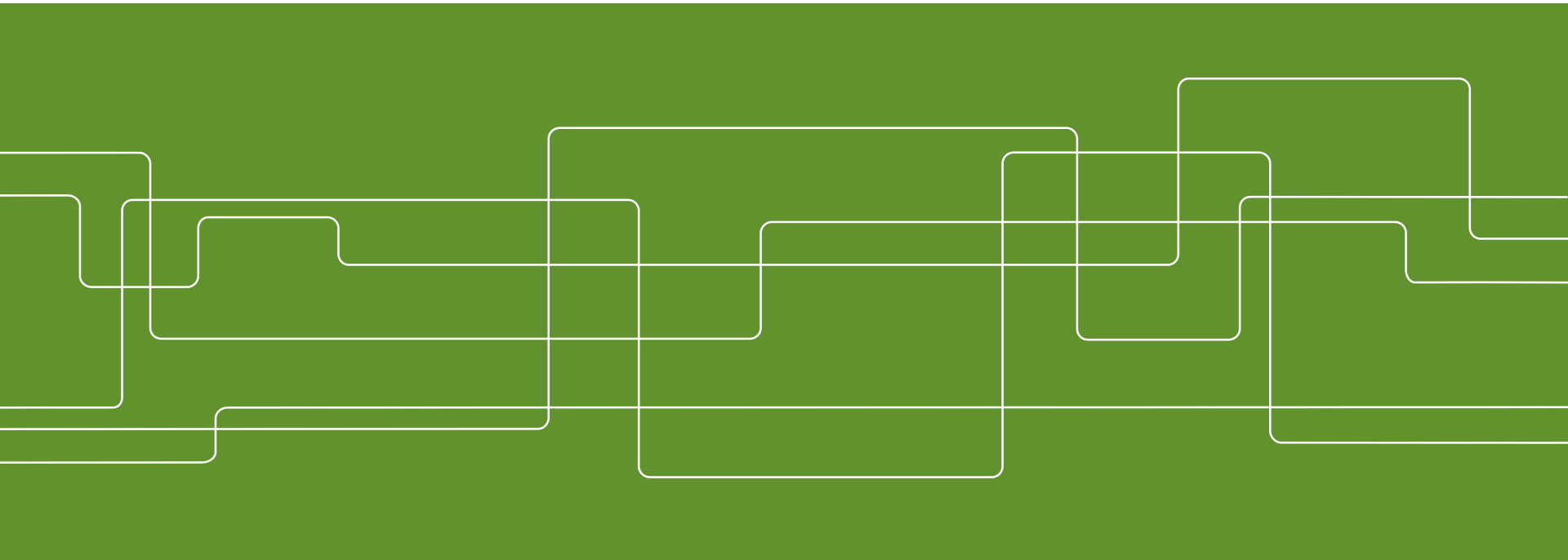
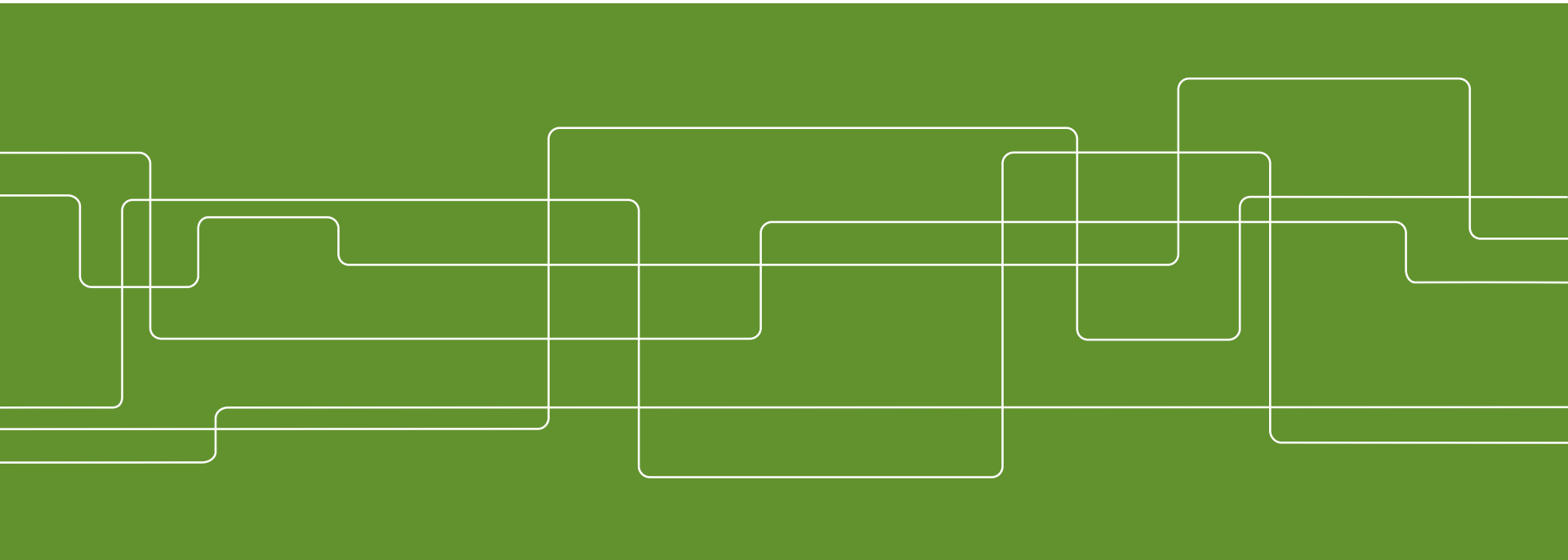
IK2215 Advanced Internetworking

Lecture 10—IPv6 and IP QoS

Markus Hidell



Part I: IPv6



Part I: IPv6

Covered briefly in the course book

- 6th ed: Section 4.4.4
- 7th ed: Section 4.3.5

IPv6

Changes since IPv4 was developed (mid 70's)

