IPclip: An Architecture to restore Trust-by-Wire in Packet-switched Networks

Harald Widiger, Stephan Kubisch,
Peter Danielis, Jens Schulz, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
18051 Rostock, Germany
Tel./Fax: +49 381 498-7276 / -1187251
Email: {harald.widiger;stephan.kubisch}@uni-rostock.de
Web: http://www.imd.uni-rostock.de/networking

Thomas Bahls, Daniel Duchow
Nokia Siemens Networks
Broadband Access Division
17489 Greifswald, Germany
Tel./Fax: +49 3834 555-642 / -602
Email: {thomas.bahls;daniel.duchow}@nsn.com

***Abstract*—During the last decades, the Internet has steadily developed into a mass medium. The target group radically changed compared to, e.g., the 90s. Because virtually everyone has access to the Internet, threats due to insecurity and anonymity reach critical levels and have to be tackled by both carriers and Internet Service Providers. Regaining Trust-by-Wire, comparable to classic fixed line telephones, could mitigate or even solve problems like Spam, Phishing, and the localization of VoIP emergency calls. This paper presents the hardware implementation of a new and highly flexible solution—Internet Protocol-Calling Line Identification Presentation—which provides additional support for new services to restore people's confidence into the Internet. Supported services are VoIP emergency calls, Spam detection and prevention, and phishing prevention. Already in the access network, the hardware adds unambiguous location information on the packet's origin to IP packets. We document the hardware design of the solution. Furthermore, hardware consumption and performance of a prototype are presented.**

***Index Terms*—Trust-by-Wire, Security, Packet-switched Networks, Internet Protocol, Emergency Calls, Location Information.**

I. Introduction

Due to an ever growing number of customers that subscribe for broadband Internet connections, the Internet has developed into a mass medium. Besides the advantages of using the Internet for business, communication, and information retrieval, it also has its downsides such as a high degree of insecurity and anonymity, which reduce the confidence of the user in the medium Internet. "Black sheeps" are hard to identify. Thus, the problem is to recover and preserve the Internet's reputation because nobody subscribes for insecurity, unavailability, and other unsolicited side-effects like spam.

New services replace existing but outdated systems, e.g., VoIP replaces the traditional fixed line telephone networks, email replaces standard mail, and online banking replaces branch banks. Customers expect these new services to offer at least the same features and Quality-of-Service (QoS) as present conventional techniques. This is only possible with additional mechanisms and support until completely new approaches for internetworking will be matured.

Regarding VoIP, an actual topic that has recently been paid much attention to is emergency calls (ECs) [1], [2]. VoIP services have the great advantage that they can be accessed from anywhere in the Internet. For subscribers, this means that it is possible to be reachable and have service from any place at any time. However, this high mobility poses the problem of providing precise location information (LI) of the caller, which is a vital information in case of ECs. The so-called "Trust-by-Wire" model is *not* given in mobile, nomadic VoIP environments. Neither is it given in other scenarios as fixed line access networks. For fixed telephone lines, the location of the terminals is well known. But for VoIP, this is not the case. Even if a subscriber device possessed a unique IP address and port number, this would not be sufficient to physically locate a caller. Neither do IP addresses provide the same geographic unambiguousness as fixed line telephones, nor have IP addresses been designed for specific purposes like mobile services [1]. Thus, without trustworthy references to a caller's position, an EC cannot be directed to the responsible Public Safety Answering Point (PSAP) and the caller cannot be located and helped [3].

The e-mail service replacing standard mail also sufffers from problems like spam. There is a wide variety of spam types such as picture spam, phishing spam, or commercial spam [4]. Usually, spam mails pretend to origin from every location except from where they really come from. Legitimate email on the other hand typically comes from where it says it comes from. If each IP packet contained precise and first of all true LI that could help to classify incoming emails as either solicited or unsolicited. Identification of spam could be done by analysis of user provided LI (in case there is any) and its (in)validity [5].

During a phishing attack, e.g., initiated by email spam, a phisher pretends to be a trustworthy person or institution. The intention is to get sensitive data such as user names and passwords for online banking or credit card information. To achieve this, manipulated hyperlinks and websites are presented to the potential victim. For example, with online banking, a user cannot be sure that it is really the financial institution's website

that he is going to connect with to use, e.g., a cash remittance service. Providing a mechanism to compare a website's LI with public LI of a finance company would greatly increase security. A user connecting to such a website compares the public LI with the website's LI, which is received. The LI can be used as a criterion to decide if the connection to a sensitive website is secure. Only if the site is considered trustworthy, the connection to the website will be established. Thus, besides other authentication mechanisms, LI is an additional security feature in this scenario [6].

The three examples show: A manageable solution to provide **trustable and accurate LI** is required in packet-switched IP networks. We therefore propose the *Internet Protocol-Calling Line Identification Presentation (IPclip)* mechanism. It associates trustworthy information on the sender's location with every IP packet in form of IP options, which are added to IP packets at the ingress points of the carriers' networks. The entire mechanism is implemented in hardware for non-blocking operation at wire speed.

The remainder of this paper is organized as follows: Section II introduces the concept of the IPclip mechanism and the format of the LI to be added. Afterwards, Section III details IPclip's hardware architecture and the functional blocks, it consists of. In Section IV the details of an implemented prototype and both ist performance and hardware costs are presented. The paper concludes in Section V.

