called *service mobility*; the ability to get access to the same user services while moving or changing devices and/or networks. SIP defines methods (*REFER* message) to perform *session mobility* maintaining an ongoing media session while changing terminals. Finally, SIP has means to support *terminal mobility*. When a mobile host moves from one network to another, after acquiring a new topologically correct IP address (e.g., via DHCP), ongoing SIP sessions may be resumed by sending a *Re-INVITE* message to the corresponding node (CN) informing about the updated network contact information. However, *terminal* mobility mechanisms using SIP alone present well-known shortcomings [22]. First, TCP connections will break in case of IP change. Second, overall handoff delay has been shown to be larger than lower layer mobility solutions.

Our proposal uses a location management mechanism based on a Rendezvous Server (RVS) where mobile nodes update and query identity to locator mapping. In this way, moving nodes use this Registry service to remain globally reachable. Additionally, moving nodes can directly inform CNs about the mobility event (e.g., via *Redirect* message). Every node periodically updates its locator and DID in the serving RVS. Due to our optimized proposal of having DIDs, only DR registry information and domain mobility events need to be propagated towards the Locator Service in the core.

The mobility management of our architecture recalls SIP mobility mechanisms (SIP *REGISTER* vs. *RVS Update*, SIP *Re-INVITE* vs. *Redirect/Relocate*). Once again, the fundamental difference is that it operates at the identity layer. Supporting internetworking natively in the architecture allows mobility events become transparent to applications, which can maintain an unchanged NID as transport endpoint identifier.

The inherent mobility support of the id/loc splitted NGI architecture enhances communication services and eases application development by avoiding application specific mobility features. Further mobility scenarios and details on our prototype's mobility support can be found in [12].

D. Performance

We ran several performance tests on the scenario described in Section 3 to proof the SIP operations over our NGI instantiation. The goal of the performance tests was twofold; first, to validate the prototype NGI implementation and second, to quantify the overhead introduced by our architectural proposal and operational framework. We certainly do not aim at providing reference values for real world inter-domain SIP communications.

We used dedicated Pentium4 3.0 GHz machines to instantiate the two domain routers, the two RVS and the Location Service in the core. A WLAN device was used to emulate the connectivity of the mobile node. DNS and DHCP servers in addition to the SIP Proxy Servers [24] and UAs under test were deployed in both the originating and destination domains. A traffic generator [23] was used to evaluate the RTP traffic flow carrying voice packets using standard coding schemes such as G.711 and G.729.

A signaling analysis of the SIP session establishment over the prototype reveals the additional interactions required by the architecture (see Table 2). The signaling overhead is caused by the extra FQDN queries for the originating and destination UA identifiers, the Registry and Locator Services, and the end-to-end Security Association (SA). In our low latency testbed environment, the variation of the SIP session establishment time over the prototype was negligible.

Table 2. Signaling overhead of a SIP session over the proposed NGI represented by the required amount of query/response interactions

	DNS	RVS	DHT	SA
SIP over IP	2	-	-	-
SIP over the NGI	4	4	1	2

SIP sessions including RTP audio streams and SIP TCP connections survived a network address re-configuration caused by a communicating node moving to a new domain. Table 3 summarizes the mobility tests results. An experiment snapshot of the mobility event is illustrated in Fig. 7.

Table 3. Mobility results over 10 experiments, with RVS *Update* every 3s and G.729 (20ms) coded RTP payload.

	Mean	Std. dev.
L2 association	30,2ms	0,8ms
DHCP	266,7ms	284ms
NGI handoff	1,6s	0,9s
Total handoff	1,9s	0,7s
Pkts lost	93	36

Moreover, support for domain mobility is an additional architectural feature of our prototype much more convenient and efficient than having every communicating host in the domain performing the handoff procedures.

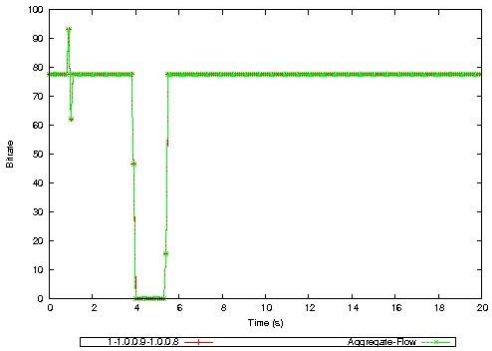

Fig 7. The RTP stream is seamlessly restored natively by the NGI architecture after the mobility event.

