

Development of high-speed FreeBSD-based solution for Internet traffic management

Nikolay Aleksandrov
(razor@blackwall.org)

I. Setting Up The Environment

- The "Internet" in Bulgaria
- Network design and topology
- Average user speeds offered
- Rivalry

Our problem:
Multi-gigabit traffic
even with small
number of clients

I. Setting Up The Environment

- Low, middle and large size Bulgarian ISPs
- Major solutions for our problem
- Common solutions for Bulgarian ISPs
- Shortcomings of the above

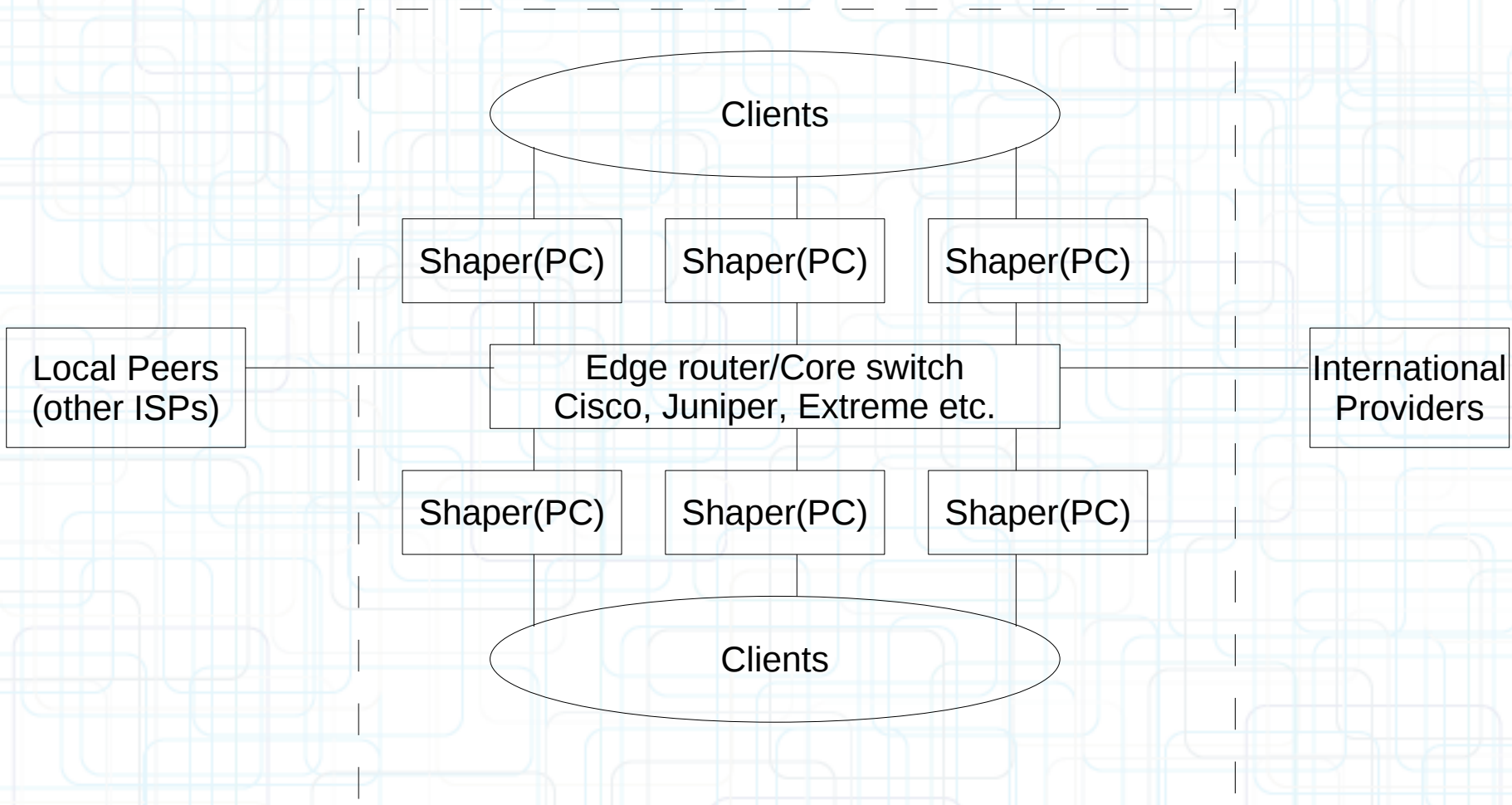


Our solution:

