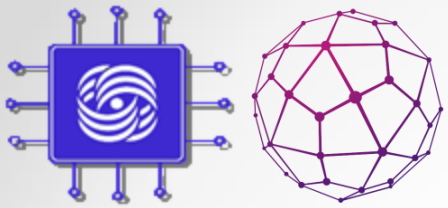
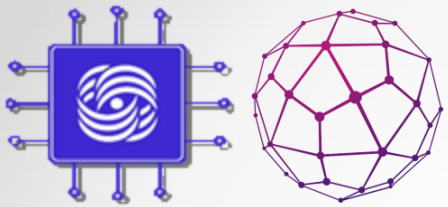


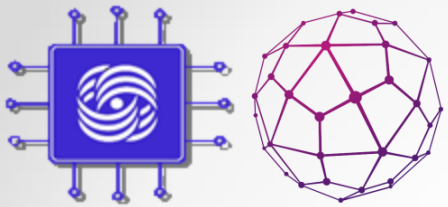
Quality of Service: hardware level resource allocation

E.P. Stepanov



Distribution of resources between flows

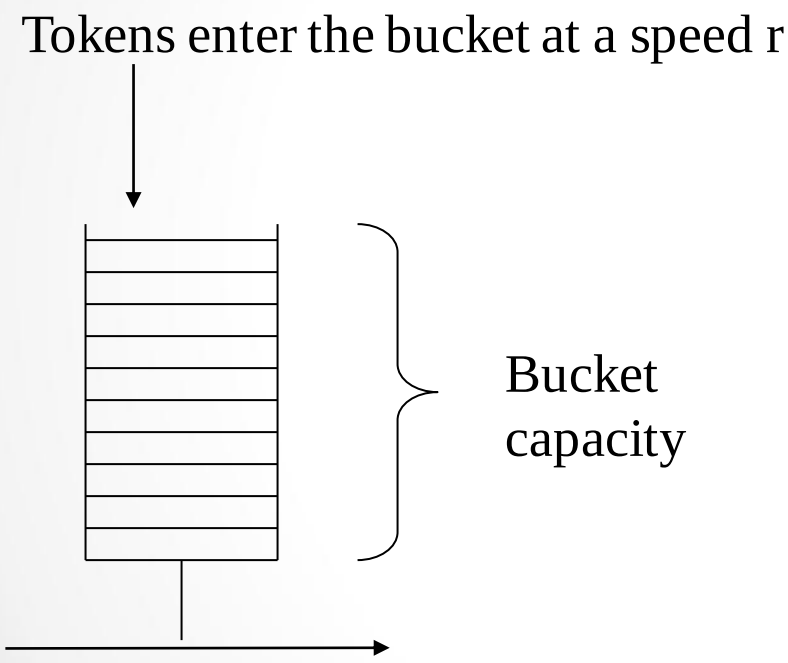
- The quality of service for flows is directly related to the **number of resources** allocated for its servicing: fractions of channel bandwidth, buffer memory, and processor time
- Connection quality management is achieved on switches using :
  - Flow queuing disciplines
  - Traffic shaping



Rate control

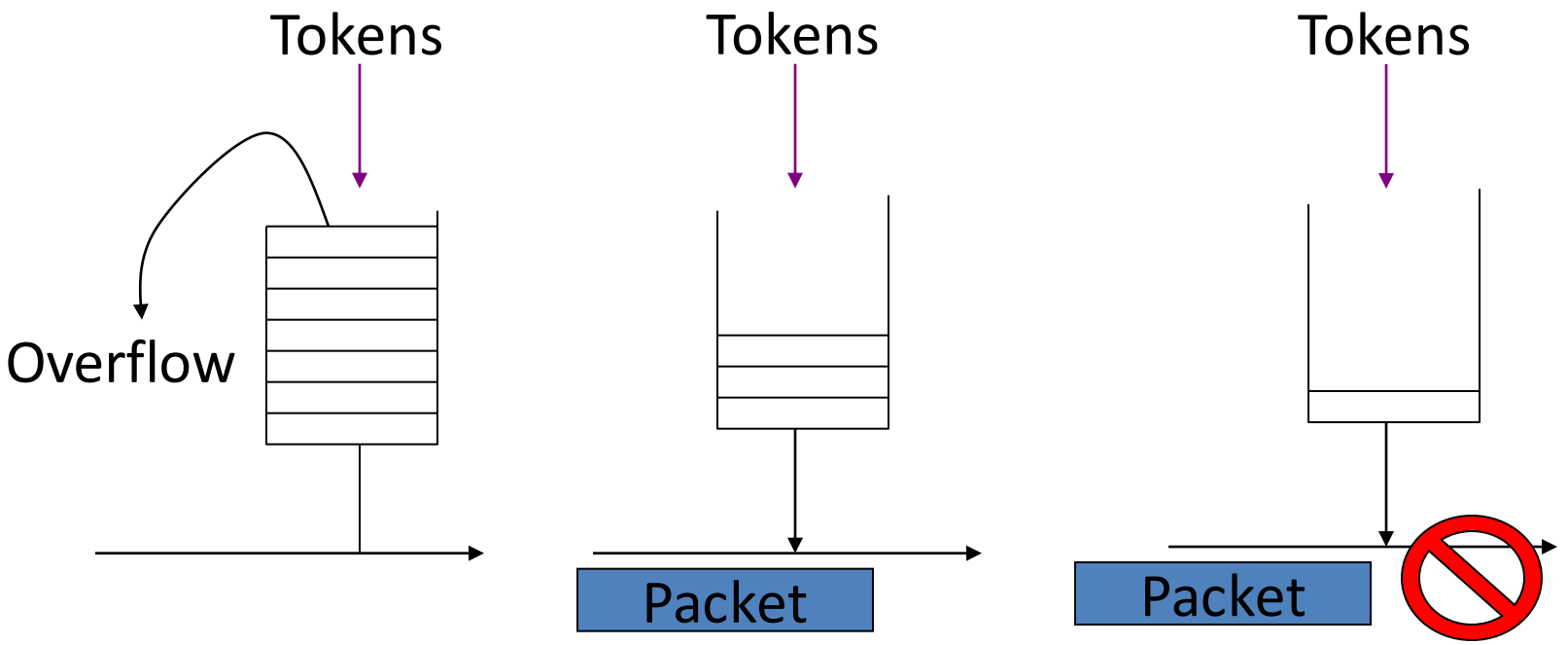
- ***The flow profile*** is determined by the intensity of its data transfer (speed, bursts, etc)
- ***Traffic Shaping & Policing*** allows you to give flow profiles the desired profile using the switch capabilities
- If the flow rate exceeds the profile specification:
  - shaping delays packet processing
  - policing discards packets

Token Bucket algorithm



- If the bucket is full, incoming tokens are ignored
- When sending a packet of size N bytes, N tokens are extracted from the bucket
- If there are not enough tokens in the bucket, then packet sending is delayed
- OpenFlow Metering

