CS 524 A

Introduction to Cloud Computing

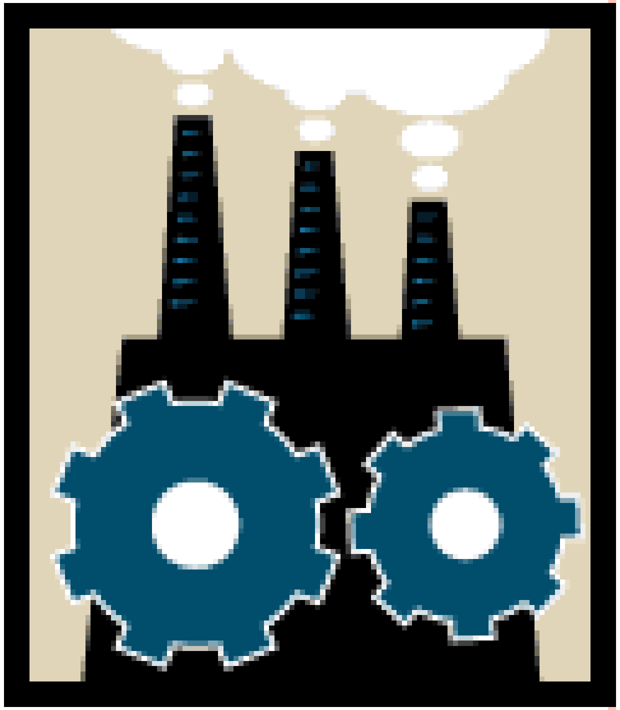
Lecture 7 Data Networking and Distributed Computation (Part 3)

Cloud *pipes*

Outline

- The business side: New Cloud service Use Cases and examples
- A bit more detail on the Internet network layer
  - Subnets (a different meaning) and masks
  - Autonomous Systems and Border Gateway Protocol (BGP)
  - NATs
  - IPv6
- QoS
  - Packet Scheduling Disciplines
  - Integrated Services
  - Differentiated Services
  - Multi-Protocol Label Switching (MPLS)
  - Generalized MPLS
  - Virtual Private Networks
- Software-Defined Networks (SDN)
- IP Security

Done



INTEGRATED SERVICES

1. Guaranteed service
2. Controlled load service

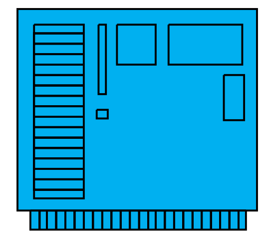
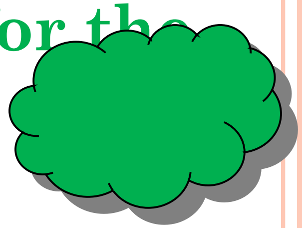
Guaranteed Service

Guaranteed service provides guaranteed bandwidth and bounds on the end-to-end queuing delay for *conforming flows.*

(In essence, guaranteed service is like, a bit "flexible," *virtual circuit*.)

The application invokes guaranteed service by specifying ***Traffic Descriptor (TSpec)*** and ***Service Specification (RSpec)***

TSpec and RSpec

- **TSpec** describes **the traffic source obligation** to the network; it contains five parameters:
  
  - *token rate* ***r*** (bytes/sec)
  - *peak rate* ***p*** (bytes/sec)
  - *token bucket depth* ***b*** (bytes)
  - *minimum policed unit* ***m*** (bytes) (if a packet is smaller, it will still count as *m* bytes)
  - *maximum packet size* ***M***
- **RSpec** describes the **service requirements for the network**; it contains two parameters:
  
  - *service rate* ***R*** *(bytes/sec)*
  - *slack term* ***S*** *(*microseconds*)* (the delay a node can add while still meeting the end-to-end delay bounds)

One more definition: Error Terms Due to Fluctuation from the *Fluid Model*

Let

C_i be the overhead delay—in the fluid model vs. *store-and-forward* one—a packet experiences in a router ***i*** due to the packet length and transmission rate;

D_i (measured in microseconds) be a rate-independent delay a packet experiences in a router ***i*** (due to flow identification, pipelining, etc.)

So, how is the Guaranteed Service guaranteed?

End-to-end worst case delay is (RFC 2212)

$$\frac{(b-M)(p-R)}{R(p-r)} + \frac{M+\sum_{i\in Path} C_i}{R} + \sum_{i\in Path} D_i$$

$$(p > R \geq r),$$

or

$$\frac{M+\sum_{i\in Path} C_i}{R} + \sum_{i\in Path} D_i$$

$$(R \geq p \geq r).$$

from TSpec:

r : *token rate*
p : *peak rate*
b : *token bucket depth*

M : *maximum packet size*

from Rspec:
R : *service rate*

Controlled Load Service

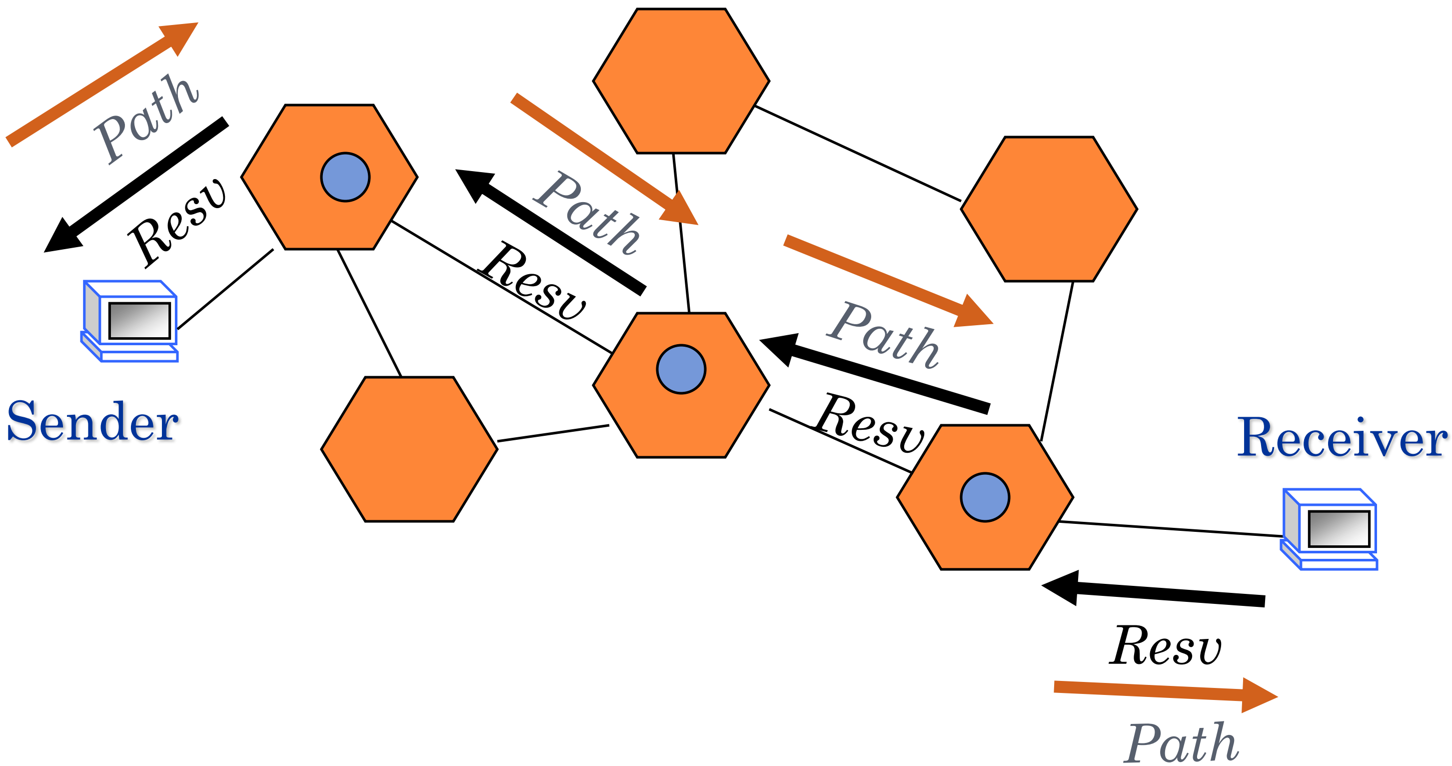
- The ***Controlled Load Service*** is best described in terms of what it does *not* allow to happen:
  - visible queuing delay
  - visible congestion loss
- The definition is left quite ambiguous—no quantitative guarantees—because admission control is left to implementation. (Sometimes, this service is called a *Better than the Best Effort* service.)
- The idea of the controlled load service is to avoid costly worst-case reservations and rely on statistical mechanisms.
- Consequently, only **TSpec** (but not **RSpec**) is specified for the controlled load service

Resource reSerVation setup Protocol (RSVP) Features

RSVP is

- Designed primarily for multicast (although it can be used for point-to-point reservations)
- Receiver-oriented (only receivers request reservations—Why?)
- *Simplex* (a reservation is made for a one-way path toward a receiver)
- Independent of a choice of routing (*unicast* or *multicast*) protocols
- *Policy-independent*

RSVP AT WORK (A SIMPLE EXAMPLE)



RSVP MESSAGES (SUMMARY)

MESSAGE	Direction
PATH	downstream (sender to receiver)
RESV	upstream (receiver to sender)
PATHerr	upstream (in response to *PATH*)
RESVErr	downstream (in response to *RESV*)
PATHTear	downstream
RESVTear	upstream
RESVConf	downstream (in response to a specific request within *RESV)*

The *PATH* messages install the path state; They may contain

- *Sender Template* (to identify the flow)
- *Session* (destination address, port, and protocol ID)
- *Time value* (required for refreshing)
- *Policy data (information for local decision on reservation)*
- *Sender Tspec* (r, p, b, m, M) for the sender's traffic
- *Adspec* (optional)
- *Integrity*
- $\sum_{i \in CurrentPath} D_i$, $\sum_{i \in CurrentPath} C_i$ and other path-related data

Note: *Adspec* is optional; it is used only in the *Open Path With Advertisement,* where the receiver is notified about the path capabilities. (A simpler method is called *One Path.*)

The *RESV* messages carry the reservation requests; They may contain

- *Style* (explained on the next slide)
- *Flow descriptor*
  - Flowspec (Service class, RSpec [for guaranteed service], and TSpec)
  - Filterspec (defines the flow for treatment specified in Flowspec)
- Time value (required for refreshing)
- RSVP Hop (NHOP)
- *Policy data (information for local decision on reservation)*
- *Integrity* (for authentication)
- *Scope* (a list of senders to forward the message to)
- *RESV confirm* (address of the receiver that has requested the confirmation)
- *Session* (destination address, port, and protocol ID)

