Advanced Networking Concepts
Flow and Congestion Control

Contents - Adv. Networking - Flow/Congestion Control

- General Understanding of Flow and Congestion Control
- Flow / Congestion Control in the QoS Realm
- Implicit / Explicit Signalling Concepts
- Forward / Backward Congestion Notification

- Example TCP Flow and Congestion Control
- Example RED
- Example ECN
- Example PCN

IP-based Mechanisms

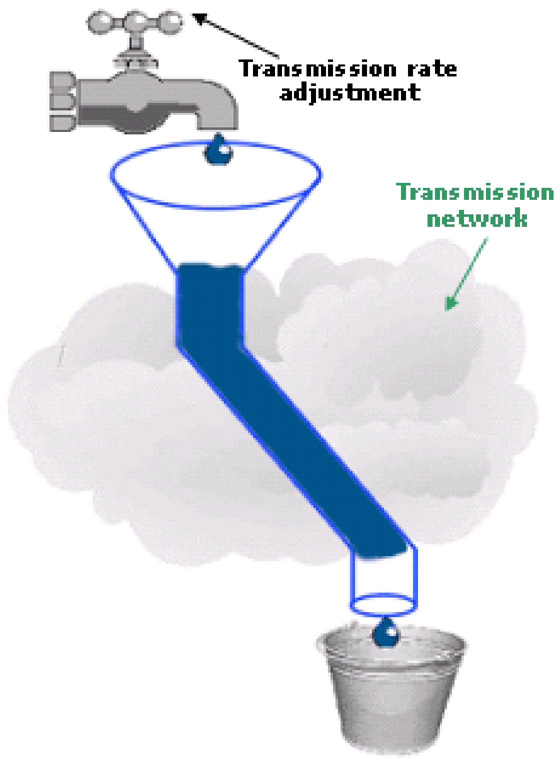
- Example FECN / BECN in Frame Relay / X25
- Example ATM (CAC with Traffic Contract + CLP + GFC)
- Example Ethernet Flow Control

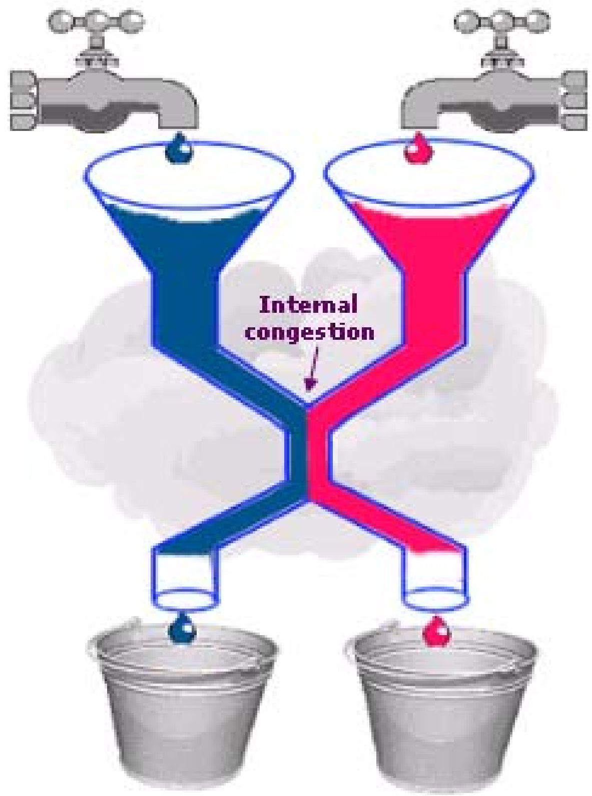
L2 Mechanisms

Flow/Congestion Control - Description

- Packet based data transmission between end systems involves sending nodes, relaying nodes and receivers.
- The aim of flow and congestion control is to **control the load situations** in relaying and receiving nodes in order to **avoid overload** situations, which **normally results in packet loss.**
  (Depending on the service qualities guaranteed by a packet network, large forwarding delay variations or other traffic contract violations might already be considered as overload/congestion).
- Relaying nodes and receivers therefore **directly or indirectly inform the sender(s)** about the (upcoming) overload and **request a reduction in the sending rate**.
- **Flow control** addresses **receiver overload** and normally requests rate **reductions at a single sender**.
- **Congestion control** addresses overload in **relaying nodes** (the network cloud) and normally requests rate **reductions at all contributing senders**.

Flow/Congestion Control - Description





- **Flow control** addresses **receiver overload** and normally requests rate **reductions at a single sender**.
- **Congestion control** addresses overload in **relaying nodes** (the network cloud) and normally requests rate **reductions at all contributing senders**.

Flow/Congestion Control - Classification

Overload Prevention (in Packet Networks)

Flow Control

End system - centric:

- reactive (closed-loop):

  Flow control
  e.g. TCP rwnd

- proactive (open-loop):

  traffic contract between end systems
  + traffic observation for conformity (statically/dynamically configured, rate-based)
  e.g. ATM connection setup

Congestion Control

Network - centric:

- reactive (closed-loop):

  Congestion control

  – **implicit** (traffic behaviour monitoring in end systems) e.g. TCP cwnd
  – **explicit** (forward or backward signalling of congestion build-up (FECN, BECN)) e.g. IP ECN

- proactive (open-loop):

  traffic contract between end systems or between network and end systems
  + traffic and connection admission control + resource reservation (statically/dynamically configured, rate-based) e.g. ATM connection setup

Flow/Congestion Control - The QoS Realm

QoS Realm

Traffic planning "**traffic control**"

- Conn.-oriented (co) & conn.-less (cl)
- Link/tunnel based
- Long term planning "Network provisioning" (months/days)
- Congestion avoidance through capacity planning

e.g. "offline" capacity planning and network (re-)design

Resource reservation w. Admission control

- co only
- Flow based
- Dynamic control (seconds)
- Congestion avoidance through managed reservations and refusal of excess connection requests or excess traffic in admitted flows

e.g. IP + RSVP,
ATM traffic contracts,
IP in circuit-switched transport networks

Prioritization (w. Admission limitation)

- co & cl
- Packet based
- Highly dynamic control (µs)
- Congestion avoidance not possible, but selective/prioritized packet dropping in relaying nodes and rate limitations

e.g. IP + DSCP,
Ethernet priorities,
MPLS priorities
ATM CLP
Frame Relay DE

(Flow) and **Congestion control**

- co & cl (limited)
- Packet based
- Highly dynamic control (µs)
- Congestion mitigation by overload guessing/signalling with resulting sender slow-down

e.g. IP + ECN,
IP + PCN (incl. CAC),
TCP flow/con. control,
Frame Relay F/BECN,
ATM F/BECI
Ethernet PAUSE