- Provider market has changed dramatically
- Immense increase in user and traffic on the Internet
- Rapid technology advancement
- Bandwidth increase from kb/s to Tb/s

IPv4 issues

- Too few addresses (though only 3-7% of address space used)
- Too large routing tables

To address these issuees IETF has standardized IPv6

- IPv6 should keep most of the characteristics of IPv4 (good design)
- Changing the address fields is the big thing with IPv6
- While modifying the header, improvements have been introduced

IPv6 vs IPv4

Changes in IPv6 compared to IPv4

- 128 bit addresses
- extended address hierarchy
- simplified header
- simpler and better support for options
- possible to extend the protocol
- support for autoconfiguration (plug-and-play)
- support for QoS treatment
- host mobility
- security
- provider selection
- no fragmentation in routers

IPv6 Simplifications

Fixed format headers

- Use extension headers instead of options

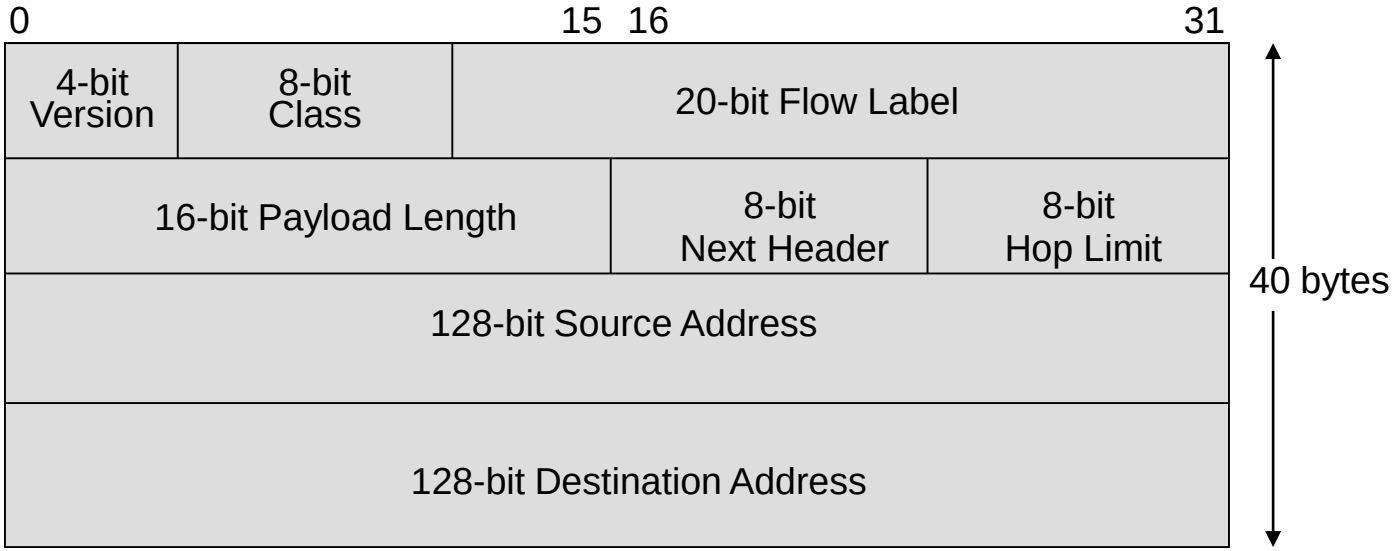
Remove header checksum

- Rely on link layer and higher layers to check integrity of data

Remove hop-by-hop segmentation

- Fragmentation only by sender due to path MTU discovery

IPv6 Header



Version	Only field identical to IPv4. Code is 6 in IPv6
Class	New field. Revised concept of priority bits. Facilitates handling of real-time traffic.
Flow Label	New field. To distinguish packets requiring the same treatment.
Payload Length	Replaces *length* field in IPv4. Gives length of data following IPv6 header
Next Header	Replaces *protocol* field in IPv4. Extension headers can be used.
Hop Limit	Replaces *TTL* field in IPv4. Hop limit more accurately reflects the use of TTL.
Src Address	Revised *source address* field. 128 bits in IPv6 vs 32 bits in IPv4.
Dst Address	Revised *destination address* field. 128 bits in IPv6 vs 32 bits in IPv4.

IPv6 Addresses

An IPv6 unicast address identifies an interface connected to an IP subnet (as is the case in IPv4)

One big difference between IPv6 and IPv4 is that IPv6 routinely allows each interface to be identified by several addresses

- facilitates management

IPv6 has three address categories:

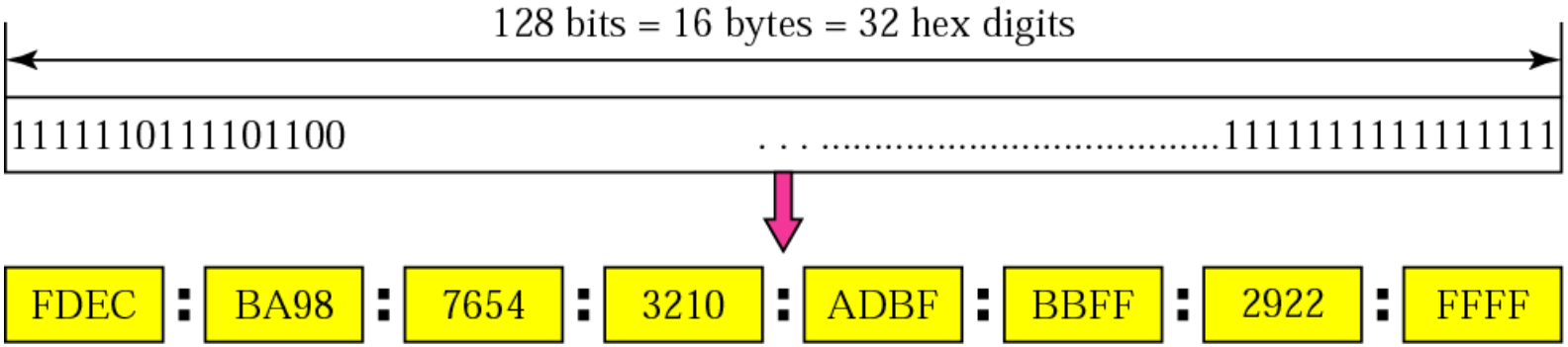
- unicast - identifies exactly one interface
- multicast - identifies a group; packets get delivered to all members of the group
- anycast - identifies a group; packets normally get delivered to nearest member of the group