Reservation Styles

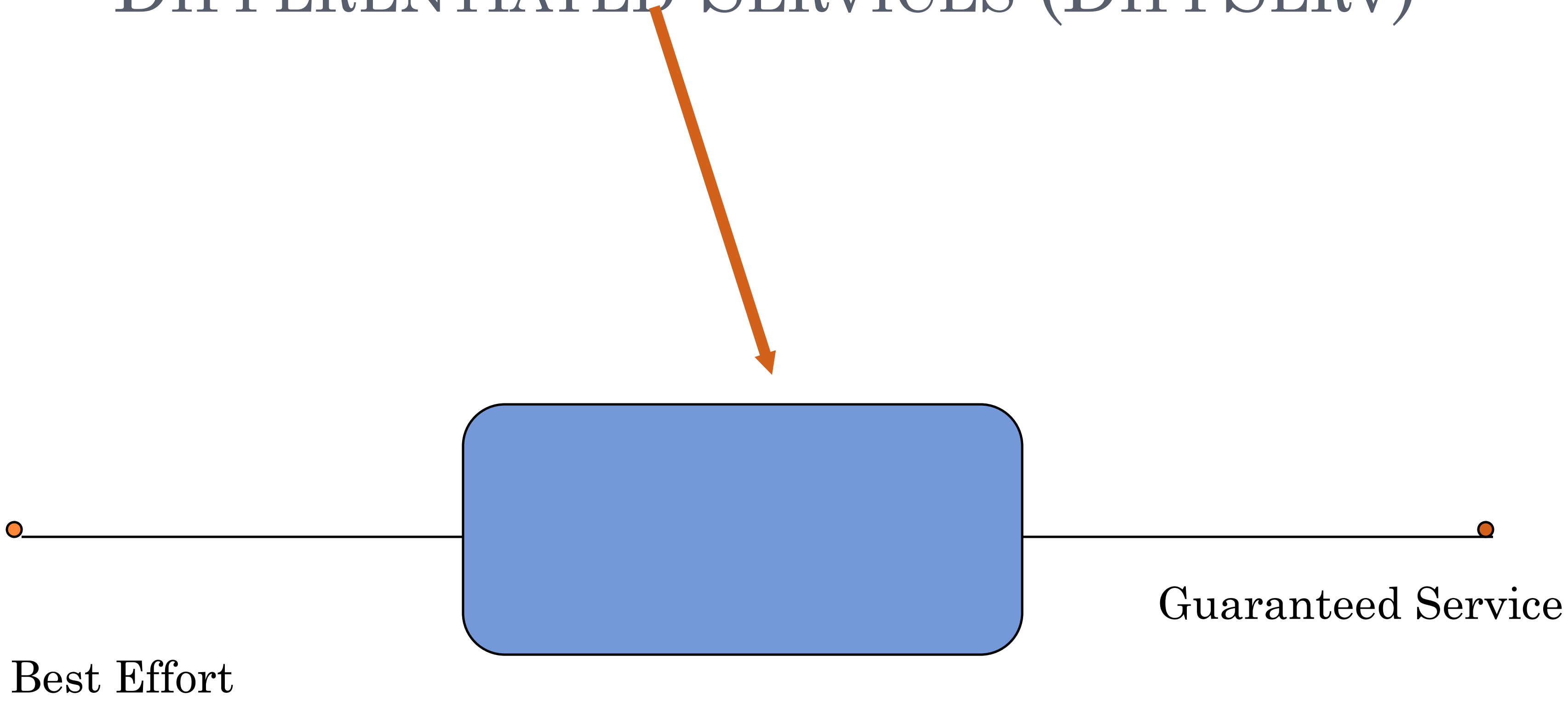
- ***Wild-card-filter (WF):*** all receivers share a single reservation (the largest of reservations requested from all receivers), which all senders can use. One flow spec *F* for all senders *(F, *)*.
- ***Fixed-filter (FF)*:** a distinct reservation is made for each specific sender *[(F1, S1), (F2, S2), ..., (Fn, Sn)]*.
- ***Shared explicit (SE):*** all receivers share a single reservation, but the senders are distinct *(F; S1, S2, ..., Sn)*.

Note: Shared reservations (WF and SE) are designed for multicast applications, where not all senders transmit simultaneously.

IntServ Summary

- The Integrated Services is a framework and a protocol (RSVP) for resource reservations (applicable to multicast)
- The architecture has been also adapted to specific link layers (IEEE 802.2, ATM)
- The reservations are receiver-based
- Packets are classified by the flows to which they belong and then scheduled for transmission according to reserved bandwidth
- RSVP embodies a hop-by-hop approach to QoS signaling (i.e., all RSVP-capable routers participate in resource reservations)
- States for RSVP reservations are *soft*
- "Pure" RSVP scalability is still questioned; although not deployed, it has a new life as RSVP-TE (addressed further)

Differentiated Services (DiffServ)



Service vs. Forwarding Treatment

- *Service* (as far as QoS is concerned) is characterized by the end-to-end behavior
- *Forwarding treatment* is characterized by the behavior (dropping, marking, and scheduling of packets) of one particular router

Through **network provisioning**, well-defined forwarding treatments can be combined to deliver new services

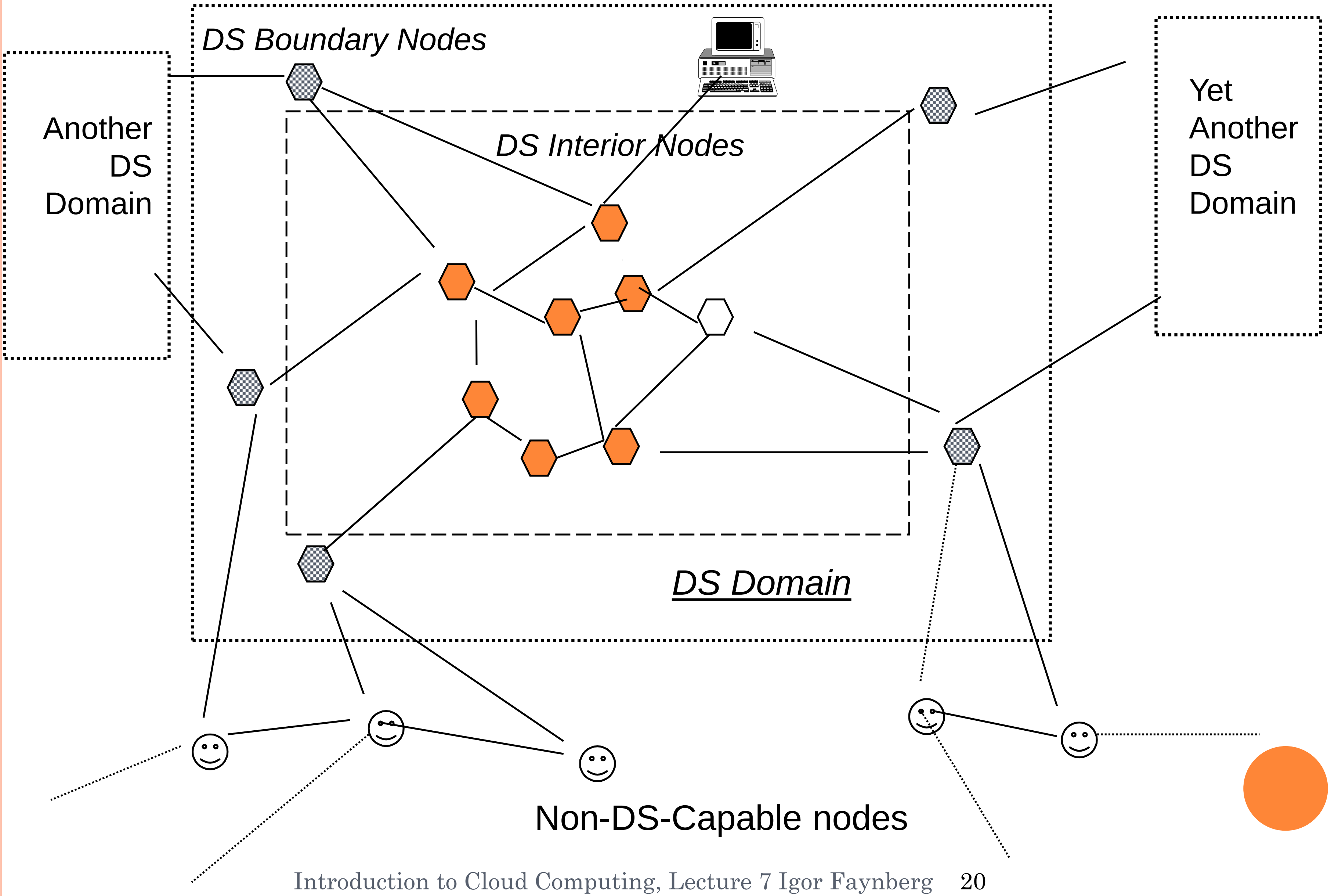
The DiffServ Concept

- Was developed to provide methods for *various* levels of QoS
- Supports a pre-defined set of service-independent building blocks rather than particular services
- Defines *forwarding* (vs. *end-to-end*) behaviors
- Breaks the traffic into *classes* (as opposite to *flows)* and allocates resources to each of these classes
- Is based on service level agreements between customers and service providers
- Employs traffic policing at the edge of the network, then class-based forwarding in the core
- Does not employ reservations or signaling (thus eliminating authentication and simplifying billing)
- Provides natural path to incremental (i.e., domain-by-domain) implementation (vs. all-or-nothing, end-to-end *IntServ* approach)

The *DiffServ* model

- QoS is defined by a per-hop-behavior (PHB)
- Each PHB is assigned a 6-bit *Differentiated Services Codepoint (DSCP)*
- PHBs *are* the building blocks: all packets with the same codepoint make a *behavior aggregate*, and receive the same forwarding treatment
- PHBs may be combined into *PHB Groups*
- Thus,
  - In the interior nodes of a domain, the origination and destination addresses, protocol IDs, and ports are irrelevant—only DSCPs are
  - When more than one PHB group is defined in a domain it is necessary to define the interaction among them

Classification of the DS Domain Nodes



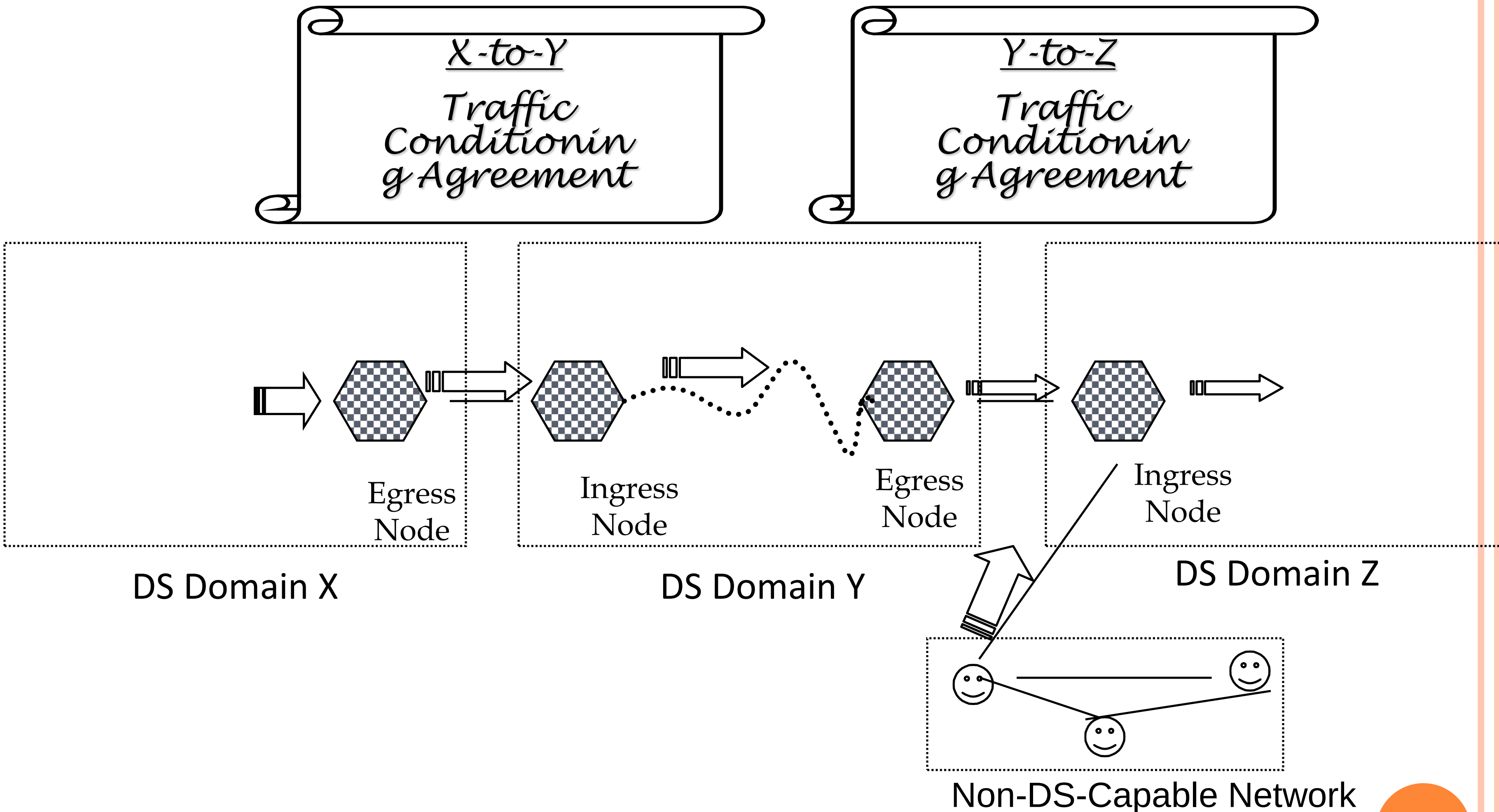
Note: *policing* has nothing to do with *policies*, but it has everything to do with *police.*