Flow/Congestion Control - Signalling

Congestion Signalling

Implicit Detection

Indirect congestion signalling through packet loss or delay variation

- Buffering → increased delay (RTT)
- Buffer overflow → loss
- Early discard → early "loss notification" for responsive ("well behaved") transport protocols

• Transport layer protocols should monitor traffic characteristics (loss, delay, delay variation) and deduce congestion forecasts
assumption: buffer usage is the only cause for delay and loss observations

Explicit Signalling

Direct congestion signalling by relaying nodes:

- Congestion notification or
- Congestion warning (early notification)

• Use usage statistics in lower layers for early indication to transport layer protocols (e.g. buffer usage level indication)

• Signalling types:
- binary indication,
- window credit and
- rates

Flow/Congestion Control - Forward / Backward Notification

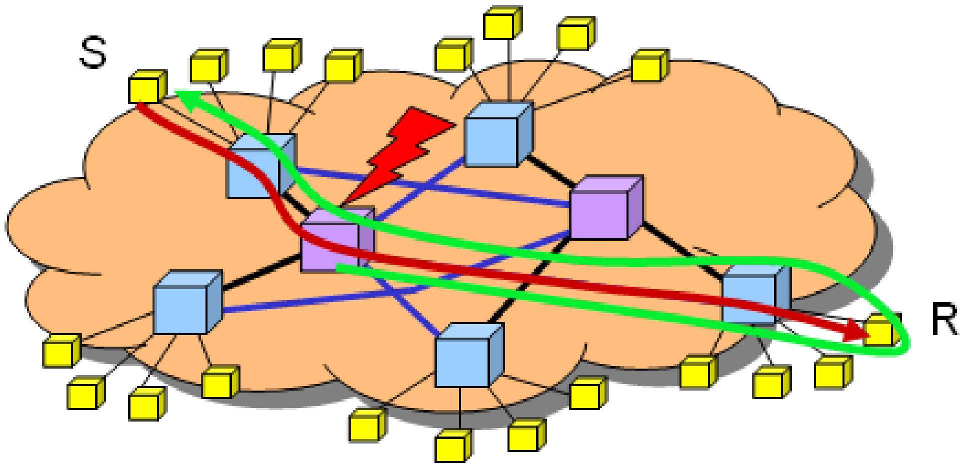
Congestion Signalling Direction

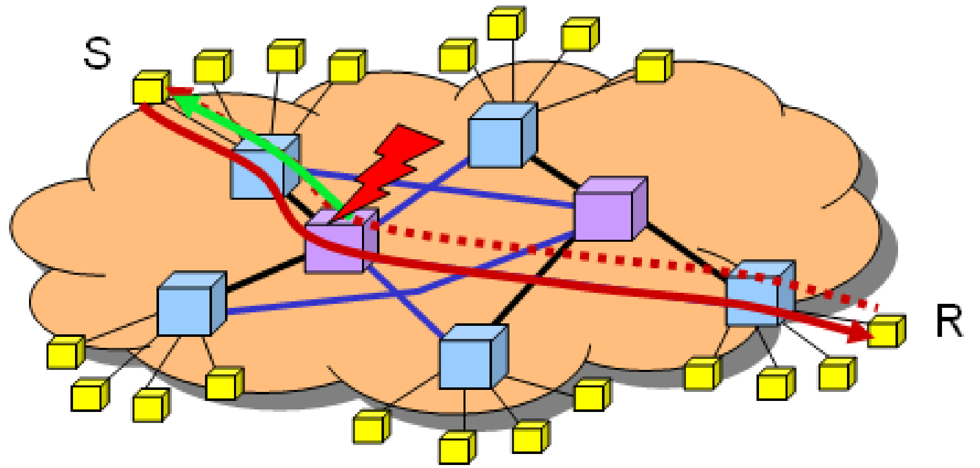
Forward ECN

- Relaying nodes inform receiver, which in turn should inform the sender
  (mark forwarded packets)
- ≈ Round trip propagation time

Backward ECN

- Relaying nodes inform sender directly (mark packets in opposite direction)
- Small (≤ single trip) propagation time





