WIRESPEED MAC ADDRESS TRANSLATION AND TRAFFIC MANAGEMENT IN ACCESS NETWORKS

Stephan Kubisch[+], Harald Widiger[+], Daniel Duchow[+], Thomas Bahls[*], Dirk Timmermann[+]

[+] University of Rostock, Institute of Applied Microelectronics and Computer Engineering
18051, Rostock, Germany, Tel./Fax: ++49 (0)381 498 - 7271/7252
{stephan.kubisch,harald.widiger,daniel.duchow,dirk.timmermann}@uni-rostock.de
[*] Siemens AG Communications, 17489 Greifswald, Germany
thomas.bahls@siemens.com

Keywords: Access Networks, Traffic Management, Reconfigurable Hardware

Abstract: Today, an increasing number of customers subscribes for a high bandwidth internet access. But not only speed is demanded. Reliability, availability, and security move more and more into the customers focus. Carriers and Internet Service Providers, too, have increasing requirements derived from new services they want to offer to their customers or use themselves. A hardware solution is presented, which provides the functionality of MAC Address Translation and Traffic Management. This solution is highly flexible and can be adapted to the providers' needs. The module is implemented on a Field Programmable Gate Array (FPGA) and offers a wide range of functionality in an Access Network for relatively low costs. Additionally, the selection of an FPGA as implementation target offers the possibility to adapt the functionality to future needs by in-field reconfiguration.

1 INTRODUCTION

Today, more and more customers subscribe for a high bandwidth internet access. Residential customers favour the usage of Ethernet-based DSL while business customers require connections with highest bandwidths and Quality of Service (QoS) [7]. Thus, future Access Networks will not only deal with single customers but with whole LANs, which are connected to the line cards' ports. Even the Plain Old Telephone System may be integrated into the Access Network because Voice over IP is settling in the customers and manufacturers minds.

As illustrated in Figure 1, the architecture of current Access Networks consists of various aggregation levels. The main aggregation points are line cards at the customers' network side supporting Gbit Ethernet (GbE), central nodes and broadband access servers at the core nets' side supporting multiple GbE streams and fibre optics. Every device in the data path has to analyse and process frame parameters like source or destination addresses, QoS information, protocol types, and checksums to direct the data streams through the Access Network.

Due to the high number of users, management of address tables within the nodes and core switches is getting difficult and MAC address table explosions can occur [2]. Furthermore, users are even able to manipulate their own MAC addresses which can result in duplicate addresses and other severe network problems like MAC Spoofing or Address Resolution Protocol (ARP) Spoofing [1].

Actually, Internet Service Providers (ISPs) are selling more bandwidth than exists, assuming not all customers to be online at the same time and using their committed bandwidth. This is commonly known as oversubscription. Thus, the customers share a common reservoir of bandwidth. But it might happen that the customers claim what they paid for because *"Internet users who pay a fixed fee have no incentive to limit their use of the network."* [5]. Hence, the network could be overloaded during peak hours. Additionally in periods of decreased demand, valuable customers

might be allowed to cause more traffic than they have subscribed for. Besides, business customers should be of higher priority than residential customers.

Software solutions for network management like VxWorks [8] or various Linux derivates can lead to an immense workload in the central nodes' CPUs preventing them from executing their primary tasks like routing or switching. This offers vulnerable points in the access architecture. Without flexible, resource-aware, and economical management solutions, this is going to be a tremendous problem in the future.

To cope with increasing demands for bandwidth and QoS and with scalability issues, solutions as proposed in this paper help to manage the growing internet traffic in an flexible, cost-effective, and high-performance way.

