Access Network Solutions

MATMUNI – MAC Address Translation, Traffic Manager, MPLS-UNI
Outline

1. Internet Growth
2. General Impacts
3. Relevance for Access Networks
4. The Solution → MATMUNI
5. Example Scenarios
6. Summary

7. Appendix
Internet Growth

- Increasing User Numbers
Internet Growth

- Increasing User Numbers
- New Technologies (xDSLs, Cable, Fiber…)

![Diagram showing data rates and distances for different technologies like FTTH, FTTC/B, FTTN, VDSL2, ADSL2, and ADSL, with copper and optical fiber options.]
Internet Growth

- Increasing User Numbers
- New Technologies (xDSLs, Cable, Fiber…)
- New fancy Services (triple-, quadruple-, …, n-tuple-Play)
Internet Growth

- Increasing User Numbers
- New Technologies (xDSLs, Cable, Fiber…)
- New fancy Services (triple-, quadruple-, …, n-tuple-Play)

Radical Change of Audience (compared to, e.g., 1980)

→ Internet = MASS-Medium!
→ Critical Impacts
General Impacts

• Traffic Load/Volume increases
  → Traffic Management needed

• QoS Demands & Workload increase
  → High-performance / Non-blocking Operation needed
  → Manageable Solutions for Traffic Differentiation needed

• Security decreases (Anonymity, typical Attack Scenarios)
  → Need for Prevention of Attack Scenarios

• Complexity increases (many Users & Addresses)
  → Scale Address Space
  → Fast Routing
Relevance for Access Networks

• What can Access Networks do about this?
  – Aggregation
  – Authorization & Authentication
  – Preprocess Traffic in Front of Core Networks

• Preprocessing Options
  – Metering, Policing, Shaping
  – Enforce Limits (e.g., number of MACs per port)
  – Enforce Identity & Uniqueness (e.g., MAT in L2 networks)
  – Inspect and apply Policy to prevent certain Attacks

• But keep it all *transparent* to End Points!

→ This is what MATMUNI is about!
The Solution - MATMUNI

- Where to place MATMUNI?
  - Logically between DSLAM and BRAS
  - Physically on DSLAM

![Diagram](image)
MAC Address Translation

- Mapping of Customer MACs to Provider MACs
- 1:1 = Security
- n:1 = Scalability
- Extra Options:
  - Bypass
  - Discard
  - ARP Handling
  - DHCP Handling
  - Configurable partial Translation
Traffic Manager

- Metering, Policing, Colormarking,
- Already in Access Area

Different Versions:
- MPLS Experimental Bits
- IP DSCP-Field
MATMUNI – Building Blocks

**MPLS-User Network Interface**

- Encapsulation Scheme
- Meant for fast Routing
  → **Here:** Simply a Container to carry Information
- Martini Encapsulation
- *Not* a full-blown LER
MATMUNI Characteristics

Major Characteristics
• Flexible Layer 2 Address Translation
• Traffic Management
• MPLS-User-Network-Interface

Other Characteristics
• Standards Compliance (Translation, no Encapsulation)
• Transparency for User Traffic (IP)
• Modular System Architecture (extensible, configurable)
• Total Throughput only limited by Hardware Constraints
• MPLS Labels = Generic Information Container
• Structured PMACs embed Hierarchy of Access Network
4. MATMUNI – Prototype

- “It really works!”
- Xilinx Virtex-4 FX20 Eval-Board with 1x GE bi-directional Channel
- GUI for Configuration & Monitoring
Now How Does MATMUNI Work?

Examples.
Problem: ARP Spoofing

- Before attack: Eve receives **no** non-owned frames
Problem: ARP Spoofing

- 1st step: Poison Alice’s ARP Cache
  → IP(Gateway) == MAC(Eve)
  → Alice’s frames go through Eve to Gateway
Problem: ARP Spoofing

- 2nd step: Poison Gateway’s ARP Cache
  - IP(Alice) == MAC(Eve)
  - All frames from Alice & Gateway go through Eve

Reason:
Automatic/dynamic update of ARP tables
→ solved by MAT on next slide
ARP Spoofing Prevention with MAT

- MAT’s FDB is not automatically updated by ARP
- Manipulated ARP frames will be dropped
  → The Clients’ and GW’s ARP caches are not poisoned

<table>
<thead>
<tr>
<th>ARP Cache Clients (auto update)</th>
<th>MAT Mapping Table (static)</th>
<th>ARP Cache Gateway (auto update)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>MAC</td>
<td>Port</td>
</tr>
<tr>
<td>IP(GW)</td>
<td>MAC(GW)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>MAC(X)</td>
<td>PMAC(X)</td>
</tr>
<tr>
<td>3</td>
<td>MAC(Eve)</td>
<td>PMAC(Eve)</td>
</tr>
</tbody>
</table>

Simplified Scenario!
Problem: Oversubscription & Congestion

- Congestions and runaway Frame Drops without Limitations through Traffic Manager
- Hard Limit @ 100% Bandwidth (BW) Utilization
  ➔ Who follows or relies on “Fair Use Policies” when subscribing for Internet Access?