128 bits results in 2^{128} addresses

- Distributed over the Earth: 665,570,793,348,866,943,898,599/m^2
- Pessimistic estimate with hierarchies: ~1,564 addresses/m^2

IPv6 Address Format

Colon hexadecimal notation (eight 16 bit hexadecimal integers)



IPv6 Address Abbreviations and CIDR

Leading zeros may be oppressed

- FDEC:BA98:0074:3210:000F:BBFF:0000:FFFF □
- FDEC:BA98:74:3210:F:BBFF:0:FFFF

Zero compression: one of a series of zeros may be replaced by ::

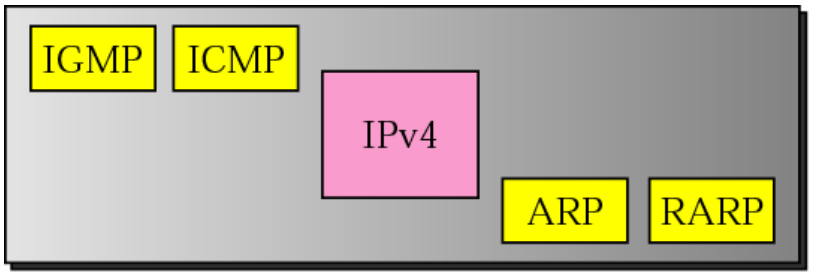
- But only once
- FDEC:0:0:0:0:BBFF:0:FFFF □
- FDEC::BBFF:0:FFFF

CIDR notation to specify the first N bits of an address

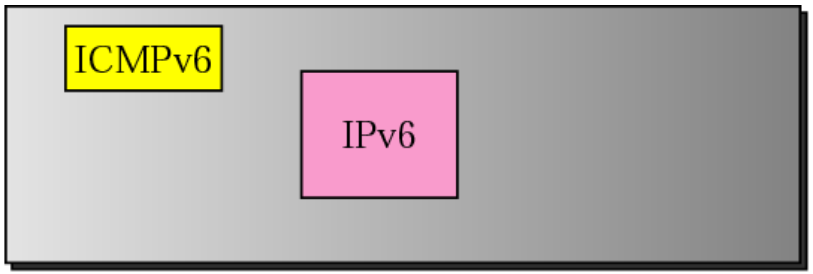
- FDEC:0:0:0:0:BBFF:0:FFFF/60

Network Layer Comparison—IPv4 vs IPv6

- ICMPv4 has been modified to be more suitable for IPv6, and thus updated to ICMPv6
- ARP and IGMP in version 4 are now part of ICMPv6
- RARP has been dropped due to limited use (DHCP does the job of RARP)
- As in ICMPv4, ICMPv6 messages are divided into 2 categories:
- Error-reporting (somewhat different messages in v6 vs v4)
- Query (rather different messages in v6 vs v4)



Network layer in version 4



Network layer in version 6

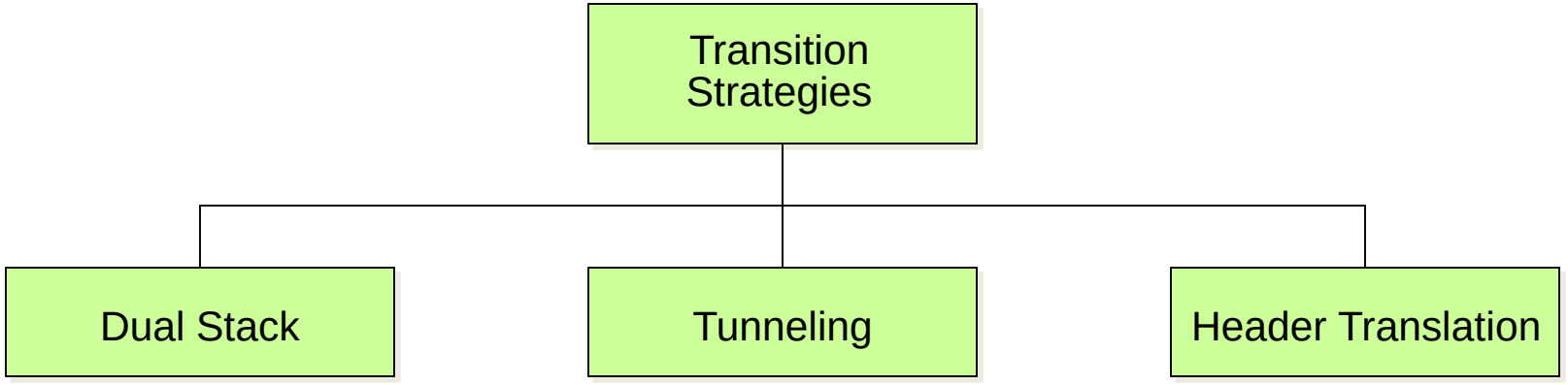
ICMPv4 vs ICMPv6

Error Report Message – Type	Ver 4	Ver 6
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Query Message – Type	Ver 4	Ver 6
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbour solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes

Transition from IPv4 to IPv6

- Because of the large number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly
- Transition should be smooth to prevent problems
- Transition strategies have been devised by IETF



IPv6 Summary

IPv6 has:

- 128-bit address space
- revised header format
- new options
- allowance for extension
- support for special handling of packet flows
- increased security measures