II. IPclip – The Mechanism in General

The name IPclip is derived from the CLIP function of Integrated Services Digital Networks (ISDN). Originally, CLIP is used as an optional feature in ISDN telephone networks. With CLIP, the number of the caller is transmitted to the person being called allowing precise identification of the caller. In case of IP, the ISDN fixed line number cannot be considered as equivalent to the user's IP address, because an IP address may not uniquely define a physical line. Furthermore, IP addresses *do not* provide LI in any case. Whereas fixed lines *do have* a defined and known origin. Therefore, this paper discusses the reuse of selected, principle aspects of the ISDN CLIP function in IP-based packet-switched networks to facilitate enhanced and new services.

With the IPclip mechanism, a customer and his actual geographic location are identified using a tuple, which consists of the current IP address and extra information. While the IP address might identify a user, his position must be part of the additional data. Preferably, a standardized format of LI is used. It can be interpreted for analysis, for classification, for generation of syslog-calls to induce further exceptional actions, or to send help to a person that requires medical assistance in case of VoIP ECs.

Network ingress—also known as access network in a telecommunication environment—is considered to be the most reasonable place in a network for implementation of the IPclip mechanism. Access networks comprise customer premise equipments (CPEs) and access nodes such as DSL access multiplexers (DSLAMs). The latter usually consists of linecards

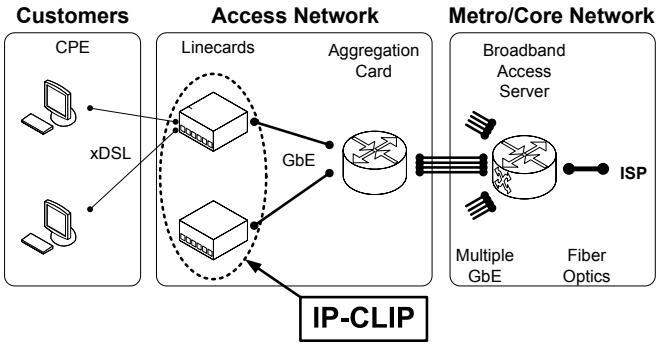

Fig. 1. Access network with IPclip

and aggregation cards as shown in Figure 1. While aggregation cards provide high-bandwidth interfaces towards metro or core networks, linecards aggregate the various subscriber lines. Furthermore, only linecards can provide sublimentary port information. Due to these reasons, IPclip is implemented on the linecards as highlighted in Figure 1. Following, traffic from the CPE towards the core network is referred to as upstream. Traffic towards the CPE is referred to as downstream.

To provide LI on a global scale, IPclip provides three major functionalities.

Provision of LI: In upstream, the IPclip mechanism provides LI for every incoming packet by inserting that information as IP option. Dependent on the application scenario, LI may be inserted only into selected IP packets, e.g., only packets containing VoIP data. The physical location of the linecard in conjunction with the linecard's port and the aggregation card ID are used as LI. IP spans the whole Internet and provides end-to-end connectivity between users. The structure and size of IP options are specified in the protocol standard. Thus, by using IP options, IPclip is a standard-compliant solution to convey extra information. IP options as part of the IP header can have a maximum length of $40\ \mathrm{byte}$ that can contain arbitrary additional information [7]. Thereby, network devices can either process this IP option or ignore it. But in any case, devices must be capable of parsing each option for reasons of interoperability.

Verification of LI: In case upstream IP packets already include an IP option containing LI, the IPclip mechanism is capable of verifying the plausibility of that information. IPclip recognizes user provided IPclip options. But since users may intentionally try to conceal their true location—a CPE is not trustworthy—it needs to be validated. IPclip can identify incorrect LI to a certain degree since only customers geographically near to a specific access node are connected to it. The plausibility of the provided LI is validated by comparing it with the well known LI of the network device, which implements the IPclip functionality. Thereby, incorrect LI—due to mobility or even due to intentional manipulation—can be replaced with true and trustworthy LI of the network device itself. The replacement of incorrect information ensures valid LI at any time. In incoming upstream packets, flags are set as result of the validation procedure. These flags may serve as additional triggers for VoIP ECs, Spam detection as well as

for fighting Phishing threats.

Removal of LI: As an optional feature, IPclip provides the possibility to remove IPclip options from IP packets in downstream. This could be necessary if users should not receive sensitive data like LI about a packets origin.

From a user's perspective, the IPclip mechanism is fully transparent when LI is neither added by nor forwarded to the CPE. Put in a nutshell, constraints and requirements for the operation of IPclip are:

- LI in form of an IP option must be added into every packet.
- If existent, user provided LI must be identified, verified, and validated.
- Optionally, LI can be deleted from packets in downstream direction to not pass it to the CPE.

A. Interaction with Security Architectures

There is no interaction between IPclip and Transport Layer Security and Secure Socket Layer respectively. These systems are independent from each other since these architectures work above layer three. However, IPclip and IPsec do influence each other. If IPsec is used in tunnel mode, the position of the endpoints of IPsec is decisive for compatibility. IPclip options must already be part of the packet, when IPsec headers are created. A subsequent addition or manipulation of an IPclip option is not possible. When using IPsec in transport mode, there are no problems if Encapsulation Security Payload is used as IP header and options are not encrypted. When using Authentication Header (AH), the compatibility depends on which parts of the IP header are part of the encryption. Depending on the relative position of IPclip towards the IPsec endpoints, IP options must not be authenticated by AH.