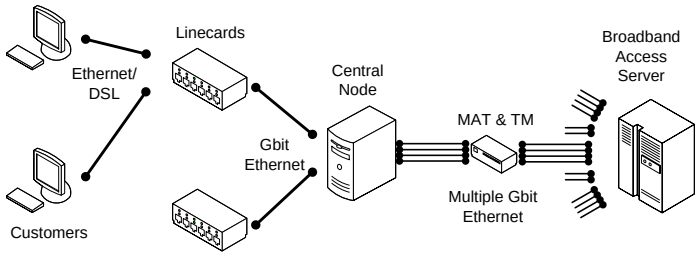

Figure 1: Access Network Architecture, Location of the Hardware Module

The remainder of the paper is organised as follows. Section 2 focuses on the functionality of the MAC Address Translation (MAT). Section 3 addresses the Traffic Manager (TM) module. In Section 4 and 5, necessary submodules for a real implementation are detailed and the benefits of a combined solution for both modules are pointed out. Finally, Section 6 shows some implementation results before the paper is concluded in Section 7.

2 THE MAC ADDRESS TRANSLATION MODULE

Below, the term *upstream* refers to traffic from the customers towards the ISPs, the term *downstream* refers to the reverse direction, and the abbreviations *MAC* and *IP* refer to data link and network layer addresses.

MAT is a technique that replaces the Customer MAC (CMAC) of a data frame with a Provider MAC (PMAC) and vice versa. In the upstream, source MACs are replaced. In the downstream, destination MACs are replaced.

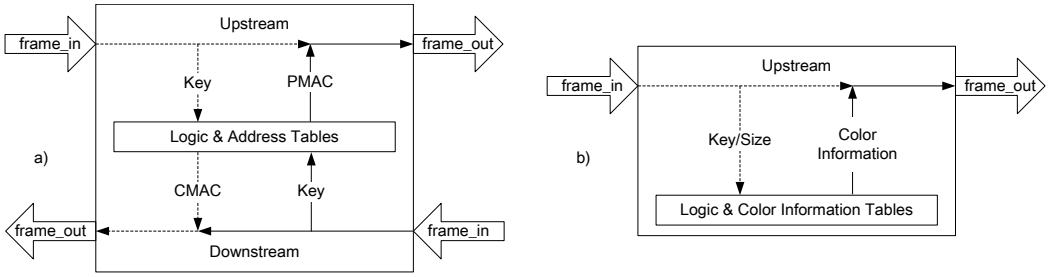

Figure 2: a) MAC Address Translation b) Traffic Management

2.1 Basic Functionality

As sketched in Figure 2, a key is extracted from each customer frame in the upstream data path. This key is processed and a truly distinct PMAC is returned to replace the CMAC. In the downstream data path, the PMAC is replaced by the original CMAC, which is stored in the module's address tables. The dimensions of the address tables in the memory will still be manageable because the intended location of the module will be within the Access Network before the main aggregation points. The elements of the key are flexible and can be configured at compile time. Source and

destination MACs, source and destination IPs, and VLAN tags (simple or stacked VLAN) are possible key parameters. The address tables contain entries for each learned or preconfigured key. Different relations between CMACs and a specific PMAC like 1:1 or n:1 are possible to reduce the workload within the core networks and to address different aspects such as security (1:1) or scalability (n:1).

If desired, only a certain portion of the CMAC may be replaced by the MAT module. This can be useful if partial MAT is performed at more than one position in the Access Network. A PMAC can be hierarchically structured and consist of different parts, which again can depend on different key sets. Therefore, a specific bitmask is used for each MAT module to identify the bits to be changed by the MAT module.

2.2 Exceptions

Besides the regular replacement of the source or destination MAC in the data link layer header, certain circumstances may occur where the replacement of the source or the destination MAC is not useful, not desired at all, or supplementary replacements are required.

On the one hand, an implicit Black List and a White List are realised. Two bits are used within the status field of each address table entry. These bits indicate whether a certain MAC is to be replaced or not. Alike the whole modules address table, both lists are managed by an administrative entity, for example a CPU or an administrator. In case an affected frame belongs to the White List, it will just be forwarded without any modification. If a frame corresponds to the Black List, it will be discarded.

On the other hand, some protocols like ARP or Reverse ARP (RARP) require a special treatment. Here, a frame contains MACs not only in the data link layer header but also at a defined position within the protocols' payload. Depending on the (R)ARP operation code and whether the frame is sent in upstream or downstream direction, the target or sender MAC within the protocol payload has to be replaced as well to achieve consistency.