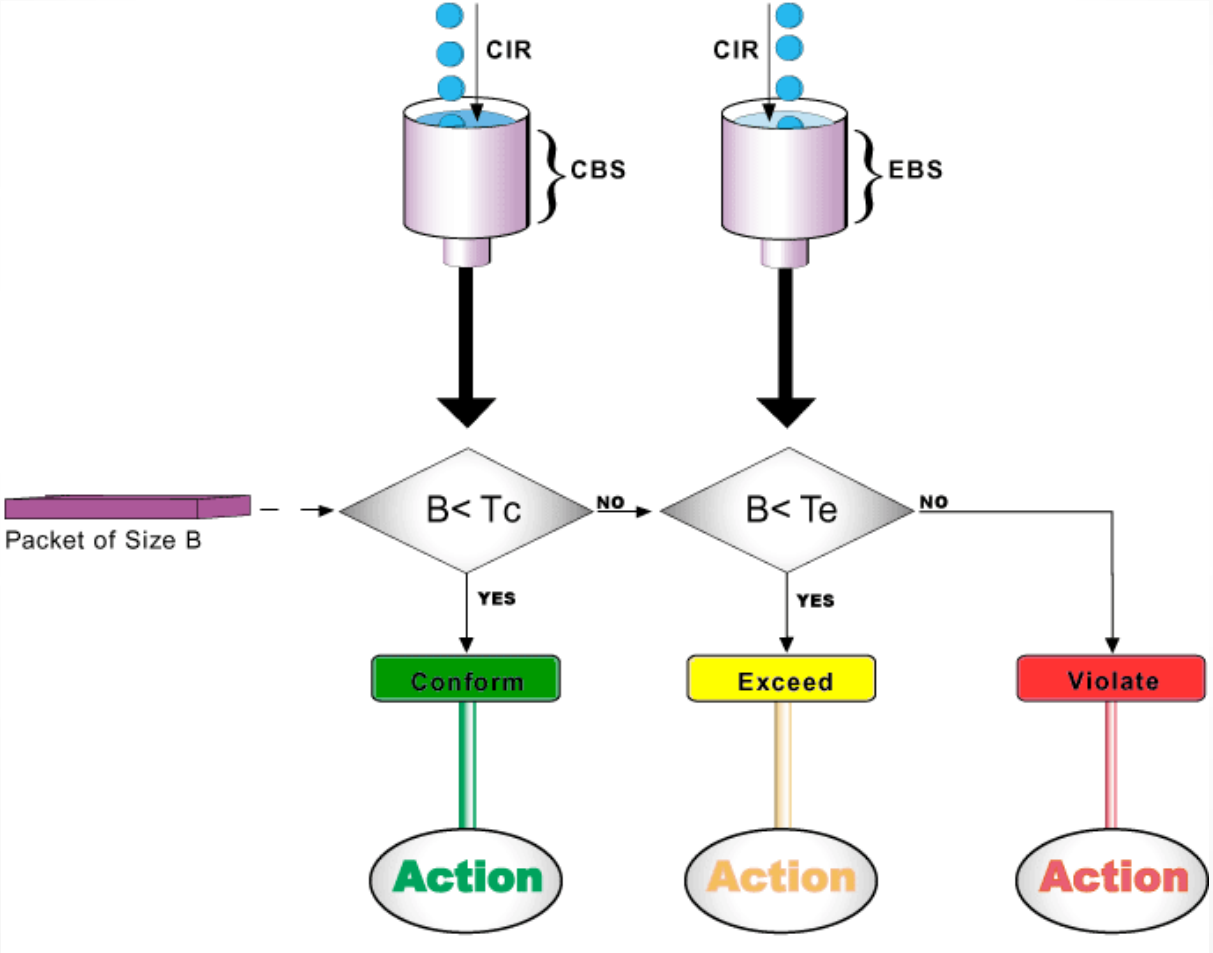
Token Bucket in Pictures

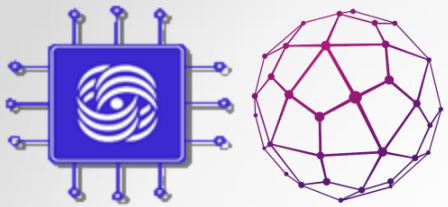


If there are enough tokens, the packet is sent, and the number of tokens decreases

Otherwise, the packet waits until new tokens accumulate in the bucket

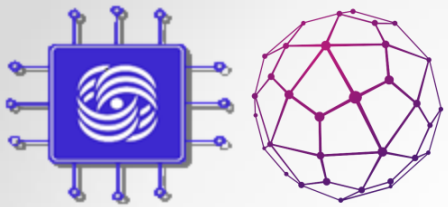
Using multiple token bucket algorithms





Queueing discipline

- Queuing discipline includes :
  - Planning for packet fetching from the queue
  - Packet discard policy
- The choice of queueing discipline determines:
  - Channel bandwidth allocation between flows - which packet will be sent next?
  - Buffer memory allocation - which packet will be discarded if there is not enough memory
- Queuing rules significantly affect latency
- OpenFlow Enqueue

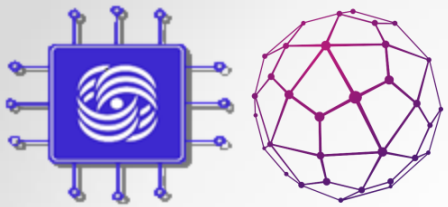


Typical Queuing Discipline

The simplest discipline:

FIFO + tail-drop

- ***FIFO (first-in-first-out)***: packets are selected from the queue in the same order in which they arrived
- ***Tail-Drop***: if there is no free space in the queue, then the packet sent to it is discarded, regardless of its importance