Because of the timing constraint (< 150 ms one-way delay) of real time communications, VoIP packets are typically small e.g., 20-240 bytes long payload depending on the codec. Consequently, bandwidth consumption overhead (ratio of header size to payload size) in case of VoIP traffic is especially critical as shown in Fig 8 and Table 4. Furthermore, building the additional headers and applying the necessary time consuming cryptographic functions to the payload introduce additional delay to packet transmission. However, in our testbed scenario with high performance machines and without additional computational load this overhead was not an issue for the end-to-end delay of the RTP stream.

Table 4. Bandwidth consumption overhead of VoIP

	G. 729 (20ms) 24 kbps codec bitrate		G. 711 (20ms) 80 kbps codec bitrate	
	Effective BW	Overhead	Effective BW	Overhead
RAW (No Sec.)	47,2 kbps	0.37	88,0 kbps	0.11
SRTP	52,0 kbps	1.17	108,0 kbps	0.35
IPSEC	72,8 kbps	2.03	128,8 kbps	0.61
Our NGI	80,8 kbps	2.37	136,0 kbps	0.71

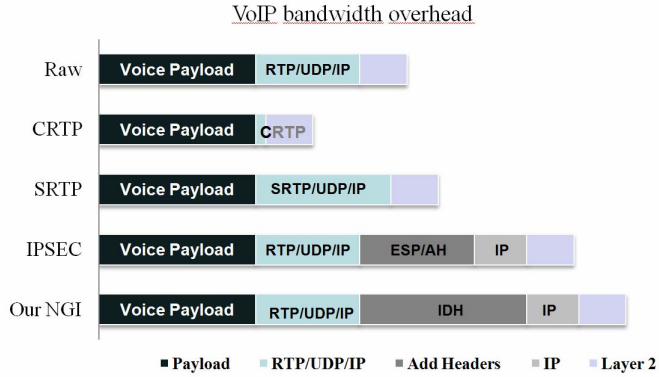

Fig 8. Overhead introduced by various protocols for a 60 bytes payload, an typical packet length for voice traffic (20ms G.729).

The overhead introduced by our architecture is comparable to other approaches for secure VoIP communications. Thanks to the "NID logical channel" established between the end parties, typical hop-by-hop header compression schemes (e.g., IPHC, CRTP, ECRTP, ROHC) could be considered to reduce the bandwidth consumption overhead.

We may conclude that the overhead introduced by the architecture is affordable when considering the benefits of embedded security and mobility-enabled communications.

4. Final considerations and outlook

Most of the final considerations can be drawn from the implications of having implemented id/loc separation [7]. It has been shown that IP layer mobility becomes easier as well as multi-address multi-homing. Moreover, it allows a further degree of freedom to routing opening new possibilities to re-consider the division of information between addresses and routing tables. However, further studies on flat routing support are required.

As a consequence of having static unique global identifiers, end-to-end connectivity is restored at layer 3.5. Security is enabled through self-certified identifiers that avoid application-specific security infrastructure or add-ons. We hope to publish soon the HIP-inspired security model of our NGI prototype (in a similar way to our mobility support presented in [12])

A common concern when trying to validate new Internet architectures is test implementations. Our prototype shows that id/loc separation is implementable with reasonable efforts, guaranteeing support for existing applications and enabling the validation of new internetworking paradigms. The NIDs of our operational framework are transparently used by legacy applications and transport protocols in place of real host addressees. Using NIDs as end-host identifiers in this manner brings the following advantages: a) NIDs are stable and unique, b) NIDs can be used with the kernel's Berkeley Socket API without changes and c) applications get "out-of-the-architecture" NAT traversal, mobility, multi-homing and secure associations to correspondent nodes.

By having a transport-layer end-point bound to the NID, a connection between two hosts can traverse many addressing realm boundaries. The IP addresses are used only for routing purposes and may be changed freely during packet traversal. The transition from IPv4 to IPv6 can definitely not happen overnight and is expected to happen in an incremental manner. Id/loc separation approaches promise easing the IPv6 transition and any further evolution of the network technology. It should be remarked that our architectural approach can be gradually deployed; another key incentive for adoption.

In a future publication we will provide more details of our DID/NID based routing approach, thereby revisiting flat-identifiers routing approaches and considering scalability, heterogeneity, and domain mobility issues.

Finally, we have shown that our framework is SIP/SDP compliant and perfectly fits with SIP´s reliance on DNS resolutions and SIP URI name forms as application-level end-point identifiers. Regarding SIP and the benefits of id/loc split, we plan to explore the SIP-based IP Multimedia Subsystem (IMS) architecture on top of our NGI prototype.