SERVICES

- Services are defined in *Service Level Agreements (SLAs)* between a customer and a service provider **as well as** between two adjacent domains. An SLA specifies the traffic as well as security, accounting, billing, etc.
- A central part of an SLA is a *Traffic Conditioning Agreement (TCA)*, which defines traffic profiles and **policing**

  actions, such as
  - token bucket parameters for each class
  - throughput, delay, and **drop** priorities
  - actions for non-conformant packets

Traffic Conditioning at the Edges of DS Domains



SLAs

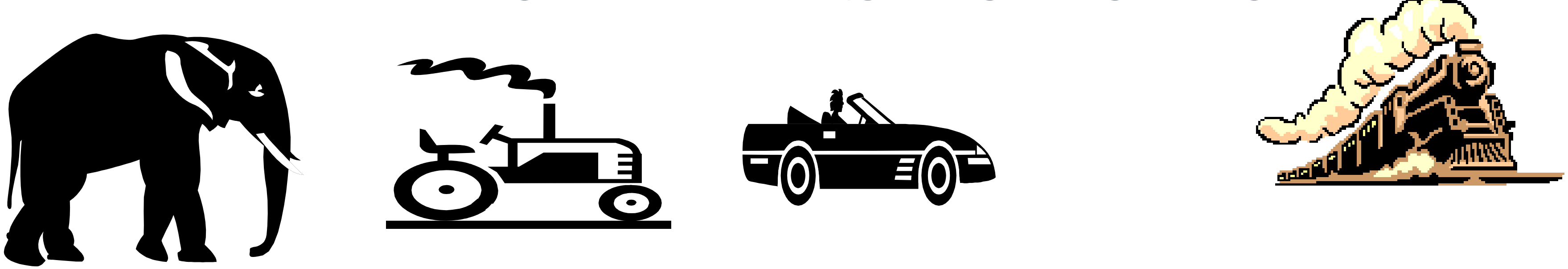
- can be *static*— provisioned once before the services are started
- can also be *dynamic*—they can be changed via real-time negotiations from a customer's node
- must be translatable to- and from a Cloud Service SLA

Support of the dynamic SLAs can be effected only with the help of the network management systems

Standardized Per-Hop Behaviors (PHBs)

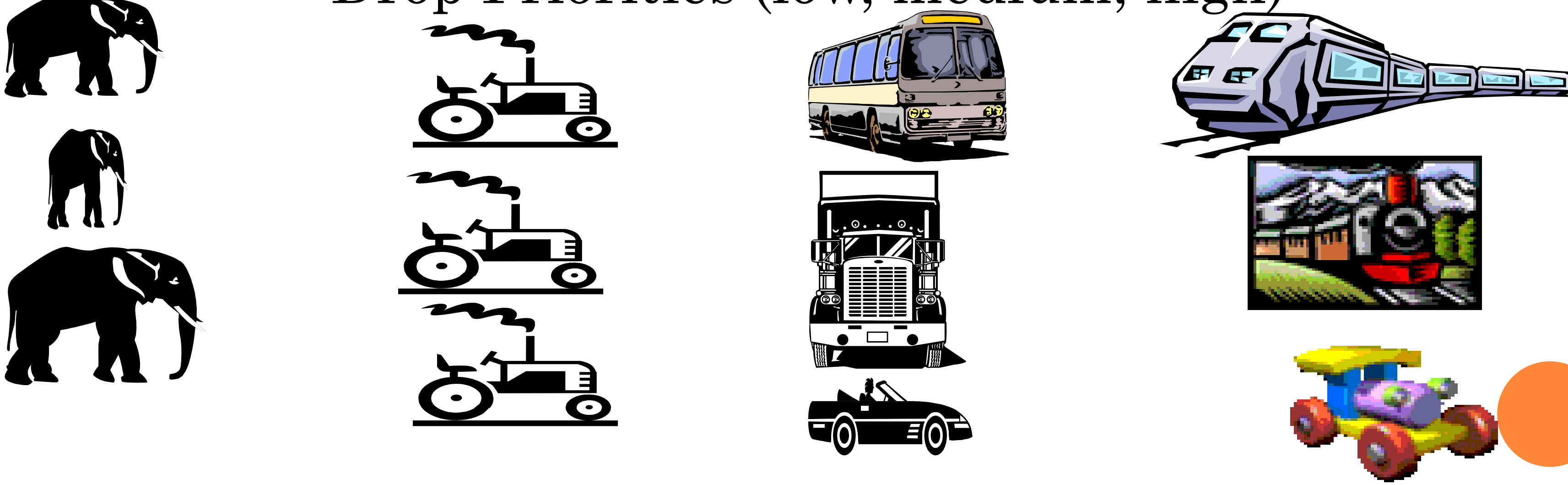
- The *Default PHB group* corresponds to…the good old Best-Effort treatment.
- The *Class Selector (CS) PHB group* (enumerated CS-1 through CS-8, in the order of ascending priority) is compatible with current implementations based on priority queuing.
- The *Expedited Forwarding (EF) PHB group* guarantees the departure rate of the aggregate's packet (to be no less than the arrival rate). It is assumed that EF traffic may preempt any other traffic.
- *The Assured Forwarding (AF) PHB group* allocates (in ascending order) priorities to four classes of services and also defines three dropping priorities for each class of service (as the treatment for out-of-profile packets).

An Example of an AF Specification



Class 1 Class 2 Class 3 Class 4

Drop Priorities (low, medium, high)



IPv4 Header

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="4">Version</td><td colspan="4">IHL (Header Length)</td><td colspan="8">TOS (Differentiated services)</td><td colspan="16">Total length (in bytes)</td></tr>
<tr><td colspan="16">Identification (common to all fragments)</td><td>Unused</td><td>DF</td><td>MF</td><td colspan="13">Fragment offset</td></tr>
<tr><td colspan="8">Time to live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td colspan="32">Source Address</td></tr>
<tr><td colspan="32">Destination Address</td></tr>
<tr><td colspan="32">Options (0 to maximum length)
…</td></tr>
</table>

IPv6 (MAIN) HEADER (REMINDER)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				Differentiated services								*Flow Label*																			
Payload length																*Next Header*								*Hop limit*							
Source Address																															
(128 bits)																															
Destination Address																															
(128 bits)																															

How it is done: The DS Field

- In the IPv4 packets: *8*-bit *Type of Service (TOS)* field.
- In theIPv6 packets: *Traffic Class* (TC) octet.)
- DiffServ uniformly *redefines* the respective octets, using the first six bits for the *Differentiated Services Code Point (DSCP)*, and reserving the remaining two bits (marked *Currently Unused [CU]*) for future use.

Codepoint Allocation within the DSCP field (general)

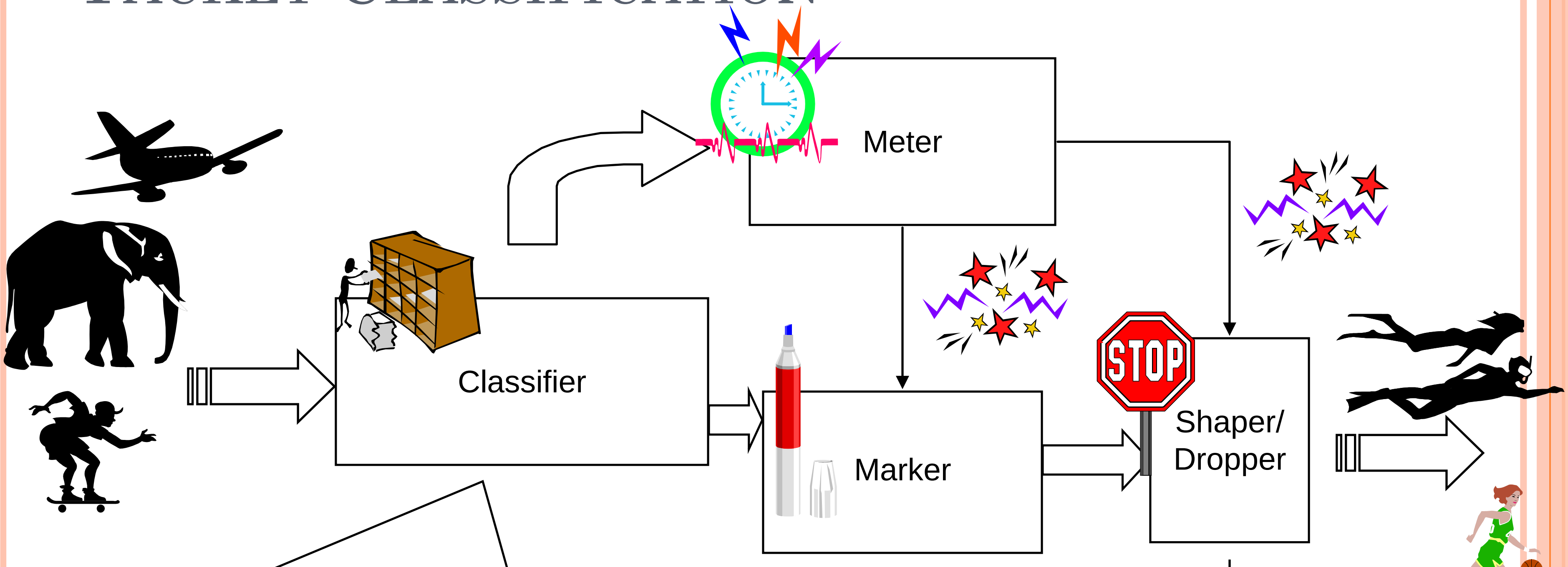
- Bit 5 is set to 0 (XXXXX0) to indicate that the codepoint is standardized, thus leaving exactly 32 standard codepoints
- Bits 4 and 5 are set to 01 (XXXX01) to indicate experimental use (with a caveat that these may be reclaimed is the standards run out of codepoints)
- Bits 4 and 5 are set to 11 (XXXX11) to indicate experimental and local use and may not be reclaimed by standards

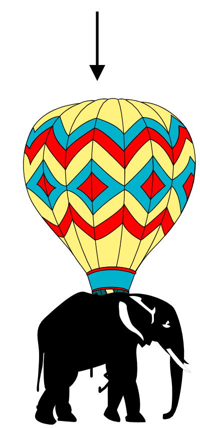
Codepoint allocation to specific PHBs

- A special codepoint (000000) is assigned, for backward compatibility, to the Default Per-Hop Behavior (PHB)
- A set of codepoints to maintain compatibility with current practices (XXX000) is assigned to Class Selector (CS) PHBs (enumerated CS-1 through CS-8, according to the codepoint numerical value
- The codepoint (101110) is assigned to Expedited Forwarding (EF) PHB
- The Assured Forwarding (AF) PHB group is assigned the following set of codepoints:

Drop Precedence	**Class 1**	**Class2**	**Class3**	**Class 4**
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