**A fast, highly customizable software
using general purpose computer
hardware**

I. Setting Up The Environment

The most common solution



II. Our solution – version 1

- Basic requirements
 - Per interface VLANs
 - ~ 10 queues per person
 - MAC per IP per VLAN protection
 - VLAN ARP proxy to shape local traffic and use addresses more efficiently
 - Efficient traffic analysis for problem solving
 - **SINGLE PC SHAPER/ROUTER**

II. Our solution – version 1

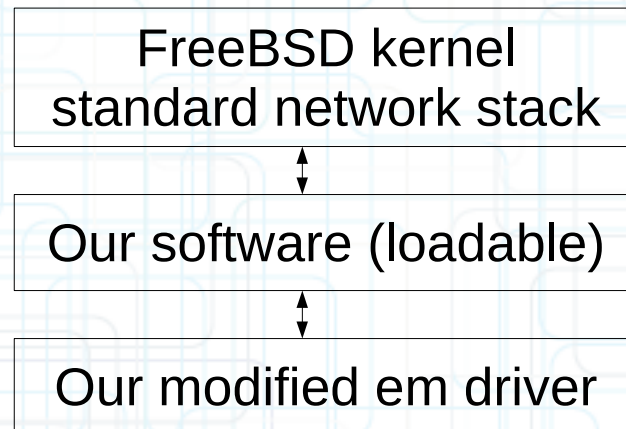
- Hardware used
 - Arima Xeon mainboard with two sockets
 - Intel Xeon 2.8 GHz CPU x 1
 - 2 x 512 MB RAM (for dual channel operation)
 - Intel Server Ethernet Adapters (Dual/Quad- port)
- First step: testing and optimizing the standard em device driver in FreeBSD 5.x (ver. 1.7.x at that time)
- Choosing software design for the router based on our requirements and observations

II. Our solution – version 1

- Graph-based design with extremely small and simple node types. For example:
 - Match node – params: offset, mask, size, algorithm
 - BPF node – connection with the BPF, counters
 - LRCV node – stands for Local Receive
 - Tag node – attaches VLAN tag
 - ETH node – params: src, dst, proto
 - ARPRPL node – params: mac address
 - VIFACE node – virtual Interface for ifconfig
 - SHAPER/MATH nodes
 - ...