Congestion signalling
Data traffic

Flow/Congestion Control - Scope

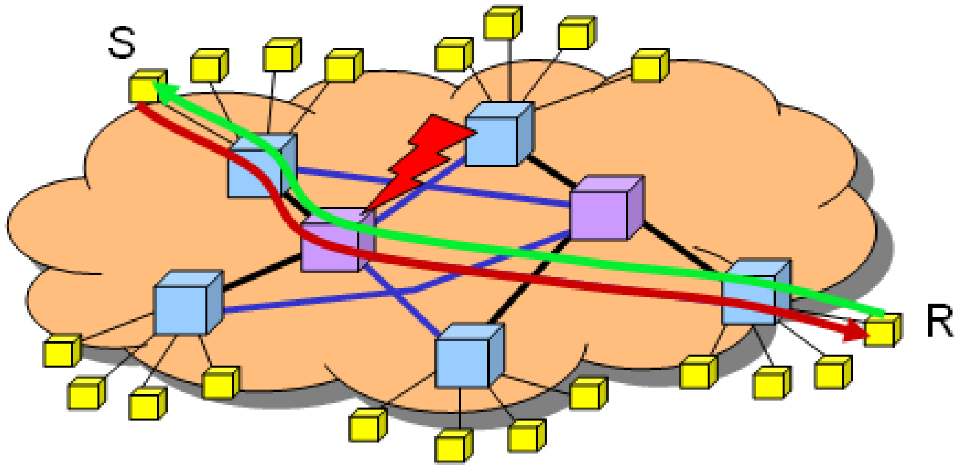
Congestion Signalling Scope

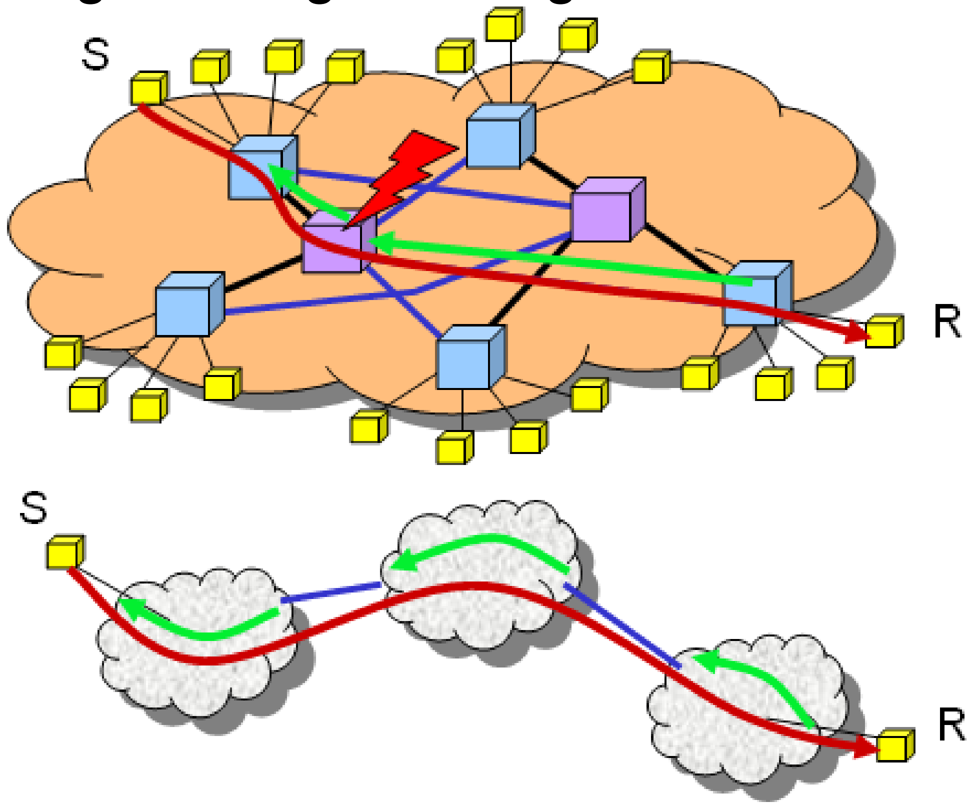
End-to-End

Congestion signalling is **triggered at relaying nodes or end systems**, but **action** is only taken **at end systems**

Link-/Segment-wise

Congestion signalling is **triggered at relaying nodes**, and **action** is taken **at neighbouring link / segment nodes**

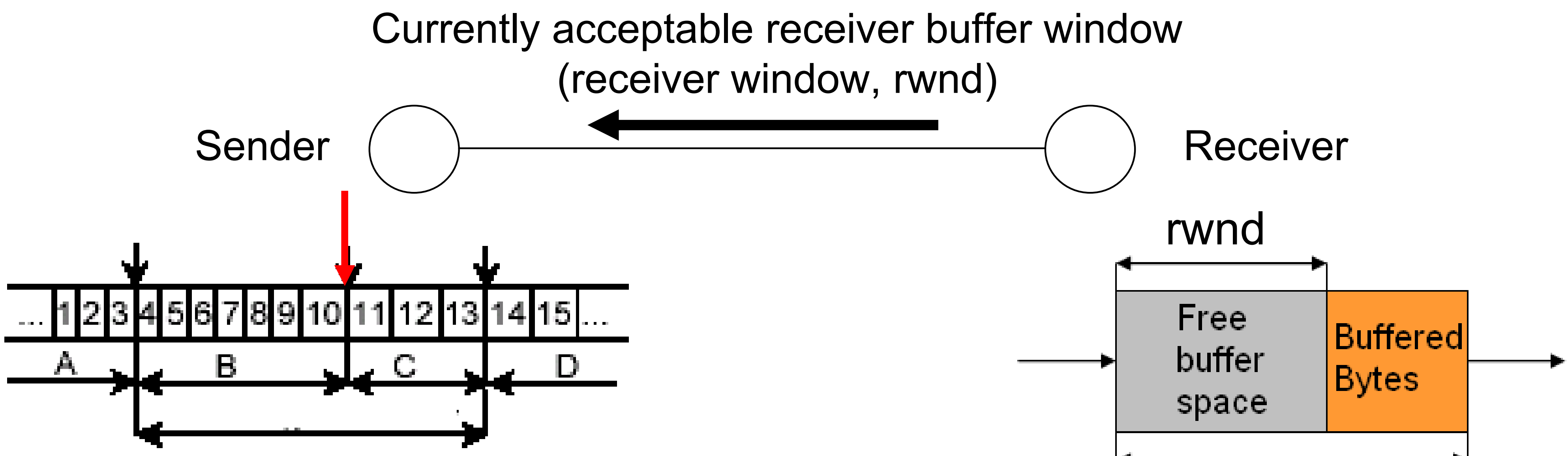




Congestion signalling
Data traffic

Flow/Congestion Control - Example TCP

- TCP implements Flow and Congestion control
- **TCP Flow control** → sliding window with receiver signalled "**rwnd**"