2.3 Scalability

As already pointed out in the introduction, the number of internet users is continuously increasing. Thus, the number of addresses in the network is increasing, too. By nature, the addresses are concentrating in the core networks. Thus, core network devices had to handle an extensive workload and address table explosions could occur. To scale down the huge number of addresses and to ease the management of address information within the core network, the MAT functionality is used. If desired, a number of n CMACs can be mapped to one PMAC to address the scalability problem. At the position of the MAT module itself within the Access Network as shown in Figure 1, address table explosions are not an issue. The network devices just have to handle a manageable number of addresses of the attached customers. But here, the frames have already to be processed to reduce the workload in the core network and to globally scale the internet traffic. Besides scalability concerns that lead to an earliest-possible implementation in the data path further criteria may apply. A MAC translation variant that translates based on the subscriber port may be required, while such port information is unlikely to be known further in the network.

2.4 Protection against ARP Spoofing

A beneficial side-effect of MAT is the improved protection against ARP spoofing based attacks as detailed in [1]. In an Ethernet network, switching decisions are based upon data link layer addresses. Using the (R)ARP protocol, address information is automatically kept up-to-date within each network device to avoid error-prone and cumbersome manual address configuration in nowadays large network environments. But the automatic update of the address tables is the weak point within this core network protocol because attackers can simply poison the address caches with manipulated (R)ARP frames. The enhanced security arises from the static nature of the address tables within the MAT module. Static means that address entries are not automatically updated by incoming frames. As already mentioned, the address tables are managed and updated by an external administrative entity. If a key, which is extracted from an incoming frame, has no match in the address tables, the information is sent to this administrative entity to further deal with the new key. Such frames may be discarded, a new valid entry may be inserted in the address table, or further action according to individual policies and functional logic, such as generation of an SNMP trap or a syslog() call, may be applied. For example, a new key can be a specific MAC & IP combination. Every frame will be masked with a

distinct and trustworthy PMAC. Thus, the providers' core networks are kept clean from manipulated MACs and even manipulated MAC & IP pairs. This kind of protection only secures the access and core network behind the MAT functionality on the provider side. Related security issues or effects of misconfiguration on the customer side are not affected and have still to be handled in a different way.

2.5 Standards Compliance

Different MAC encapsulation schemes exist. All schemes relieve the workload of the core switches because switching decisions are no longer based on the CMACs but on the PMACs, which are inserted in the Access Network. MAT has similarities with MAC-in-MAC encapsulation (MiM) [6] and MAC stacking (MAS) [2]. But the difference is the replacement of the addresses instead of adding new header fields to the frame and thereby extending the header size. MiM adds a complete MAC header and MAS adds at least 12 Bytes (destination and source PMAC). This leads to larger frames which may easily exceed the maximum frame size and thereby result in additional delays. Moreover, MAT is feasible to be integrated in every Ethernet-based network. It does not demand any functional software extensions to existing switching hardware except for system integration and configuration purposes. It is fully transparent and conforms to the IEEE 802.3 standard.

3 THE TRAFFIC MANAGER MODULE

The Traffic Manager (TM) module targets the problem of managing excessive load in the core network. Frames within the data path have to be discarded to resolve congestions in the network. It is desired not to discard frames randomly but to do so in a fair way. For this purpose, the TM module is used. The module extracts a key from the headers of each incoming data frame. The key identifies the customer. It can consist of different fields as already listed for the MAT module in Section 2. The colour information corresponding to the key is searched in a memory. Depending on the retrieved information, the frame is coloured. Furthermore, the colour information in the memory is updated to perform metering. Like the MAT module, the TM module performs its tasks with wirespeed in hardware. It was implemented using the Very high speed integrated circuit Hardware Description Language (VHDL) and is to be inserted as a Field Programmable Gate Array (FPGA) between the central nodes and the broadband access servers in an Access Network as shown in Figure 1.