Packet Classification

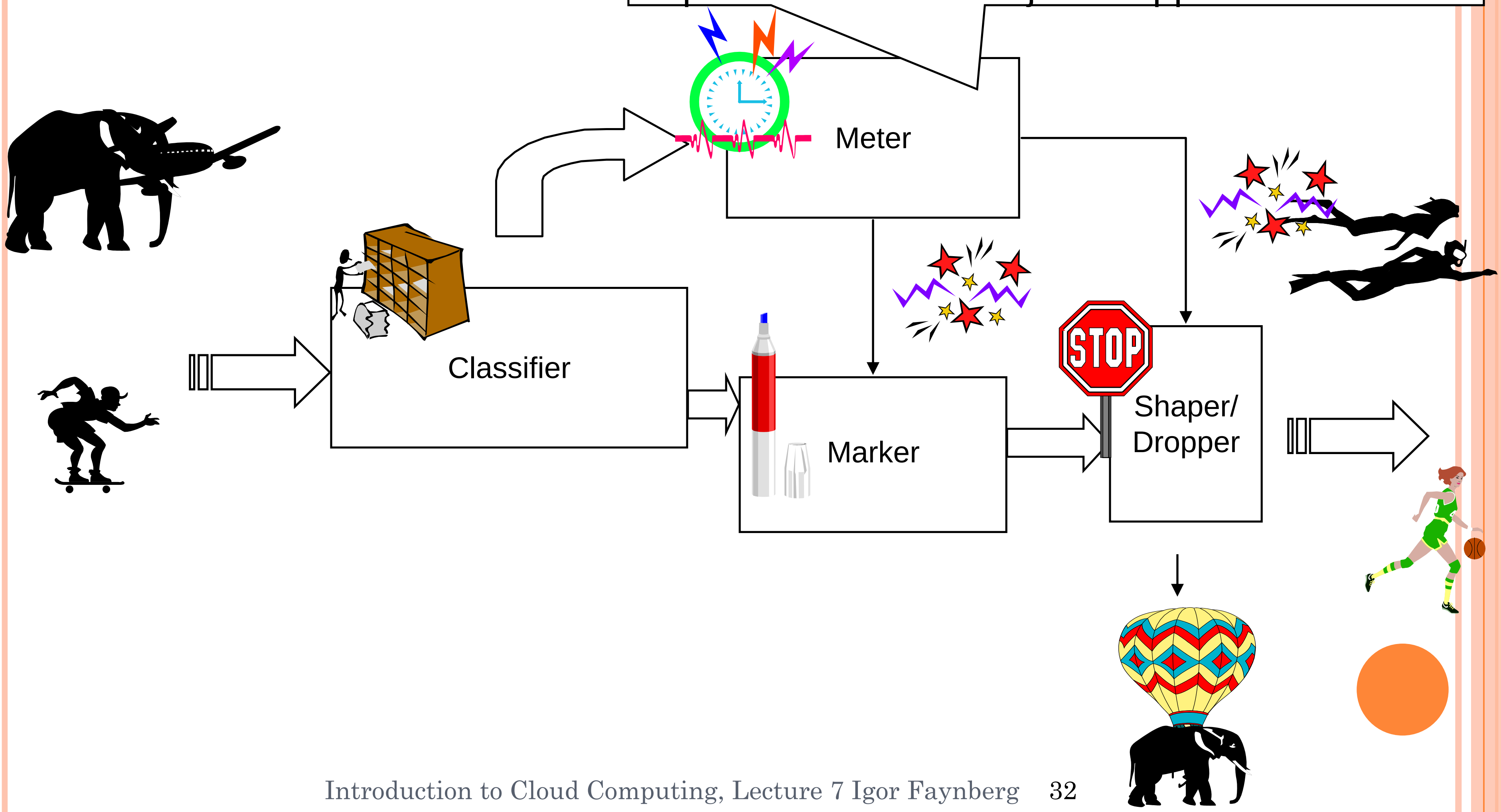




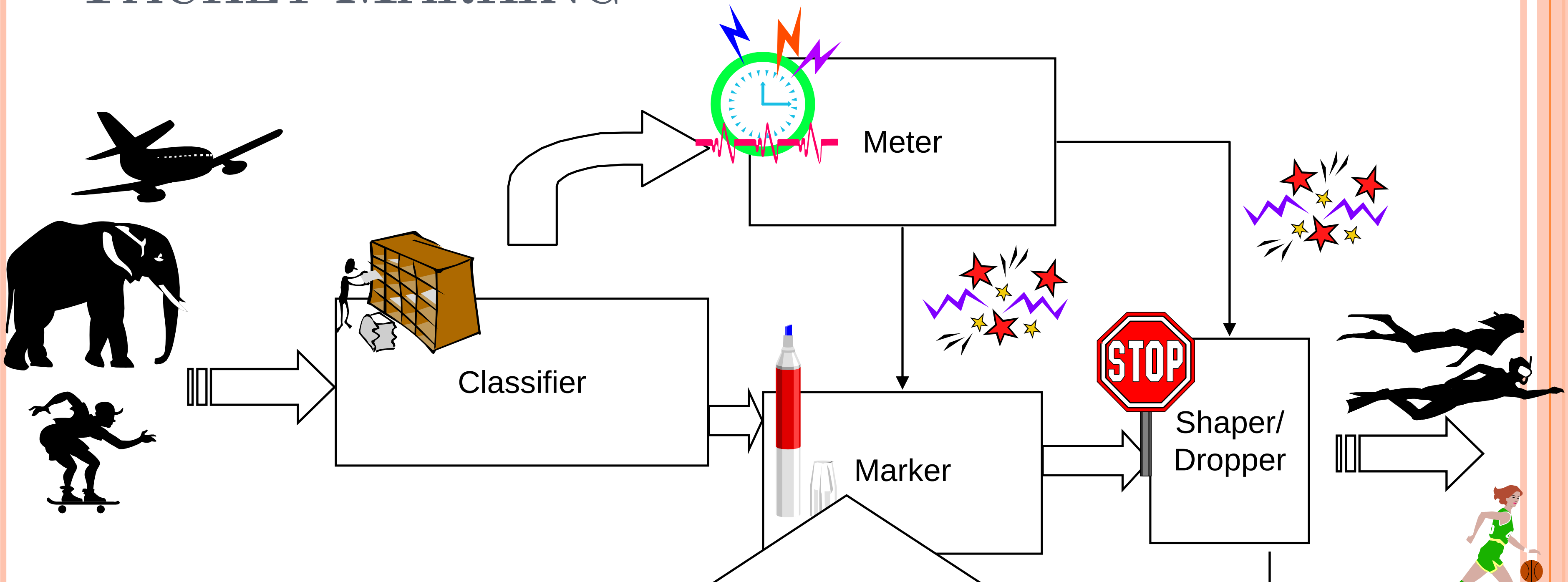
- At the edge of the network, the Classifier typically looks at *multifield* (such as flow tuple) of a packet and determines the aggregate to which the packet belongs and the respective SLA
- Everywhere else in the network, the Classifier examines just the DSCP values

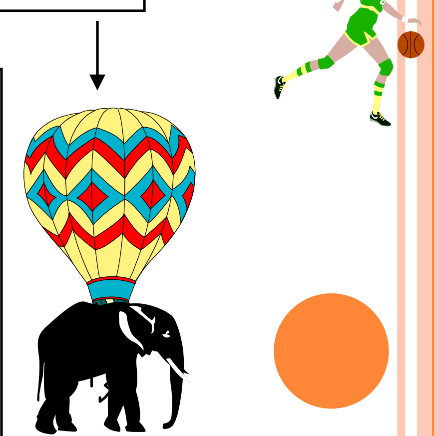
Traffic Metering

- The Meter checks the aggregate (to which the incoming packet belongs) against the Traffic Agreement Specification and determines whether it is *in* or *out* of profile
- Depending on particular circumstances, the packet is *marked* or just dropped



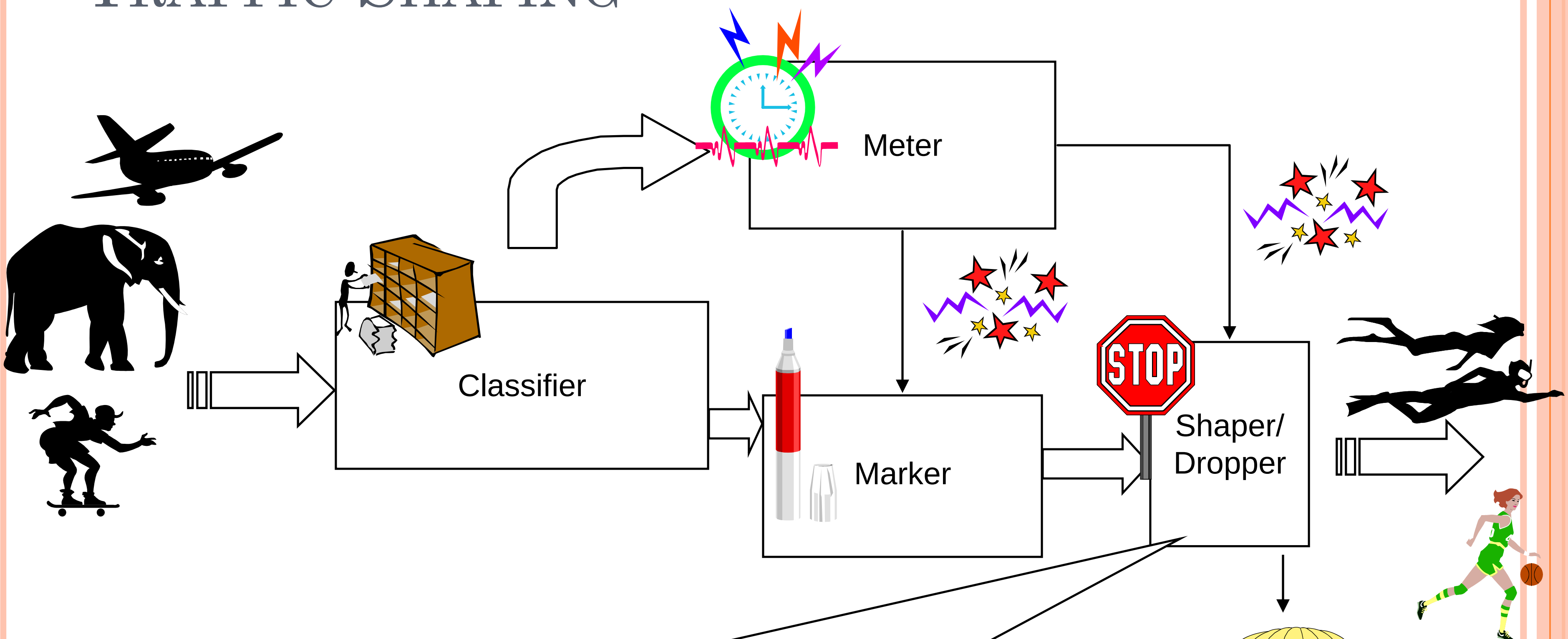
Packet Marking

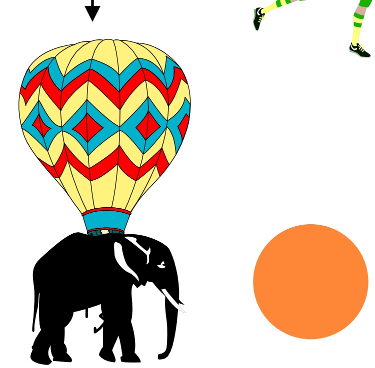




- The marker writes the respective DSCP in the DS field (marking may be done by the host, but it is checked and may be changed when necessary by the boundary nodes)
- Sometimes, special DSCPs may be used to mark non-conformant packets, which may be dropped later due to congestion. Packets may be also promoted or demoted.

TRAFFIC SHAPING





- The Shaper delays non-conformant packets until it brings the respective aggregate in compliance with the traffic
- Shaping often needs to be performed between the egress and ingress nodes

DiffServ Summary and Comparison to *IntServ*

- *DiffServ* allocates resources based on a few classes of services rather than individual flows (as in *IntServ)*)
- The *DiffServ* standard maps *Per-Hop Behaviors (PHBs)* into *DS codepoints* of the IP packets. Traffic is classified and conditioned only at the edges of *DS Domains* rather than at all transit node (as in *IntServ*)
- Resource allocation is achieved through network provisioning rather than dynamic reservation mechanism (as in *IntServ);* after network is provisioned, performance is assured through traffic prioritization and conditioning as well as a static (thus ineffective) form of admission control
- A combined use of *DiffServ* (in core networks) and *IntServ* (in edge networks) exploits the strengths of the two approaches (cf. RFC 2998)

New Topic: Multi-protocol Label Switching

- … Virtual circuits in (mostly IP) networks

Multiprotocol Label Switching (MPLS)

The Principle:

Establishing the Label-Switched Path (LSP)—an equivalent of a virtual circuit and then *switching* (rather than routing) IP packets without ever looking at the packet itself. Furthermore, an LSP can be established along any path (including those that do not traverse the shortest distance), a feature that helps traffic engineering.

The Consequences:

- ***Theoretical*:** another score in the never-ending battle of *connection-oriented* vs. *connectionless*
- ***Political:*** problematic in the *old* Internet community
- ***Practical*:** enabled peer-to-peer communication with ATM and Frame Relay switches, support of Internet Traffic Engineering (off-load, rerouting), acceleration of forwarding (although that may become less significant with advances in silicon), protocol-independent forwarding, consistent tunneling (VPN) discipline, and support of traffic engineering.