(Sender) window size = B+C = rwnd

Receiver buffer, rbuf

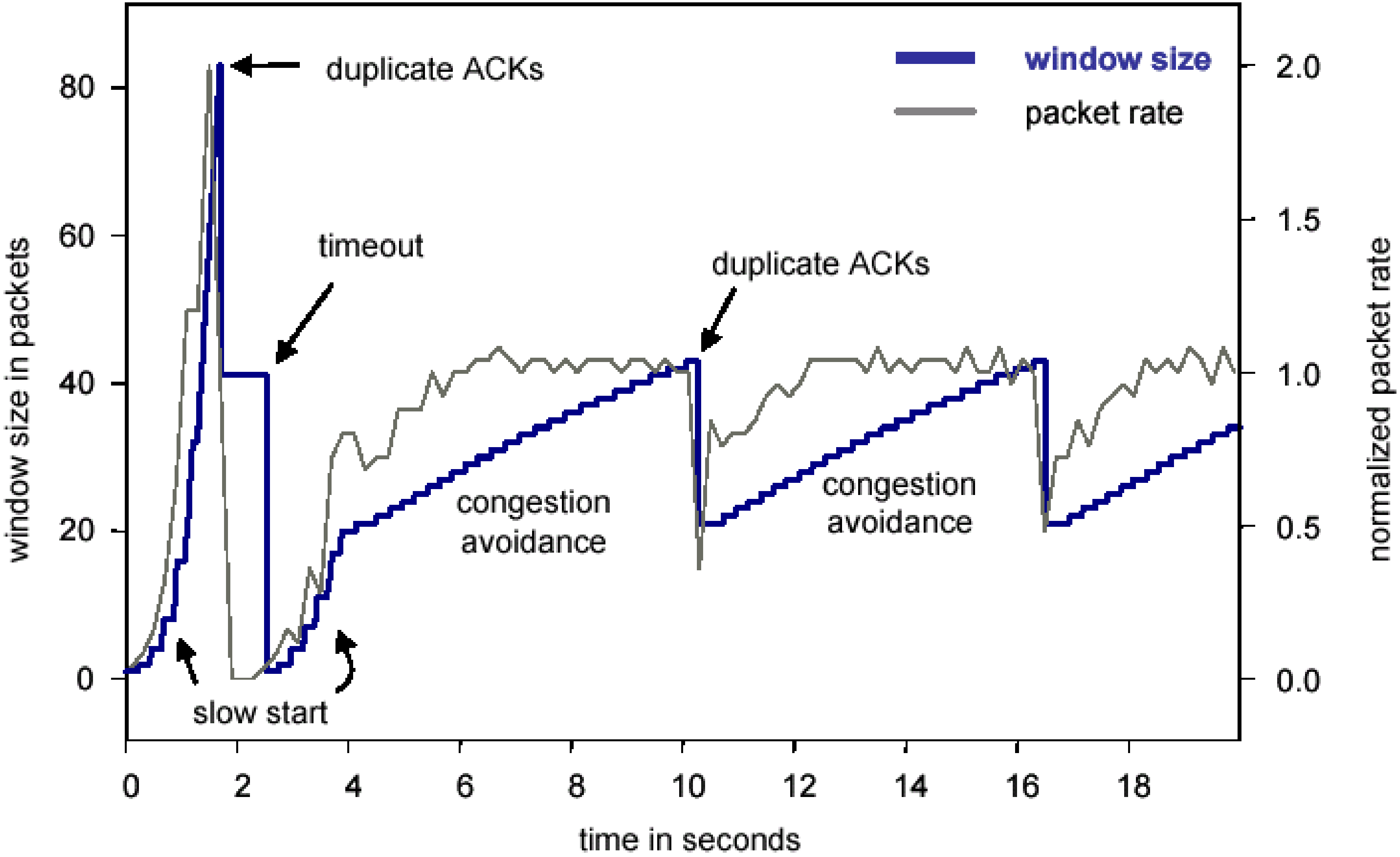
A: Bytes, sent and acknowledged
B: Bytes, sent and not yet acknowledged
C: Bytes, about to be sent without prior ACK clearance
D: Bytes, outside the window – waiting for sliding window shift

Flow/Congestion Control - Example TCP

- **TCP Congestion control** → slow start / congestion avoidance phase
  → sliding window with sender gauged "**cwnd**"
  → adaptive slow start cwnd threshold (sst) for phase change decision
- ***Sending rate* = min (*rwnd*, *cwnd*)**
- Congestion "detection" under the **assumption of packet loss** through:
  - **Retransmit timeout (RTO)** or
  - **3 Duplicate ACKs (DUP)**
- sst and cwnd adoption during **slow start**:
  no loss detected: cwnd = cwnd + 1 per acknowledged segment
  loss detected (RTO): sst = cwnd/2 ; cwnd = 1
- sst and cwnd adoption during **congestion avoidance**:
  no loss detected: cwnd = cwnd + 1/cwnd per acknowledged segment
  loss detected (RTO): sst = cwnd/2 ; cwnd = 1 → back to slow start
  loss detected (DUP): sst = cwnd/2 ; cwnd = cwnd/2

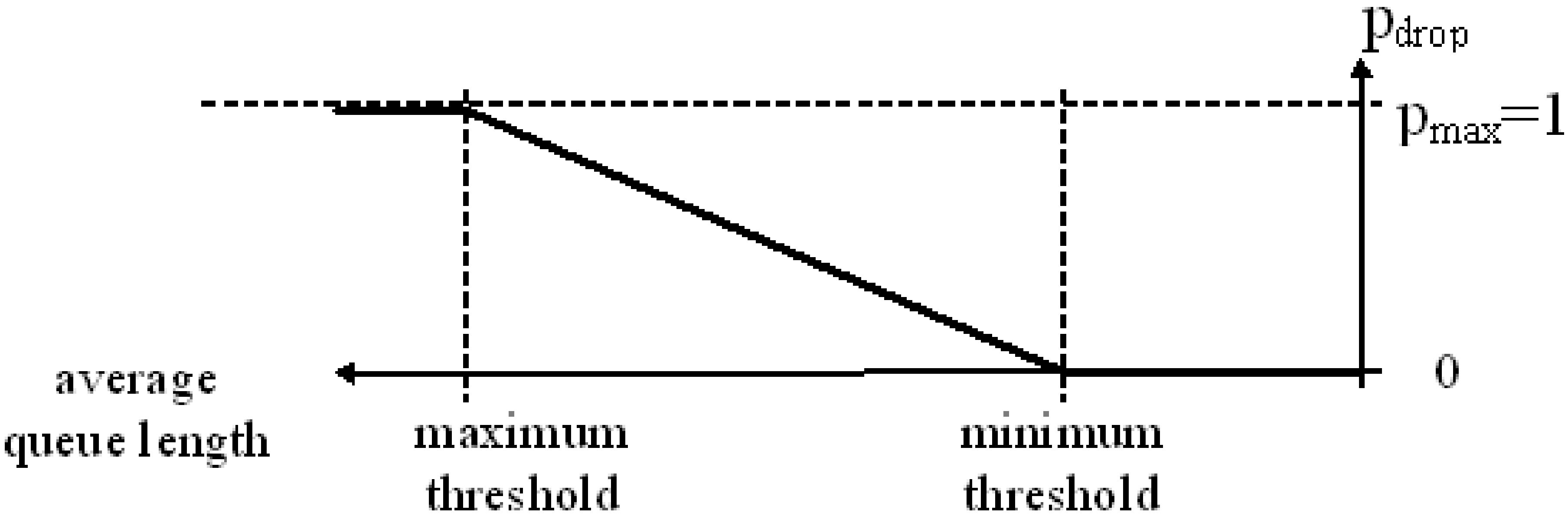
Flow/Congestion Control - Example TCP

- TCP Congestion Control → slow start / congestion avoidance phase



Flow/Congestion Control - Example RED

- **Active queue management** to pre-inform responsive transport protocols like TCP's Congestion control
  → **Random Early Detection (RED)**
  → probability based packet drops with probability increase towards the queue end
  → assumption: early "losses" will trigger congestion avoidance phase and will prevent buffer overflow