B. IPclip Option Definition

As mentioned above, the IPclip option is one possible *value* of an IP option. It must not be mixed up with an IP option! Specified by the IP protocol [7], an *IP option* can either consist of one byte that represents the option or several bytes that are structured in the commonly known Type-Length-Value (TLV) format. The type field consists of a copied flag, a 2-bit option class, and a 5-bit option number. There are already several option numbers defined [8]. At the moment, one of the undefined option numbers (26) is chosen as option number for IPclip for the prototypic implementation.

The *IPclip option* (Figure 2) consists of 1 byte for the IPclip type, 4 bit for flags, and LI of variable size. The IPclip option has to consist of an integer number of bytes. In total, the sum of all IP options in the IP header must have a length that equals

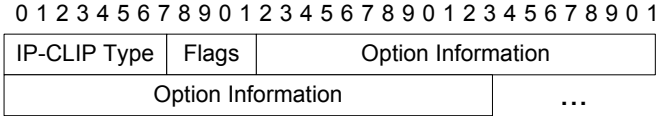

Fig. 2. Format of an IPclip option as value of an IP option

TABLE I
IPCLIP OPTION FLAGS

Value	Source / Authenticity	Option Description
xx00	user provided / untrusted	A user provided IPclip option did not pass verification.
xx01	user provided / trusted	A user provided IPclip option did pass verification.
xx10	network provided / untrusted	A User provided IPclip option did not pass verification and is replaced on the linecard.
xx11	network provided / trusted	A new IPclip option is added on the linecard.

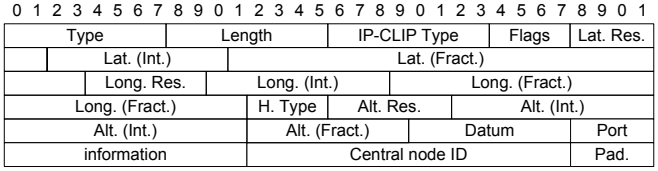

Fig. 3. IPclip option with **GLI** data plus aggregation card and linecard port IDs encapsulated in an IP option

a multiple of four bytes to correspond to the value in the IHL field. Hence, padding may be necessary.

IPclip type: Approved geographic standards, which are well-known in the field of geographic information systems, are used for the IPclip location information. Value 1 denotes Global Positioning System (GPS) location information [9], value 2 denotes Geospatial Location Information (GLI) [10], and value 3 and 4 refer to GPS or GLI plus aggregation card ID and port information. In conjunction with geographic LI, aggregation card ID and linecard port allow for more precise localization of the user. Both have a size of 2 byte. Other values have not been assigned yet.

Flags: 4 bits are used for flags. The first 2 bits are reserved. The remaining 2 bits define source and credibility of the IPclip option as sketched in Table I.

Option Information: In [10], a DHCP option for coordinate-based LI (Geospacial Loacation Information, GLI) is defined. The option comprises information about latitude, longitude, and altitude as can be seen in Table II. A possible encoding is illustrated in Figure 3. The values for the resolution denote the numbers of high-order bits of latitude, longitude, and altitude that should be considered as valid. The height type denotes the altitude to be defined in meters or in floors. Datum denotes the map datum used for the coordinates given in this option. Of the 256 possible values for the datum field, 3 have been registered with IANA. The most common format is World Geodetic System 84 (Geographical 3D) indicated by $datum = 1$. For the complete TLV-structured IP option with GLI data, up to 24 byte are needed. Another format of LI is GPS location information. It is used in the NMEA-0182 data format [9].

III. Hardware Realization

LI has to be added to every packet in the upstream. Due to the high bandwidths of multiple $\mathrm{Gbit/s}$, this would result in tremendous workloads for a CPU when done in software. This would keep network devices from executing their primary software tasks like, e.g., operation administration maintenance (OAM). Furthermore, it can expose vulnerable points in the access network infrastructure. Without flexible, resource-aware, and economical solutions, this is going to be a severe problem in the near future. Mostly, full-blown network processors (NPs) are expensive solutions. It is economically not reasonable to apply NPs on every linecard of an access node to realize the IPclip functionality. Other low-cost hardware solutions like microcontrollers do neither provide the needed functionality nor do they offer sufficient performance margins to cope with growing bandwidths due to technology improvements as for example proposed in [11]. Consequently, a hardware solution is required to fulfill IPclip's tasks. A Field-Programmable Gate Array (FPGA) was chosen as the target platform for IPclip to process traffic in a flexible (reconfigurability), cost-effective (reusability), and high-performance (parallelism) way. Available FPGA technology allows for non-blocking performance and operation at wire speed. Only a negligible delay is inserted into the data path. A hardware prototype has been developed for IPclip on a Xilinx Virtex-4 FX basis. To execute the tasks and meet the constraints mentioned in the last section, the following independent submodules have been designed for IPclip. They are combined in the IPclip prototype, which is flexibly configurable at synthesis time:

- IPoE MTU Adaptation Module (MAM)
- PPPoE MTU Adaptation Module (PAM)
- Packet Classifier (PC)
- Option Verification Module (OVM)
- Additional Information Adder (AIA)
- Additional Information Remover (AIR)

TABLE II
Geospatial Location Information as per [10]