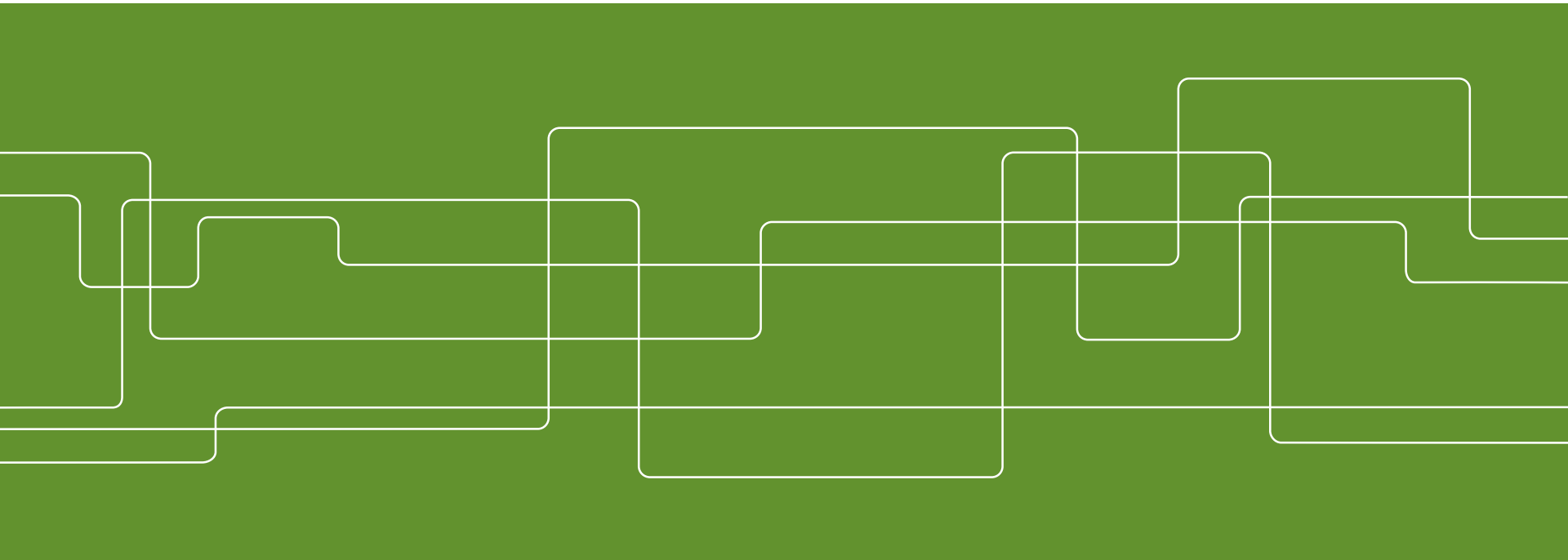
IPv6 uses hexadecimal colon notation with abbreviation methods

IPv6 has three address types: unicast, anycast, and multicast

IPv4, ICMPv4, ARP, RARP, and IGMP replaced with IPv6 and ICMPv6

IPv4 to IPv6 transition strategies are dual-stack, tunneling, and header translation

Part II: IP QoS



Outline

- IP QoS (Quality of Service)
- Integrated Services (int-serv)
  - With RSVP signalling
- Differentiated Services (diff-serv)

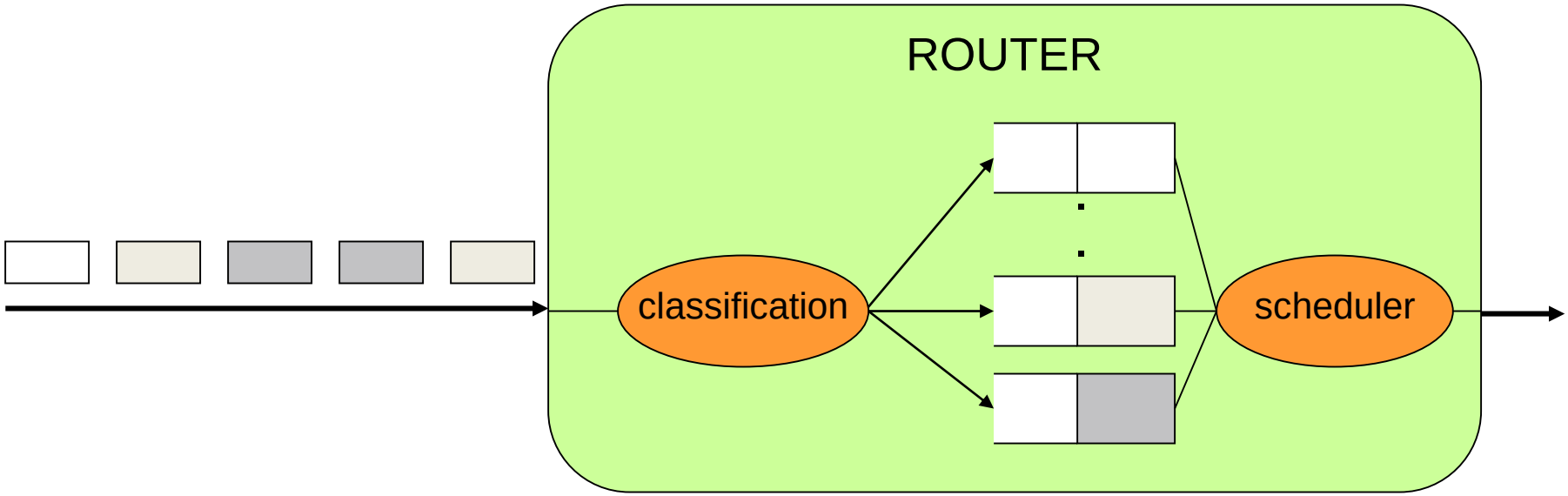
Traditional IP Networking

Vanilla flavor of IP networking:

- Connectionless best effort service
- Each packet treated independently by routers (stateless)
- Route lookup based on dst IP address and longest prefix match (LPM)
- No bandwidth guarantees
- Delay variations introduced along the path of a packet