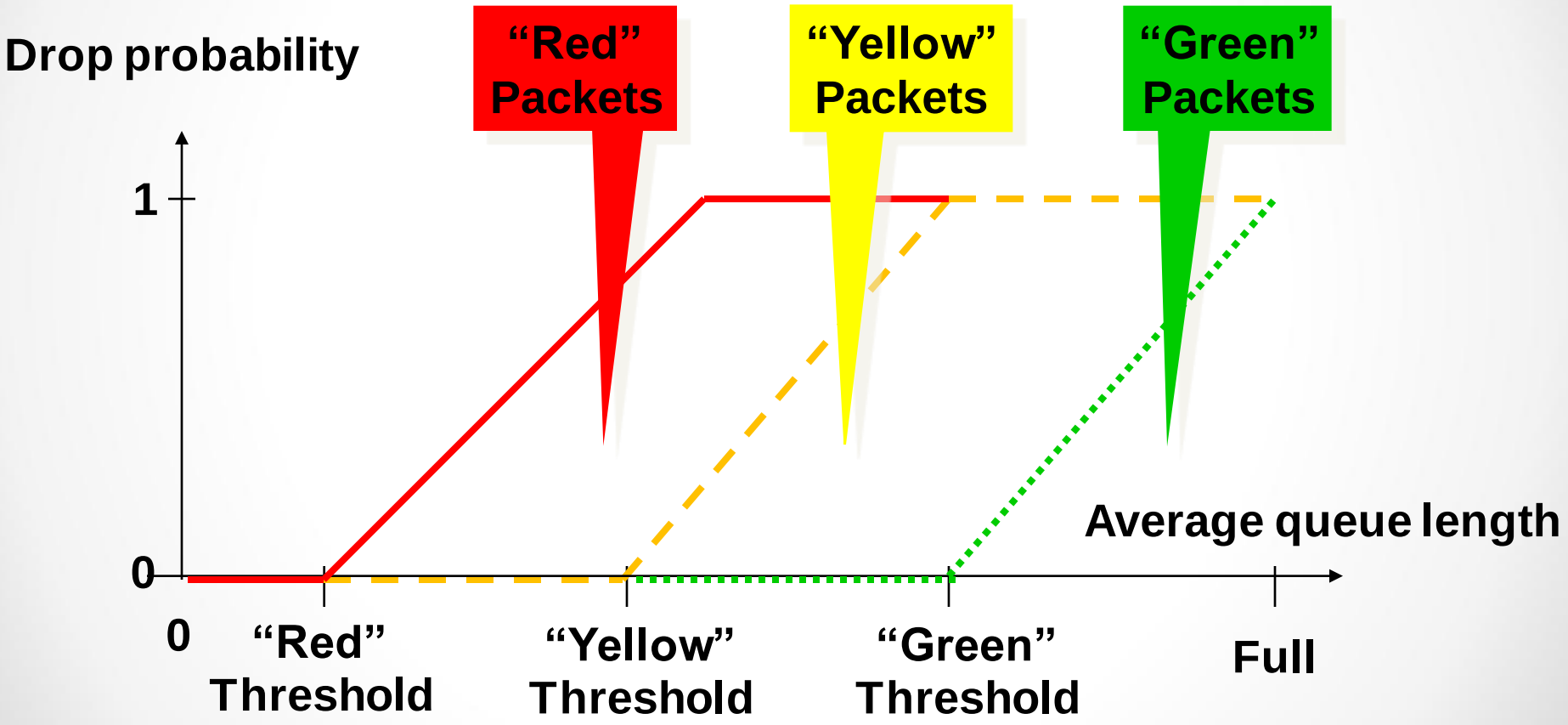
Disadvantages of FIFO + Tail-Drop

- Lock-out
  - Allows unlimited resource grabbing
  - The greater the flow intensity, the more resources it receives
- Flows are processed with the same quality
- Full Queue Problem
  - Results in end-to-end delay increase
  - There is an effect of synchronization of TCP traffic - the queue is either full or idle
- Flows with large bursts are harmed more than others

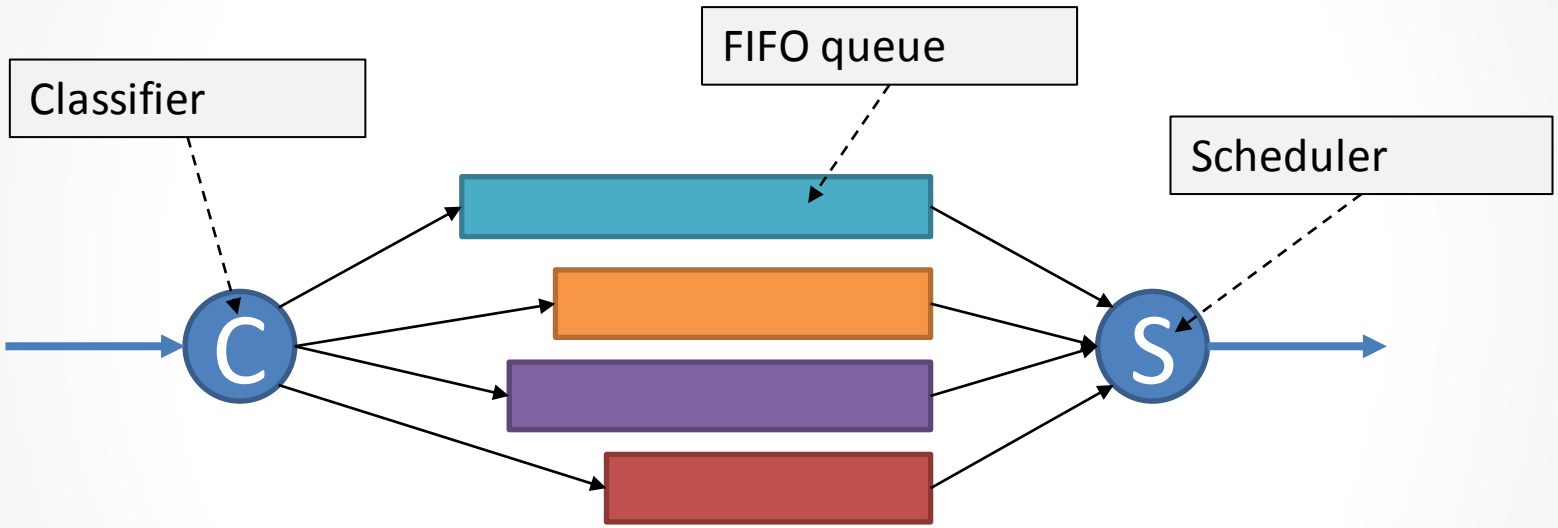
Bypassing lock-out issues

- Random drop
  - If a packet is directed to the queue and there is no place for it in it, a random packet is deleted from it
- Packets are discarded before queue overflow (random early detection)
  - Calculates the average queue load x_n
  - If $x < x_{min}$, then packets are not discarded
  - If $x > x_{max}$, then the incoming packet is discarded
  - Otherwise, the packet is discarded with a probability linearly dependent on the proximity to the threshold values

RED with multiple thresholds

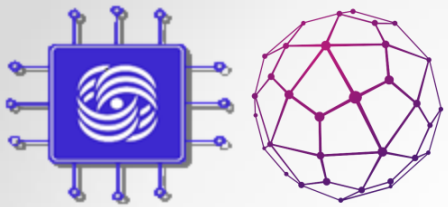


Buffer block



The classifier distributes packets in queues

Scheduler selects packets from queues

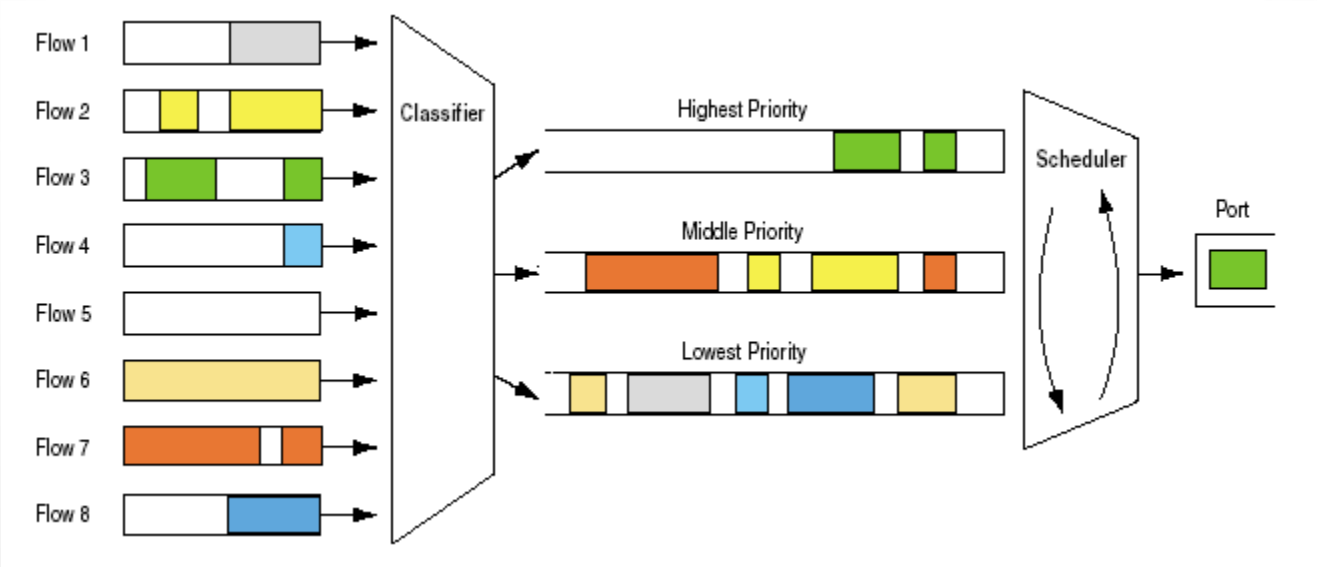


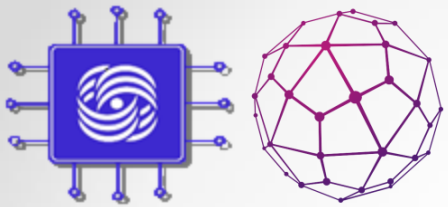
Common Queuing Disciplines

- First-In-First-Out (FIFO)
- Priority Queuing (PQ)
- Fair Queuing (FQ)
- Weighted Fair Queuing (WFQ)
- Weighted Round Robin (WRR)
- Shared Round Robin (SRR)
- Deficit Weighted Round Robin (DWRR)