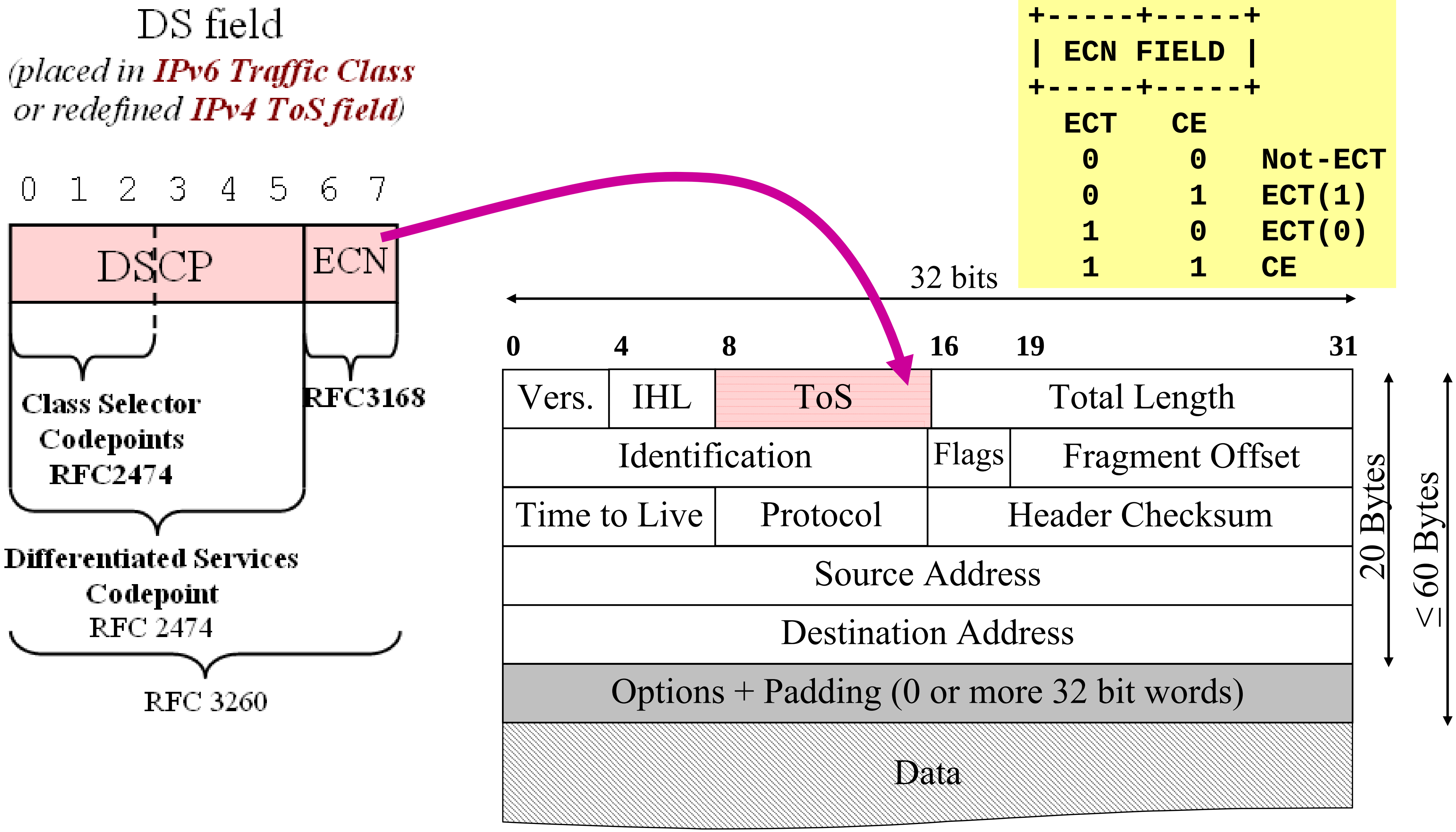


Flow/Congestion Control - Example ECN

- IP layer - Explicit Congestion Notification (ECN) – RFC 3168
- Reuse RED (or other incipient congestion detection mechanism)
  BUT: do not drop packets ! → mark them with congestion notification
- Work principle:
  - ECN capable sender marks IP packets as “ECT = ECN capable transport” packets
  - congested forwarding nodes (router) set “CE: congestion experienced” indication
  - receivers inform senders about the reception of CE marked packets
- Currently, ECN is focussed on use with TCP
- ECN capability negotiation during TCP connection setup
- 2 additional TCP flags:
  - “ECN-Echo (ECE)” → CE notification from receiver to sender
  - “Congestion Window Reduced (CWR)” → adoption ACK by sender
- CE indications are handled like packet drops in the cwnd adoption