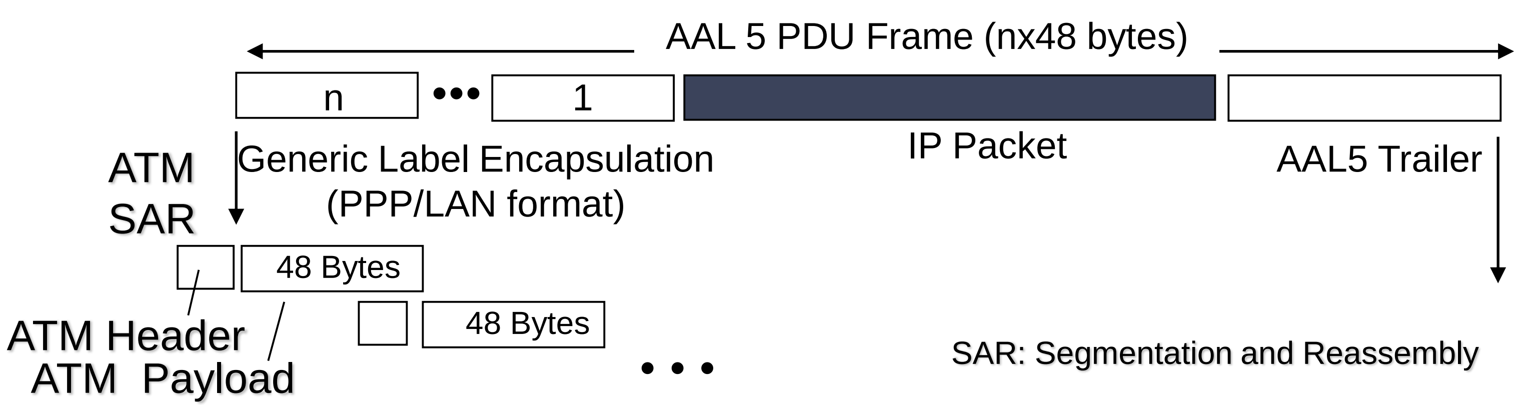
MPLS Components

- Definition of the Forwarding Equivalence Classes (FECs)
- Definition of the label format
- Definition of the hierarchical label stacking mechanism
- Specification of the protocols for label distribution
  - Label Distribution Protocol (LDP)
  - RSVP Traffic Engineering extensions (RSVP-TE)

Where is the Label?

The Layer 2 (i.e., PPP, 802.2, or HDLC) Frame

The ATM Adaptation Layer 5 (AAL5) PDU



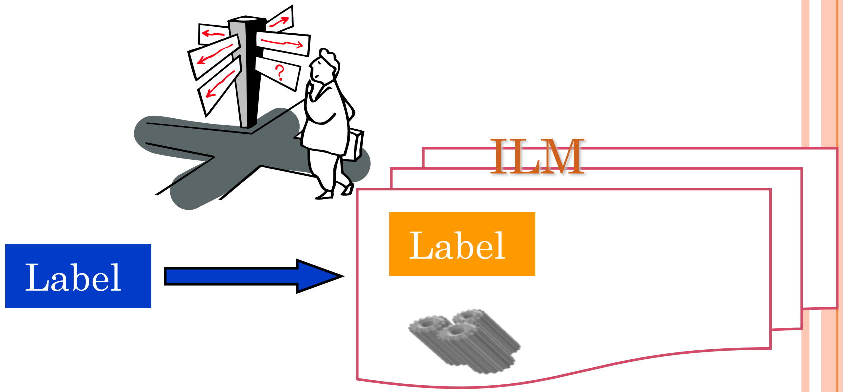
WHAT *IS* A LABEL?

0 … 20	20 … 23	23 24	24 … 31
Label Value	EXP	S	Time to Live (TTL)

- ***Label Value:*** used for forwarding by the Label-Switched Router (LSR) (**Note:** There could be a stack of labels). The label value determines the whole Label-Switched Path (LSP)—the virtual circuit. Values (*0* through *15* are reserved; the values *0* and *2* have a particular significance of terminating MPLS—popping the stack and continue forwarding using the packet's IPv4 (0) or IPv6 (2) address.
- ***EXP:*** reserved for experimental use
- ***S:*** set to *1* if this is the last label in the label stack; set to *0* otherwise.
- ***TTL:*** replaces the IP TTL (which is invisible to MPLS) to detect loops

How is the Label Used?

- The label value serves as an index into the Incoming Label Map (ILM) of an LSR, which, among other things contains
  - Outgoing label value
  - Next hop
  - State information
- The outgoing label value is typically substituted for the new one (hence label switching) and the packet is forwarded to the next hop address

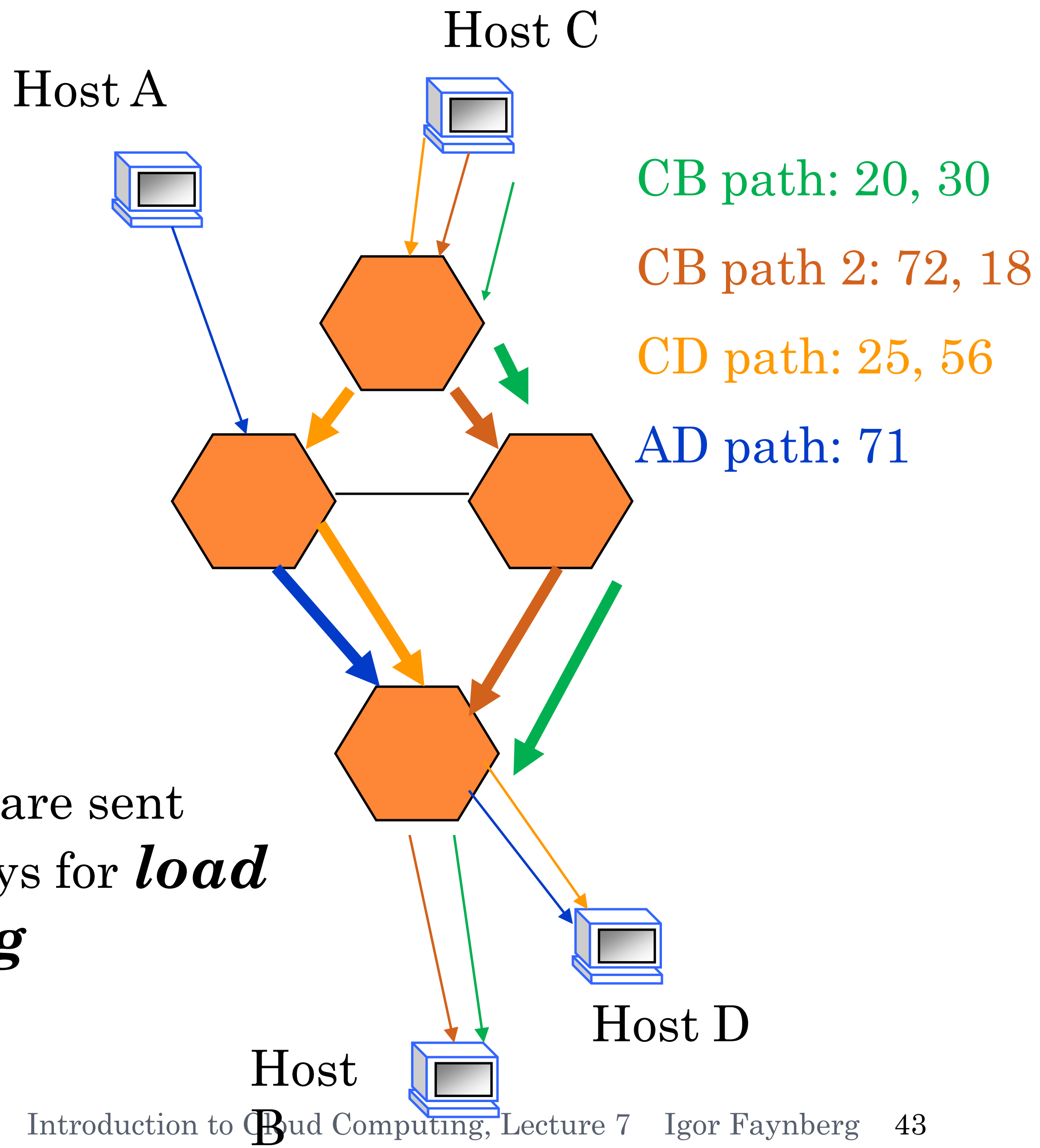


Forwarding Equivalence Class (FEC)

- Each label in the Label-Switched Path (LSP) has a one-to-one association with a *Forwarding Equivalence Class (FEC)*, which is in turn associated with the treatment the packets receive
- An FEC is defined by a set of rules. MPLS presently supports the following FECs:
  - Packets that match a particular IP destination address
  - Packets destined to the same egress router
  - Packets that belong to an application flow (as defined by *IntServ*)
- Naturally, different FECs have different levels of scalability

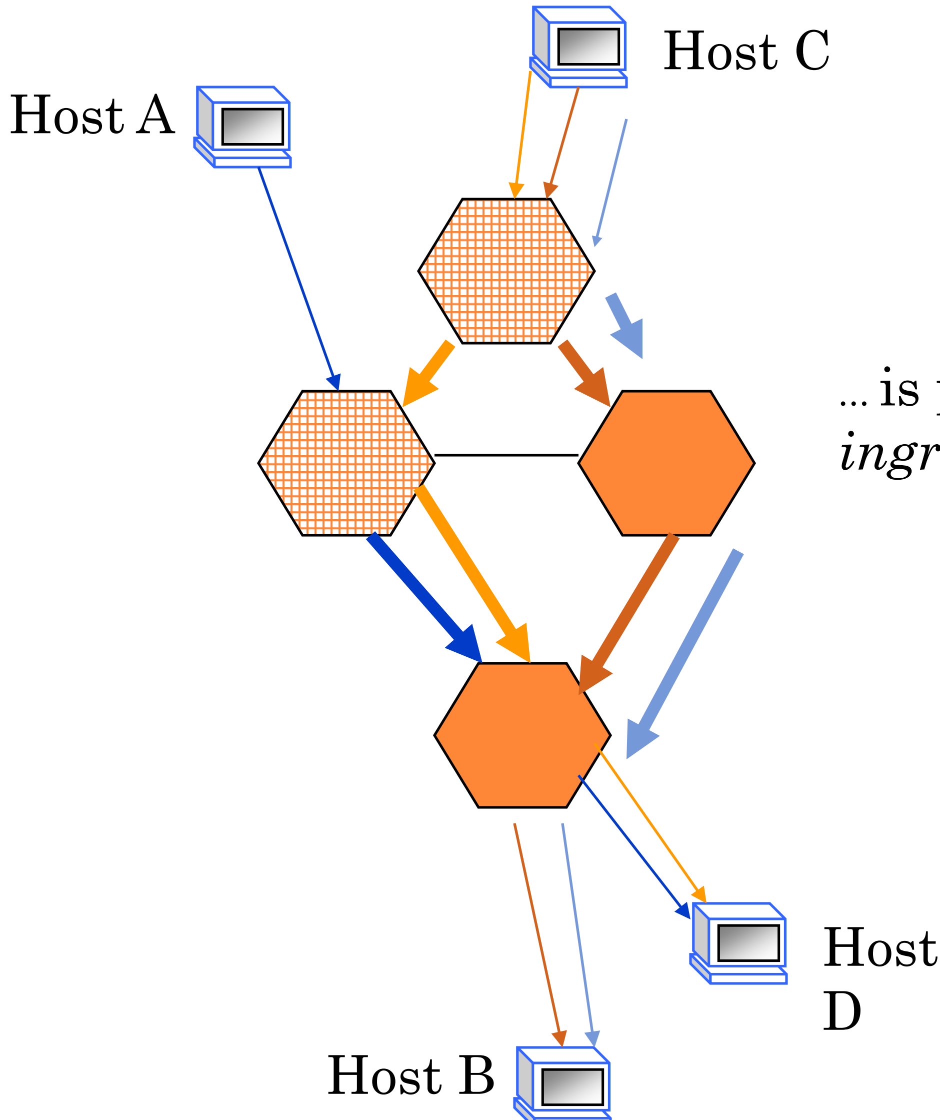
Note: The traffic associated with an LSP is simplex.