Advanced Packet Handling

- Identifying packet flows in the network – classification
- Give special treatment to each flow – higher degree of service than BE
- After classification packet is placed in a certain queue for an outgoing interface
  - More specific than the traditional route lookup (dst IP addr and LPM)

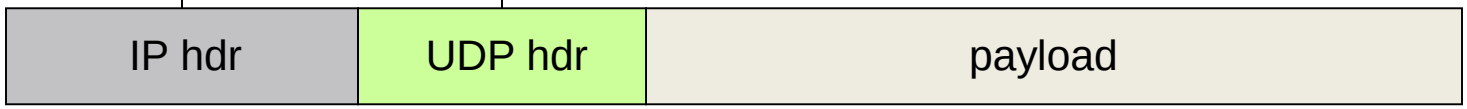


Defining a Flow

- To recognize flows in the network, classification information is needed
- This information typically includes src/dst IP addresses and src/dst port numbers
- Routers along the path examines both IP and transport level headers to identify the flows

A flow can be specified by any combination of these fields

- Src IP addr
- Dst IP addr
- Protocol field
- ToS field
- Src port number
- Dst port number



Other types of classification info:

- **Link level header information**
- **VLAN header information**
- **Other extra headers that can exist**

Functions in IP QoS

Classification

- Identifying the packets belonging to a certain traffic flow

Policing

- Ensure that the flow conforms to a traffic specification

Shaping

- Smoothing out packet bursts (traffic is often bursty)

Scheduling

- Manage packets in queues so that they receive desired service

Admission control

- Check that there are enough resources to accept a new traffic flow

IP QoS

There are basically two approaches by IETF for IP QoS:

Integrated services (int-serv)

- QoS architecture produced by IETF in the mid 1990s
- End-to-end guarantees for applications
- Uses a signaling protocol, RSVP, to make requests for QoS
- Includes service class definitions

Differentiated services (diff-serv)

- Introduced by IETF in the late 1990s
- More coarse-grained model of QoS
- Allocates resources on a per class basis

Integrated Services (int-serv)

RFC 2210

Reservation of resources (bandwidth, buffers) for application flows

- src/dst IP addr, IP protocol field, UDP/TCP port numbers

RSVP is used to signal the reservation for each flow

Three *service classes* for applications to choose from

- Guaranteed Service – hard guarantee of bandwidth and delay
- Controlled Load – lower level of service
- Best Effort – traditional IP service

Guaranteed Service

RFC 2211

Guarantees:

- Bounded delay
- Bandwidth
- No loss

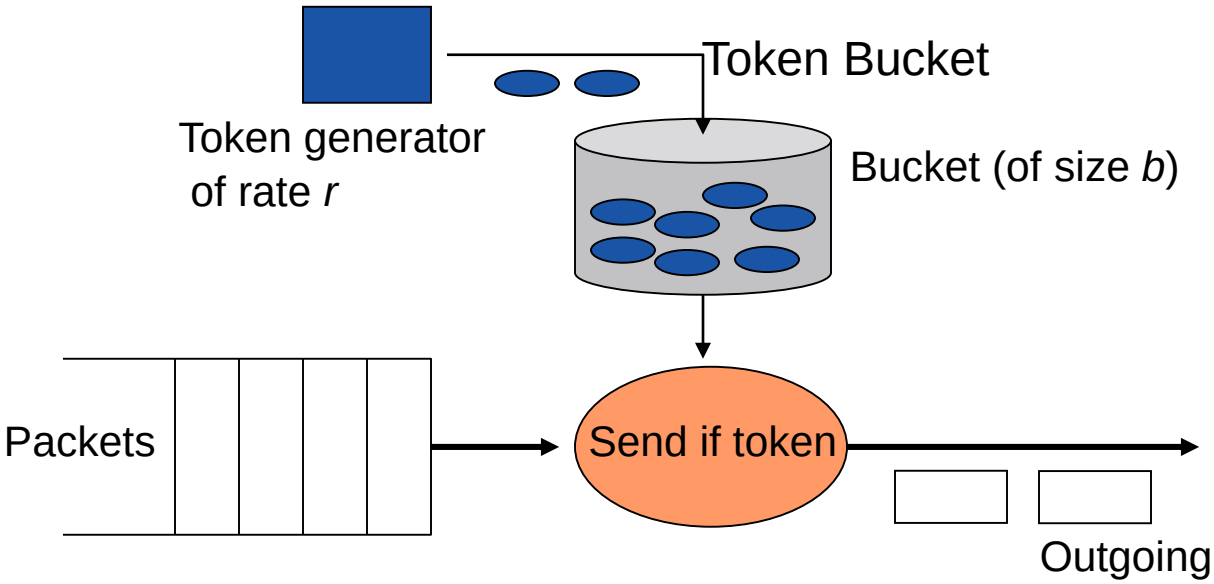
Traffic described with:

- Peak rate
- Max packet size
- Burst size
- Token bucket rate

Guaranteed service comes at a cost:

- Every flow using the service must be queued separately
- Often results in rather low network utilization

- Token bucket specification is a standard way to represent the bandwidth characteristics of an application that generates data at variable rate
- A traffic flow is characterized by a token bucket of rate *r* and burst size *b*, if for any time interval *T*, it sends no more than *rT* + *b* bytes



In practice, token bucket is implemented in bytes. In order to send a packet of n bytes, n tokens are needed

Controlled Load

RFC 2212

Lower cost compared to guaranteed service

Approximation of best effort in a lightly loaded network

Network elements (routers) ensure that

- There are enough resources to provide the specified QoS (admission control)
- The flows are queued and scheduled in a way that prevents other flows from degrading with their performance (all end-to-end flows do not have to be queued separately)

Resource Reservation Protocol

RSVP is a network control protocol used to express QoS requirements

- RSVP binds a QoS request to a flow
- RSVP does not carry application data
- RSVP is an important component in IETF int-serv
- RSVP is also used in other scenarios
  - Traffic Engineering/VPN