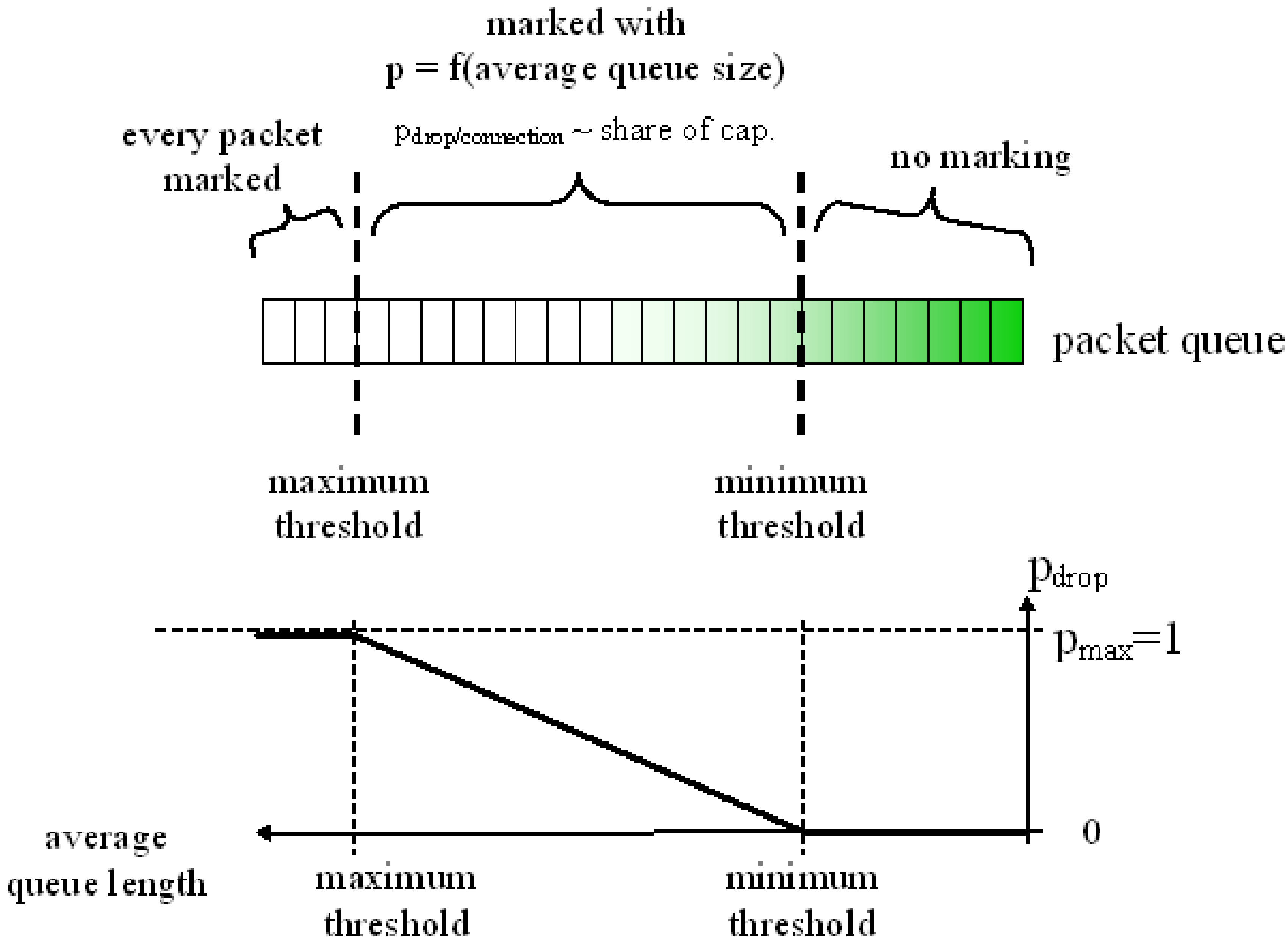
Flow/Congestion Control - Example ECN

- IP layer - Explicit Congestion Notification (ECN) – RFC 3168



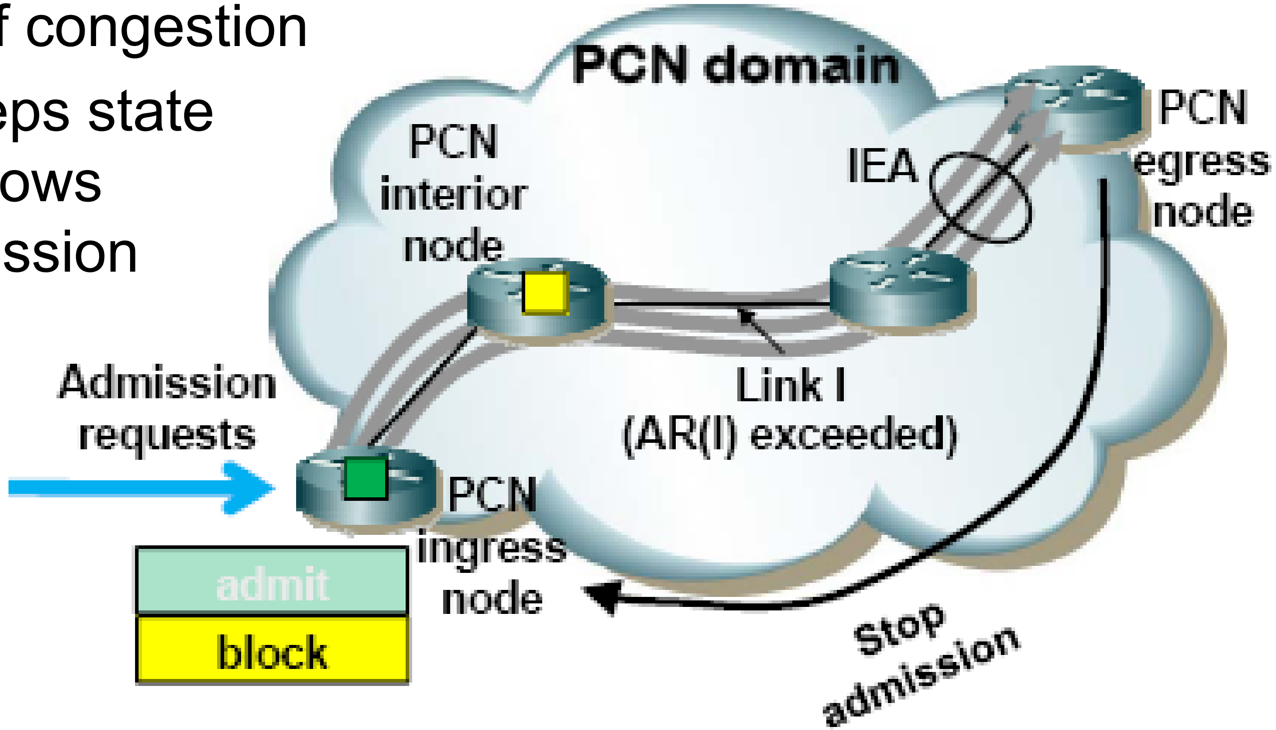

```
+-----+-----+
| ECN FIELD |
+-----+-----+
  ECT   CE
   0     0    Not-ECT
   0     1    ECT(1)
   1     0    ECT(0)
   1     1    CE
```