An Example of Label Assignment to Flows and LSPs



Note: Flows are sent different ways for ***load balancing***

Packet Classification and Mapping to FECs



… is performed only by the *ingress* routers

LABEL ASSIGNMENT IN LABEL-SWITCHING ROUTERS (LSRS)

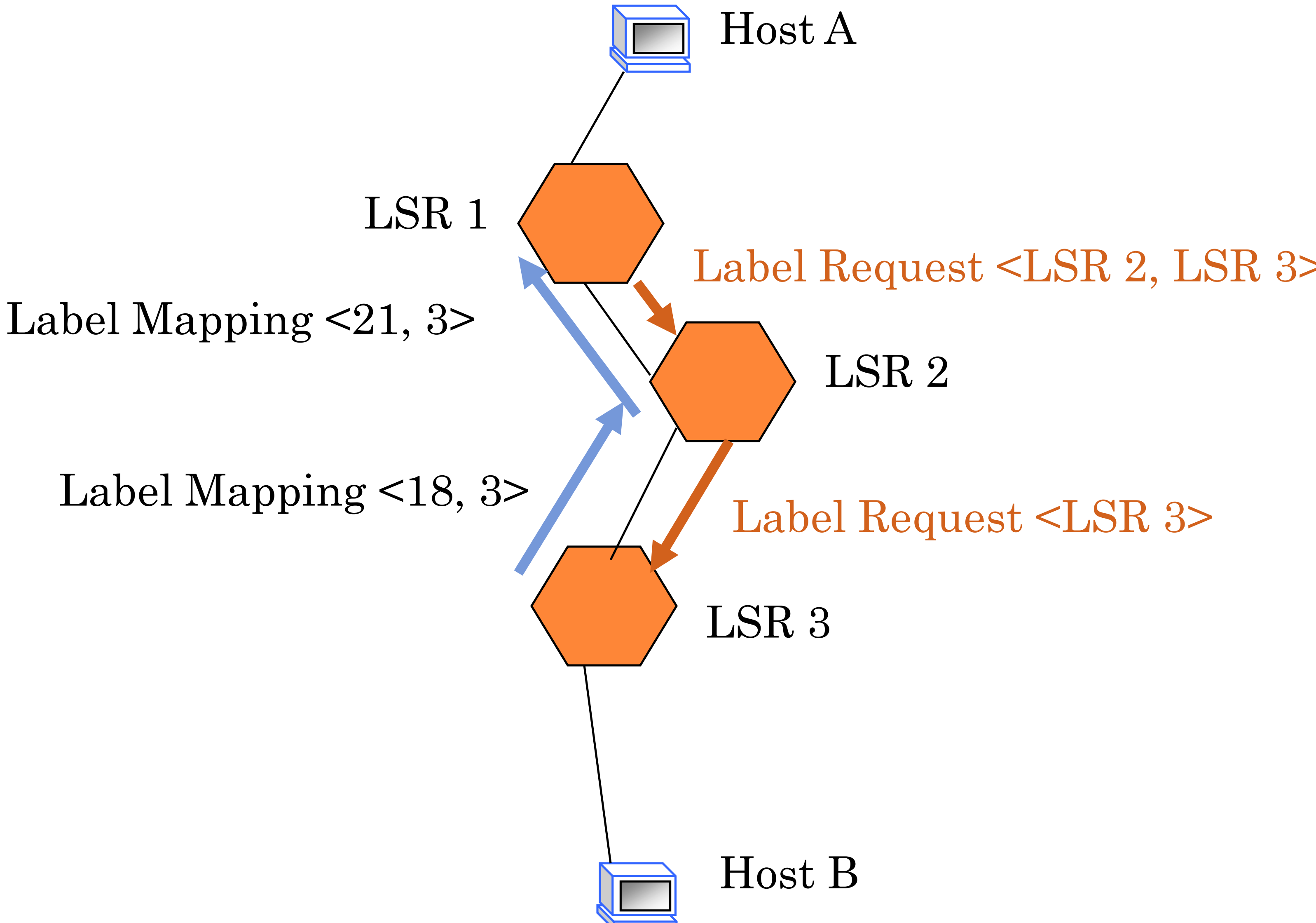
Labels are assigned *upstream:*

An LSR knows better how to index its own Incoming Label Map (ILM)

Label Distribution Protocols (there are three):

- LDP
- Constraint-based Routing LDP (CR-LDP)
  An LDP extension based on traffic engineering
- RSVP-TE
  An RSVP extension based on traffic engineering

An Example of Explicit Route Setup with CR-LDP



RSVP-TE

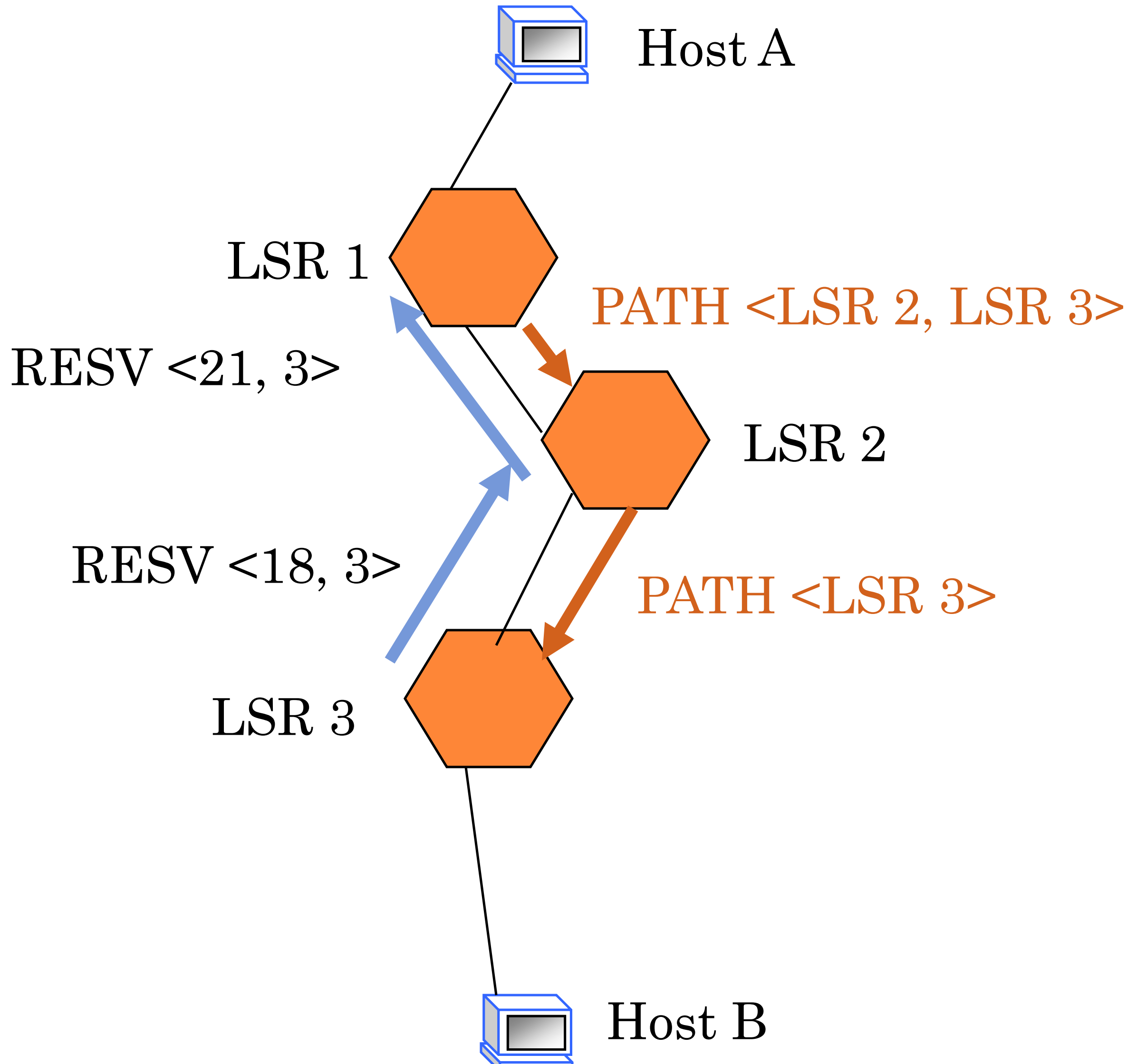
RSVP-TE extends the RSVP to support resource allocation to LSPs and also support

- Label distribution (with explicit routing)
- LSP tunnel rerouting (with its "make before break" technique)
- Preemption options

It introduces a new message (*HELLO)* for rapid node failure detection and several new "objects":

- LABEL_REQUEST, EXPLICIT_ROUTE, RECORD_ROUTE
  - to be used in *PATH*
- LABEL, RECORD_ROUTE
  - to be used in RESV

TE


Host A
LSR 1
PATH <LSR 2, LSR 3>
RESV <21, 3>
LSR 2
RESV <18, 3>
PATH <LSR 3>
LSR 3
Host B

How CR-LDP and RSVP-TE Compare?

	CR-LDP	RSVP-TE
Protocol Transport	TCP, UDP	Raw IP
Security	IPSec	RSVP Authentication, IPSec
LSP State	Hard	Soft
LSP Refresh	None	Hop-by-hop
LSP Preemption	Supported	Supported

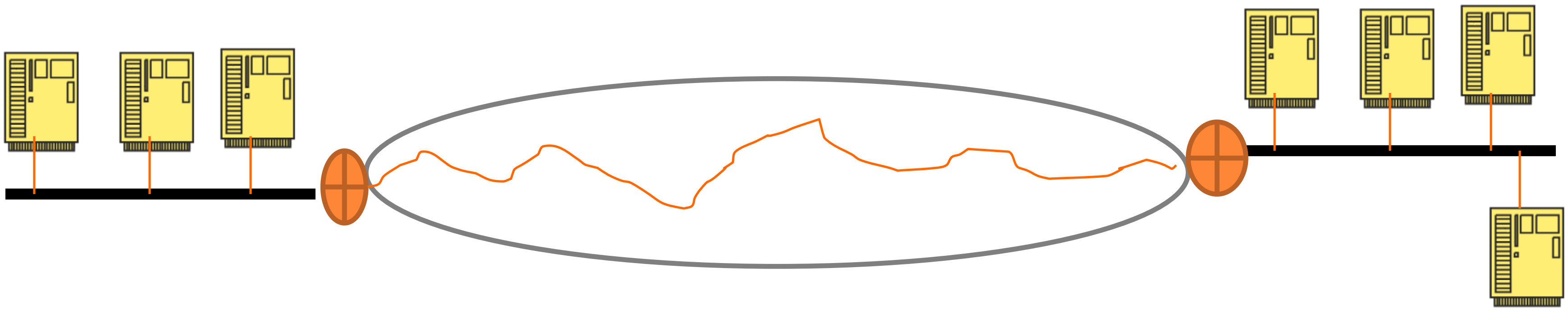
MPLS Summary

- MPLS is using label switching, which allows to establish virtual circuits (called label switching paths [LSPs]), which switches IP packets based solely on short (compared to the IP header size) labels outside the IP packet. One benefit of this approach is that it naturally eases internetworking with ATM
- The LSPs can be established over other than shortest paths in IP networks. Traffic Engineering can use this feature for load balancing. Using label stacking, tunnels for IP VPNs can be built
- Each LSP is mapped onto a Forward Equivalence Class (FEC). FECs can specify forwarding granularity
- Three label distribution protocols have been in existence: LDP, CR-LDP, and RSVP-TE
- MPLS signaling is used for optical control plane

