Master Course Computer Networks IN2097

Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.

Chair for Network Architectures and Services
Institut für Informatik
Technische Universität München
http://www.net.in.tum.de


Christmas Party
Chair for Network Architectures and Services (I8)
Date: Tuesday 20 December 2010
Time: starting at 18:00 Uhr
Where? FMI, 3rd Floor, Room 03.07.023
Please register via www.net.in.tum.de "News / Weihnachtsfeier"

Registration for Christmas Party

Please register via www.net.in.tum.de "News / Weihnachtsfeier"

News11.11.11

Einladung zur Weihnachtsfeier am 20.12.2011 ab 18:00 Uhr in Raum 03.07.023.

- **Anmeldeformular zur Weihnachtsfeier**

  Anrede:

  Name:

  Vorname:

  E-Mail:

  Mitbringsel:

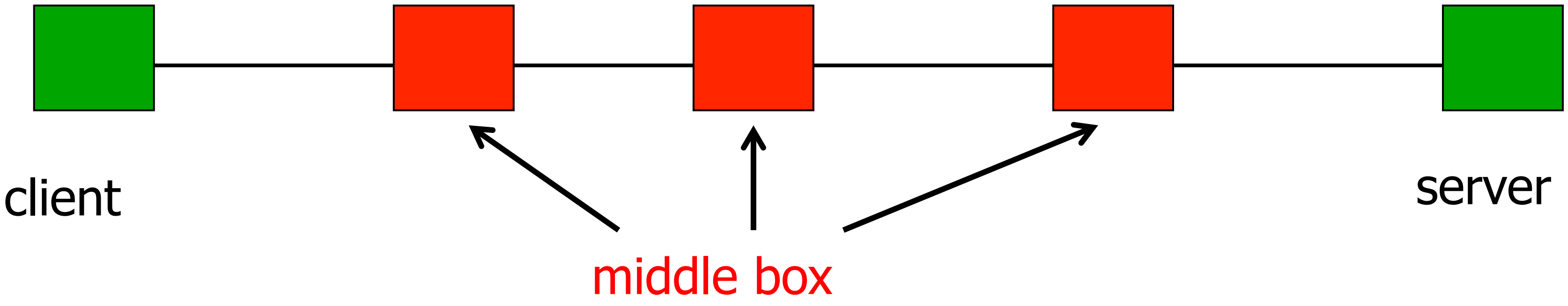
Middleboxes

RFC 3234 - Middleboxes

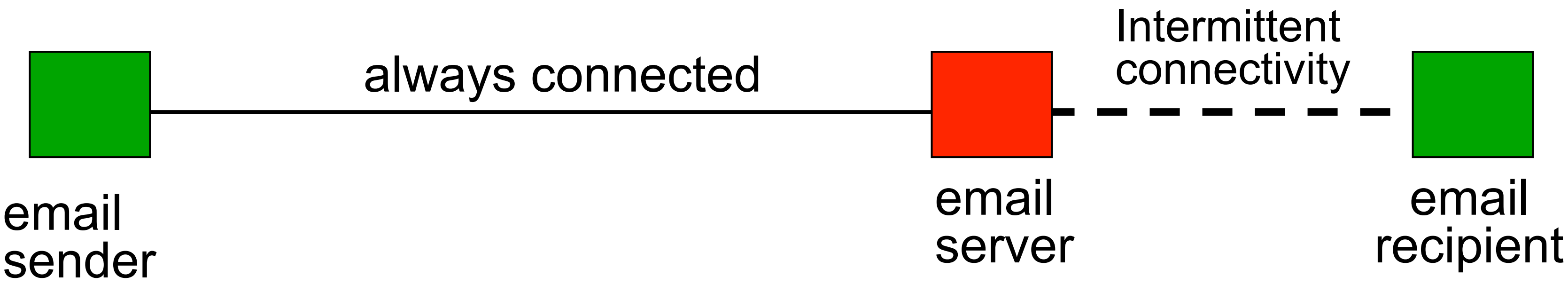
- The phrase "middlebox" was coined by Lixia Zhang as a graphic description of a recent phenomenon in the Internet.

Lixia Zhang, UCLA

What are *middle* boxes?

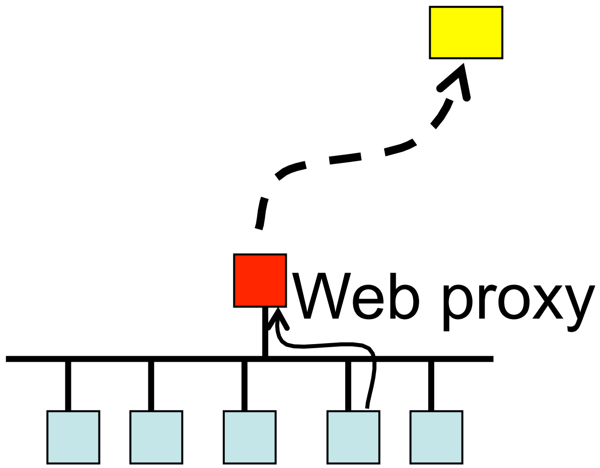


- data is no longer delivered between the two end boxes by *direct* IP path
- The first middleman: email server

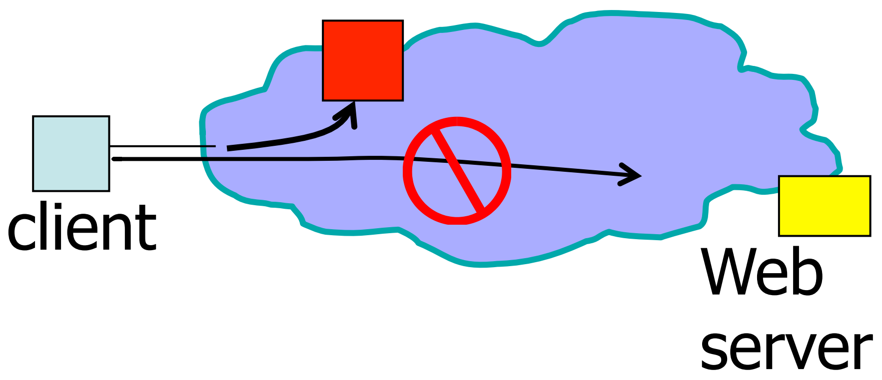


Middleboxes

- Web proxies



- "transparent" Web caches



Packet hijacking!("for your benefit")

Middleboxes Address Practical Challenges

- IP address depletion
  - Allowing multiple hosts to share a single address
- Host mobility
  - Relaying traffic to a host in motion
- Security concerns
  - Discarding suspicious or unwanted packets
  - Detecting suspicious traffic
- Performance concerns
  - Controlling how link bandwidth is allocated
  - Storing popular content near the clients

Layer Violation Boxes

- Peek into application layer headers
- Send certain packets to a different server
- Proxy certain request without being asked
- Rewrite requests

- Result: unpredictable behaviour, inexplicable failures
- c.f. RFC 3234

RFC 3234 - Middleboxes: Taxonomy and Issues

- A middlebox is **defined** as any intermediary device performing functions other than standard functions of an IP router on the datagram path between a source host and destination host.