Flow/Congestion Control - Example ECN

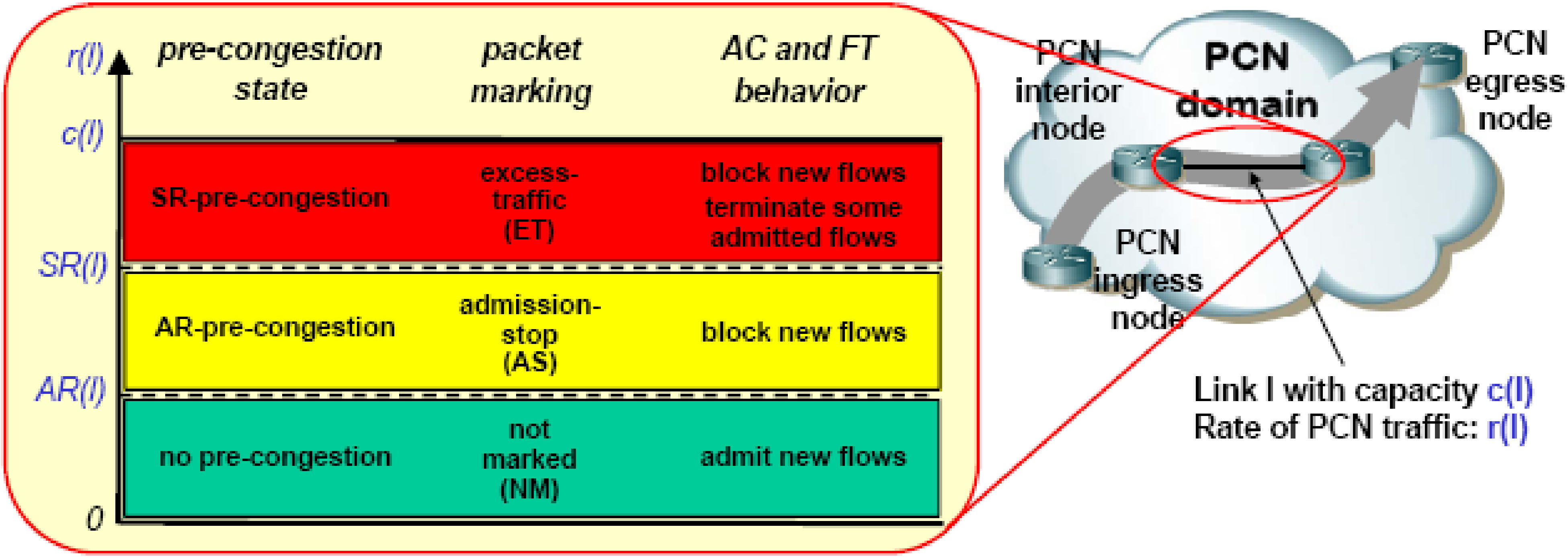


Flow/Congestion Control - Example PCN

- Pre-Congestion Notification (PCN)
- Lightweight Admission Control in PCN capable clouds including Flow Termination capabilities
- PCN reuses the ECN marking bits of the IP header for (and only for) the DSCP „VOICE-ADMIT“ = ‘101100’
- PCN performs admission control for packet flows and supports up to 3 level markings of congestion
- PCN ingress keeps state for all admitted flows and adopts admission based on egress congestion notifications



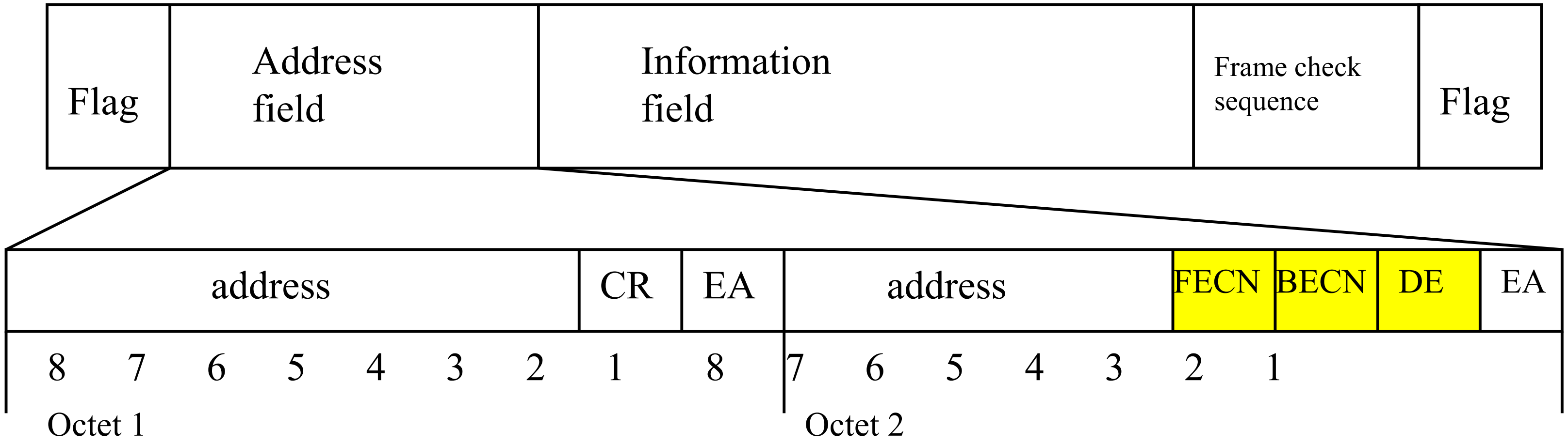
Flow/Congestion Control - Example PCN



- Admissible rate AR(l)
  - AR(l) < c(l)
  - Block flows if r(l) > AR(l)
- AR-Overload
  - Traffic rate exceeding AR(l)
- Supportable rate SR(l)
  - AR(l) < SR(l) < c(l)
  - Terminate admitted flows if r(l) > AR(l)
- SR-Overload
  - Traffic rate exceeding AR(l)