5. Conclusion

Our operational framework enables the instantiation of next generation Internet architectures that put into practice novel networking concepts such as id/locator split. We conclude that existing applications like SIP are seamlessly supported and can clearly benefit from the proposed architecture in terms of security and mobility by transparently using permanent cryptographic–enabled node identifiers uncoupled from the actual network location. Easing the transition to IPv6 and the possibilities for partial deployment are further incentives for the adoption of identifier/locator separated architectural approaches. Performance results in our testbed environment validated our operational framework and the proposed concepts towards a next generation Internet architecture.

REFERENCES

[1] J. F. Shoch, "Inter-Network Naming, Addressing, and Routing." *In Proceedings of IEEE COMPCON*, Fall, 1979.

[2] J. Chiappa, "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture", [Online]. Available: *http://users.exis.net/~jnc/tech/endpoints.txt*, 1999.

[3] R. Jain, "Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation," *Military Communications Conference* MILCOM, Washington, DC, October 23-25, 2006.

[4] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet Indirection Infrastructure," *In Proceedings of SIGCOMM 2002*.

[5] M. Caesar, K. Lakshminarayana and et al. "ROFL: Routing on Flat Labels". *In Proceedings of SIGCOMM 2006*.

[6] B. Ahlgren, J. Arkko, L. Eggert and J. Rajahalme. "A Node Identity Internetworking Architecture". *In Proceedings of the IEEE INFOCOM 2006 Global Internet Workshop*, Spain, 2006.

[7] P. Nikander. "Implications of Identifier / Locator Split", Helsinki University of Technology (TKK) NETS 1a morning coffee, Dec. 2004.

[8] D. Farinacci et al. "Locator/ID Separation Protocol (LISP)". IETF Draft, *draft-farinacci-lisp-02* (work in progress), July 2007.

[9] ITU-T, "Separation of IP into identifier and locator in NGN", Draft Recommendation Y.ipsplit, Beijing, China, 8-12 January 2007.

[10] S. Schuetz, R. Winter, L. Burness, P. Eardley and B. Ahlgren, "Node Identity Internetworking Architecture", IETF Internet-Draft *draft-schuetz-nid-arch-00* (work in progress), September 2007.

[11] W. Wong, R. Pasquini, R. Villaça, L. de Paula, F. L. Verdi and M. F. Magalhães, "A Framework for Mobility and Flat Addressing in Heterogeneous Domains", *In 25° Brazilian Symposium of Computer Networks and Distributed Systems 2007*, SBRC 2007, Brazil May 2007.

[12] W. Wong, R. Pasquini, R. Villaça, L. de Paula, F. L. Verdi and M. F. Magalhães, "An Architecture for Mobility Support in a Next Generation Internet", In *IEEE 22nd International Conference on Advanced Information Networking and Applications (AINA),* Japan, March 2008

[13] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002

[14] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[15] B. Ahlgren, L. Eggert, B. Ohlman, J. Rajahalme, and A. Schieder, "Names, addresses and identities in ambient networks". 1st ACM Workshop on Dynamic interconnection of Networks, September 2005

[16] J. Rosenberg and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[17] M. Handley, V. Jacobson and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[18] J. Y. H. So, J. Wang, and D. Jones, "SHIP Mobility Management Hybrid SIP-HIP Scheme," *In Proceedings of Sixth SNPD/SAWN International Conference*, USA, 2005.

[19] H. Tschofenig, J. Ott, H. Schulzrinne, T.Henderson, and G. Camarillo, "Interaction between SIP and HIP", *draft-tschofenig-hiprg-host-identities* (work in process), Internet-Draft, IETF, June 2007

[20] D. Geneiatakis et al. "Survey of Security Vulnerabilities in Session Initiation Protocol", *IEEE Communications Surveys and Tutorials, vol. 8 (3), IEEE Press*, 2006, pp. 68–81.

[21] H. Schulzrinne and E. Wedlund, "Application Layer Mobility using SIP", *ACM Mobile Computing and Communications Review, vol. 4,,* July 2000.

[22] D. Le, X. Fu and D. Hogrefe, "A Review of Mobility Support Paradigms for the Internet", *IEEE Communications Surveys and Tutorials*, Jan 2006.

[23] A. Botta, A. Dainotti and A. Pescapè, "Multi-protocol and multi-platform traffic generation and measurement", *INFOCOM 2007* DEMO Session, May 2007, Anchorage (Alaska, USA).

[24] Open SIP Express Router, [Online]. Available: http://www.openser.org/

[25] SIPp, traffic generator, [Online]. Available: http://sipp.sourceforge.net/