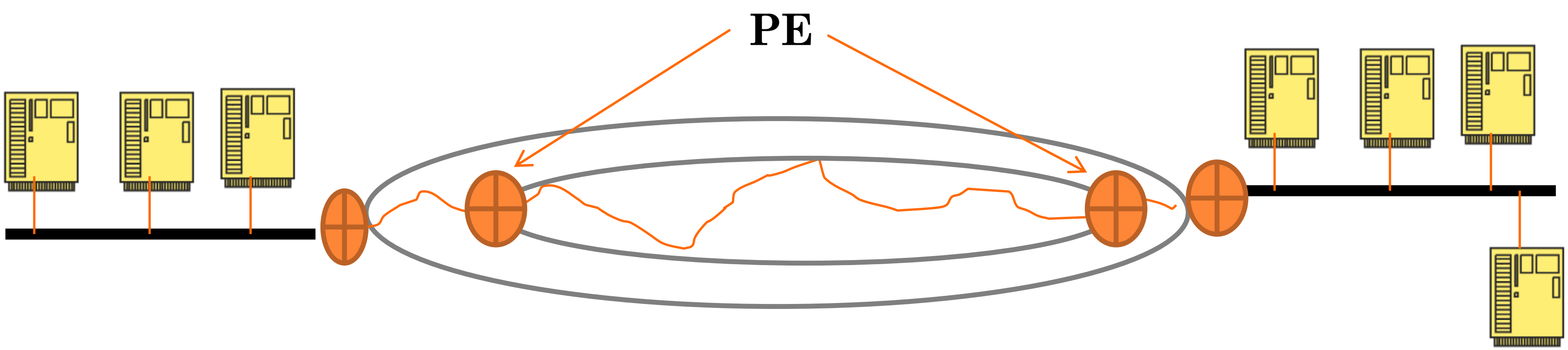
Generalized MPLS

- Extends MPLS to support control of (including signaling and routing aspects) the following types of switching layers:
  - Packet switching
  - Layer 2 switching
  - *Time division multiplexing*
  - *Lamda switching*
  - *Fiber Switching*
- Is based on MPLS-TE (i.e., RSVP-TE and CR-LDP) for signaling extensions, including
  - Generalized labels
  - Bi-directional LSPs with downstream or upstream label assignment
  - Explicit label control
  - Event notification (to any node, adjacent or non-adjacent)
  - Control and data channel separation
  - Handling of control channel failures

Label Stacking in provider-supported VPN



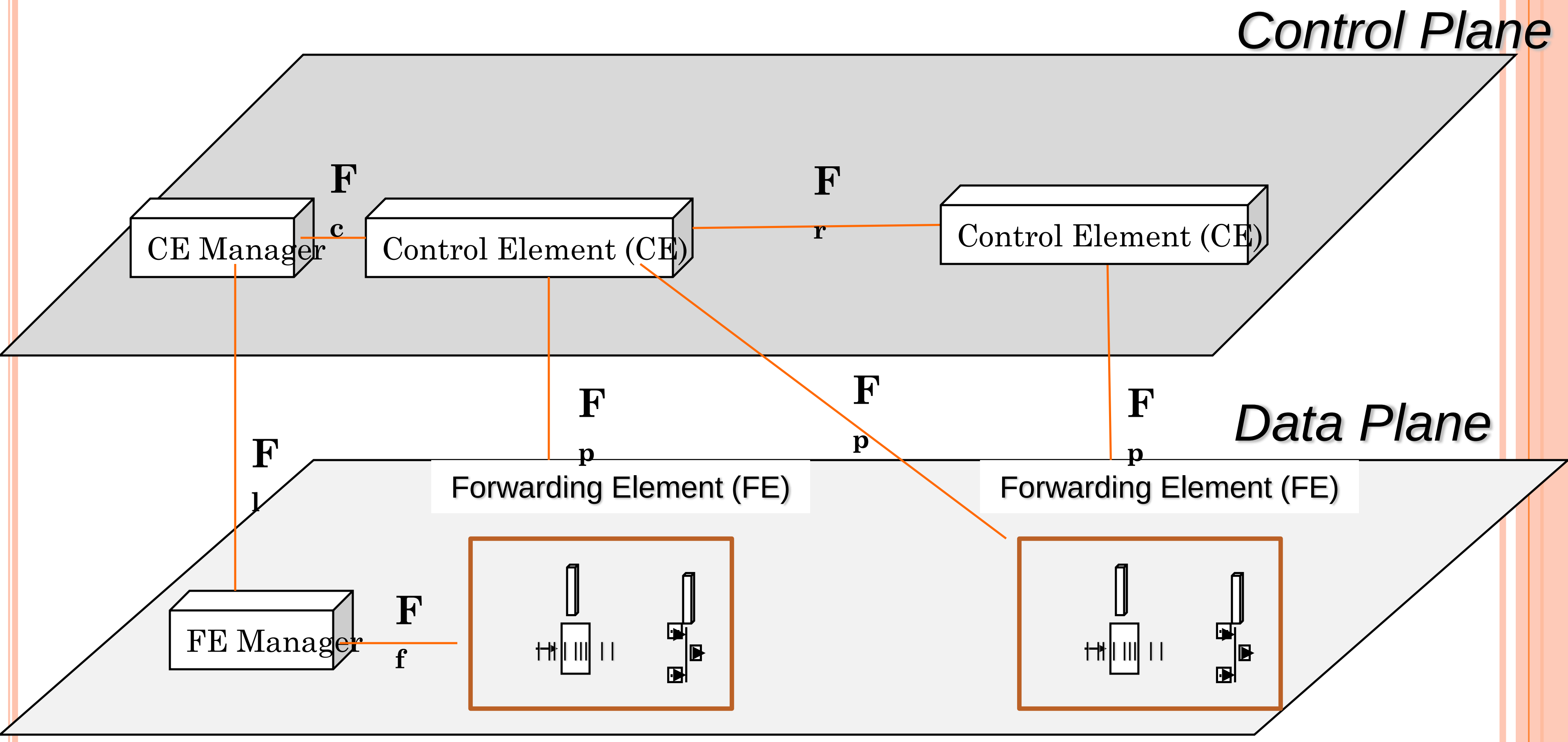
a) LSP in a single network



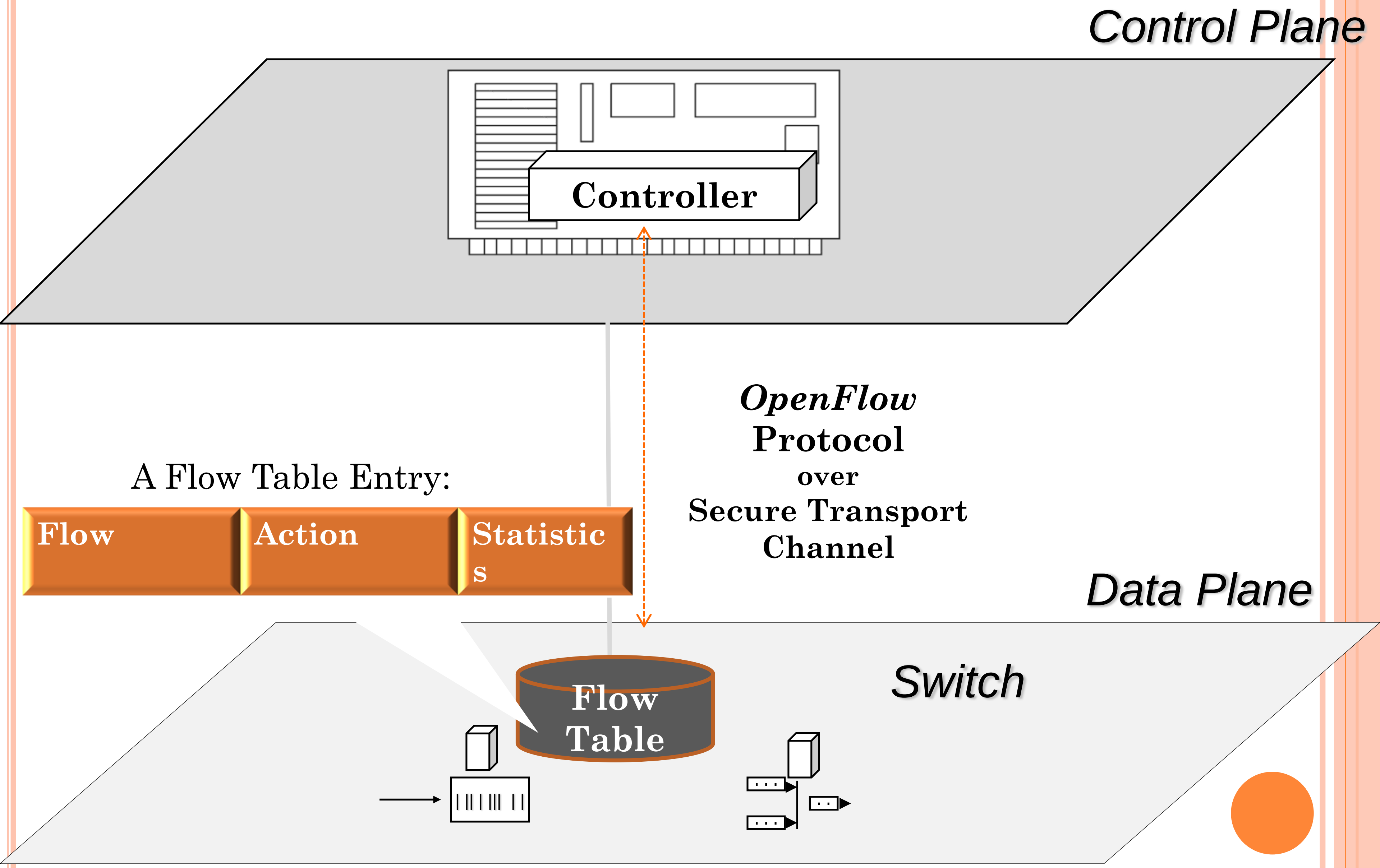
a) LSP traversing a provider network

Software-Defined Network (SDN)

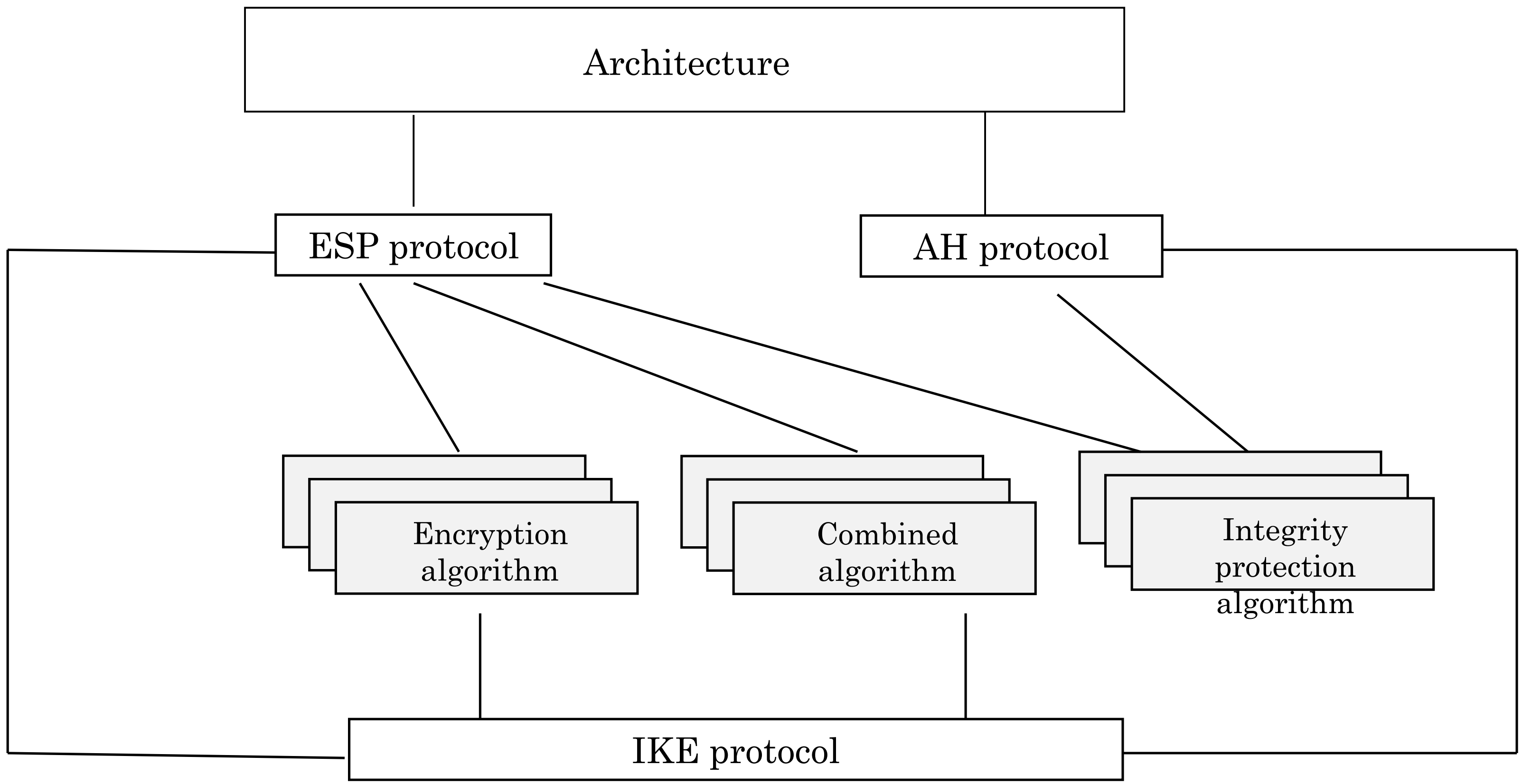
ForCES Architecture (after RFC 3746)