Information	Range	# of bit
Option Information		128
Latitude		45
Resolution	0...34	6
Degree (Integer)	(-90)...(+90)	9
Degree (Fraction)	0...(1-2^{-26})	25
Longitude		45
Resolution	0...34	6
Degree (Integer)	(-180)...(+180)	9
Degree (Fraction)	0...(1-2^{-26})	25
Altitude		40
Height Type	0...2	4
Resolution	0...30	6
Value (Integer)	0...(2^{22}-1)	22
Value (Fraction)	0...(1-2^{-26})	8
Datum	1...3	8

Figure 4 shows the coarse structure of the IPclip system. The functional modules MAM, PAM, PC, OVM, AIA, and optionally AIR are serially connected. MAM carries out the adaptation of the MTU (configurable at runtime) for IPoE. PAM intervenes with the MTU negotiation in case of PPPoE. PC maps port numbers to every IP packet. OVM identifies and verifies user provided LI by comparing it with LI of the linecard. The insertion of LI is done by AIA if the *add*-signal is set. Optionally, AIR can be inserted into the downstream to remove IPclip options from incoming packets.

The interfaces of the main data path in the IPclip system are standardized. It has already been used in other packet processing systems [12], [13]. Frames and packets byte-serially pass through the functional modules. The first byte is signalized with a Start-of-Frame (SOF) flag. The last byte is signalized with End-Of-Frame (EOF). Due to this simple structure and interchangeability between the elements, new functional modules for further tasks can easily be integrated into the data path.

A. MTU Adaptation

Especially for high volume data streams, which already exploit the maximum payload size, the allowed MTU is likely to be exceeded when adding extra information in the magnitude of 15 or $24\,\mathrm{byte}$ as IP option. In this case, the packet has to be either fragmented or discarded. Fragmentation should be avoided since signaling and retransmission deteriorate the performance of the respective communication channel [14]. Firstly, the processing overhead for CPU and memory increases. Secondly, the receiver needs to reserve more memory for fragments, which need to be reassembled. This is of relatively little importance on a host because sufficient time and memory resources are usually available there. However, reassembling fragments within routers is inefficient because routers are primarily intended to forward incoming information as quickly as possible. They are not intended to hold on to packets. Finally, the complete IP packet has to be retransmitted if only one fragment is missing or has errors.

If no fragmentation shall take place, the MTU of all packets must be adapted so that additional information fits into the packets. Regarding the IPclip mechanism, special MTU adaptation components execute these tasks transparently. For IPoE, the MTU adaptation is carried out periodically. That means, adaptation is done whenever a packet is received that exceeds the maximum MTU. For PPPoE, the MTU is adjusted

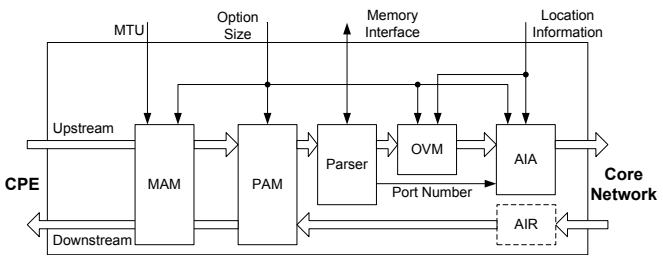

Fig. 4. IPclip Hardware Prototype - Architecture

during a negotiation phase. Principles for both MTU adaptation schemes are examined below.

IPoE MTU Adaptation: In IP-based networks, the so-called Path MTU Discovery (PMTUD) is used for dynamic adaptation of the packet's MTU to the smallest MTU in the data path [15]. If PMTUD is supported by a host, the Don't Fragment (DF) bit of the IP header is set. That is, no fragmentation of packets exceeding the MTU shall take place. If such a packet reaches a router, which cannot forward the packet due to a smaller own MTU, the packet is discarded as the DF flag is set. Then, an Internet Control Message Protocol (ICMP) message is sent to the packet's source. This ICMP message passes information about the size of the allowed MTU to the sending host. On reception, this host will store the updated MTU and send subsequent packets with the appropriate payload size.

ICMP is encapsulated in IP and has a 4-byte header and an optional data field [15]. If an incoming packet in the upstream would become bigger than the allowed MTU after the insertion of the IP option, an ICMP packet is generated by MAM. The size of this option (IP_OPTION_SIZE) is an integer number of bytes, which equals the length of the IP option to be added. It is supplemented to the next multiple of 4 because the size of the IP header is fixed to multiple of 4 byte by the IHL field (Internet Header Length). Thus, only packets complying with the path MTU after insertion of the IP option are sent to the upstream.

PPPoE MTU Adaptation: PPPoE describes how a PPP session over Ethernet is established and how PPP packets are encapsulated in Ethernet frames [16], [17]. A PPPoE connection is negotiated between two communicating entities. The flow of the PPPoE protocol falls into three phases: PPPoE discovery phase, PPP session phase, and PPPoE termination. During the PPP session phase, the MTU is negotiated as part of the link establishment using Link Control Protocol (LCP) configure packets. Here, PAM has to intervene by changing the size of the MTU in the respective LCP packets. As part of the LCP header, the code field defines an LCP Configure Request (0x01), a Configure Acknowledge (0x02), or a Configure Not Acknowledge (0x03). With the type set to 0x01, an LCP Configure Request indicates MTU negotiation. In these packets, the MTU value has to be changed.