- Standard IP router: transparent to IP packets

- End-to-end principle: asserts that some functions (such as security and reliability) can only be implemented completely and correctly end-to-end.
- Note: providing an incomplete version of such functions in the network can sometimes be a performance enhancement, but not a substitute for the end-to-end implementation of the function.

Properties

- Middleboxes may
  - Drop, insert or modify packets.
  - Terminate one IP packet flow and originate another.
  - Transform or divert an IP packet flow in some way.
- Middleboxes are never the ultimate end-system of an application session

- Examples
  - Network Address Translators
  - Firewalls
  - Traffic Shapers
  - Load Balancers

Concerns

- New middleboxes challenge **old protocols**. Protocols designed without consideration of middleboxes may fail, predictably or unpredictably, in the presence of middleboxes.

- Middleboxes introduce **new failure modes**;
rerouting of IP packets around crashed routers is no longer the only case to consider. The fate of sessions involving *crashed middleboxes* must also be considered.

- **Configuration** is no longer limited to the two ends of a session; middleboxes may also require configuration and management.

- **Diagnosis** of failures and misconfigurations is more complex.

RFC 3234: Middlebox Classification

1. Protocol layer (IP layer, transport layer, app layer, or mixture?)
2. Explicit (design feature of the protocol)
   or implicit (add-on not by the protocol design)
3. Single hop vs. multi-hop (can there be several middleboxes?)
4. In-line (executed on the datapath) vs. call-out (ancillary box)
5. Functional (required by application session) vs. optimising
6. Routing vs. processing (change **path** or create side-effect)
7. Soft state (session may continue while middlebox rebuilds state)
   vs. hard state
8. Failover (may a session be redirected to alternative box?)
   vs. restart

Specific Middleboxes

- **Packet classifiers**
  - classify packets flowing through them according to policy
  - either select them for special treatment or mark them
  - may alter the sequence of packet flow through subsequent hops, since they control the behaviour of traffic conditioners.
  - {1 multi-layer, 2 implicit, 3 multihop, 4 in-line, 5 optimising, 6 processing, 7 soft, 8 failover or restart}

1. Protocol layer (IP layer, transport layer, app layer, or mixture?)
2. Explicit (design feature of the protocol) or implicit
3. Single hop vs. multi-hop (can there be several middleboxes?)
4. In-line (executed on the datapath) vs. call-out (ancillary box)
5. Functional (required by application session) vs. optimising
6. Routing vs. processing (change packets or create side-effect)
7. Soft state (session may continue while rebuilding state) vs. hard state
8. Failover (may a session be redirected to alternative box?) vs. restart

Specific Middleboxes

- **IP Firewalls**
  - Inspects IP and Transport headers
  - configured policies decide which packets are discarded, e.g.:
    - Disallows incoming traffic to certain port numbers
    - Disallows traffic to certain subnets
  - Does not alter forwarded packets
  - Not visible as protocol end-point
  - {1 IP layer, 2 implicit, 3 multihop, 4 in-line, 5 functional, 6 routing, 7 hard, 8 restart}

1. Protocol layer (IP layer, transport layer, app layer, or mixture?)
2. Explicit (design feature of the protocol) or implicit
3. Single hop vs. multi-hop (can there be several middleboxes?)
4. In-line (executed on the datapath) vs. call-out (ancillary box)
5. Functional (required by application session) vs. optimising
6. Routing vs. processing (change packets or create side-effect)
7. Soft state (session may continue while rebuilding state) vs. hard state
8. Failover (may a session be redirected to alternative box?) vs. restart

Specific Middleboxes

- **Proxies**
  - Intermediary program that acts as client and server
  - Make requests on behalf of client and then serves result

- **Application Firewalls**
  - Act as a protocol end point and relay (e.g., Web proxy)
  - May

    (1) implement a "safe" subset of the protocol,

    (2) perform extensive protocol validity checks,

    (3) use implementation methodology for preventing bugs,

    (4) run in an insulated, "safe" environment, or

    (5) use combination of above

Middlebox Types according to RFC 3234

1. **NAT**
2. **NAT-PT**
3. **SOCKS gateway**
4. **IP tunnel endpoints**
5. **packet classifiers, markers, schedulers**
6. **transport relay**
7. TCP performance enhancing proxies
8. **load balancers that divert/munge packets**
9. **IP firewalls**
10. **application firewalls**
11. application-level gateways

bold - act per packet
- do not modify application payload
- do not insert additional packets

12. gatekeepers / session control boxes
13. transcoders
14. (Web or SIP) proxies
15. (Web) caches
16. modified DNS servers
17. content and applications distribution boxes
18. load balancers that divert/munge URLs
19. application-level interceptors
20. application-level multicast
21. **involuntary packet redirection**
22. **anonymizers**

Assessment of Middlebox Classification

1. Protocol layer (IP layer, transport layer, app layer, or mixture?)
2. Explicit (design feature of the protocol) or implicit
3. Single hop vs. multi-hop (can there be several middleboxes?)
4. In-line (executed on the datapath) vs. call-out (ancillary box)
5. Functional (required by application session) vs. optimising
6. Routing vs. processing (change packets or create side-effect)
7. Soft state (session may continue while rebuilding state) vs. hard state
8. Failover (may a session be redirected to alternative box?) vs. restart

Of 22 classes of Middleboxes:

- 17 are application or multi-layer
- 16 are implicit
- 17 are multi-hop
- 21 are in-line; call-out is rare
- 18 are functional; pure optimisation is rare
- Routing & processing evenly split
- 16 have hard state
- 21 must restart session on failure

