A Generalized Label-Forwarding Architecture for the Future Internet

Achim Friedland
Ilmenau University of Technology / SONES GmbH
Email: {achim.friedland} @sones.de

***Abstract*—It is common sense that *security*, *privacy*, *mobility* and *quality-of-service* are the vital parts of future Internet architectures. Nevertheless these aspects are still well separated fields of research resulting in a complex and highly redundant protocol stack within todays communication. Contrary to this our *Future Internet* design combines these requirements in a integrated and generalized approach. It is based on an *label-forwarding architecture* using so called *extended labels* and expanding the well-known *end-to-end*-concept to the application layer. This paper outlines a preliminary protocol design for further research on this approach.**

I. Introduction and Motivation

In the last decades communication networks have emerged from intelligent telephone networks to todays connectionless (dumb) internet design. The main design principle behind the actual internet architecture was the efficient interconnection of supercomputers and nomadic workstations. For these applications a *packet-swichted* network is more efficient than the former *circuit-switched* networks, especially in terms of redundancy and fault-tolerance (*"fate-sharing"*). This design paradigm [1] was sufficient for the implementation of a wide range of new and unexpected applications and was able to scale from the small number of hosts within the ARPANET till todays billions of Internet users [2].

Future Internet designs might again change this paradigm as the main application of the Internet is no longer a *„best-effort bit pipeline"* between two hosts (e.g. a client and a server), but a dissemination of data with high service demands (*security*, *privacy* and *quality-of-service*) between multiple mobile peers. Additionally and with respect to the widespread virtualization technologies even these peers become irrelevant as applications may also migrate along the network. In order to enable these concepts the lower layers of the communication networks need a certain redesign as todays overlay-implementations like BitTorrent are limited by the restrictions of the underlying networking protocols.

Unfortunately in the past *security*, *privacy*, *mobility* and *quality-of-service* were well-separated fields of research and had often been reinvented at multiple positions within the networking stack. This division of concerns is proven to be useful as it lowers the complexity and error-proneness of these aspects and especially their implementations, but nevertheless we are sure that this strict division should be softened, as the achievable synergy effect outperforms the drawbacks of this decision. We expect that this will not be possible without a soft *green field* approach and a slightly more statefull network design adding contextual information to the individual packets of a data stream or flow. This will shift the emphasis from todays *packet switching* to something more like a *stream switching* approach. However this additional state must be kept minimal for saving the scalability of the network in terms of its size and the number of transported streams.

II. Basic Idea

Our idea is based on a *label-switched* network design, comparable to MPLS networks, but extends this concept in order to use the intermediate state within the network nodes to implement *security*, *privacy*, *mobility* and *quality-of-service* functions. The *label-switched path* is expanded to reflect an *application-to-application*-networking concept following *Van Jacobson's* network channels [3] and superseding traditional upper layer functions like loss detection, packet ordering or encryption in todays networking protocols. Our approach makes an intensive use of a *label stack* to aggregate the increased number of micro-flows using newly defined *extended labels* which can also be defined as *"globally unique"* to support classical high-redundancy routing and to increase the robustness of the forwarding while privacy issues are neglected.

III. Proposed Architecture

The architecture is called *"Extended Label Stream Switching Architecture (ELSSA)"* as the design was heavily influenced by the demand to aggregate packets to streams within a more stateful network than todays Internet. It is based on traditional *label-forwarding* protocols like MPLS, but was extended by functions from security protocols like *IPSec ESP*, *IEEE 802.11i WPA* and *IEEE 802.1ae MACSec* in order to support security and privacy features but also by the motivation behind IEEE 802.1Q VLAN, IEEE 802.1ad "VLAN-in-VLAN" and IEEE 802.1ah "MAC-in-MAC".

The packet format of the ELSSA protocol is shown in figure 1 and consists of the following entities:

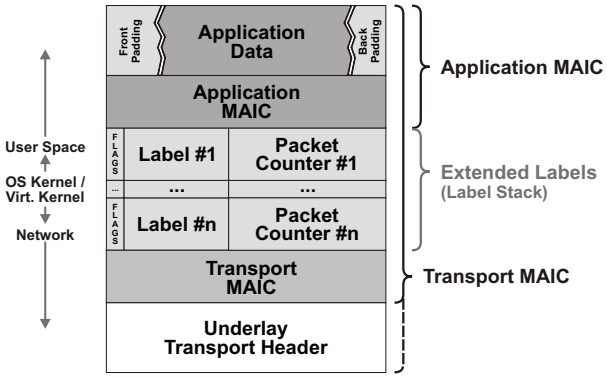

Fig. 1. Packet Format of the Extended Label Stream Switching Architecture (ELSSA)