Figure 5 depicts the signaling between the client's host and the broadband access server of the respective ISP with interposed PAM functionality. For MTU negotiation, a client sends an LCP Configure Request, which proposes a certain MTU value (MTU_{client}). PAM has to increase that value by IP_OPTION_SIZE because the size of each packet will increase by the size of the IP option to be inserted. If necessary padding the IP option to a multiple of 4 bytes is possible. The broadband access server receives the LCP Configure Request with $MTU_{client} + IP\_OPTION\_SIZE$ and responds with an LCP Configure Acknowledge. This response contains the confirmed value $MTU_{client} + IP\_OPTION\_SIZE$, which has to be scaled by PAM to MTU_{client} and is forwarded to the client (see Figure 5a). Then, the client's host will send packets with a size of MTU_{client}.

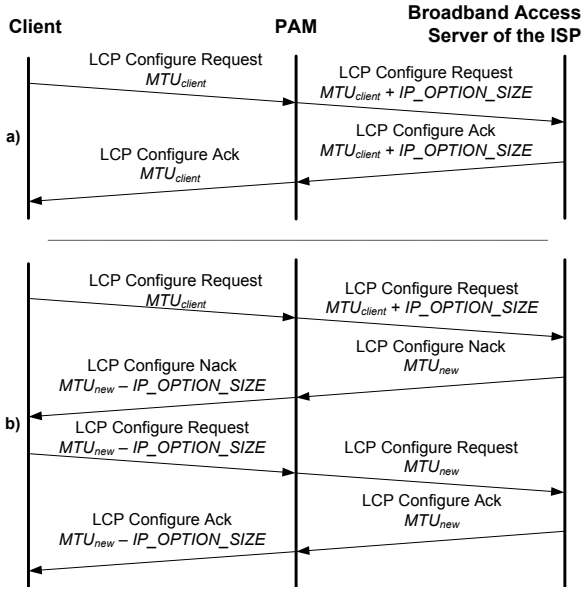

Fig. 5. Sequence of control messages with interposed PPPoE MTU Adaptation Module (PAM)

If the broadband access server does not agree with the suggested MTU_{client}, it sends an LCP Configure Not Acknowledge together with its own suggestion for a different MTU value (MTU_{new}). This value has to be modified to $MTU_{new} - IP\_OPTION\_SIZE$ by PAM. Having received this LCP Configure Not Acknowledge, a LCP Configure Request is sent again by the client, which contains $MTU_{new} - IP\_OPTION\_SIZE$. PAM updates this value to MTU_{new}, which the broadband access server receives and acknowledges. The message flow is drawn in Figure 5b. As a result, the client will send packets with a maximal size of $MTU_{new} - IP\_OPTION\_SIZE$.

B. Packet Classifier

In case LI with additional Access Node ID and port information is required, a PC is part of IPclip's architecture. PC's task is the classification of each incoming IP packet. For every packet the corresponding host is identified in order to assign the correct port number as part of the LI. To perform the task, the PC parses each packet for a key. That key is sent to a search engine, which searches a memory for the corresponding port information. When the information is found, both port number and IP packet are presented to the AIA module, which adds the LI including port number to the packet.

The PC is capable of parsing different fields of incoming packets. Possible fields are source and destination MAC address, VLAN tags, Ethertype, source and destination IP address, and the DSCP field in the IP header. Any combination of these fields can be configured at compile time. For the actual IPclip functionality hosts are identified by source IP and VLAN tag.

The memory, where keys and port numbers are stored can be implemented either with external memory or with the FPGA's internal Block RAMs. That depends on the size and number of necessary entries in the memory. Each memory entry consists

of the key (SRC-IP+VLAN—48 bit) and the corresponding port information (16 bit). Thus, 8 Byte are required for each entry. All entries are stored in a sorted manner in the memory. That reduces the complexity of finding a memory entry to a binary search $(O\,(log\,(N)))$. The implemented search algorithm cannot analyse a memory entry in each clock cycle. In fact two clock cycles are required. After setting the read address of a certain memory entry, it takes one wait cycle until the memory outputs are valid. Then a comparator can decide if an entry is the correct one or the upper respectively lower half of the unsearched memory has to be searched further. To increase look up speed a prediction is executed. After setting a read address, the next probable memory address is assigned in the following clock cycle. According to the search algorithm, the following address in the upper half of the unsearched memory is selected as next address. In half of the cases, this assumption is correct. This way, in average 1.5 instead of 2 clock cycles per memory lookup are necessary.

Sorting the memory entries of course has the disadvantage of increased time consumption for insertion and deletion. Both operations have a time complexity of $O\,(N)$. In average, $\frac{N}{2} + 2 \cdot ld\,(N-1)$ memory accesses are required. As changes in the look up table occur quite seldom in access network environment, these additional costs are acceptable when increasing lookup speed from $O\,(N)$ to $O\,(log\,(N))$.

C. IPclip Option Verification

The IPclip Option Verification Module (OVM) identifies user provided LI and verifies it by checking if that LI is within a pre-defined subscriber catchment area (SCA). For defining the SCA of the linecard, a logical rectangle is spanned around it. The edge length of the rectangle is configurable and specified in meter $[\mathrm{m}]$. It depends on the desired precision and the spatial dimension of the SCA. The linecard is located in the rectangle's center. The linecard's exact position is defined by the "LI" input. It can be either configured or taken from a connected GPS receiver. Both possibilities are implemented. The *Option size* input defines the length of the configured LI. To configure the location of the linecard, GPS LI or GLI can be used. In either case, the input data is converted to GLI as shown in Figure 6.