Priority Queuing (PQ)

- Packets are distributed in several queues
- Each queue is assigned its own priority
- The scheduler retrieves a packet from the queue only if all queues with high priority are empty
- Each of the queues is served by FIFO discipline





Priority Queuing

Advantages

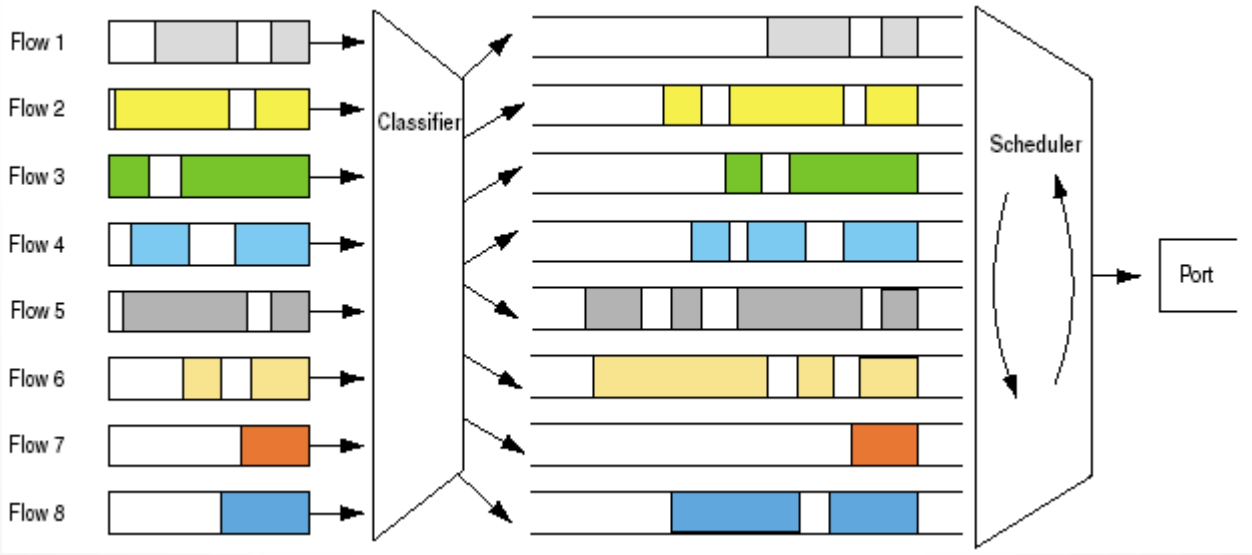
- Allows you to organize traffic differentiation in an easy to implement way
- Ability to transmit data with low latency

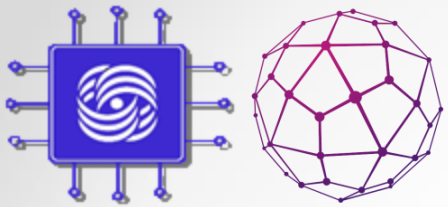
Disadvantages

- There is a risk of a flow starvation - it is better to use additional rate-control mechanisms
- Low-priority traffic can experience significant delays
- The struggle between flows directed in the same queue is not vanished

Fair Queuing (FQ)

- Each flow has its own queue for temporary storage of packets
- Packets are selected from the queues cyclically



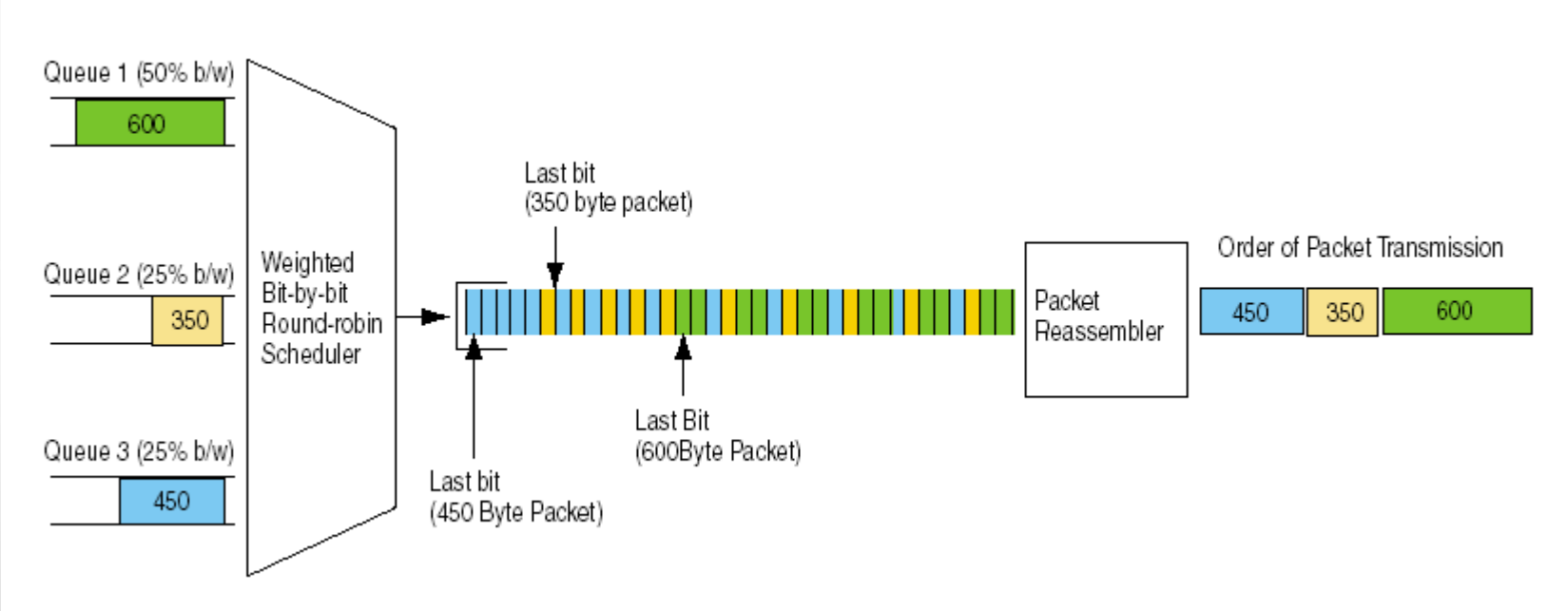


Fair Queuing (FQ)

- Flows are isolated from each other
- Implementation based on hardware queues is difficult (emulation is possible by distributing using hashes)
- Can be used in conjunction with other queueing disciplines
- Differentiation between data flows is not performed, flows with different bandwidth requirements are not supported
- There are no mechanisms for transmitting real-time traffic
- Flows with large size packets take advantage