II. Our solution – version 1

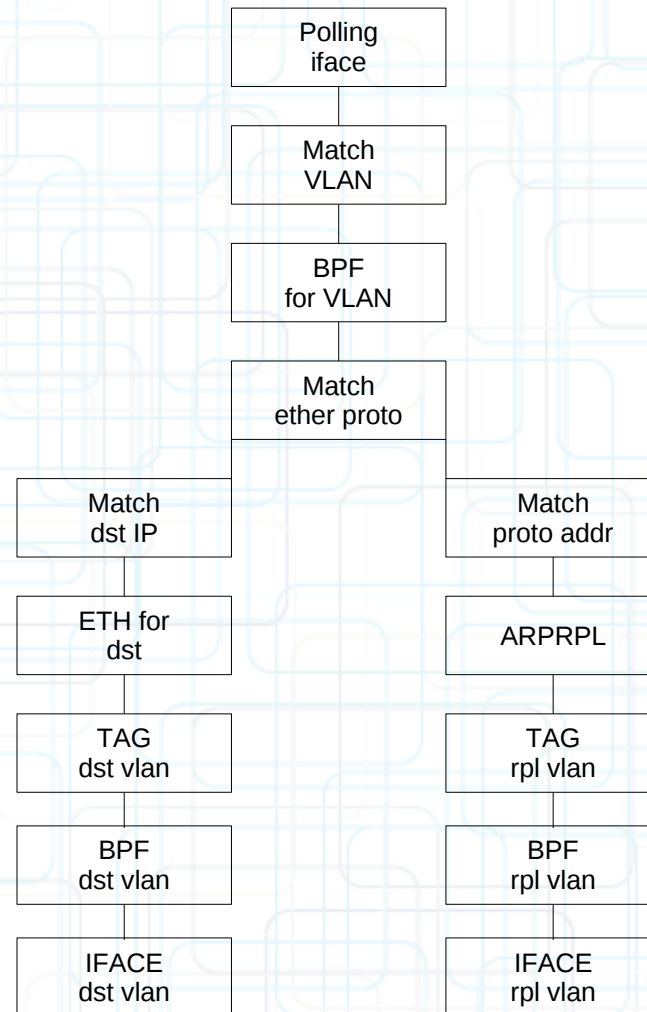
- Software design from the point of view of FreeBSD



- Loadable kernel module, integrated polling for the ethernet cards. Registers as callout which executes every clock tick (usually HZ > 2000). Registers syscall for control and information retrieval

II. Our solution – version 1

- Our view for packet transition – passing packet simple example



II. Our solution – version 1

- Theoretical limits
 - Mainboard bus limits
 - Slot/card limits
 - CPU limits
- Classifying performance based on limits
 - Missed/dropped packets
 - Bandwidth (MB/s – PPS)
 - CPU load average, increase (every instr counts)
 - Overall responsiveness
- Usual tests are far from reality

III. Our solution – version 2

- Hardware used:
 - Intel S5000 V/P mainboards with 2 sockets
 - Intel Xeon Quad-Core (1.6/2.5 GHz) x 1
 - Intel Server Ethernet Adapters (Dual/Quad)
 - Dual channel memory

III. Our solution – version 2

- Going multiprocessor (why?)
- Upgrading the kernel to 6.x
- New polling mechanism
- New node types and functionality

