# Secure Access Node: an FPGA-based Security Architecture for Access Networks

Jens Rohrbeck, Vlado Altmann, Stefan Pfeiffer, Dirk Timmermann
*University of Rostock*
*Institute of Applied Microelectronics and Computer Engineering*
*Rostock, Germany*
{*jens.rohrbeck;va031;sp385;dirk.timmermann*}*@uni-rostock.de*

Matthias Ninnemann, Maik Rönnau
*Nokia Siemens Networks GmbH & Co. KG,*
*Broadband Access Division*
*Greifswald, Germany*
{*matthias.ninnemann;maik.ronnau*}*@nsn.com*

*Abstract*—**Providing network security is one of the most important tasks in today's Internet. Unfortunately, many users are not able to protect themselves and their networks. Therefore, we present a novel security concept to protect users by providing security measures at the Internet Service Provider (ISP) level. Already now, ISP are using different security measures, e.g. Virtual Local Area Network tags, MAC limitation, or MAC address translation. Our approach extends these security measures by a packet filter firewall and a deep packet inspection engine. A firewall and a deep packet inspection system, at the ingress of the network, offers security measures to all connected users, especially to users with limited IT expert knowledge. Adjustments can be made only by the ISP administrator. Consequently, our security system itself is secured against attacks from users and from the network side. Our approach includes a powerful Packet Classification Engine, a high speed Rule Set Engine without using Content Addressable Memory and control stages in reconfigurable hardware. Our goal is to be able to control network traffic at wire speed.**

*Keywords*-**Access Network, Hardware Firewall, Intrusion Detection, Web Filter**

## I. Introduction

Firewalls and anti-virus programs provide basic protection for Internet-enabled devices. Normally, these security measures are installed on computers of users. But installing security measures at the users' side has two serious drawbacks. Firstly, the threat detection is done on the target machine. Secondly, the users must install, upgrade, and maintain these security measures without professional support. Other measures such as a Web filter and a deep packet inspection engine like snort are often not installed and require additional maintenance. In addition, the majority of Internet users is missing the necessary expertise to configure their security software so that it provides optimal protection. Furthermore, because of negative experiences like phishing attacks targeting online banking, many users have lost their confidence in online services and the Internet itself. Therefore, it is mandatory to disburden respectively to support users in issues of Internet security.

A trustworthy place for the placement of security measures is the ingress of the network — the access network. Each user, referred to as subscriber by Internet Service Providers (ISPs), is connected to the Internet through the access network. The access network itself consists of access nodes (AN).As ANs are transparent for subscribers, these components are safe from, e.g., Denial of Service Attacks. To reestablish the subscribers' confidence into the Internet and moreover, to even protect the Internet itself, it is useful to establish additional security services at ANs. With these additional security features, two objectives can be achieved. On the one hand, the subscriber is offered a higher security service without the need to care about security measures himself. On the other hand, outgoing traffic from subscribers can be verified. Thus, the network is protected as well.

However, although an ISP can take up new security measures in its portfolio, various challenges have to be addressed. On an AN, higher traffic rates (e.g., 1 Gbit/s or higher) have to be processed than in single Internet connections — both in up- and downstream. Furthermore, rules for up to 32k connections should be supported. Due to hardware restrictions, we dispense with connection tracking and the control of protocols' communication sequences. Our approach referred to as Secure Access Node (SecAN) extends the currently available security measures on an AN by a packet filtering firewall, Web filtering, and intrusion detection system. Thereby, these functionality moves from the subscriber to the ISP.

To fulfill these tasks under the conditions described, a very powerful packet classification [1] and packet processing are required. Due to these requirements, pure software solutions are not applicable. Therefore, we use a hardware/software solution on a XILINX evaluation board with a FX70T Field Programmable Gate Array (FPGA). In our solution, we do not use CAM memory. Already for 224 connections (these approximates ca. 0.7% of all connections), over 90% of available block ram ressources or 23% of slice register would be needed. Without using CAM, our solution is able to control traffic at wire speed. Briefly summarized, the main contributions of this paper are the following:

- We present a novel hardware/software approach of a packet filter, Web filter, and an intrusion detection system placed onto an access network.
- Our solution is able to control traffic in up- and downstream direction simultaneously. Thus, we can protect the connected subscribers and the network itself.