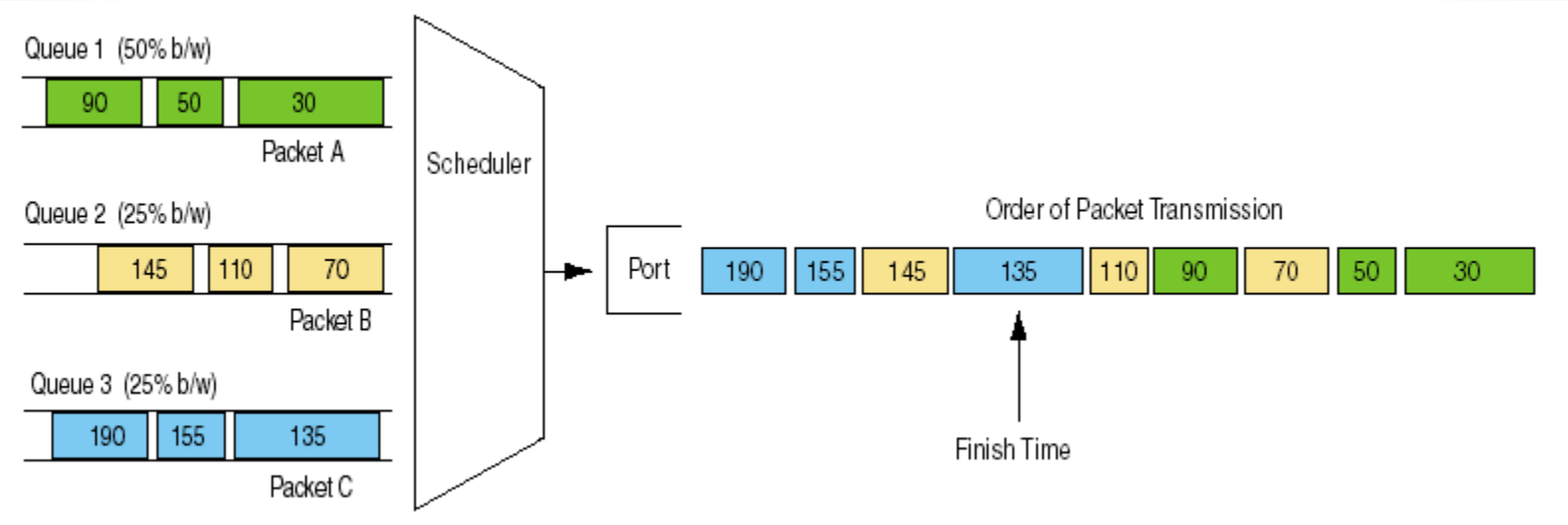
Queuing on the principle of fluid model

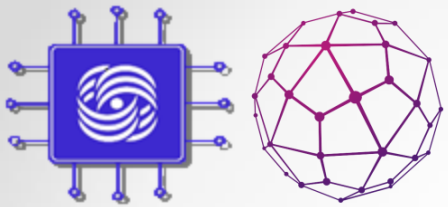
- Uses weights to support flows with different bandwidth requirements
- Considers the packet size while selecting packets from the queue



Weighted Fair Queuing (WFQ)

- Abstraction of fluid model is approximated by calculating the completion time of the packet transmission
- The scheduler selects the packets with the lowest estimated transmission completion time.





Weighted Fair Queuing (WFQ)

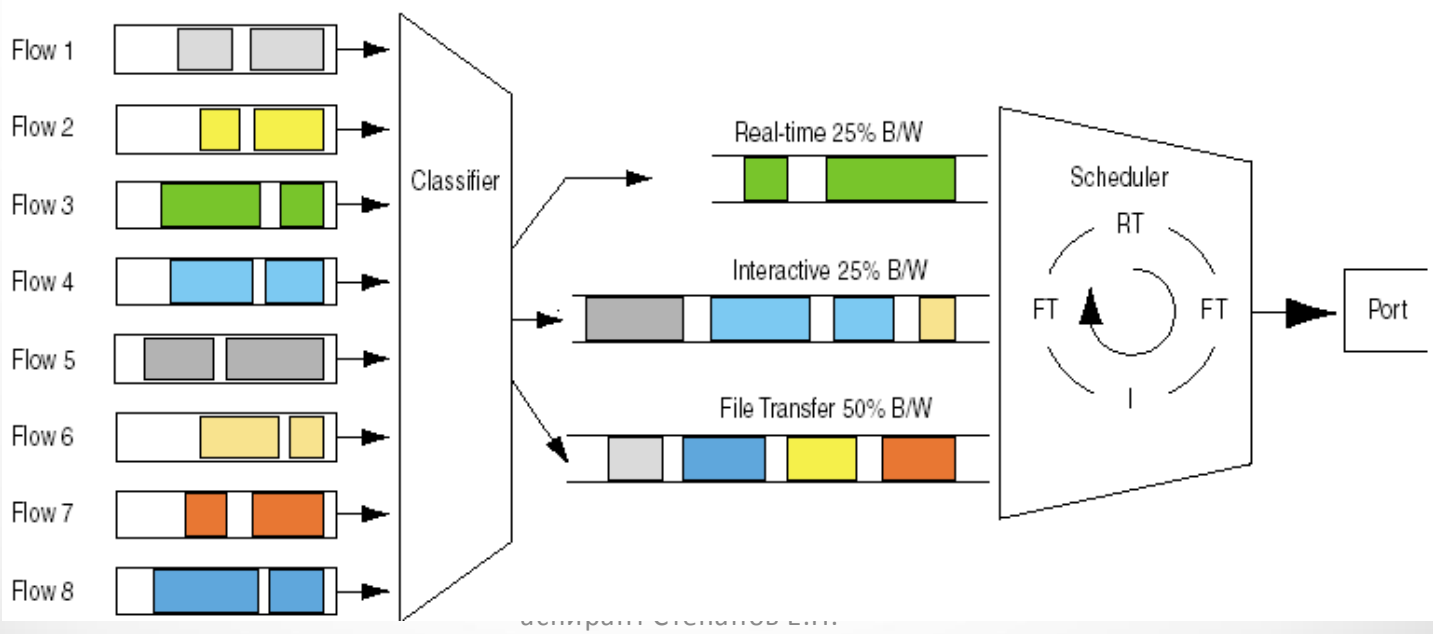
- Hardware implementation complexity
- High complexity of the algorithm - for each queue it is necessary to maintain a state - a time stamp, and update its value every time a packet arrives or is sent
- Poor scalability

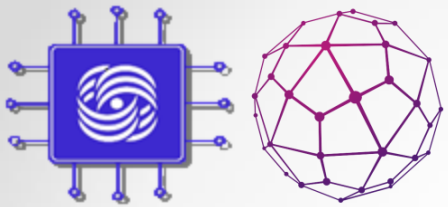
There are more advanced analogues:

- Self-clocking Fair Queuing (SCFQ)
- Worst case Fair weighted Fair Queuing (WF^2Q)
- Worst case Fair weighted Fair Queuing+ (WF^2Q+)

Weighted Round-Robin

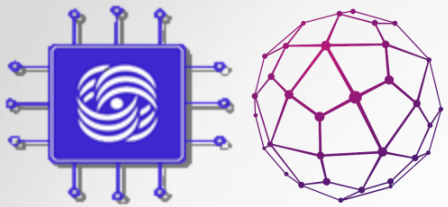
- Flows are distributed into several classes, each class has its own queue
- Queues are serviced cyclically
- The number of packets extracted from the queue corresponds to its weight coefficient