3.1 Functionality

The functionality of the TM is based on a Single Rate Three Colour Marker as described in [3]. The TM is able to meter the traffic of each customer. Depending on the actual data rates and two stored values, the committed information rate (CIR) and the burst information rate (BIR), the module colours every frame *green*, *yellow*, or *red*. For each customer, CIR and BIR are stored together with corresponding counters for metering the actual data rates.

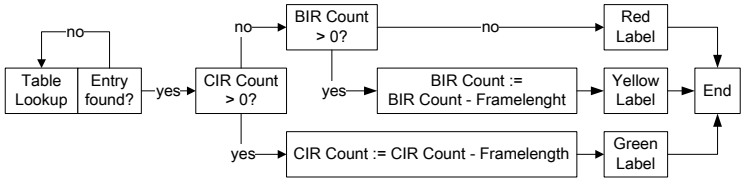

Figure 3: Colouring Algorithm

The algorithm illustrated by Figure 3 works as follows: CIR and BIR are the number of bytes the customer is allowed to transfer in a certain period of time. Two counters will periodically be reset to the CIR and BIR value if this time runs out. In the metering process, each frame's length value is subtracted from the counter values. Frames will be marked green if the customer does not exceed his CIR, meaning the corresponding counter is greater than zero. If he does and stays below his BIR, the frames will be marked yellow. Otherwise, the frames will be marked red. All CIR and BIR counters are reset in intervals of 100 ms to their initial values less a potentially used over-consumption.

The CIR counter is reset to the stored CIR value, the BIR counter is reset to the stored BIR value respectively. Thus, extra memory accesses are required. Every entry must be read and altered in periods of 100 ms. To avoid periodic congestions every 100 ms, the update cycles of all customers are time shifted. If there are 1000 customers, i.e., one update operation is executed every 100 ns. Thereby, regular lookup operations are just marginally affected. Figure 4 shows the periodic replenishment of the counters.

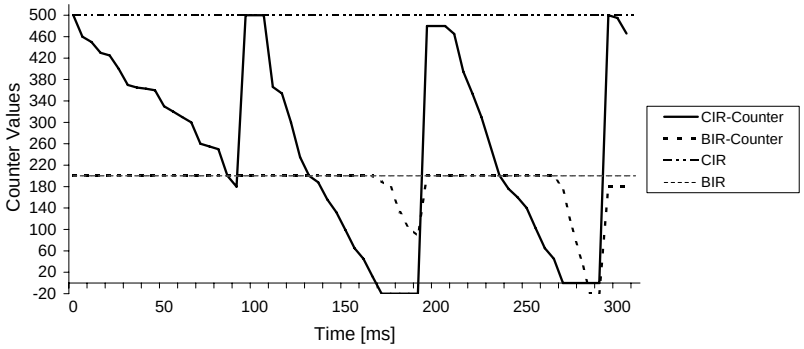

Figure 4: Time Flow of the BIR and CIR Counters

With the colour information coded into the frames, adjacent systems are able to discard red frames first. If desired, this can also be done already within the TM module. Second, all yellow frames would be discarded to resolve congestions before a green frame would need to be removed. Thus, the green frames of all users are the last which have to be deleted assuring the CIR for all users as long as possible.

The colour information can be coded into different parts of the frame. If the frames are MPLS-labelled, the EXP-field of the label is used for the coding. Alternatively, the colour can be coded into the DSCP field portion of the respective IP header if the frame contains an IP packet. The TM can operate in upstream, as shown in Figure 2, and in downstream direction.

3.2 Advantages

The Traffic Manager provides very useful functionality to manage Access Networks. With every incoming frame marked with a colour, it is possible to perform congestion control within the Access Network in a fair way. With the possibility to program the TM to discard frames with a red colour, the TM can even perform policing tasks at the very ingress point of a network. ISPs are now able to offer customers a higher quality of service. They can nearly grant high bandwidths and low loss rates in an oversubscribed network environment to premium customers.

4 SUBMODULES

In order for the presented functional modules to operate properly, certain submodules are necessary to set up MAT and TM in a real Ethernet environment. The structural assembly is outlined on the left side in Figure 5. Following, the individual entities are described in detail.