Each incoming packet is inspected by OVM for a present IPclip option. If there is no IPclip option, the packet is simply forwarded to AIA. If there is an IPclip option containing LI

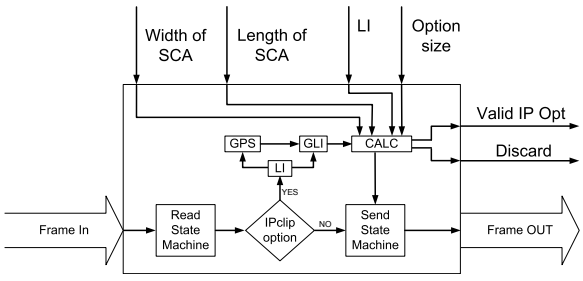

Fig. 6. Location verification with OVM

(which can be GPS LI or GLI), the plausibility if verified. All calculations performed by OVM are based on GLI. Therefore, GPS LI must be converted to GLI whereas the altitude is assumed to be zero. For the calculation, geographic units are used given in degree $[^\circ]$, minutes $[']$, seconds $['']$ and angular milliseconds $[ams]$ for latitude and longitude. If the desired precision is $1\ \mathrm{m}$, the edge length has to be converted to ams. Thereby, the conversion factor between linear and angular measurement for the latitude is globally constant whereas the conversion factor for the longitude depends on the latitude. For example, an alteration of the longitude of about $1''$ in the polar regions ($\pm 89^\circ$latitude) matches a linear distance of approximately $0.54\ \mathrm{m}$. But the same alteration of the longitude in the equatorial region ($\pm\ 1^\circ$) makes for approximately $31.0\ \mathrm{m}$. This relation will be considered by OVM.

Depending on the geographic position, the input values of '*Width of SCA* and *Length of SCA* are calculated in ams. This guarantees the highest resolution and smallest calculation error. In the next step, OVM converts the configured LI data $(Conf\_Lat, Conf\_Lon)$ and the LI from the incoming packet $(User\_Lat, User\_Lon)$ to ams. Now, all values for the calculation are given in ams. For the comparison whether the LI data from the incoming packet are within the SCA, two results are required—one for latitude and one for longitude. The results $(\Delta Lat, \Delta Lon)$ are calculated as shown in Formulas 1 and 2:

$$\Delta Lat = |Conf\_Lat - User\_Lat| \tag{1}$$

$$\Delta Lon = |Conf\_Lon - User\_Lon| \tag{2}$$

ΔLat and ΔLon are compared to SCA border values (ΔSCA_Lat, ΔSCA_Lon) as shown in Formula 3 and 4:

$$Lat\_Res = \begin{cases} true & \text{for} \quad \Delta Lat \leq \frac{\Delta SCA\_Lat}{2} \\ false & \text{for} \quad otherwise \end{cases} \tag{3}$$

$$Lon\_Res = \begin{cases} true & \text{for} \quad \Delta Lon \leq \frac{\Delta SCA\_Lon}{2} \\ false & \text{for} \quad otherwise \end{cases} \tag{4}$$

Only if Lat_Res and Lon_Res are true, the user provided LI is valid. In this case, the *Valid IP Opt* signal takes the value of the option counter to indicate valid LI to AIA. If one of the calculated results is false, *Valid IP Opt* keeps the old value, e.g., zero. If the IP header is analyzed completely, the packet is sent to AIA. If no valid LI is found or only invalid LI is found, *Valid IP Opt* remains at zero. In this case, AIA adds the linecard's LI as IPclip option if there is enough space in the IP header. If there is not enough space, the *Discard* signal is set to one and the whole packet will be discarded by AIA. While sending the IP packet, OVM takes care of sending the right IPclip option Flags. If at least one valid IPclip option was found, the flags are set to "01" (user provided / trusted). Otherwise, the flags are set to "00" (user provided / untrusted) as shown in Table I.

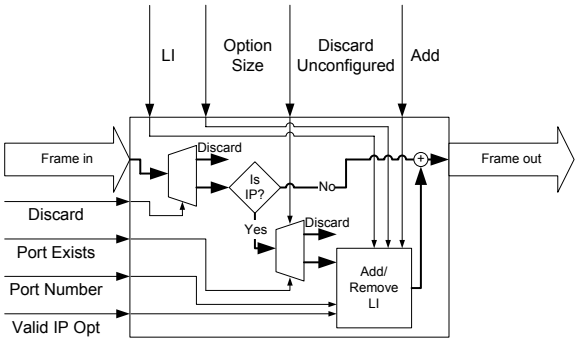

Fig. 7. Additional Information Adder (AIA)

D. Additional Information Adder and Remover

AIA's task is to provide IPclip options, i.e. LI for every IP packet in the upstream. This is done as long as there is enough space in the IP options field of the IP header and no valid user provided IPclip already exists. Optionally, just Access Node ID and port numbers can be added in addition to the existent LI. This information allows for a more precise localization of users.

In Figure 7, a block diagram for AIA is depicted. When AIA receives a packet, it checks if the packet contains IP. If it does not, the packet is forwarded without any modification. For every packet the *Discard* signal, coming from the OVM, is checked. If *Discard* = '1', a packet contains IP but does not have sufficient space in the options field to insert an IPclip option. Suchlike IP packets shall be dropped. That is required to be consistent on the one hand. On the other hand a self-contained and terminated IPclip domain must be assured. Beyond, ISPs, access providers, or carriers may apply a tailored policy.