Weighted Round-Robin

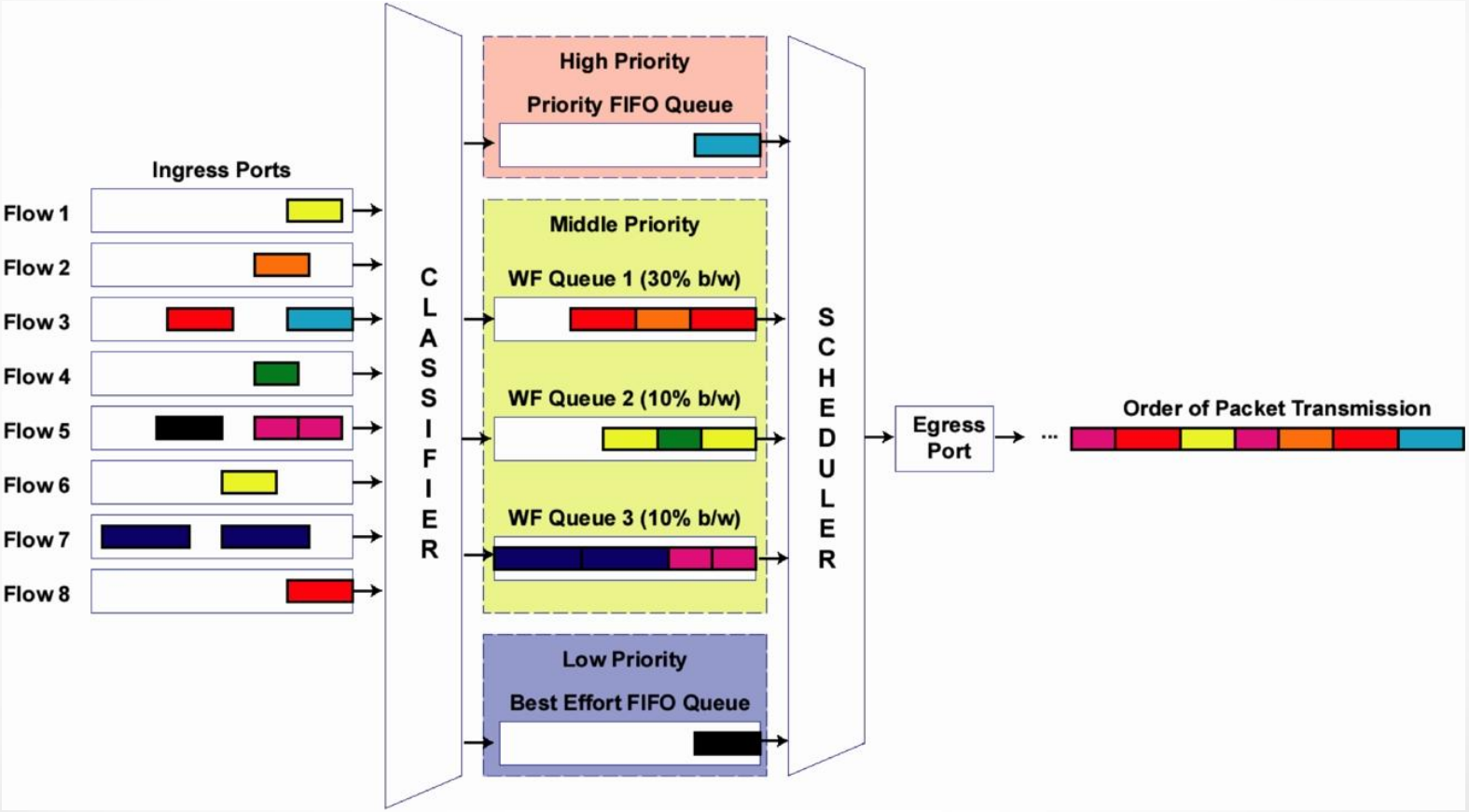
- Simple model - the algorithm can be implemented in hardware
- Queues for traffic of medium and low importance are not ignored - each class of flows receives its share of resources
- Works well only if the packet sizes are equal
- It doesn’t mix traffic very well, preserving consecutive packets from one flow

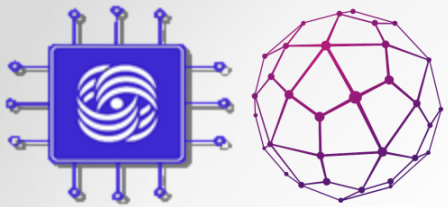


WRR improvements

- SRR – shaped round-robin
  - The selection takes place in rounds
  - A queue is excluded if its weight is less than the round number
  - Only one packet from each of the queues participating in this round is selected per round
  - If there are no non-empty queues left in the round, start from the first round
- DWRR – deplicit weighted-round robin
  - Sampling is not by packets, but bytes - the problem of packets of different sizes is solved

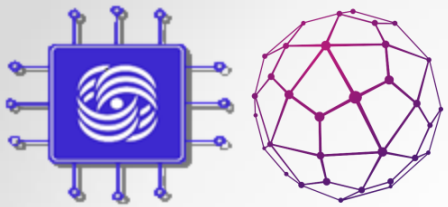
Combining multiple disciplines





Quality of Service: network resource allocation models

E.P. Stepanov



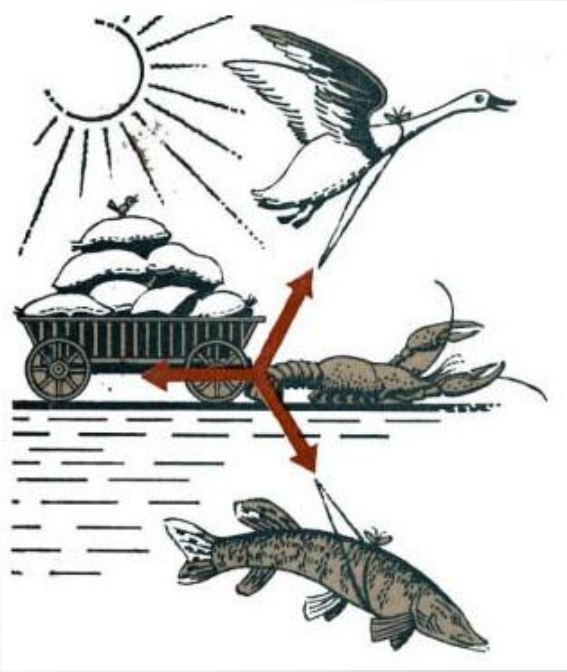
Quality of Service in Internet

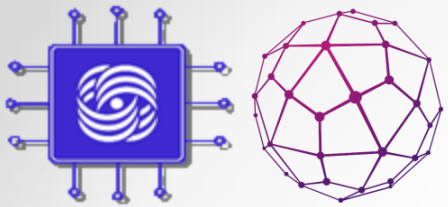
- If QoS management is not supported, then network connections are served according to the **best effort** principle.
- Data flows are served equally
  - Why do you need a different quality of service?
  - Differentiating flows by quality: is it always a good idea?

David D. Clark, K. Sollins, J. Wroclawski, R. Braden, ***Tussle in Cyberspace: Defining Tomorrow's Internet*** Proceedings of SIGCOMM 2002, ACM Press, 2002

Methods of QoS management in the network

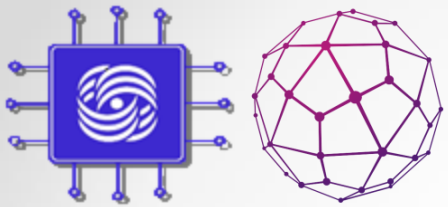
- Meeting application needs
  - What connection parameters can be set?
  - Is there a guarantee of compliance?
- The complexity of implementation and operation
  - What additional features should the equipment have?
  - What kind of support is required from the hosts?
- Usage overhead
  - How effective is the network?
  - How many resources will be involved?
  - How many resources will be idle?