4.1 Synchronisation

Within the functional modules, an 8-bit wide data bus is used to handle one byte per cycle as shown on the right side in Figure 5. For a throughput of one Gbit per second, the proposed system must achieve a frequency of at least 125 MHz. The data stream has to be sync'd into and out of the MAT and TM module at the ingress and egress points. At the ingress points, the data stream is delivered by a medium access controller which is connected to a physical layer transceiver device. At the egress points, the data stream is transmitted by another medium access controller. Even if the clocks to and from the medium access controllers have the same frequency, they will not be phase aligned anyway. For synchronisation and clock domain crossing, simple and small First-In-First-Out buffers (FIFOs) are used. Because of independent read/write ports, a FIFO is inherently convenient for these purposes.

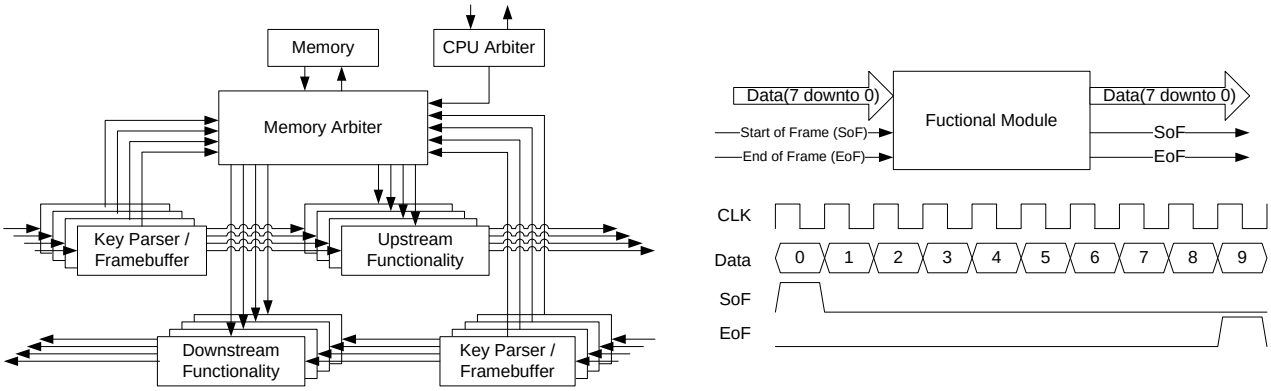

Figure 5: Wired Functional Module and Data Path Interface

4.2 Framebuffer

The first submodule in the data path is a so-called Framebuffer (FB). The structure is shown in Figure 6. Its tasks are to temporarily buffer received frames in a FIFO and to extract the keys from the frame headers with an integrated Key Parser (KP). The key identifies a customer, a group of customers, or a flow. The structure of the key is flexible and specified at synthesis time. The key can be composed of the source and destination MACs and IPs, the inner and outer VLAN tags, the Ethertype, and the DSCP field in the IP header. This way, the FB requires just as much logic as really needed for a certain key. Furthermore, the FB will request a lookup at the Memory Arbiter (MA) for each extracted key or will directly redirect the frame to the CPU if the target address matches the configured system address. If no match can be found for a certain key, the frame will be sent to the CPU to further handle the unknown key. For example, an update of the memory is initiated as described in the next paragraph. Otherwise, the frame is forwarded to the MAT or TM module together with the information which is returned from the memory.

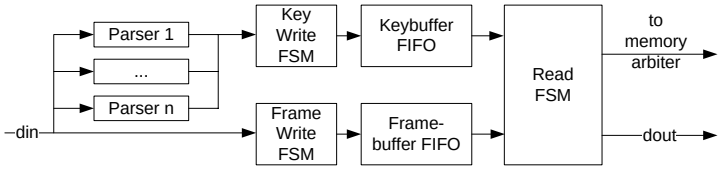

Figure 6: Framebuffer and Key Parsers

4.3 Memory Arbiter

At the intended location within the Access Network, an interface with four independent GbE channels is existent. Thus, four parallel data paths will be used. As there are four KPs in both up- and downstream direction, eight independent lookup requests may be generated concurrently. That is why an MA is required. Any frame arriving via the four GbE links from any side has to be classified. Thus, the number of memory lookups that can be performed in a given time is the bottleneck of the system and defines the possible throughput!