RSVP delivers QoS reservations along a path from source to destination(s)

- no routing—routing protocol computes path
- uses soft state—periodical refresh

RSVP Building Blocks

RSVP must carry the following information:

Classification information

- Flows with QoS requirements must be recognized within the network
- Includes src and dst IP addr and UDP/TCP port numbers

Traffic specification (TSpecs)

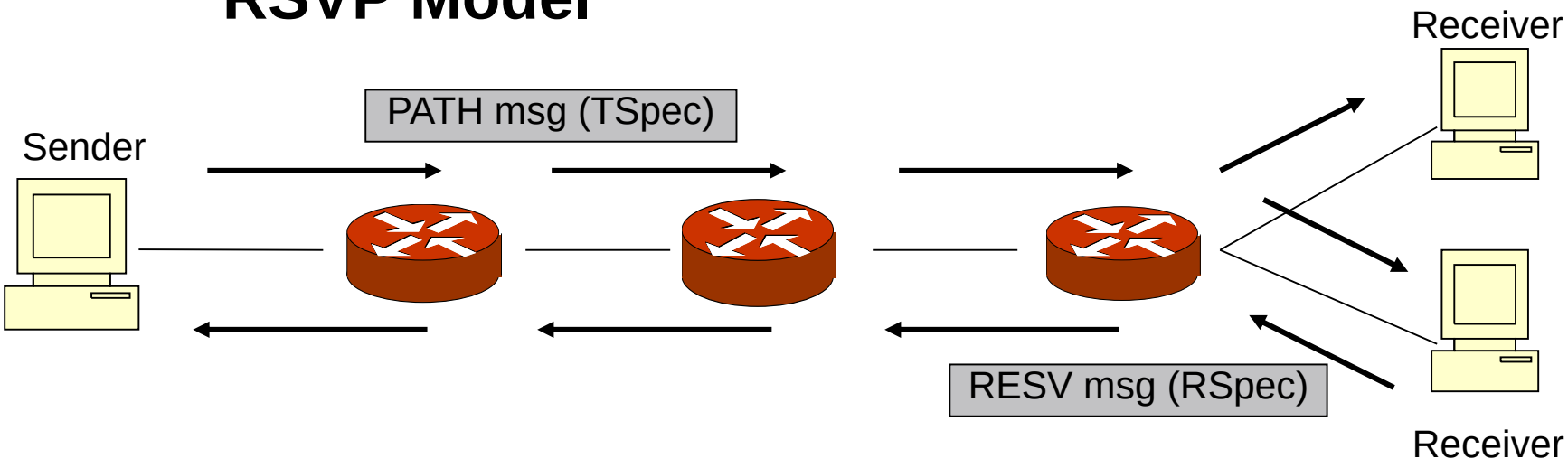
QoS requirements (RSpecs)

- Including desired service (guaranteed or controlled load)

from hosts to all routers along the path

RSVP is explicitly designed to support multicast.

RSVP Model



Two types of messages to set up reservations:

- PATH message
  - From a sender to one or several receivers, carrying TSpec and classification information provided by sender
- RESV message
  - From receiver, carrying RSpec indicating QoS required by receiver
- Token bucket specification used in these messages

RSVP Model cont'd

A reservation is unidirectional

The messages (PATH and RESV) are intercepted and forwarded by each router along the path

Each router must take actions to do the actual resource allocation at each hop

Once the reservation has been established, the routers along the path recognize packets by inspecting IP and transport protocol headers

The packets recognized in this way is called a *reserved flow*

Different receivers of the packet flow can have different QoS requirements (RSpecs)

Functionality

RSVP is receiver oriented protocol.

- The receiver is responsible for requesting reservations.

RSVP handles heterogeneous receivers.

- Hosts in the same multicast tree may have different capabilities and hence need different QoS.

RSVP adapts to changing group membership and changing routes.

- RSVP maintains “Soft state” in routers. The only permanent state is in the end systems. Each end system sends their RSVP control messages to refresh the router state.
  In the absence of refresh message, RSVP state in the routers will time-out and be deleted.

RSVP is **not** a routing protocol.

- A host sends IGMP messages to join a multicast group, but it uses RSVP to reserve resources along the delivery path(s) for that group.

RSVP Soft State

RSVP maintains “soft state” in hosts and routers

Any state will automatically expire after some time unless it is refreshed periodically

Routing protocol determines path dynamically

Soft state is created by PATH and RESV messages

Soft state is refreshed by PATH and RESV messages

Time-outs clean up reservations

RSVP and Router Operation

At each node, RSVP applies a local decision procedure “admission control” to the QoS request. If the admission control succeeds, it sets the parameters to the classifier and the packet scheduler to obtain the desired QoS. If admission control fails at any node, RSVP returns an error indication to the application.

Each router in the path capable of resource reservation will pass incoming data packets to a packet classifier and then queue these packet in the packet scheduler. The packet classifier determines the route and the QoS class for each packet. The scheduler allocates a particular outgoing link for packet transmission.

The packet scheduler is responsible for negotiation with the link layer to obtain the QoS requested by RSVP. The scheduler may also negotiate a “CPU time”.