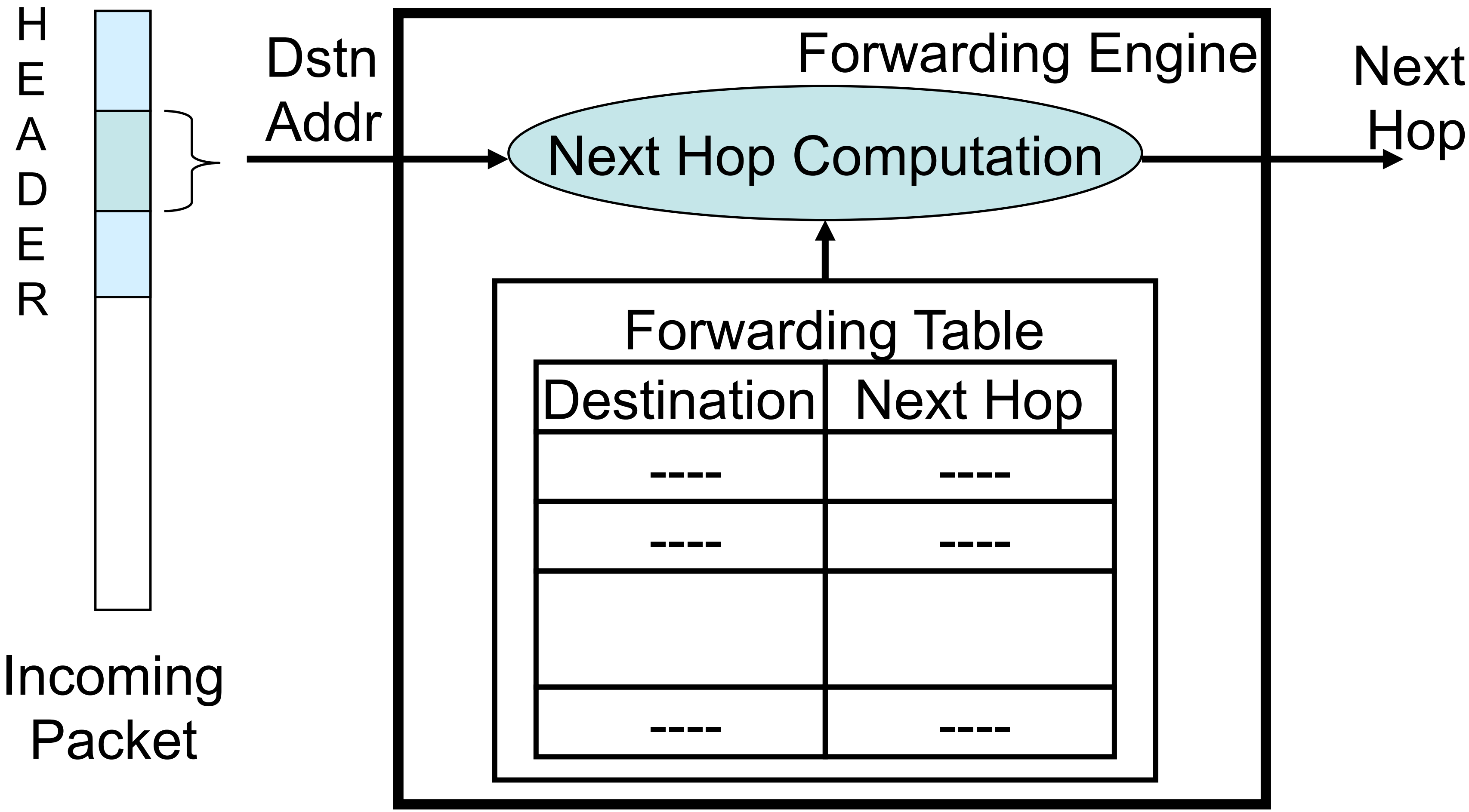
Assessment

- Although the rise of middleboxes has negative impact on the end to end principle at the packet level, it is still a desirable principle of applications protocol design.
- Future application protocols should be designed in recognition of the likely presence of middleboxes (e.g. network address translation, packet diversion, and packet level firewalls)
- Approaches for failure handling needed
  - soft state mechanisms
  - rapid failover or restart mechanisms
- Common features available to many applications needed
  - Middlebox discovery and monitoring
  - Middlebox configuration and control
  - Routing preferences
  - Failover and restart handling
  - Security

Chair for Network Architectures and Services – Prof. Carle
Department for Computer Science
TU München

The Evolution of IP Routers and Middleboxes

IP Router: *Lookup*



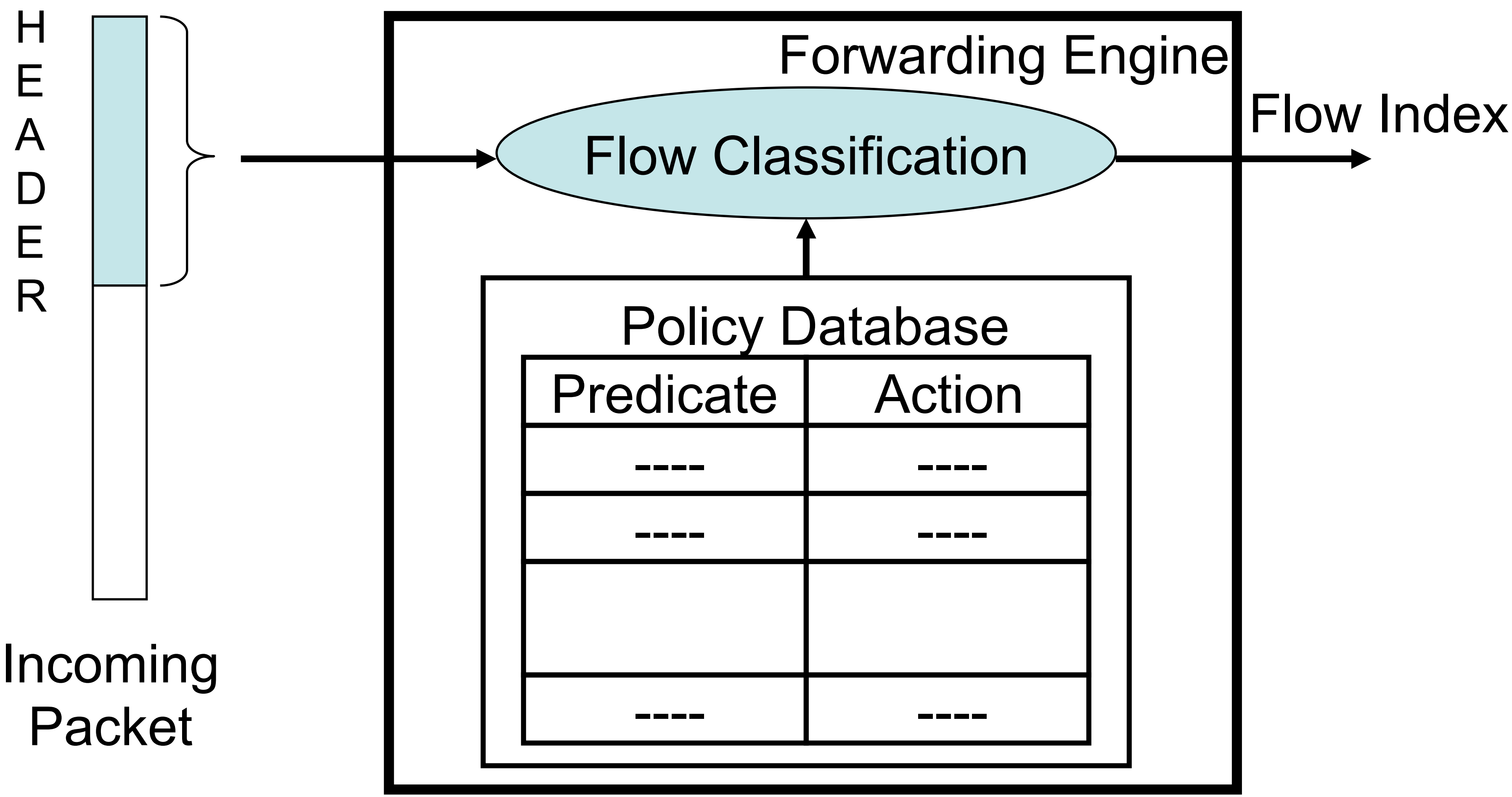
IPv4 unicast destination address based lookup

Need more than IPv4 unicast lookups

- IPv6
  - 128-bit destination address field

- Multicast
  - PIM-SM (Protocol-Independent Multicast, Sparse Mode)
    - Longest Prefix Matching on the source (S) and group (G) address
    - Start specific, subsequently apply wildcards: try (S,G) followed by (*,G) followed by (*,*,RP)
    - Check Incoming Interface
  - DVMRP:
    - Incoming Interface Check followed by (S,G) lookup