The MA decides which of the lookup requests is performed next. This decision is based on the Least Laxity First (LLF) algorithm. The LLF, which is usually used for process scheduling in operating systems, assigns the memory access to the request with the smallest slack. In process scheduling, the smallest slack is computed as the difference between two parameters of the process, the deadline to meet and the computation time required. In case of two equal slack values, the process with the smallest deadline is scheduled. In our application, the deadline derives from the space left in the FB. The computation time derives from the number of frames that are already stored in the FB. Both parameters are presented as four bit values along with the extracted key to the MA. It has been considered to use the Enhanced LLF algorithm [4] but the benefit of avoiding ”thrashing” (meaning unnecessary task switches) is of no use here. In contrast

to task switches in operating systems, no costs arise from changing the KP needlessly.

4.4 Memory

The memory, where the information for the functional elements is stored, can be implemented either with external Dynamic RAM or with the FPGAs internal memory resources dependent on the size and number of necessary entries in the memory. A memory entry is composed of the key, which consists of maximal 216 bit, and the corresponding information. The information stored for MAT consists of the PMAC (48 bit). For the TM module, CIR, BIR, CIR counter, and BIR counter (82 bit) are stored. The entries are arranged in a sorted manner in the memory. Thus, the lookup complexity is reduced to $O(\log 2(N))$. Sorting the memory entries, of course, has the drawback of increased time consumption for management operations like insertions or deletions. Both operations have a time complexity of $O(N + \log 2(N))$. In average, $N/2 + 2 \cdot \log 2(N-1)$ memory accesses are required. As changes in the look up table occur quite seldom in Access Network environment, these extra costs are acceptable when increasing the lookup speed from $O(N)$ to $O(\log 2(N))$. The TM functionality requires some additional logic in the memory module. This logic has the purpose of updating the counters regularly without interfering with the regular lookup functionality.

4.5 CPU Arbiter and CPU Interface

The CPU interface is bidirectional and allows parallel operation to and from the CPU, thereby facilitating data retrieval from a management system as well as configuration actions such as an address table update. The CPU Arbiter and the CPU Interface are responsible for managing the data paths to and from the administrative entity connected to the functional module, whether it is MAT, TM or both. The kind of the CPU interface is not previously pinpointed and must be adapted to the particular requirements. For example, a PCI interface can be used to connect to an external CPU. Alike the memory, several so-called clients demand access to the CPU. Client classes are for example the upstream or downstream FBs, which may want to send frames containing the module's own address or with an unknown key to the CPU at the same time. These accesses have to be scheduled, too. This is done via a simple priority and Round-Robin (RR) scheduler. The scheduler prioritises different client classes. Multiple requests within the same class are scheduled with an RR algorithm.

Because frames are sent to the CPU in upset conditions or when addressed to it, attackers may try to cause damage in the system by encumbering the CPU with exceeding workload. To protect the CPU from such Denial-of-Service based attacks, two mechanisms are implemented. First, the CPU arbiter obstructs every access to the CPU until the CPU signals to have become ready again. When the internal buffers of the FB are fully utilised, additional frames will be discarded in the buffers anyway. Second, frames in the upstream data path, which are addressed to the CPU, have lowest priority. Generally, no frame in the upstream data path should be addressed to the administrative unit. Regarding the downstream data path, management frames may be addressed to the CPU and therefore have high priority. In both directions, frames with an unknown key have middle priority to maintain an efficient but also secure update of the address tables.