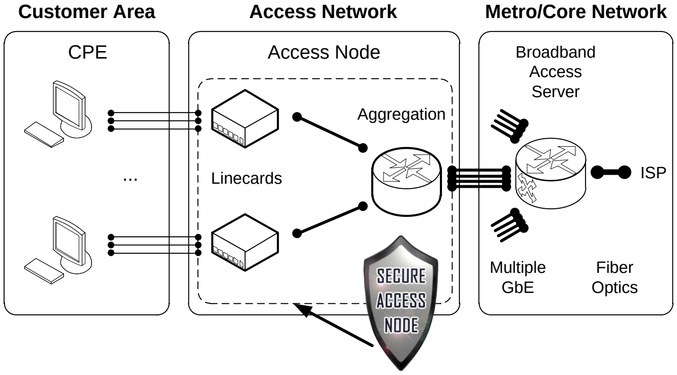

Fig. 1. Access Network containing ANs. The Secure Access Node is an extension of an AN.

- We aspire an individual classification for each connection, e.g., we do not want to limit the number of configurable rules as described in [1].
- As target platform, we use a XILINX evaluation board with an FX70T FPGA. As we do not using CAM, we are able to control traffic with 2 Gbit/s in wire speed, e.g., we control each packet without packet loss.

The remainder of this paper is organized as follows: Section II describes security measures available in the access network today. In Section III, our hardware design is presented. Here the various modules and their functions are explained. Before the paper concludes in Section V, we introduce our software solution for flexible configuration in Section IV.

## II. Security Measures In The Access Network

Each subscriber achieves access to the Internet through the access network. Access networks comprise subscriber premise equipments (CPE) and access nodes such as DSLAMs. The latter usually consists of linecards 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.

Although the network ingress is transparent to the traffic from and to subscribers, ISPs have to protect the access network. Today, security measures mainly include passive measures on OSI layers II and III [2], [3]. For example, ISPs are using security measures like:

- Port isolation - subscriber may not communicate via an AN
- MAC antispoofing - a Source MAC address is allowed only at one port at a time
- MAC address limitation - to limit the number of MAC addresses per port
- MAC address translation - subscribers MAC address is translated to an ISP MAC address
- VLAN tags - to separate subscriber and services
- IP antispoofing - only the IP address - assigned by the ISP - in combination with the requested MAC address, is allowed pair Source IP and Source MAC at a special port

To ensure a minimum necessary level of security when connecting to the Internet, the already introduced security measures must be integrated into the access area by means of the Secure Access Node.

## III. SecAN - Architecture

### A. Hardware Overview

To emulate the SecAN on an AN, we use the XILINX ML507 evaluation board with an FX70T FPGA [4]. Thereby the FPGA is the main component. We also utilize the 1MB Static Random Access Memory (SRAM) and the 512MB large Double Data Rate Synchronous Dynamic Random Access Memory (DDR2-SDRAM). To control traffic in upstream and downstream direction, we use two 1 Gigabit Ethernet transceivers.

### B. The System In General

Each Ethernet transceiver of the evaluation board is able to process data with 1 GBit/s. If we want to process all data from both directions, we have to process 16 bits/cycle. To avoid the discarding of any uncontrolled data frame and due to the internal delay during frame processing, we have decided to increase the internal bandwidth to 32 Bit/cycle.

The basic components of the SecAN system are the packet classification engine (PCE), rule set engine (RSE), and packet processing engine (PPE) (see Figure 2).

### C. The Frame Configuration and Processing In General

- Before the system can process traffic, it must be configured. The components that need to be configured are the PCE, RSE, Web filter, and the DPI control stage. All configuration data is solely written to the hardware and read from it by the ISP. The configuration flow is shown by dashed arrows in Figure 2.
- After configuration, frames reach the inner system. The frame multiplexer chooses the next frame to be processed by the PCE. The PCE separates flow data from the frame and requests the individual rule set from the RSE. The rule set is an individual collection of rules, which are necessary to evaluate a frame. After identifying the right rule set, it has to be forwarded to the PCE. If the rule set reaches the PCE, the rule set, the data frame, and collected frame parameters have to be sent in the direction of the PPE - to the control stages. In the control stages, the rules from the rule set are applied. The control stages are able to discard or forward frames or replace frame values like IP addresses. If a frame is not discarded, it leaves the PPE and is forwarded to the right output interface.