III. Our solution – version 2

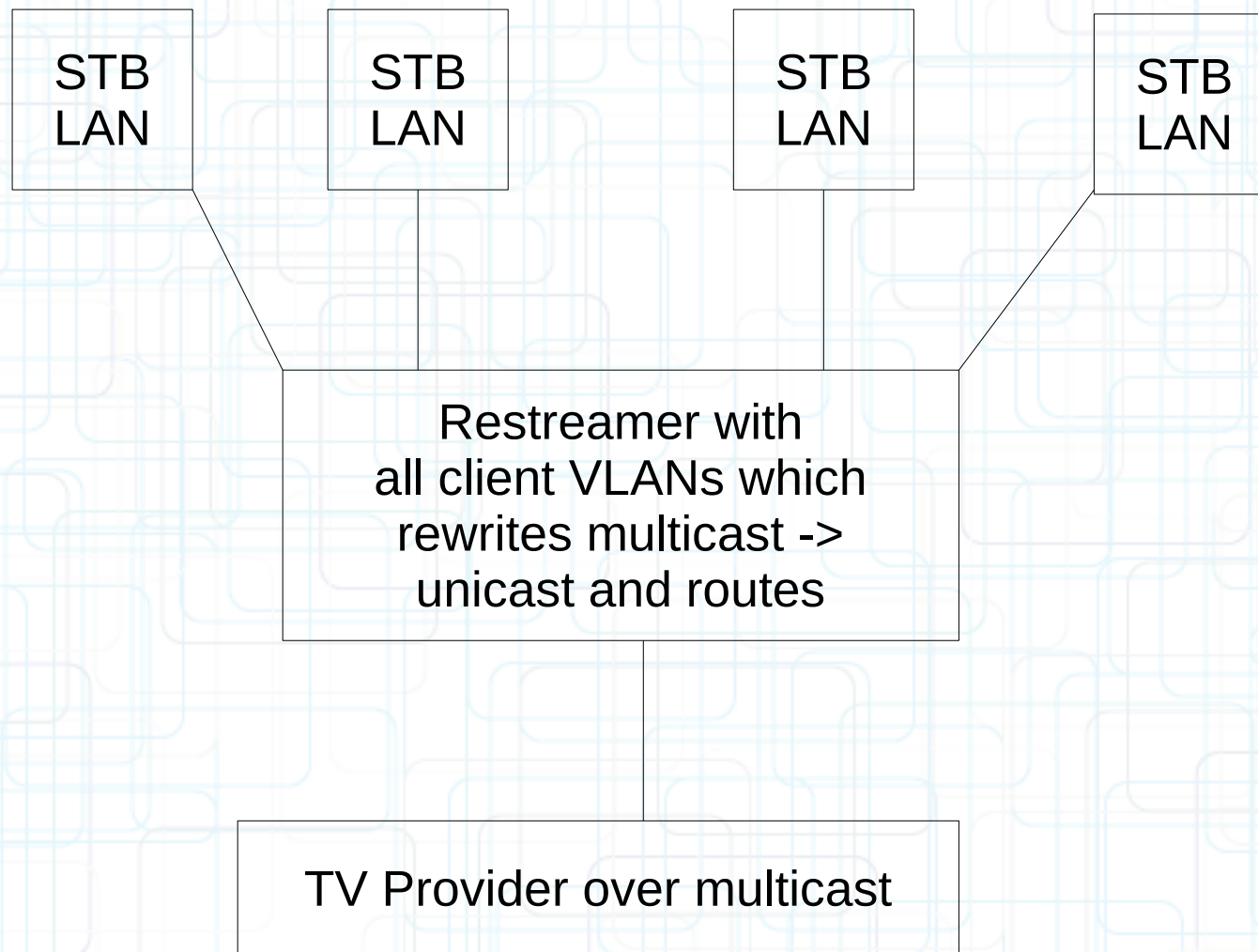
- Bottlenecks and problems
- Removing ALL locks in the driver
- New queueing mechanism (DMFQ)
- Kernel threads – good, callouts – bad

IV. Statistics

ISPs	Number of clients	Hardware/Software	Traffic	Overall load
A	15 000	Core 2 Duo/Xeon Quad X 30, FreeBSD 7.x, 2 queues per client 2 edge FreeBSD routers With missed/dropped packets 500 users per machine max 2 rules per person	3.5 Gbps overall	60-75 %
B	20 000	Cisco SCE X 3, 76xx edge router	3 Gbps overall	Cisco maximum
C	10 000	Cisco 76xx edge, Athlon 64 X2/Dual Opteron X 30 FreeBSD 7.x/8.x 4 rules per person X 300 users per machine	800 Mbps per machine max with ~ 80 kpps (on the Athlons), 1 Gbps with 180 kpps on the dual Opteron	80-90 %
D	60 000	Juniper MX edge router, > 70 machines (mixed, desktop/server), Debian Linux (shape)	~ 800 Mbps per machine	> 70 %
E	4 000	6 Quad-core desktop PCs (Q9550), Linux Debian, shape + route	150 mbps peak per direction overall	15-25 %
OnlineDirect (that would be us)	12 000	1 X edge MP router (single quad-core xeon), 3 X Quad-core 2.5 GHz (1 active core per machine) 10 queues per person, personal firewalls, packet inspection & classification, 1000 VLANs, ARP proxy	30 Gbps overall, 10 Gbps per shaper, ~ 20 Gbps on the edge...	40-50 %