If *Discard* = '0', *Port Exists* (set by PC) and *Discard Unconfigured* (configurable at run time) are evaluated. If *Port Exists* = '0', i.e. no port number could be assigned for an incoming packet by the classifier, and *Discard Unconfigured* = '1', the IP packet is discarded. By setting *Discard Unconfigured* to '0', suchlike IP packets are not discarded. In this case, the port number in the IPclip option is set to a default value, e.g. 0xFFFF.

After passing the previous checks, the *Add* input (configurable at run time) is checked. If *Add* = '0', no additional information is added and the IP packet is forwarded. If *Add* = '1', AIA adds an IPclip to the IP packet. Therefore new values for IHL, Total Length, and Checksum fields of the IP header are calculated. That is because those fields change due to an added IP option. In case one or more IPclip options already exist, OVM checks, which of them is contains LI and counts the number of IPclip options with LI. (The default case is that only one user provided IPclip option contains LI.) If one LI is valid, *Valid IP Opt* denotes that LI (e.g. if there are two IPclip options with LI and the second is valid, then *Valid IP Opt* = 2). If there is no user provided IPclip option with LI at all or no valid user provided IPclip option with LI, then *Valid IP Opt* = 0. Basically, if user provided LI is existent in an IP packet, there are four possibilities:

- One IPclip option with LI and successful validation: The packet is left unchanged. In that case, either no further information is added, or Access Node ID and port number are added if configured.
- One IPclip option with LI and unsuccessful validation: LI it is removed and, if *Add* = '1', replaced by AIA with a distinct and trustworthy LI, which is hard-coded in the linecard.
- More than one user provided IPclip option with LI in a packet and (at least) one is valid: *Valid IP Opt* denotes that IPclip option, the last valid one is left unchanged and every other IPclip option with LI is removed. Optionally, Access Node ID and port number can also added as well.
- More than one user provided IPclip option with LI in a packet and none is valid: Each and every IPclip option with LI is removed. If *Add* = '1', a distinct and trustworthy LI, optionally with Access Node ID and port number is inserted.

IP packets in downstream can be left unchanged or all IPclip options containing LI are removed. The behaviour is configurable. Removing IPclip options is done by the AIR submodule. It assures that the striped IP packets contain valid IHL, Total Length, and Checksum fields after removal.

TABLE III
RESOURCE CONSUMPTION OF THE IPCLIP PROTOTYPE

Module	Slices	BRAMs
MAM	786	1
PAM	163	0
PC	832	11
AIA	1019	4
OVM	2491	2
AIR	519	6
EMAC + glue + prototyp related	1700	31
IPclip prototype	7486	55

IV. PROTOTYPE

A hardware prototype has been developed for IPclip on a Xilinx Virtex-4 ML405 FPGA development board [18]. The whole architecture presented in Section III has been fully implemented. The prototype system requires 7486 slices of logic and 55 BRAMs. The used Virtex4-FX20 FPGA is thus utilized fairly.

To evaluate the performance of the system with a $1\ \mathrm{Gbit/s}$ data path the worst case scenario was implemented. Different sized Ethernet frames containing IP packets were induced with 100% data rate (minimal inter frame gap). Data was sent both with and without IP options containing LI. As derives from Figure 8, with minimal frames the loss rate of the system reaches 25% when LI has to be inserted in each frame. That cannot be avoided as the size of each frame increases in size by $25\ \mathrm{byte}$. When sending realistic frames (35% $64\ \mathrm{byte}$, 11% $594\ \mathrm{byte}$, 10% $1518\ \mathrm{byte}$, 44% random) the loss rate is reduced to 6% without LI. However, when the frames already contain LI, there is no frame loss at all. In average, IPclip

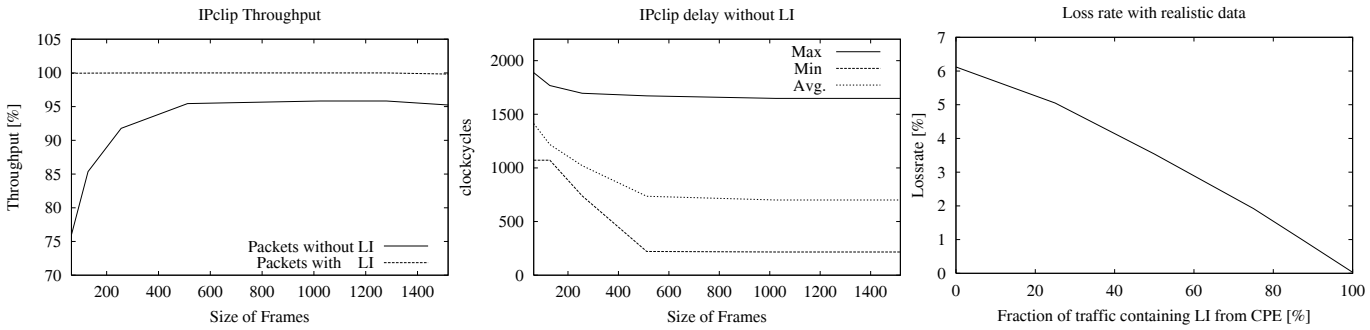

Fig. 8. Behaviour of IPclip with a 1 $\mathrm{Gbit/s}$ data path

hardware system introduces a delay of 700 clock cycles into the data path, when there is already LI. That equals 5.6 µs. As the system cannot compute all data frames when no LI exists, the delay is higher because of the internal buffers, which are filled to the maximum. In that case the delay reaches up to 1900 clock cycles. That equals 15.2 µs, which is still fast. When simulating traffic with 50% LI the average delay is at about 5.5 µs.