Flow Classification



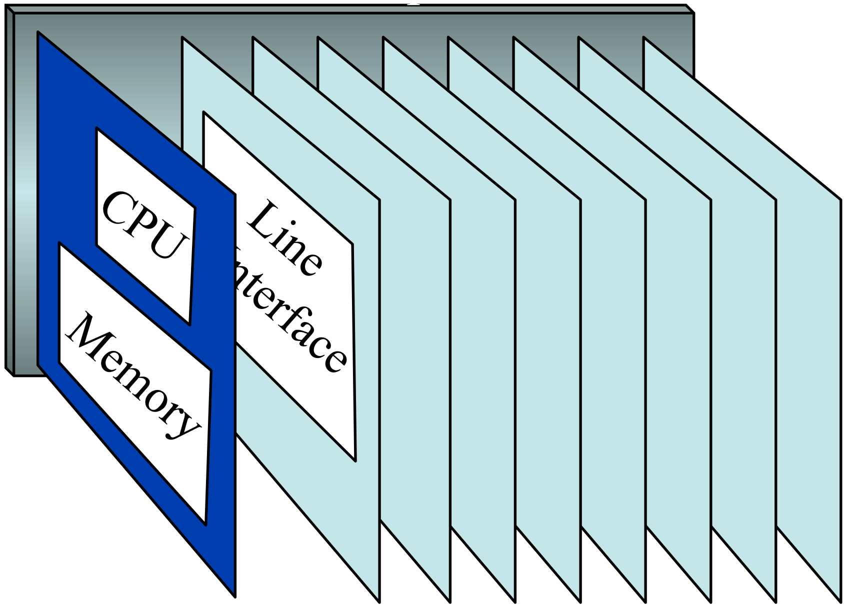
Providing Value-Added Services: Some examples

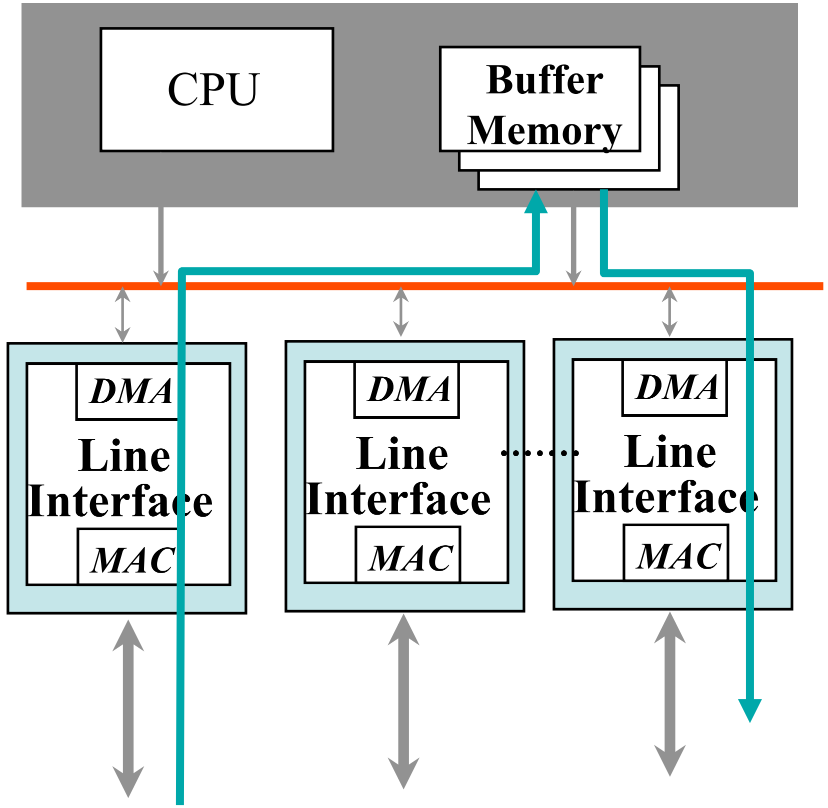
- Differentiated services
  - Regard traffic from AS#33 as `platinum-grade'
- Access Control Lists
  - deny tcp host 1.1.1.1 eq 68 host 2.2.2.2 eq 34
- Committed Access Rate
  - Rate limit WWW traffic from interface#739 to 10Mbps
- Policy-based Routing
  - Route all voice traffic through specific MPLS path
- Peering Arrangements
  - Restrict the total amount of traffic of precedence 7 from MAC address N to 20 Mbps between 10 am and 5pm
- Accounting and Billing
  - Generate hourly reports of traffic from MAC address M

Network Technology and Packet Rate

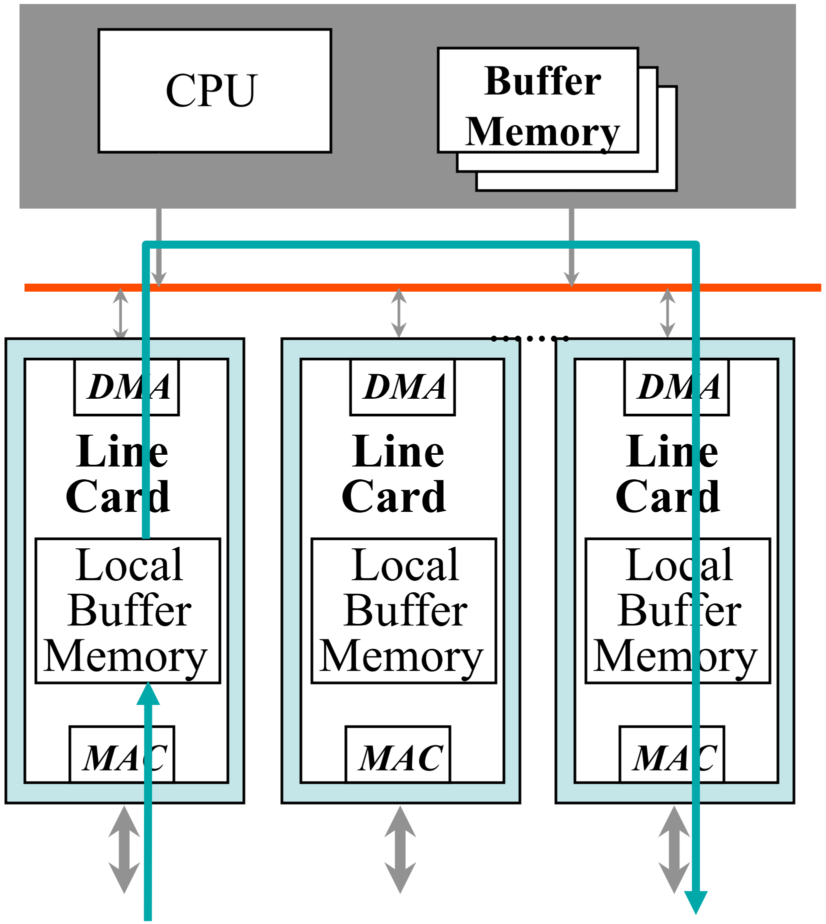
Data rate	Transm. Delay (1kbyte)	Transm. Delay (125 byte)	Packet Rate (125 byte)	CPU cycles per packet (1 GHz)
1 Mbit/s	8 ms	1 ms	1 Kpps	10^6
10 Mbit/s	0,8 ms	100 us	10 Kpps	100.000
100 Mbit/s	80 us	10 us	100 Kpps	10.000
1 Gbit/s	8 us	1 us	1 Mpps	1.000
10 Gbit/s	0,8 us	100 ns	10 Mpps	100
100 Gbit/s	80 ns	10 ns	100 Mpps	10