- **Underlay Transport Header**
  The *Transport Header* is not further defined as it can be the header of any communication protocol within the ISO/OSI layer model. Therefore the *ELSSA protocol* can be transported across nearly all networks even as an overlay. If requested this header can be included in the calculation of the following *Transport MAIC*.
- **Transport MAIC**
  The *Transport Message Authentication and Integrity Code* secures the whole packet with or without the *transport header*. The further layout of this MAIC is not part of the *ELSSA* specification, but the size of this field must be a multiple of 64 bit.
- **Extended Labels**
  A *Extended Label* consists of the fields *flags*, the actual label and a packet counter which counts all packets sent with the given label. The purpose of the *counter* is to support the *initialization vector* for encrypting and signing the packet but also to detect packet lose or reordering. The size of the *flags* is 8 bit. The size of the remaining fields is defined within the first bits of the *flags* according to the following table:

extended label size	flags	label size	counter size
64 bit	000x xxCU	8 bit	48 bit
64 bit	**100x xxCU**	**16 bit**	**40 bit**
128 bit	010x xxCU	32 bit	88 bit
128 bit	110x xxCU	56 bit	64 bit
256 bit	001x xxCU	160 bit	88 bit
256 bit	101x xxCU	184 bit	64 bit
...	...	...	...

The penultimate bit "C" of the *flags* marks the use of the *packet counter*. If this bit is zero the size of the *label* is enlarged by the size of the counter. The final bit "U" of the *flags* is defined as the *"global uniqueness"*-bit. If this bit is true the label is assumed to be globally unique like a traditional ip address, a network coordinate [4], a DIFF-Serv tag, a security label or a checksum. This *label stack* is no longer only used within the network like in the case of MPLS, but may range from the network till the user space of an application.

- **Application MAIC**
  The *Application Message Authentication and Integrity Code* secures the application data. The first two bits of the *MAIC* defines wether a *Front- or Back-Padding* is used. The further layout of this MAIC is not part of the *ELSSA* specification, but the size of this field must be a multiple of 64 bit.
- **Application Data**
  The *Application Data* consists of the actual user data and additional management data of traditional layer 4 transport protocols like flags or an *acknowledgement number*. This implies that in our network architecture at least this management data is part of an user space library and no longer part of the operating system kernel. Even parts of the *label stack* may already be defined within the user space. This might be useful for multiplexing data streams within the application, comparable with the multiplexing of data streams in modern transport protocols like SCTP [5].
- **Front-/Back-Padding**
  The state in the network nodes of the communication define wether a *Front- or Back-Padding* is used to camouflage the real size of the *application data*. This can be important as this might leak information about the nature of the data or communicating hosts and narrow the privacy of the transmission [6]. If the paddings are present they are defined in a similar but elongated format like defined by Bruce Schneier [7].

As the use of a *label-forwarding* protocol implies, this protocol relies on network nodes keeping a state for all incoming labels. We extent the traditional state of network switches – which just implements forwarding-specific information – in several ways e.g. by adding packet counters, cryptographic keys and associated methods which allows us to imitate the functionality of the *Encapsulated Security Payload* of IPSec using the label as *Security-Parameters-Index* for finding the appropriate security context. Further state entities are dealing with *quality-of-service*, mobility and multi-path delivery.

IV. Properties and Comparisons

From the best of our knowledge ELSSA is the only Future Internet approach making use of a combination of a more statefull network and a redefined label stack with integrated support of *security*, *privacy*, *mobility* and *quality-of-service*. By this it accounts the requirements of *Future Internet* designs given by projects and initiatives like FIND, EIFFEL [8], eMobility [9] and NewArch [10]. These requirements include the following aspects:

- *Separation of location, addressing and routing:* As our protocol does not rely on any given underlay and is aware of multiple different addressing schemes by including abstract *globally unique* labels, it meets these demands even using in-band signaling. Such a *globally unique* label might e.g. be an embedded IP address allowing the forwarding plane to use normal IP routing for location lookup and to deliver the packets without any prior label setup.
- *Support of heterogeneous subnetworks and autonomous subsystems:* Like MPLS our protocol can use an additional label/layer for the traversal of a subnetwork. As long as this subnetwork can handle the lowest label these subnetwork will be abstracted to a single link for all upper layer labels.
- *Build-in security and privacy features:* The ELSSA protocol was designed according to multiple security protocols and thus provides both requirements. However, it must be pointed out, that the the implementation of privacy demands is not part of the basic protocol definition but part of an future *privacy-aware* label distribution and path maintenance protocol called *Traffic Analysis Security (TASec)* (figure 2). In comparison with *IPSec ESP* our protocol can provide a similar security but its integrated approach avoids complicated *security policies* especially for assigning multiple layers of encryption in the case of a communication with hosts inside of *Virtual Private Network*.
- *(Micro-)Mobility:* As a communication protocol relying on a more statefull network than todays Internet, it is possible to support macro mobility schemes like MobileIP as well as micro-mobility schemes like I-MPLS [11] and a general multi-path delivery option by the use of an upper layer path maintenance protocol.
- *Modularity and soft migration:* As the history of IPv6 shows even (widely) accepted communication protocols will not be implemented in the field if there is no soft migration path available. Therefore modularity is one of the major goals of our protocol design as it simplifies any migration. Both the protocol and the intermediate state within the network nodes can be striped down to a lightweight *label-forwarding* protocol like MPLS just transporting classical IP datagrams.

V. Future Project Development

Our next steps are to extend and verify the specification of the ELSSA protocol and the needed forwarding-state within the network nodes by the help of communication use-cases. Afterwards we want to define consecutive protocols especially for security and privacy-aware label distribution, security key-distribution, path maintenance and TCP/SCTP adaption (figure 2). We plan to implement the resulting protocols within a simulation testbed and a user-space library in order to prove our approach and further ideas in comparison and coexistence with existing protocols and related Future Internet designs.

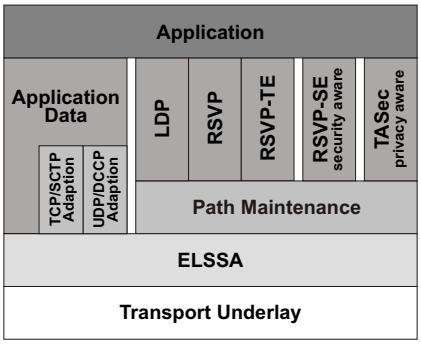

Fig. 2. ELSSA Protocol Architecture

VI. Conclusion

Our proposed protocol was designed to fill the gap between *security*, *privacy*, *mobility* and *quality-of-service* demands of the future Internet in a still very flexible and modular way, which could not be achieved with any of todays communication protocols. To obtain this goal our approach supersedes the traditional layer-model with a more generic label stack and includes mandatory security and quality-of-service aspects. The specification of this protocol is not yet finished, but already opens a lot of possibilities for the development of new communication protocols.

References

[1] David D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM SIGCOMM*, 1988.

[2] Marjory S. Blumenthal and David D. Clark, "Rethinking the Design of the Internet: The end-to-end arguments vs. the brave new world," *ACM Transactions of Internet Technology*, vol. 1, 2001.

[3] Van Jacobson and Bob Felderman, "Speeding up Networking," in *Linux.conf.au 2006*, Dunedin, NZ, 2006.

[4] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris, "Vivaldi: A decentralized network coordinate system," in *In SIGCOMM*, 2004, pp. 15–26.

[5] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," *RFC 3286*, May 2002.

[6] David Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[7] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition*, Wiley, October 1995.

[8] Mähönen, Trossen, Papadimitriou, and Polyzos, "The Future Network Society - A White Paper from the EIFFEL Think-Tank," .

[9] Andersen, Berndt, Ambramowicz, and Tafazolli, "Future Internet - From Mobile and Wireless Requirements Perspective," *eMobility Technology Platform Whitepaper*, 2007.

[10] David Clark, Karen Sollins, John Wroclawski, Dina Katabi, Joanna Kulik, and Xiyowei Yang, "Newarch: Future Generation Internet Architecture," Tech. Rep., MIT Computer Science & Artifical Intelligence Lab, 2003.

[11] René Böringer, A. Saeed, Ali Diab, Andreas Mitschele-Thiel, and M. Schneider, "I-MPLS: A Transparent Micro-Mobility-enabled MPLS Framework," *11th European Wireless 2005*, April 2005.