The problem of efficient resource allocation

- The quality assurance problem is related to the problem of distribution of network resources between data flows
  - The problem of resource allocation can be formalized as an optimization problem
  - The more resources involved in servicing a flow, the higher the quality of its connection
  - The more resources that the distribution model allows you to use, the higher the network efficiency, and the greater the utilization level



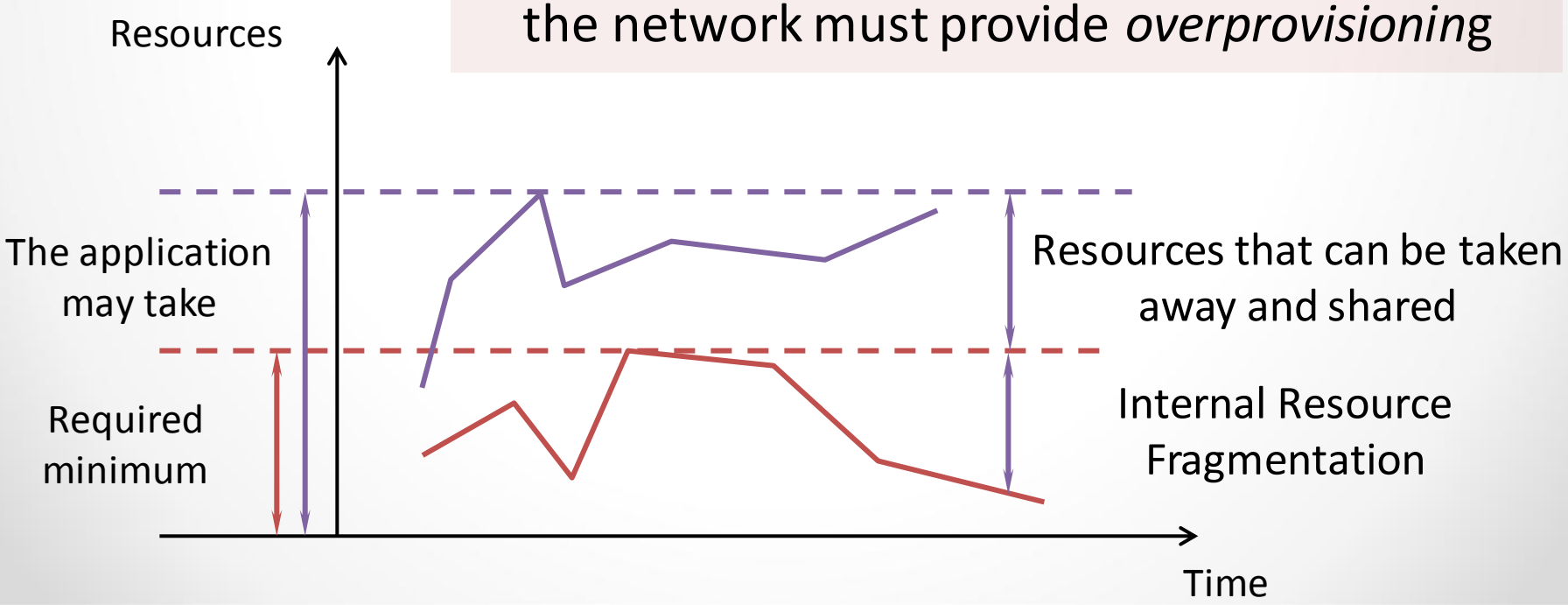
Why is it good to have a high level of utilization?

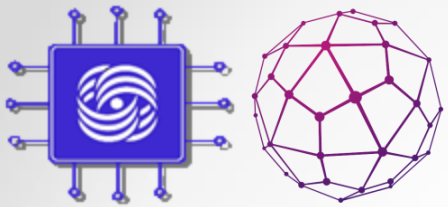
- Applications are not always able to use all network resources
  - Spanning Tree Protocol – with Fat Tree topology
  - Backbone networks – performance reserve
  - The difference in the equipment characteristics
- The higher the utilization level, the better the ratio of network performance to the cost of network infrastructure
  - Provider cost reduction

Low utilization with poor QoS management

The network works correctly if each application is provided with the required amount of resources

To prevent violation of quality requirements, the network must provide *overprovisioning*



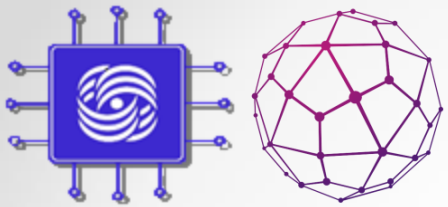


Integrated services (IntServ)

Multimedia traffic in the network:

- How to protect TCP traffic from multimedia data transmitted over UDP?
- How to ensure the quality of the connection?

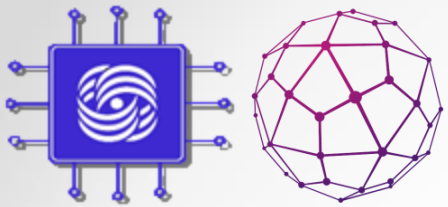
A guaranteed level of quality can be ensured only by **reserving resources** - securing a part of the network resources to a specific data flow



Integrated services (IntServ)

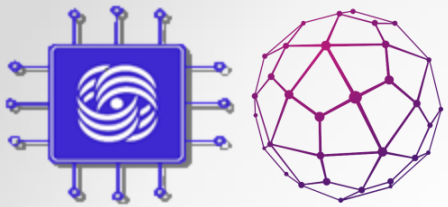
The main idea is to construct routes with a given quality by pre-booking resources of network equipment

- Model only extends the Internet architecture, maintaining compatibility with the *best-effort*
- The model is especially effective in multicast data transmission
- Overhead costs for pre-routing are allowed
- ***all or nothing*** **-** the model either guarantees a connection of the desired quality, or refuses to provide any connection