First-Generation IP Routers

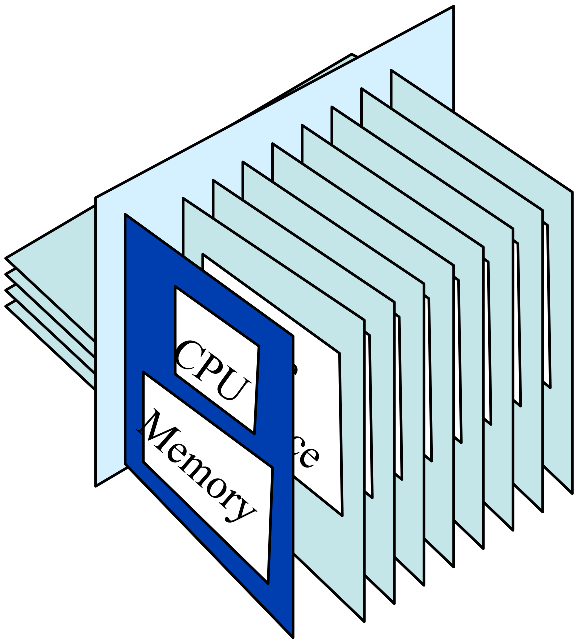


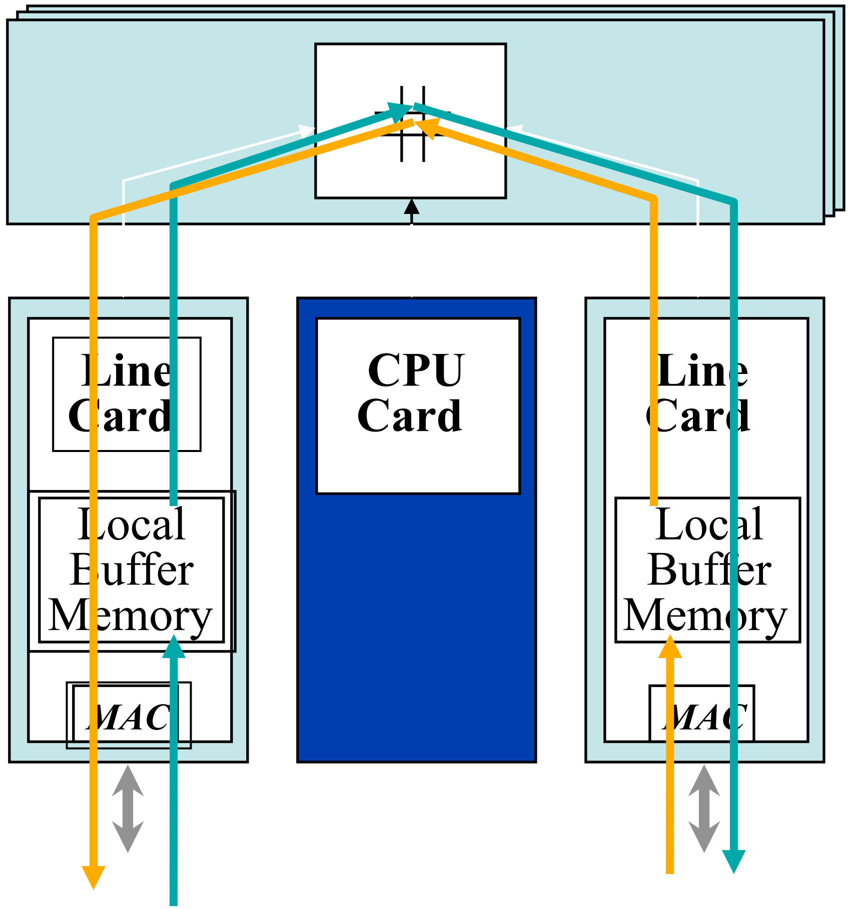


Second-Generation IP Routers



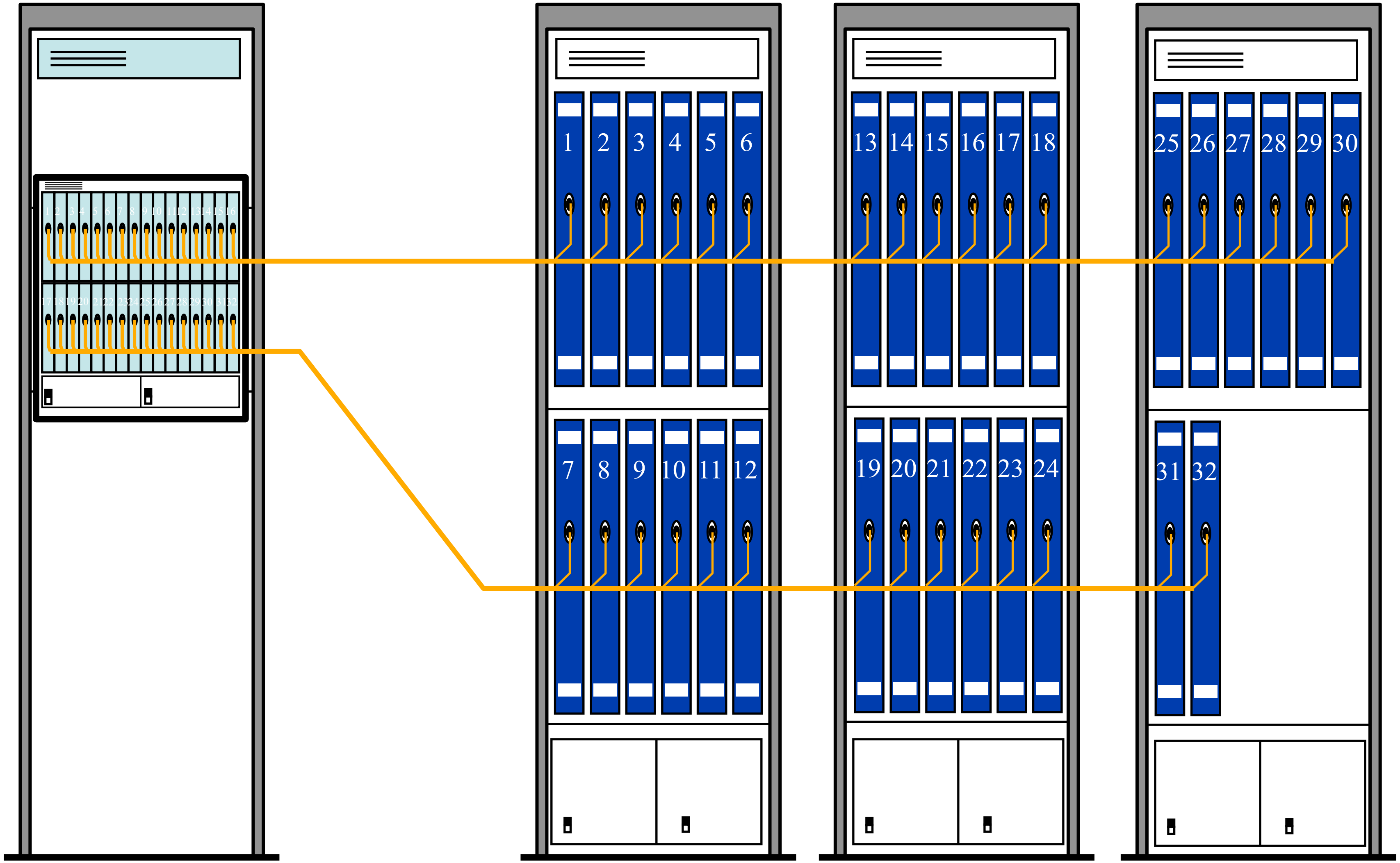
Third-Generation Switches/Routers





Fourth-Generation Switches/Routers

Clustering and Multistage



Research Issues

Internet Trends and Innovative Concepts

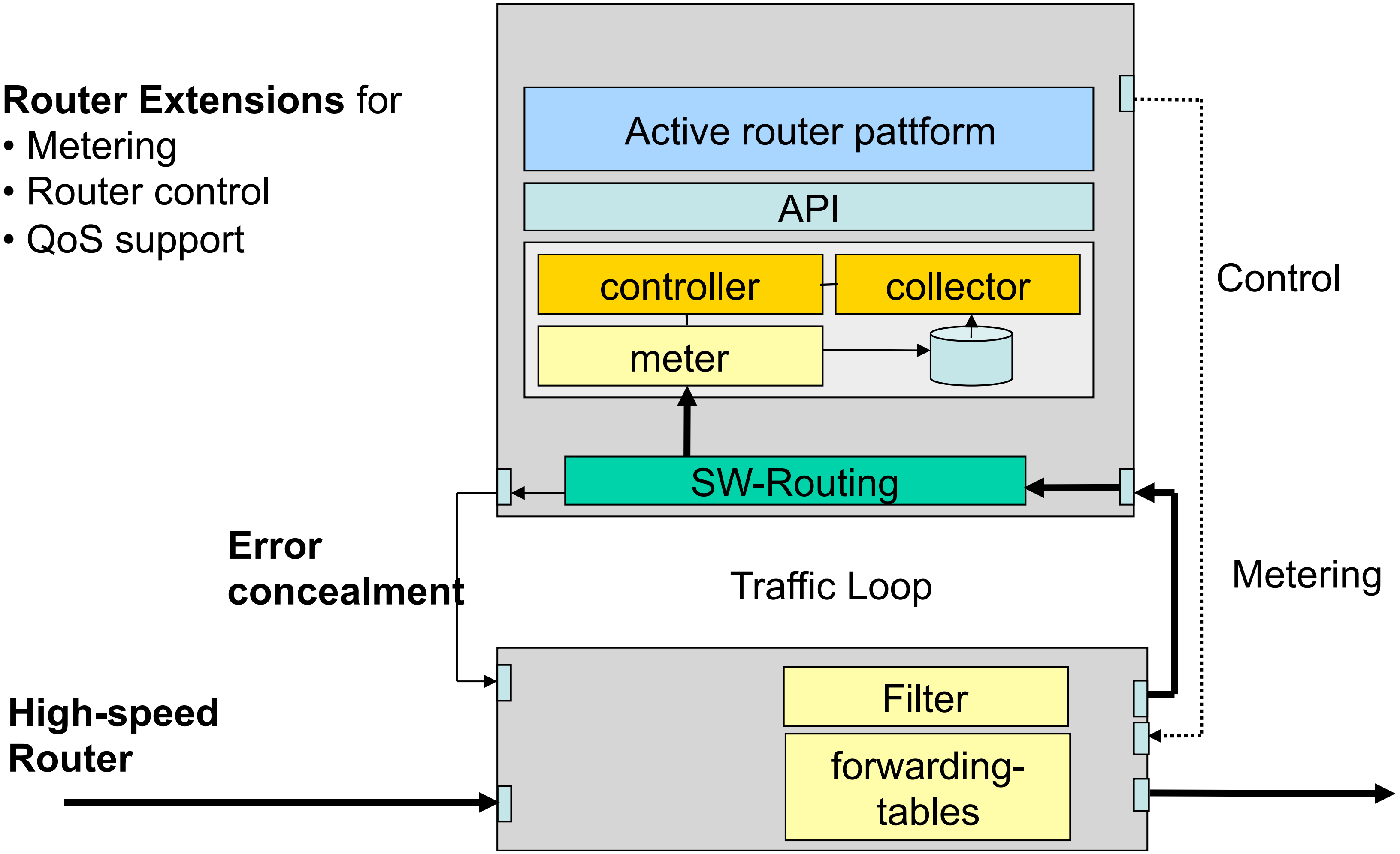
- Innovative approaches
  - Inspection-and-action boxes (Katz - UC Berkeley)
  - Knowledge plane (Clark - MIT)
  - Autonomic Networking (c.f. Dagstuhl perspectives seminar: Carle, Katz, Plattner)
  - NSF GENI (Global Environment for Networking Innovations)
    FIND (Future Internet Network Design)

- Relevant components
  - Instrumentation of the network
  - Intelligent processing
  - Initiating actions based on derived information
  - ⇨ Concept „Measuring – Processing – Reacting“