Maximum BW (100%)

Possible BW Limitation set by TM (e.g., 20% BW Utilization allowed per Users)

Peaks over 100% would be discarded ➔ potentially unfair
Fair Bandwidth Utilization with TM

- With TM, Limitation of each Users Traffic Load to the subscribed and configured Value (20% in the Example)
- Still extra BW available for other Services
- Fair

---

Maximum BW (100%)

Possible BW Limitation set by TM (e.g., 20% BW Utilization allowed per Users)

Policing & Frame Drops on a per-Customer Basis

Bandwidth Demands for 4 Users (simplified)
Summary

- Scalability
  - N:1
  - n:1 MAT
  - Reduce FDB Size

- Traffic Manager
  - Fair Bandwidth Allocation & QoS

- 1:1 MAT
- Identity (Translation Scheme)
- Attack Prevention

- Expand MPLS into Access Area
- Generic Information Container

- FPGA Hardware Solution (wire speed)
- Flexibility by Reconfiguration
Results are internationally accepted!

“sMAT – A Simplified MAC Address Translation Scheme“
15th IEEE Workshop on Local and Metropolitan Area Networks, 2007, Princeton, NJ, USA

“Configuration Tool and FPGA-Prototype of a Hardware Packet Processing System“
Design, Automation and Test in Europe Conference and Exhibition, 2007, Nice, France

“An integrated Hardware Solution for MAT, MPLS-UNI, and TM in Access Networks“
31st Annual IEEE Conference on Local Computer Networks, 2006, Tampa, FL, USA

“Wirespeed MAC Address Translation and Traffic Management in Access Networks“
World Telecommunications Congress 2006, Budapest, Hungary,

“A Simplified, Cost-Effective MPLS Labeling Architecture for Access Networks“
World Telecommunications Congress 2006, Budapest, Hungary

Thomas Bahls, Stephan Kubisch, Daniel Duchow, Harald Widiger
Appendix

(Technical Information)
System Architecture

- 2-8 main GbE Data Paths
- Vertical Control Paths for Memory Operations (on- or offboard) and System Configuration (CPU)
- Serially linked functional Modules
Synthesis Results

- Xilinx Virtex 4 FX20-11 FPGA, ISE XST Synthesis Flow
- \( \text{min} = \) minimal Key configured
- \( \text{max} = \) all possible Keys configured
- 125 MHz required for non-blocking GbE Operation

<table>
<thead>
<tr>
<th>Hardware Module</th>
<th>Speed in MHz</th>
<th>Slices min/max</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATMUNI Core System</td>
<td>140</td>
<td>2300/4300</td>
</tr>
<tr>
<td>MAT (UP &amp; Downstream)</td>
<td>220</td>
<td>2x210</td>
</tr>
<tr>
<td>TM (UP &amp; Downstream)</td>
<td>190</td>
<td>2x240</td>
</tr>
<tr>
<td>MPLS-Labeler</td>
<td>180</td>
<td>129</td>
</tr>
<tr>
<td>MPLS-Delabeler</td>
<td>320</td>
<td>101</td>
</tr>
<tr>
<td>Memory Arbiter</td>
<td>160</td>
<td>282/1023</td>
</tr>
<tr>
<td>CPU Arbiter</td>
<td>160</td>
<td>640</td>
</tr>
<tr>
<td>Key Parser &amp; Framebuffer</td>
<td>150</td>
<td>336/723</td>
</tr>
<tr>
<td>Framebuffer</td>
<td>180</td>
<td>205</td>
</tr>
</tbody>
</table>
Simulation Results

- 8 independent GbE channels
- Artificial traffic
  - only minimal frames
- Realistic traffic
  - 35% minimal
  - 11% average
  - 10% maximal
  - 44% random
  - no packet losses
- Average latency 130 cycles (~1 us for GbE)