5 Benefits of a Combined Solution

Throughout the previous sections, both modules have independently been treated. But due to different reasons, a combination of both functionalities in the data path would be beneficial. On the one hand, both modules feature the same interface allowing an easy connection within the data path. On the other hand, both modules support the same basic system structure as sketched in Figure 5. Thus, MAT, TM, and also the submodules represent functionally sound blocks in a construction kit. New modules can easily be integrated by using the same systematics. No additional resources are used for the FBs and KPs when integrating additional functional modules. The size of the KPs only depends on the set union of the independent keys for each functional module. The size of the MA depends on the number of functional modules and the key size. The size of the CPU arbiter slightly depends on the number of parallel GbE channels and not on the number of the functional modules. Thus, by sharing common submodules, e.g., the extraction of the various key fields, valuable hardware resources can be saved. A broad range of functions can be provided within just one device in the Access Network.

Module	MAT	TM	MA	CPU Arbiter	2 x FB&KP	Synchroni-sation	∑ GbE Channel
Slices	180	240	610-1920	430	1330-2580	400	2950-4580
f**(MHz)**	210	180	135	320	145	250	135

Table 1: Resource Utilisation for one Gbit Ethernet Channel

6 Implementation Results

A first implementation was done using a Xilinx XC4VFX20 FPGA although the intended target platform will be a larger member of the Virtex device family. The amount of reconfigurable resources needed for the design considerably depends on the configuration of the modules. The structure of the keys of each functional module and the number of parallel GbE channels must be defined. These configurations can be made by changing values in a VHDL configuration file. Depending on the key structure, a combined MAT and TM functionality for one GbE channel requires 2950 slices if configured with a minimal key size (only the DSCP field of the IP header). Configuring a maximum key size, the number of slices increases up to 4600. A typical implementation, where the key consists of MACs and IPs, demands approximately 3600 slices. As described in Section 4.1, some negligible glue logic is included to synchronise incoming data into the module. The required resources for the modules are listed in Table 1. Particulars on the memory are omitted because the size of the memory considerably depends on the key structure, the number of entries in the address tables, and the nature of the memory, e.g., internal memory blocks or external RAM modules. The module can operate at a speed of 135 MHz, although at least 125 MHz would be sufficient to handle GbE.

7 Conclusion

The proposed hardware solutions help to manage the scalability problems in current and future Metropolitan Ethernet Networks by processing frames already in the access area. Security issues are also addressed accordingly. The modules compute the data streams with wirespeed. Thus, nearly no additional delay is generated in the data path. Due to the fact that our solutions are designed for reconfigurable hardware, they are almost as flexible as software solutions. The functional spectrum can be broadened and adapted to future challenges. Both modules immensely relieve the workload of the central nodes and switches. Besides configuring the address and colour tables in the hardware modules, the CPU controlling the Access Network is not stressed with additional tasks. MAT mainly addresses the problems of MAC table explosions and duplicate addresses while TM addresses bandwidth management problems due to oversubscription. In consequence of similarities in the architecture of the MAT and TM modules, a combined architecture has been developed to save hardware resources. Anyhow, both modules can be implemented independently.

This work is done in cooperation with Siemens AG Communications, Greifswald.

References

[1] J. Bronson. Protecting Your Network from ARP Spoofing-Based Attacks. Foundstone, Inc., June 2004.

[2] G. Chiruvolu, A. Ge, D. Elie-Dit-Cosaque, and M. Ali. Issues and Approaches on Extending Ethernet Beyond LANs. *IEEE Communications Magazine*, pages 80–86, March 2004.

[3] J. Heinanen. A Single Rate Three Color Marker. RFC 2697, September 1999.

[4] J. Hildebrandt, F. Golatowski, and D. Timmermann. Scheduling Coprocessor for Enhanced Least-Laxity-First Scheduling in Hard Real-Time Systems. In *Proc. of the 11th Euromicro Conference on Real-Time Systems*, York, GB, 1999.

[5] L. W. McKnight and J. Boroumand. Pricing Internet Services: Approaches and Challanges. *IEEE Computer*, pages 128–129, February 2000.

[6] Nortel Networks. Service Delivery Technologies for Metro Ethernet Networks. White Paper, 2003.

[7] R. Santitoro. Metro Ethernet Services - A Technical Overview. White paper, 2003.

[8] Wind River. Platform for Network Equipment - VxWorks Edition. Product Note, 2005.