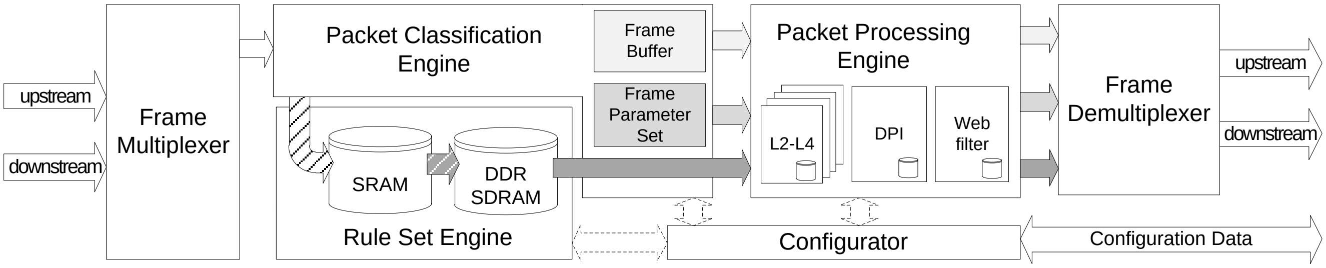

Fig. 2. Block diagram of the Secure Access Node

### *D. Configuration Of The Hardware*

Before the hardware components are not configured, no frames traverse the SecAN. During configuration, all internal processes are stopped. Configuration data is provided to the appropriate modules by the configurator. This data has a type-length-value layout.

- Type is an 8 bit field and determines the component of the hardware to be addressed. Each component has two valid type values: one for writing configuration data and one for reading configured data.
- Length is an 8 bit field and represents the number of configuration data bytes. A maximum of 256 bytes plus configuration header can be configured.
- The actual configuration data is contained in the value part. All components are assigned specific configuration values.

### *E. The Frame Processing Flow*

The frame processing flow is shown by shaded arrows in Figure 2. If the PCE is not busy, it has to receive and classify the frame. The frame multiplexer selects a frame from the internal buffer with the highest fill level. After frame classification is finished, the RSE searches an individual rule set for each frame. Rules of the rule set are applied to the frame. If the frame is not discarded by the PPE due to the rules, it is sent to the correct output interface by the frame demultiplexer.

*1) Packet Classification Engine:* Often packets are classified by five packet header fields: Both IP addresses and port numbers, and transport layer protocol [5], [6], [7]. Upon agreement with our co-operation partner, which develops end products for ISPs, we want to support a higher degree of flexibility. Thus, we extend this set by both MAC addresses, up to 2 VLAN tags, and the Ether type field to a set of 10 frame parameters. During the configuration phase, the PCE has received two so-called flow id triggers (up-/down stream trigger). These triggers describe, which of the 10 frame parameters are necessary to classify a frame. It is possible to set a new trigger by reconfiguration on the fly.

In addition to the frame, the frame multiplexer delivers information of the receiving direction of frames. Depending on the receiving direction, the flow id trigger for upstream or downstream is selected. After this, the corresponding frame parameters are combined to a unique flow id, which identifies the frame bijectively. Furthermore, all described frame parameter are extracted and stored, and the receiving frame is buffered.

During composing the flow id, we calculate an address for the rule set by CRC32 on the fly. Similar projects like [5], [6], [8] have very short flow IDs and use CAM or bloom filter approaches to increase the lookup performance as suggested in [1]. A CAM would require a disproportionate number of hardware resources and a bloom filter approach is not able to calculate an address for a rule set. We avoid these both solutions and use a two-stage approach in the RSE.

After the flow id is completely composed, a request for the individual rule set is performed by the RSE. If one of the frame parameter is not available in the frame, the flow id is not fully completed. In this case, a standard rule set is requested. If the individual rule set is received from the RSE, the frame, rule set, and parameter set is sent towards the PPE. Because only the parameter set is available in the PPE with the first cycle, a comparison with rule parameter can be done before the proper frame data reaches the PPE.

*2) Rule Set Engine:* For the Rule Set search, we use a two-stage approach with a hardware-gentle compression method. First, the mapping between flow id and the rule set is done in a sufficiently large SRAM memory. Second, the very large rule sets have to be stored in DDR2-SDRAM. To increase speed when reading and writing memory information, we use self-developed memory controllers. The rule set is sent towards the PCE and forwarded together with the frame and frame parameter set to the PPE.

*3) Packet Processing Engine:* The PPE is responsible for control and evaluation of the data stream and consists of three central components. In addition to a classic packet filtering, we have implemented a signature recognition and a Web filter. Each of the three components aims at protecting subscribers from unauthorized access from the network side and suppresses attacks from subscribers on the network.