- Example use case
  - Quality improvements for Internet telephony

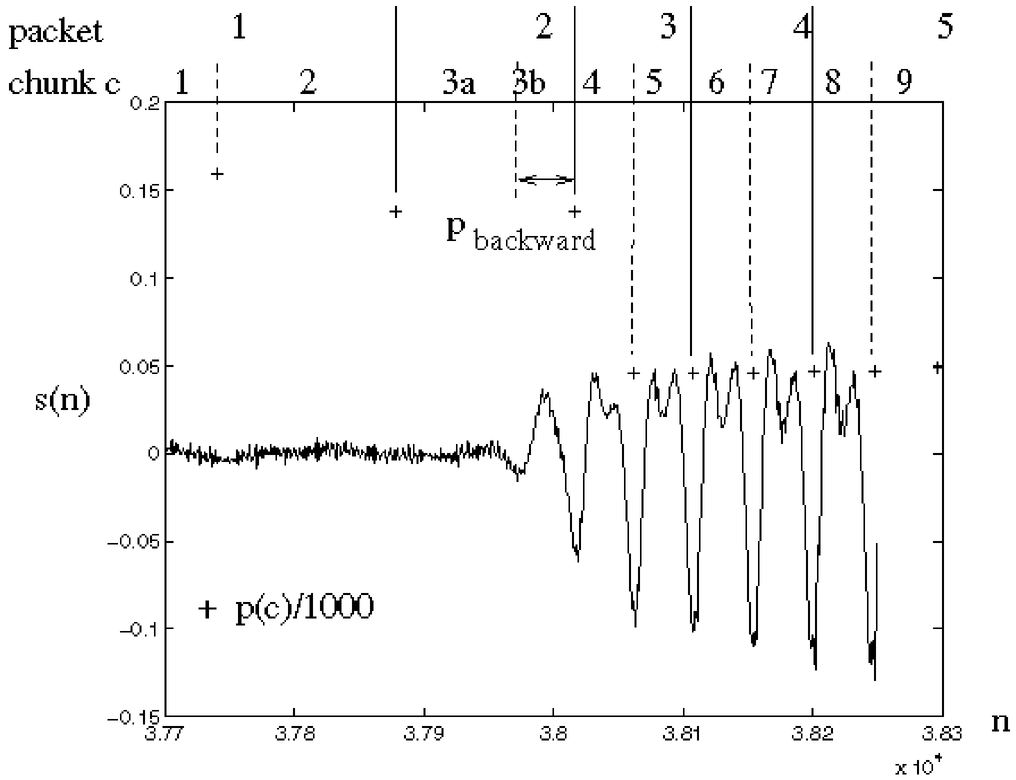
Active High-speed Router

Router Extensions for
- Metering
- Router control
- QoS support



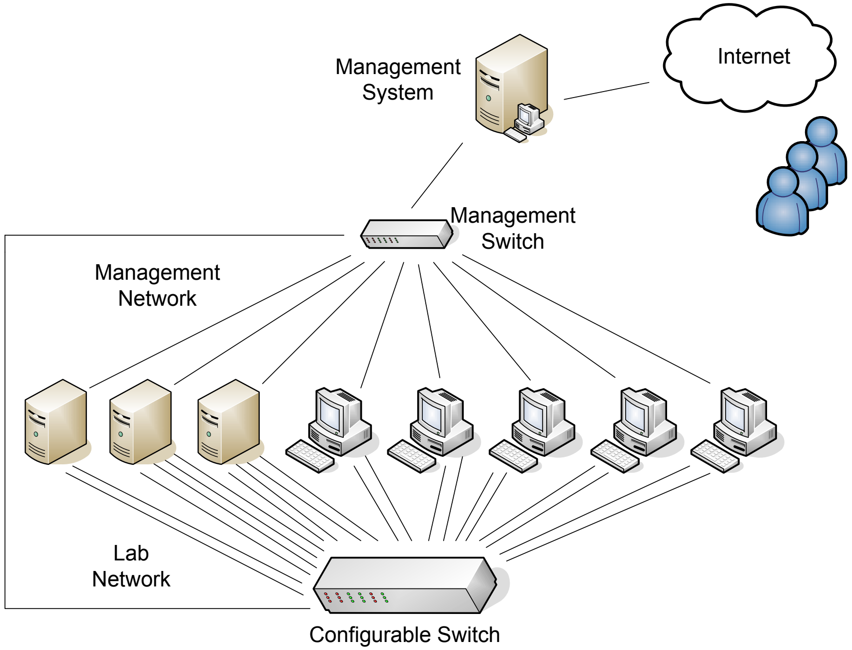
Sender: Adaptive Packetization

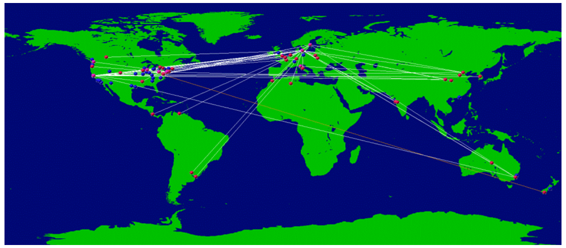
- Autocorrelation of audio signal
  - Partitioning into segments (chunks)
- Analysis of voice content
  - Detection of symmetry
  - Detection of unvoiced/voiced transitions



Evaluation

- Reproducable experiments in testbed
  - Virtual network lab, configurable topology
  - Experiments: artificial attack traffic for assessment of algorithms for attack detection

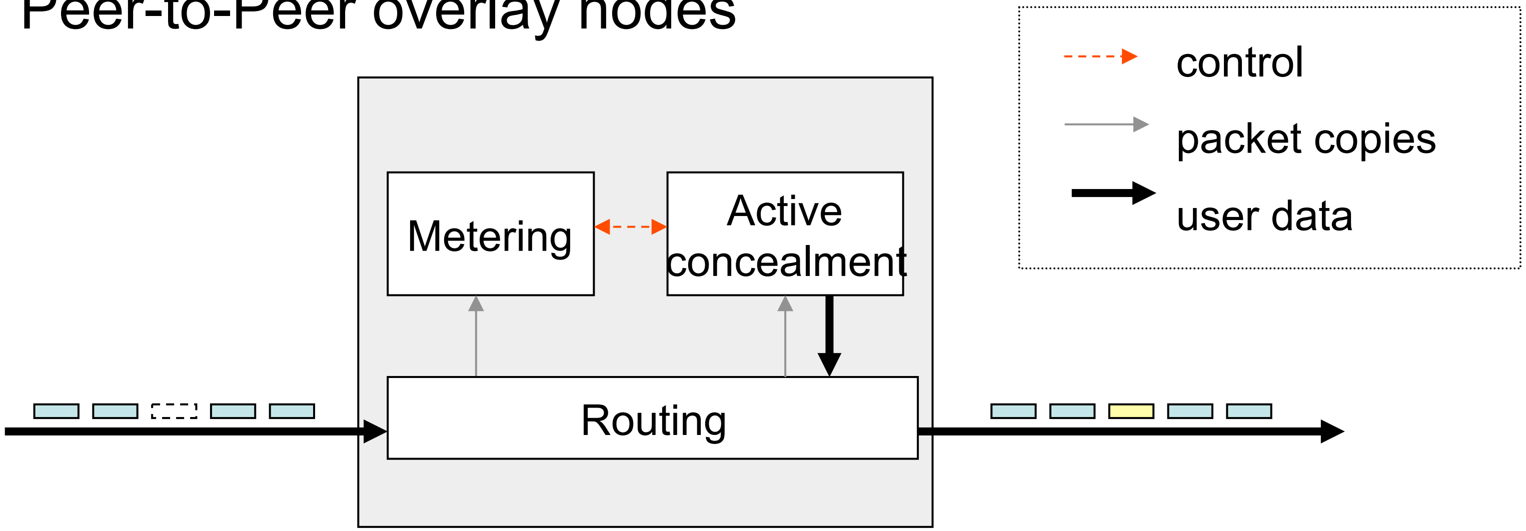




- experiments in global Internet (Planetlab etc.)