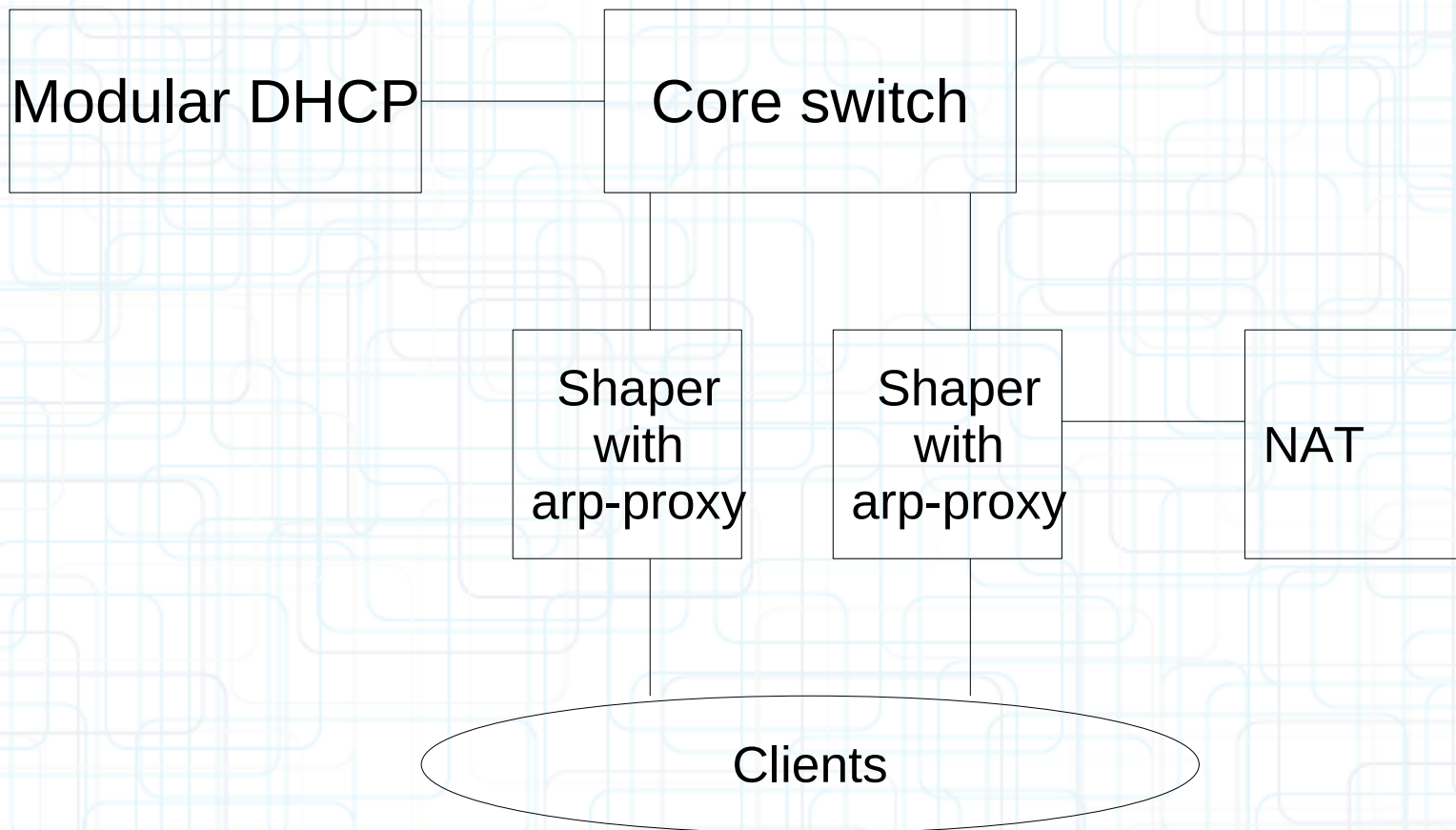
V. Case Studies

- TV service



V. Case Studies

- DHCP & NAT



VI. Conclusion & Future Work

- Extremely flexible, fast and cost-effective
- Dedicated router distribution or OS
- Nehalem & 10 gig interfaces
- XML definitions, new queueing mechanisms
modular nodes and more failover options