**Packet Filtering**

The packet filter is divided into several control stages (CS). It controls and evaluates Ethernet frames on OSI layer 2 until

4. Therefore, CS use the first rule in the rule set. Each rule has a type-length-value layout similar to configuration data. If the type of the rule is unequal to the CS' ID, the frame, rule set, and parameter set are forwarded to the next stage. Otherwise, the rule is processed by the CS and removed from the rule set. So, the next CS is able to look at the first position of the rule set. Each CS compares the data from the rule with the data of the parameter set. Because the whole parameter set is available in the first cycle of new data, the lookup increases the processing speed, especially for OSI layer 4 values. In case of a match, the rule action has to be executed. That is, the frame, rule set, and parameter set can be discarded or forwarded. Thereby, it is possible to suppress, e.g., local network shares for the Internet.

**Signature Recognition**

The signature detection starts after the header of the transport layer. As a basis, we use the snort database. Additionally, we support ISP administrator defined rules. To realize a high speed signature detection, we use bloom filters [1]. Bloom filter are space-efficient probabilistic data structures, which allow for the detection of special signatures in a set of known signatures in a very short time.

**Web Filtering**

Web filters are a very sensitive issue and have been poorly discussed in the research community. Some countries such as China, the United States, and Great Britain [9] already use Web filtering. These Web filters use external services to compare tagged domain names like the database from Internet Watch Foundation. Our developed solution works in 2 steps. First, we hashes detected domains by CRC64 and search the hash value in a preconfigured binary tree. Second, we verify matches by the onboard DDR2 memory. Thereby, we are able to control traffic in wire speed.

Matches by bloom filters or the web filter can be false positives due to different domains resulting in the same hash value. In the improbable case of a false positive (0.001 %), a match analyzer verifies the possible match at wire speed.

## IV. Configuration Software

Via a web interface, customers can set their own filtering rules. Before these rules are applied, they are verified by the ISP. The configuration of the hardware is done by platform independent software developed with QT. The graphical user interface (GUI) consists of a framework, which is able to include so called plugins. Each plugin offers a GUI to configure a separate hardware component of the Secure Access Node. When starting the GUI, the software searches in a special directory for available plugins. All plugins are loaded and appear in the software as a tab. By means of the plugins, ISP provided rule can be generated and customer rules are applied. Furthermore, the configuration software is able to interrupt the hardware processing flow for updating the hardware configuration.

## V. Conclusion

Because many subscribers do not have the necessary knowledge to maintain their own security measures, it is important to include security features at the ingress of the network. Therefore, we have designed a software/hardware co-design consisting of a packet filter firewall, a signature detection, and a Web filter module. The implementation results show a reachable speed of 142.9 MHz corresponding to 4.57 GBit/s. Furthermore, subscribers are protected by the Secure Access Node and do not need to care about their own security. Especially for the large number of customers with minor technical knowledge, this is an important feature. Because of the applied methods, the bandwidth of customers is not influenced. Furthermore, no attacker has access to the hardware. Only an ISP administrator is able to update the security mechanism. Moreover, it is possible to update the system during operation. Prospectively, a functional test with real traffic data is intented.

## Acknowledgment

We would like to thank the Broadband Access Division of Nokia Siemens Networks in Greifswald, Germany for their inspiration and continued support in this project. This work is partly granted by Nokia Siemens Networks.

## References

[1] D. Taylor and J. Turner, "Scalable packet classification using distributed crossproducting of field labels," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, pp. 269–280, 2004.

[2] A. Guruprasad, P. Pandey, and B. Prashant, "Security features in ethernet switches for access networks, TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region ," pp. 1211–1214, 2003.

[3] N. S. Networks, "SURPASS hiX 5622/25/30/35 R3.7M System Description IP-DSLAM," *System*, no. 3.

[4] XILINX, "Platform User Guide," *Evaluation*, vol. 347, pp. 1–60, 2009.

[5] G. S. Jedhe, A. Ramamoorthy, and K. Varghese, "A Scalable High Throughput Firewall in FPGA," *16th International Symposium on Field-Programmable Custom Computing Machines*, pp. 43–52, Apr. 2008.

[6] W. Jiang and V. K. Prasanna, "A FPGA-based Parallel Architecture for Scalable High-Speed Packet Classification," *20th IEEE International Conference on Application-specific Systems, Architectures and Processors*, pp. 24–31, Jul. 2009.

[7] M. Dixit, B. V. Barbadekar, and A. B. Barbadekar, "Packet classification algorithms," *IEEE International Symposium on Industrial Electronics*, no. ISlE, pp. 1407–1412, Jul. 2009.

[8] A. Kayssi, L. Harik, R. Ferzli, and M. Fawaz, "FPGA-based Internet protocol firewall chip," *ICECS*, pp. 316–319, 2000.

[9] R. Clayton, "Anonymity and traceability in cyberspace," *ACM SIGACT News*, vol. 36, no. 653, pp. 115–148, Nov. 2005.