Flow/Congestion Control - Example F/BECN in FR/X.25

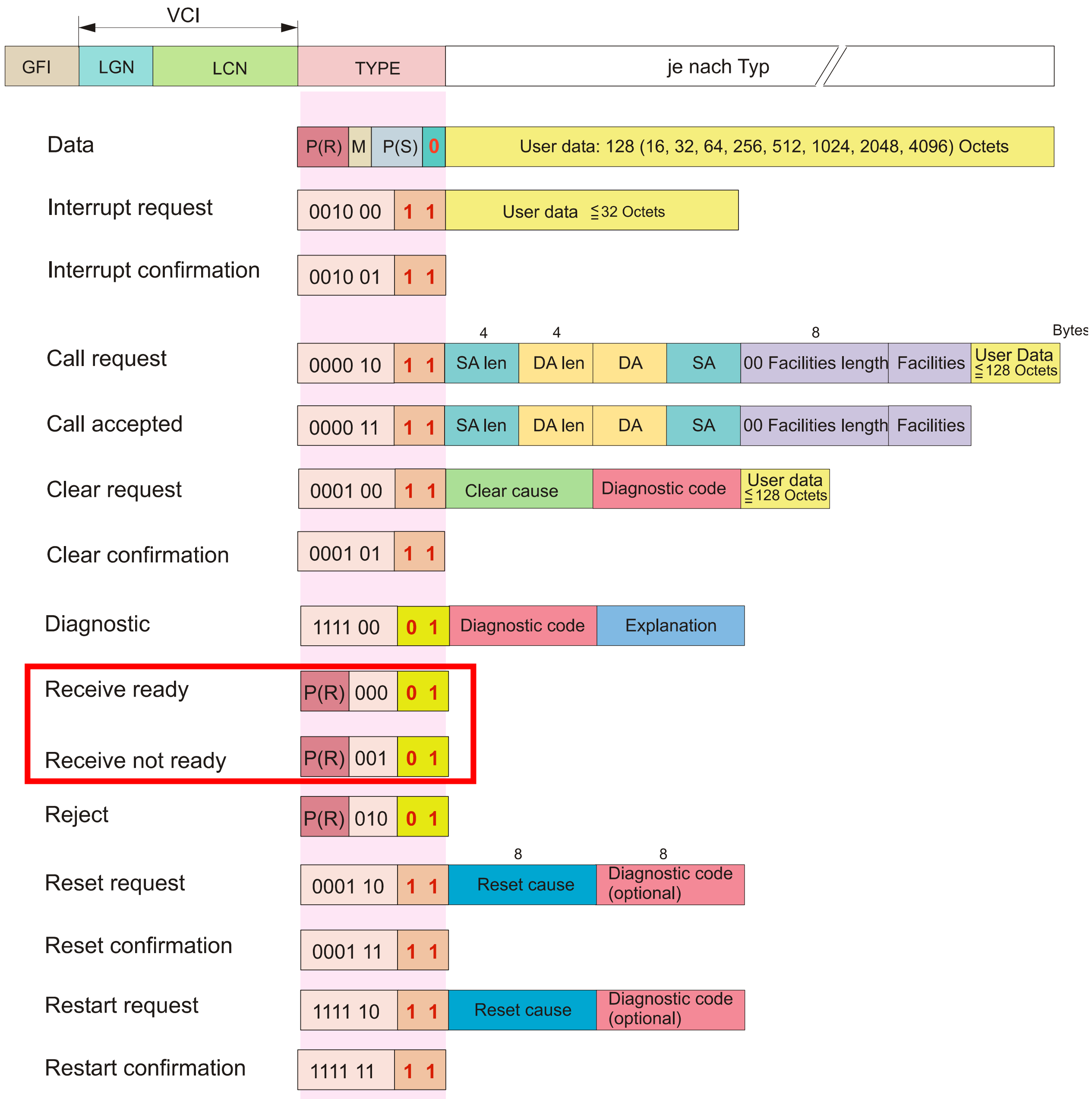
- Frame Relay (FR) – ANSI T1.618 and CCITT specification
- Frame header



- address: DLCI - Data Link Connection Identifier
- CR: 1 bit, user defined
- EA: extended address (“1” - there will be next address byte)
- **FECN: Forward Explicit Congestion Notification**
- **BECN: Backward Explicit Congestion Notification**
- **DE: Discard Eligibility → support for frame prioritization**

Flow/Congestion Control - Example F/BECN in FR/X.25

- X.25 – with control message based flow control
- No congestion marking
- Link based signalling (and ARQ)



Flow/Congestion Control - Example ATM

- Asynchronous transfer mode (ATM)
  = sophisticated QoS enabled packet network
- Connection-oriented and connectionless mode of operation with traffic class support and detailed traffic descriptions and negotiations.
- Traffic contracts are “signed” during connection setup, which define the requested/guaranteed service level.
- Connection Admission Control (CAC), traffic monitoring, traffic policing and traffic shaping enable and ensure high quality network operation.
- Congestion is mitigated through resource reservation and traffic shaping, but can not be waived completely due to the statistical multiplex of packets in intermediate nodes.
- User to Network Interface (UNI) frame format contains a “Generic Flow Control (GFC)” field of 4 bits, which was originally intended for flow control and access channel sharing → no longer used today

Flow/Congestion Control - Example ATM

4	8	16	3	1	8	384 (48 bytes)
GFC	VPI	VCI	Type	CLP	HEC (CRC-8)	Payload

User-Network Interface (UNI) frame (“cell) format

- host-to-switch format
- **GFC: Generic Flow Control (no longer used)**
- VCI: Virtual Circuit Identifier
- VPI: Virtual Path Identifier
- Type: management, congestion control, AAL5 (later)
- **CLP: Cell Loss Priority → support for frame prioritization**
- HEC: Header Error Check (CRC-8)

Flow/Congestion Control - Example Ethernet

- Ethernet “flow” control (Gigabit onwards)
- Congestion mitigation mechanism between Ethernet switches
- Slow down of upstream neighbour through special MAC control frames type = 0x8808
- Requests upstream sender to stop for “n * 512 bit times”.
- Recursive procedure, if upper stage runs into congestion itself.

Flow/Congestion Control - Example Ethernet

```
⊟ Frame 2 (64 bytes on wire, 64 bytes captured)
    Arrival Time: Jan 30, 2008 10:25:52.012139558
    [Time delta from previous captured frame: 0.036914825 seconds]
    [Time delta from previous displayed frame: 0.036914825 seconds]
    [Time since reference or first frame: 0.036914825 seconds]
    Frame Number: 2
    Frame Length: 64 bytes
    Capture Length: 64 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:macc]
    [Coloring Rule Name: Broadcast]
    [Coloring Rule String: eth[0] & 1]
⊟ Ethernet II, Src: 42networ_30:41:50 (00:0f:5d:30:41:50), Dst: Spanning-tree-(for-bridges)_01 (01:80:c2:00:00:01)
  ⊟ Destination: Spanning-tree-(for-bridges)_01 (01:80:c2:00:00:01)
      Address: Spanning-tree-(for-bridges)_01 (01:80:c2:00:00:01)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  ⊟ Source: 42networ_30:41:50 (00:0f:5d:30:41:50)
      Address: 42networ_30:41:50 (00:0f:5d:30:41:50)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: MAC Control (0x8808)
⊟ MAC Control
    Pause: 0x0001
    Quanta: 65535
```

Waiting time = Quanta * 512 bit times

Wireshark example of an Ethernet PAUSE frame

Flow/Congestion Control - Example Ethernet

- Hazard of congestion spreading through PAUSE backpressure