Error Concealment for Internet Telephony

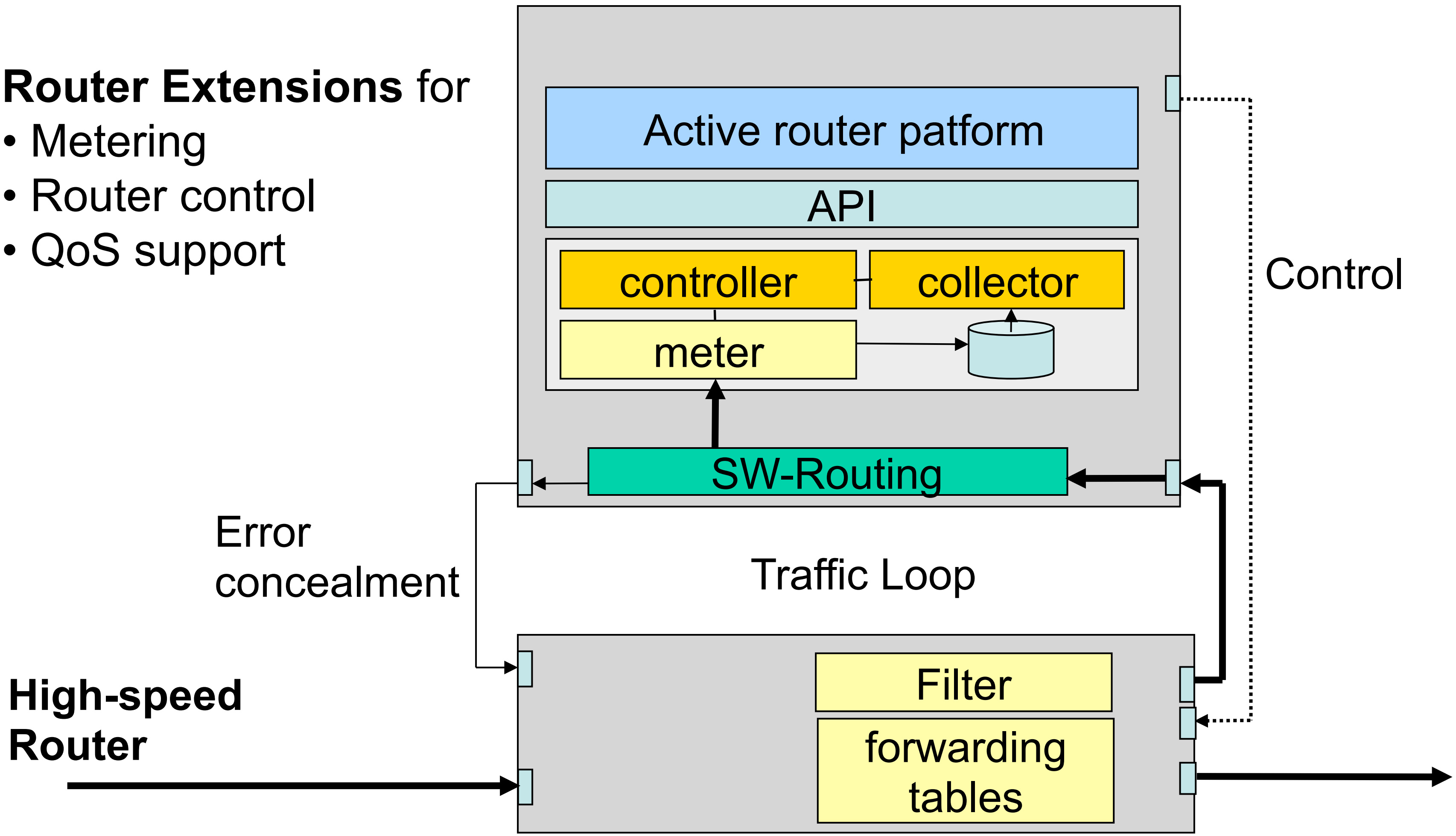
- Problem
  - Packet loss due to congestion
- Approach
  - Error concealment
- Implementation alternatives
  - Active routers
  - Peer-to-Peer overlay nodes



⇨„Measuring – Processing – Reacting“ for Internet Telephony

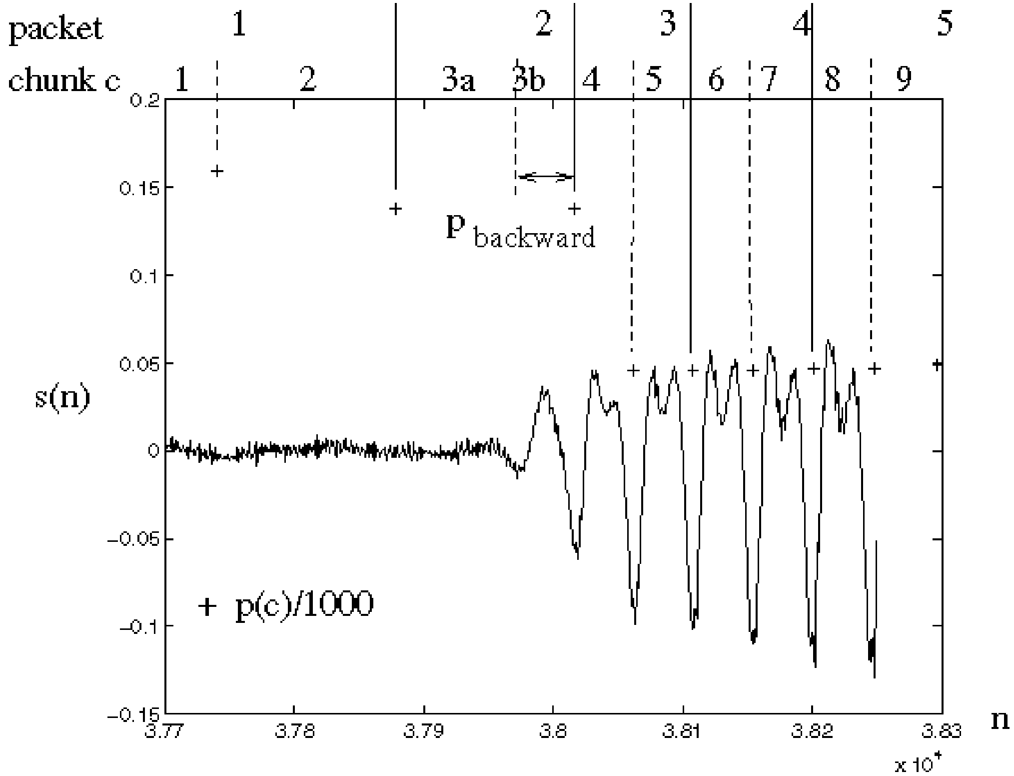
Architecture for Active High-speed Router

Router Extensions for
- Metering
- Router control
- QoS support



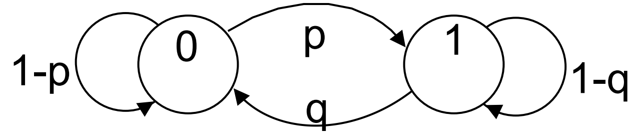
Sender: Adaptive Packeting

- Autocorrelation of audio signal
  - Partitioning into segments (chunks)
- Analysis of voice content
  - Detection of symmetry
  - Detection of unvoiced/voiced transitions
- packetisation
  - 2 Segments/packet



Performance Assessment of Error Concealment

- Error modelling with Gilbert model



- Analytical method for speech quality assessment