The *OpenFlow* Switch

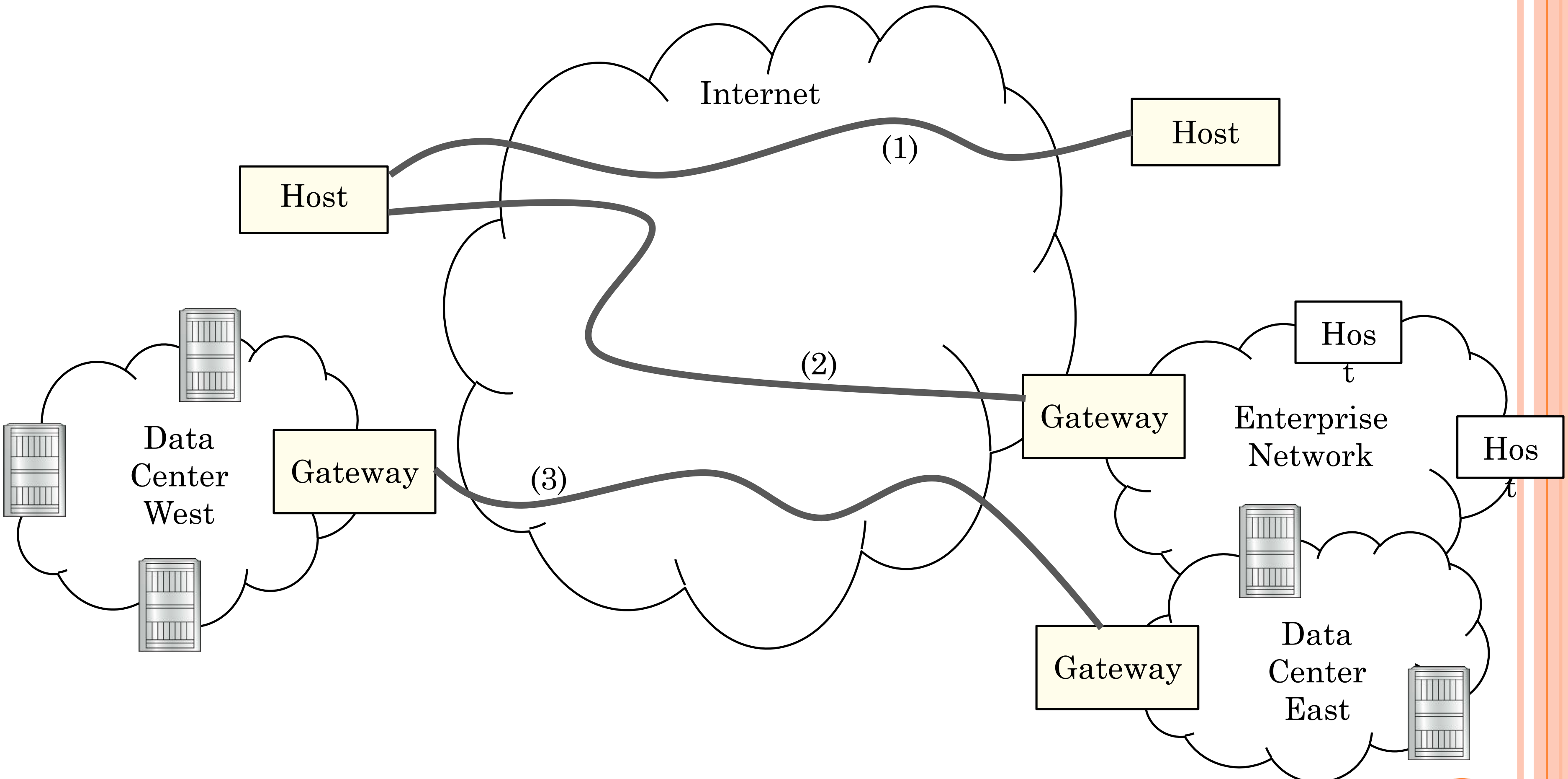


Relationship among the IPSec specifications (after RFC 6071)

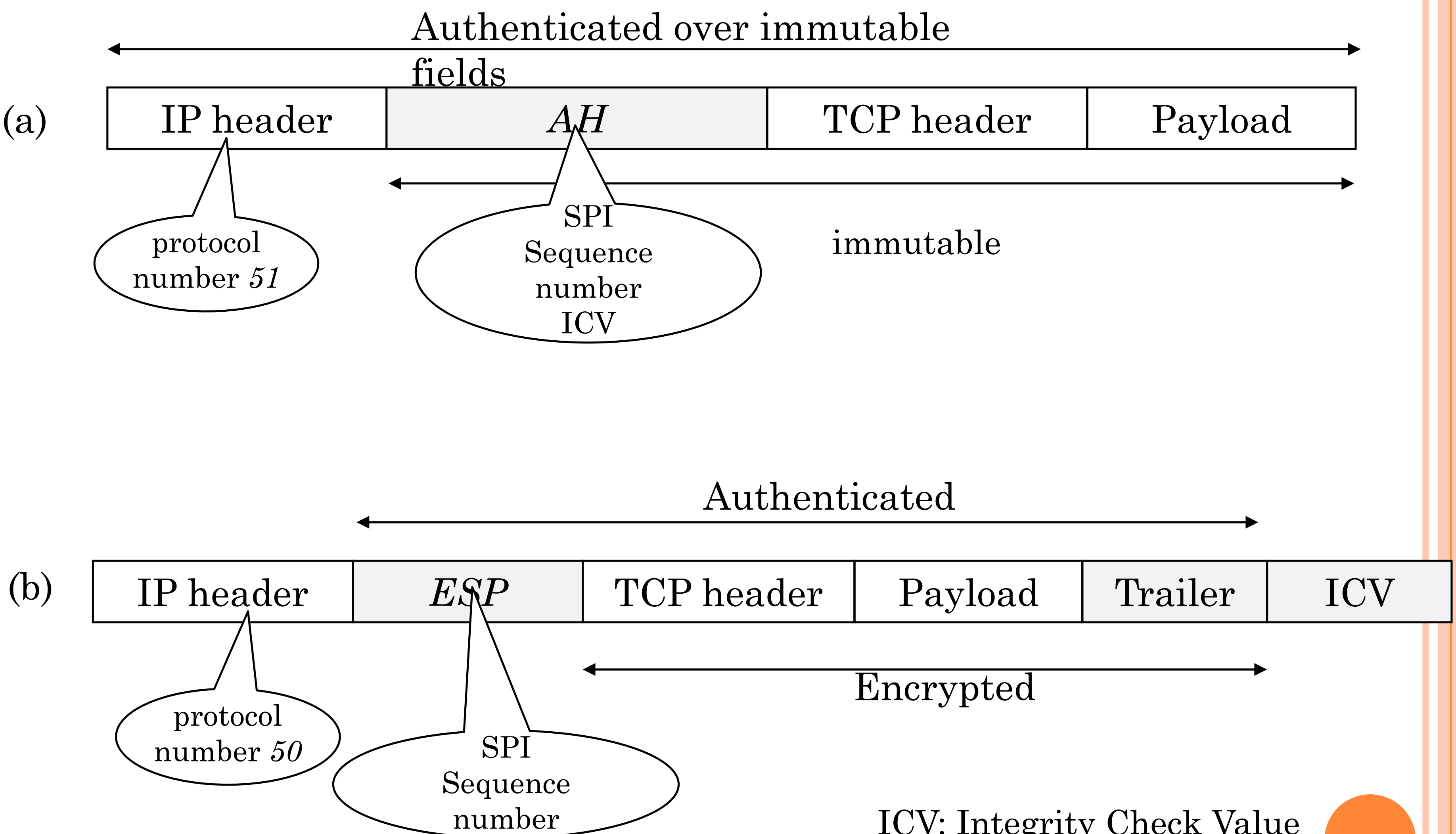


AH: Authentication Header
ESP: Encapsulating Security Payload
IKE: Internet Key Exchange

IPSec scenarios

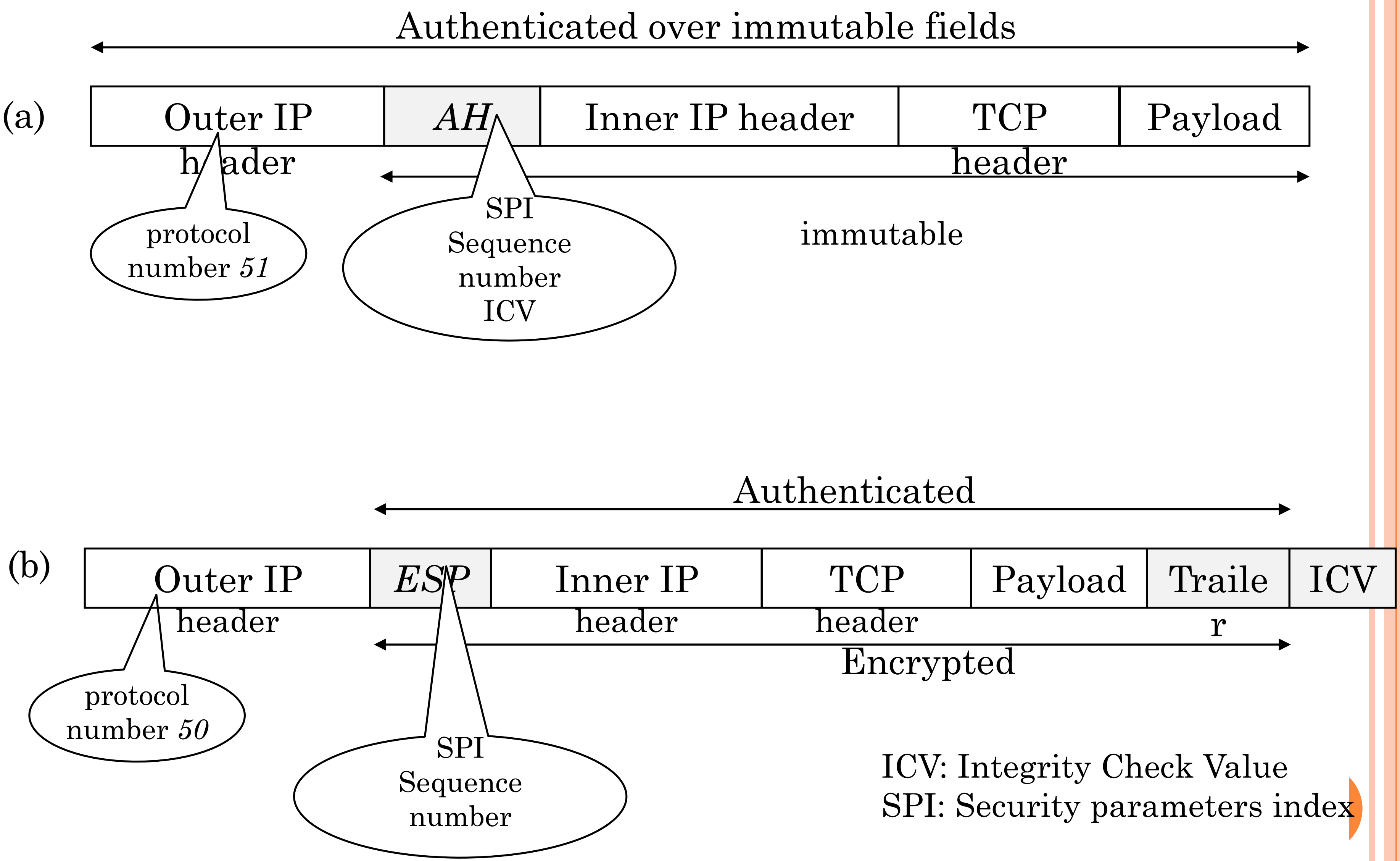


IPsec in transport mode in IPv4



ICV: Integrity Check Value
SPI: Security parameters index

IPsec in tunnel mode in IPv4



ICV: Integrity Check Value
SPI: Security parameters index