RSVP Summary

RSVP supports multicast and unicast data delivery

RSVP adapts to changing group membership and routes

RSVP reserves resources for simplex data streams

RSVP is receiver oriented, i.e., the receiver requests resources (note that IP multicast is receiver-oriented)

RSVP maintains a “soft-state” in routers

- supports gracefully dynamic memberships and automatically adapt to routing changes

RSVP provides several reservation models

RSVP is transparent for routers that do not support RSVP

Differentiated Services (diff-serv)

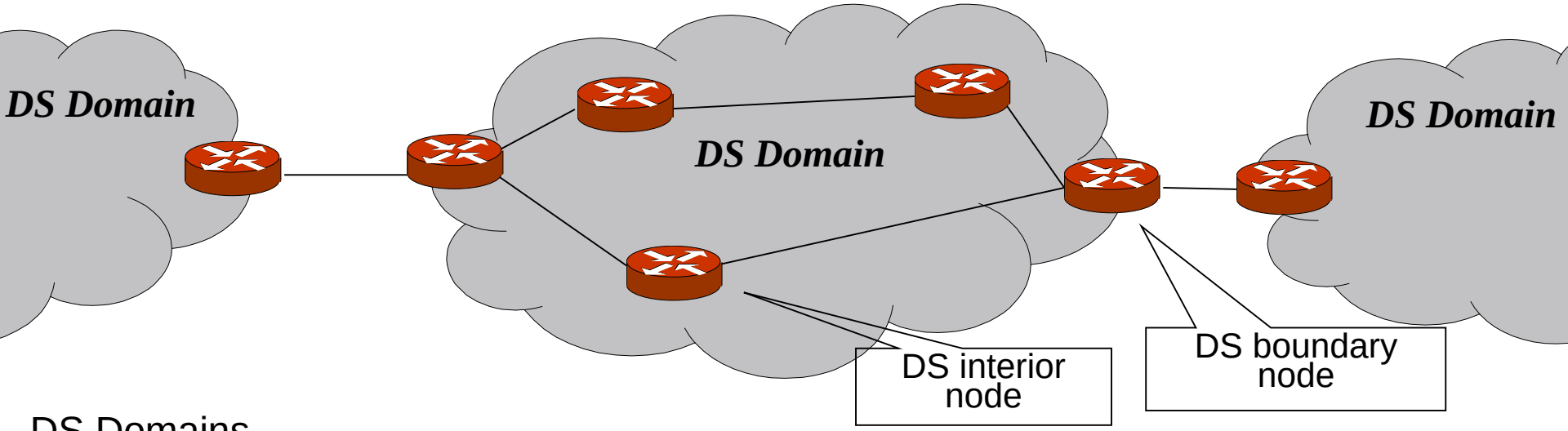
Many drawbacks of int-serv

- End-to-end connection setup and resource reservation
  - practically impossible: too many flows
- State per flow
  - costly for routers

IETF started to develop another approach—diff-serv—in 1998

- Basic definition: RFC 2474, RFC 2475
- Divide traffic into small number of classes, and allocate resources per class
- Use aggregates—No per-flow state
- Use Service Level Agreements (SLA) as contracts—No signaling
- Use bits in IP header—ToS field—to mark packets with their traffic class
- Flows are aggregated in the network so that routers need only distinguish between a small number of aggregated flows, even if those aggregates contain millions of individual flows.

Diff-Serv Architecture



DS Domains

- e.g., Autonomous Systems
- SLAs between domains: contract for diff-serv conditioning

DS Boundary Nodes

- Egress/Ingress

DS Interior Nodes

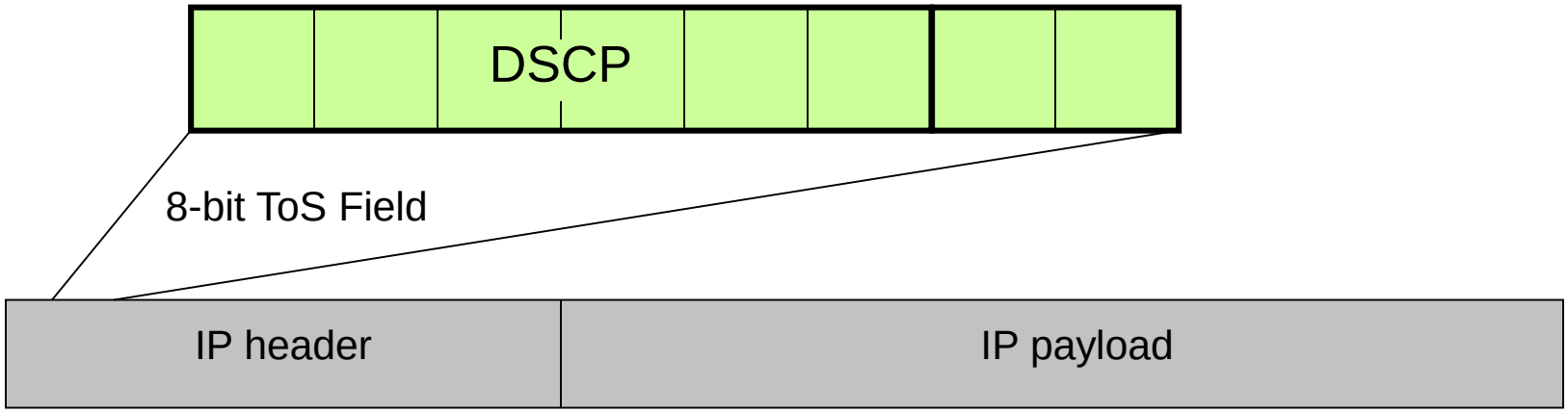
- Traffic is classified, marked and policed at ingress
- Resources are *provisioned* in the network
- Packets are queued, forwarded and dropped according to marking

Packet Marking and Aggregation

Each packet is marked with a DSCP (Differentiated Services Code Point) directly in the 8-bit IP ToS header field

- 6 bits used → 64 possible code points (in practice much less is used)
- Code points are unique within a domain – but may change at domain borders

An ingress node aggregates packets into *behavior aggregates*, each marked by a unique code point (DSCP)



Per Hop Behavior (PHB)

The code point is used to select a specific *Per-Hop Behavior* (PHB) within the domain

- A PHB defines how packet should be treated by the router
- Forwarding behavior: e.g. scheduling, shaping, etc