V. Conclusion

The paper proposed and discussed the architecture of the new IPclip mechanism. With IPclip it is now feasible to identify a physical line in IP-based packet-switched networks. A broad range of different services and security mechanisms can be derived from the general IPclip functionality.

For example, IPclip enables VoIP emergency calls by inserting distinct location information into every IP packet using standard-compliant IP options. Emergency calls in nomadic VoIP environments are thus possible. The user is no longer required to update his current location. Due to the availability of trustable location information, VoIP emergency calls can be redirected to the correct, responsible Public Safety Answering Points. Users can be reliably located wherever they are. Furthermore, IPclip also serves many other scenarios like spam identification and tracking.

A prototypic IPclip hardware implementation was presented. It is located on the linecards and processes IP traffic with wire speed. Individual submodules have been designed for the main tasks, which are LI verification, IP option insertion, MTU adaptation of the respective communication channel, and optionally LI removal. Due to the fact that the IPclip system is designed for reconfigurable silicon devices, the functional spectrum can be adapted to future needs of the fast-living networking domain.

Ongoing and future work covers research and adaptation of IPclip on further application scenarios and the migration of the mechanism to IPv6.

References

[1] Newport Networks Ltd., "Emergency Call Handling in VoIP Networks," White Paper, 2006. [Online]. Available: http://www.newport-networks.com/cust-docs/89-ECH.pdf

[2] B. Rosen and J. Polk, "Best Current Practice for Communications Services in support of Emergency Calling," Internet draft, June 2006. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-ecrit-phonebcp-01.txt

[3] S. Kubisch, H. Widiger, P. Danielis, J. Schulz, D. Timmermann, T. Bahls, and D. Duchow, "Trust-by-wire in packet-switched ip networks: Calling line identification presentation for ip," in *1st ITU-T Kaleidoscope Conference: Innovations in Next Generation Networks - Future Network and Services*, 2008, accepted.

[4] Symantec, "The State of Spam – A Monthly Report," May 2007. [Online]. Available: http://www.symantec.com

[5] S. Kubisch, H. Widiger, P. Danielis, J. Schulz, D. Timmermann, T. Bahls, and D. Duchow, "Complementing e-mails with distinct, geographic location information in packet-switched ip networks," in *MIT 2008 Spam Conference*, march 2008.

[6] ——, "Countering phishing threats with trust-by-wire in packet-switched ip networks - a conceptual framework," in *22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS), 4th International Workshop on Security in Systems and Networks (SSN 2008)*, april 2008, accepted.

[7] Information Sciences Institute University of Southern California, "Internet Protocol specification," RFC 791, September 1981. [Online]. Available: http://www.ietf.org/rfc/rfc0791

[8] IANA Home Page, "IP OPTION NUMBERS," February 2007. [Online]. Available: http://www.iana.org/assignments/ip-parameters

[9] National Marine Electronics Association (NMEA), "NMEA 0183 Standard," January 2002. [Online]. Available: http://www.nmea.de/nmea0183datensaetze.html

[10] J. Polk, J. Schnizlein, and M. Lisner, "Dynamic Host Configuration Protocol Option for Coordiante-based Location Configuration Information," RFC 3825, July 2004. [Online]. Available: http://www.apps.ietf.org/rfc/rfc3825.html

[11] J. Cioffi, "Vectored DSLs with DSM: the road to ubiquitous gigabit DSLs," in *Proceedings of the World Telecommunications Congress 2006 (WTC06)*, Budapest, Hungary, April 30 - May 3 2006.

[12] H. Widiger, S. Kubisch, D. Timmermann, and T. Bahls, "An integrated Hardware Solution for MAT, MPLS-UNI, and TM in Access Networks," in *Proceedings of the 31st Annual IEEE Conference on Local ComputerNetworks (LCN)*, Tampa, FL, USA, November 14-16 2006.

[13] H. Widiger, S. Kubisch, and D. Timmermann, "A Structural Architecture for HW Packet Processing," in *Proceedings of the 11th IEEE Pacific Rim Conference on Communications,Computers and Signal Processing (PACRIM)*, Victoria, B.C., Canada, August 22-24 2007.

[14] Cisco Systems, "Resolve IP fragmentation, MTU, MSS, and PMTUD issues with GRE and IPSEC," White Paper, October 2006. [Online]. Available: http://www.cisco.com/warp/public/105/pmtud_ipfrag.pdf

[15] J. Mogul and S. Deering, "Path MTU discovery," RFC 1191, November 1990. [Online]. Available: http://www.ietf.org/rfc/rfc1191

[16] W. Simpson, "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994. [Online]. Available: http://www.ietf.org/rfc/rfc1661

[17] L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, and R.Wheeler, "A method for transmitting PPP over Ethernet (PPPoE)," RFC 2516, February 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2516

[18] Xilinx, Inc., "ML405 Evaluation Platform." [Online]. Available: http://www.xilinx.com/products/boards/ml405/docs.htm