Standard PHBs include

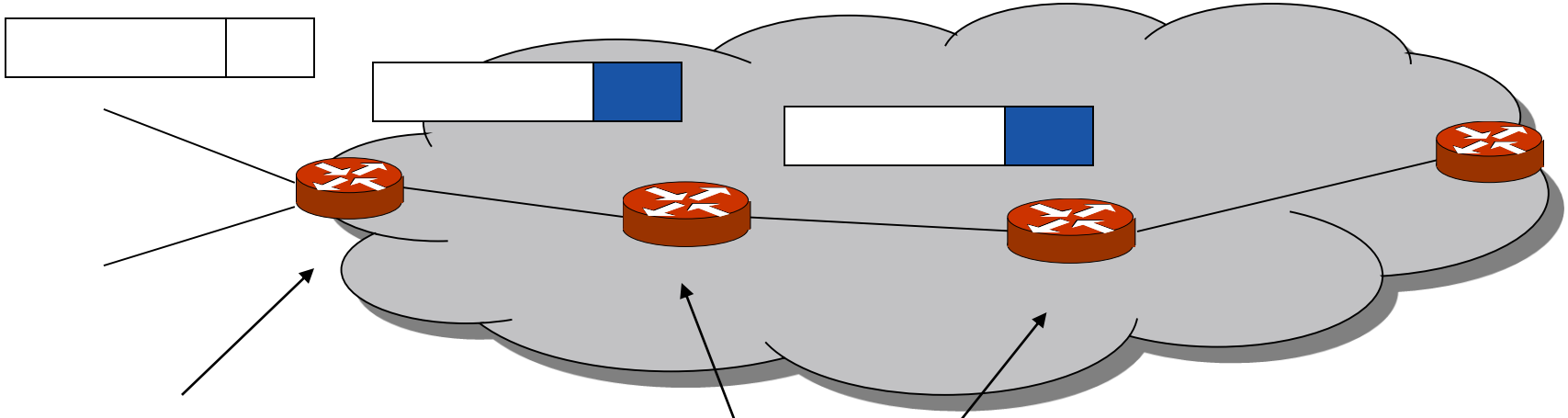
- Default (no special treatment)
- Expedited forwarding – EF (minimal delay, low loss)
- Assured forwarding – AF (four classes, separate queues, three levels of drop preferences)

PHB Group

- A set of PHBs with similar handling
- E.g. one traffic class with many levels of *drop priority (or drop preference)*

Set DSCP and Do PHB

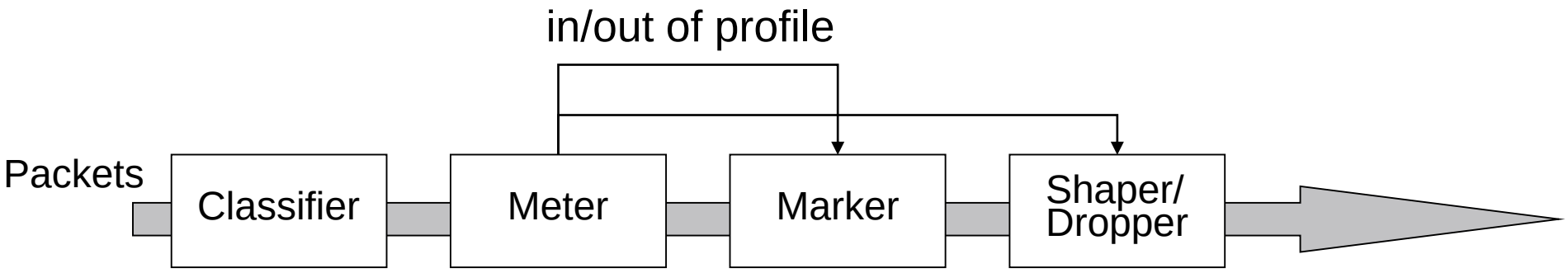
DSCP is generally set as packets cross an administrative domain



DCSP value set in packet based on local policy

PHB determined from DSCP; PHB determines QoS treatment

Traffic Conditioning



Typically, the ingress node performs traffic conditioning on incoming packets according to the SLA with the upstream domain

The metering measures the rate of packets. Consequences:

- mark packets in one PHB group to be in-profile/out-of-profile
- may result in shaping and dropping

Traffic Conditioning cont'd

Metering

- Token Bucket (specifies the traffic)
- Measure temporal properties (does traffic conform to spec)
- Three Color markers (RFC 2697, RFC 2698)

Policing: Shaping and dropping

- Delay packets to bring a stream into conformance
- Drop packets (several strategies can be used)
  - Tail drop
  - Random Early Discard

Expedited Forwarding PHB (EF)

RFC 3246

DSCP = 101110

EF marked packets should be forwarded with minimal delay and experience low loss

Specify a min-rate *r* and max-rate *R*

The rate at which the traffic is served at a given output interface should be at least *r* over a suitably defined interval, independent of the offered load of non-EF traffic to that interface.

Typically: separate queues for best effort and EF with strict priority

Assured Forwarding PHB Group (AF)

RFC 2597

Four *separate* classes

- Each class has a specific set of resources (buffers/bandwidth) and its own queue
- E.g. Gold, Silver, Bronze service

Three levels of drop preference in each class (green, yellow, red)

In-profile packets (green) get assured QoS

Out-of-profile packets get best effort sevice or are dropped

	Class 1	Class 2	Class 3	Class 4
Low Drop Pref	001010	010010	011010	100010
Medium Drop Pref	001100	010100	011100	100100
High Drop Pref	001110	010110	011110	100110

Summary

Basics about traditional best-effort IP and IP QoS

Two models proposed by IETF:

- Integrated services (int-serv)
  - Flow based – end-to-end signaling
- Differentiated services (diff-serv)
  - More coarse-grained, less complex, using service classes

Resource reservation protocol (RSVP)

Course Wrap-up

Start working on problem solving

- Old exams on course web
- Same structure of written exam (Part A + Part B), but new problems ;-)

Register for the exam

Course evaluation

- On-line in KTH Social
- Please fill out at the end of the course (after the exam)
- Course panel has been established
  - Muhammad Mahad Mufti (mmmufti@kth.se)
  - Parumita Saha (parumita@kth.se)