Main components of the IntServ model

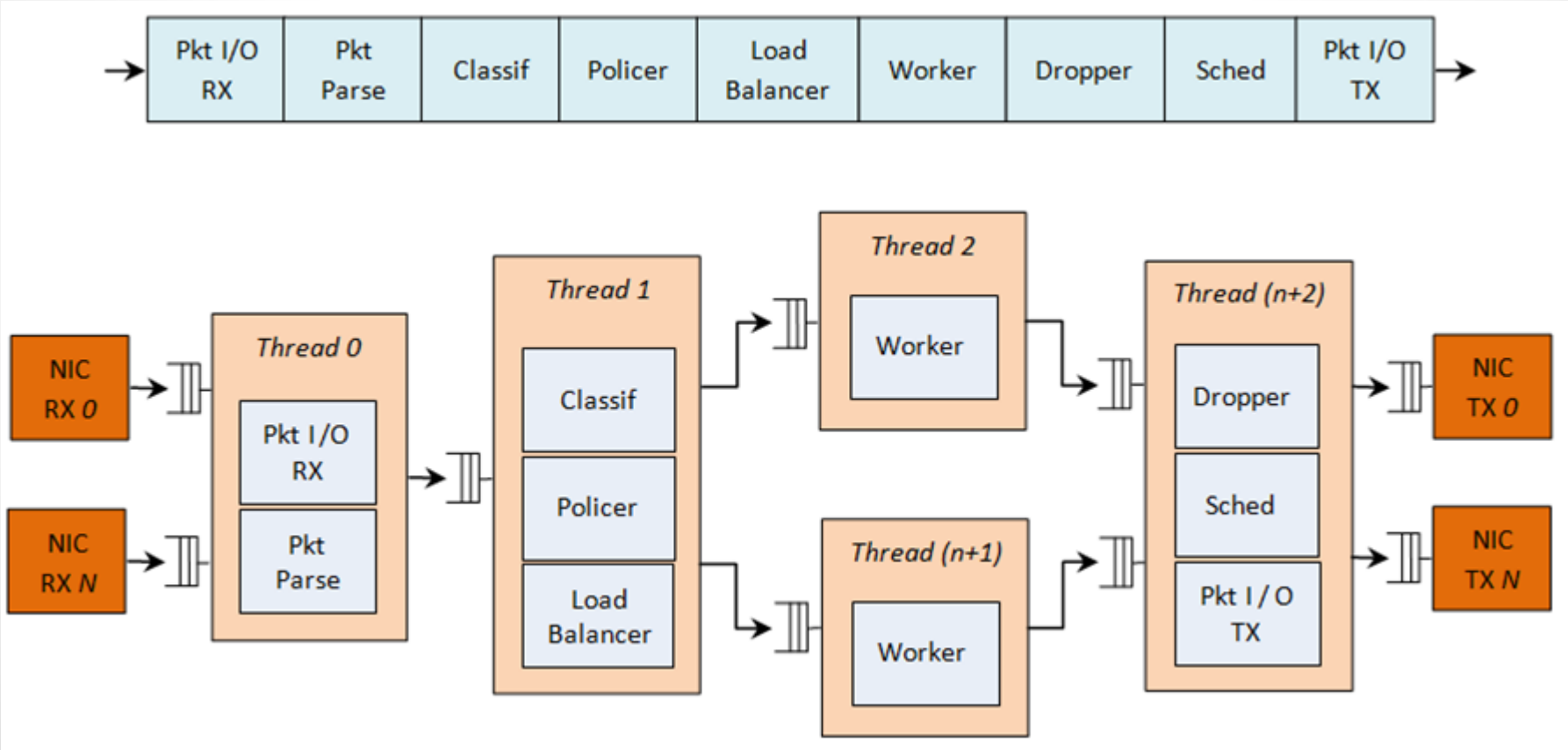
- Classifier
  - Division of packets into service classes
- Scheduler
  - Enforcing QoS Requirements
- Admission control
  - Evaluation of the ability to add flows
- Resource Reservation Protocol
  - Reserving resources along the route

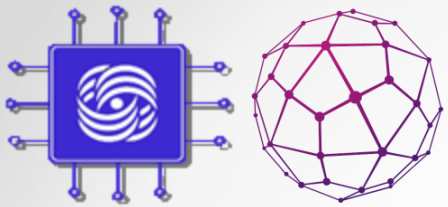


Calculation and maintenance of connection quality

- Queuing on inbound and outbound switch interfaces
- Using policing & shaping to form the desired flow profile
- Setting proper disciplines for dropping and fetching packets from queues
- Setting up switching matrix scheduling algorithms

Calculation and maintenance of connection quality

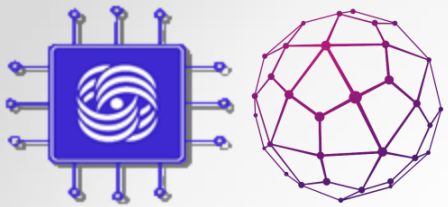




RSVP Resource Reservation Protocol

Supported traffic types:

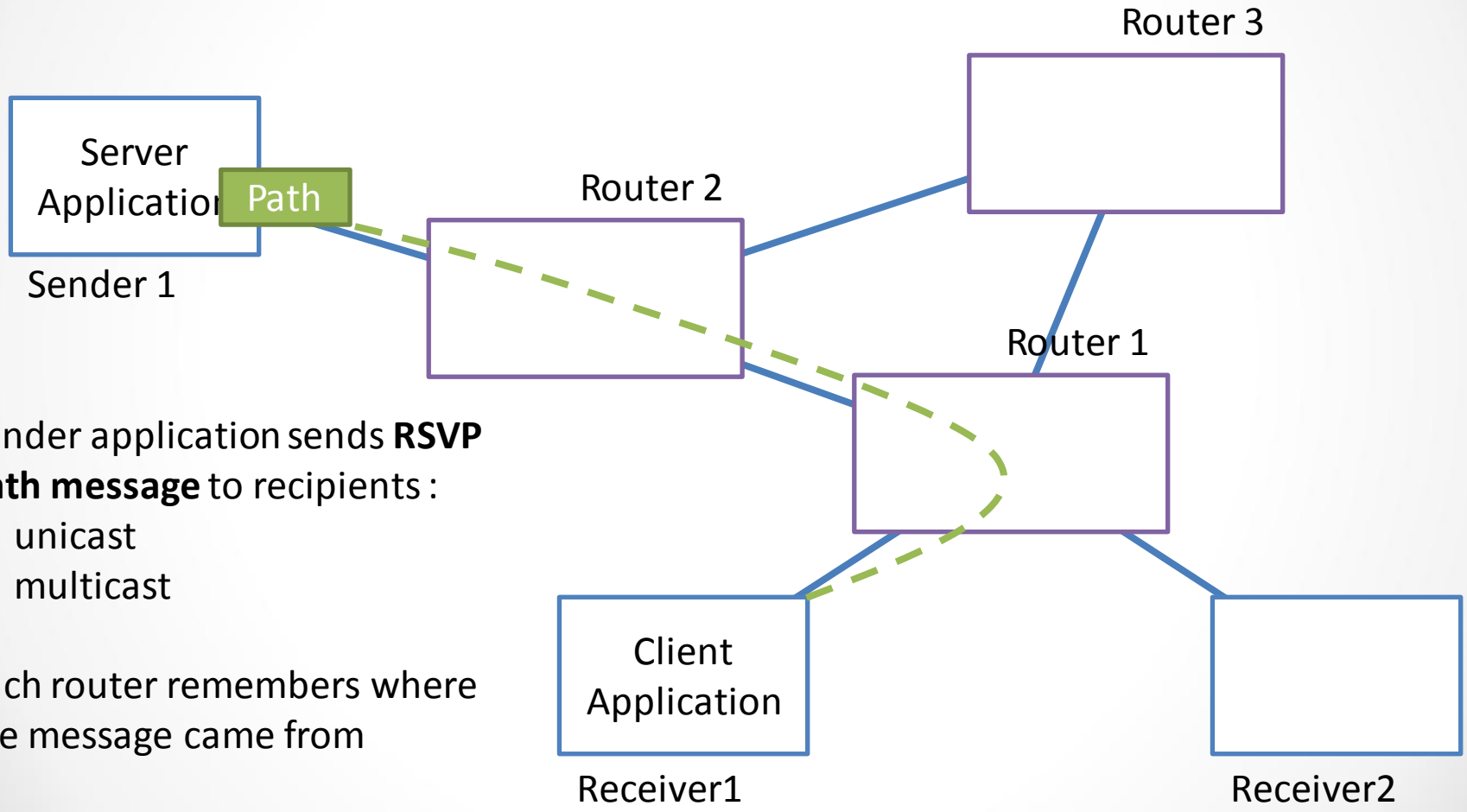
- *Best-effort*
  - File Transfer, Email Browsing, etc
- *Rate-sensitive*
  - Streaming audio and video
- *Delay-sensitive*
  - Voice Over IP, online games



RSVP Resource Reservation Protocol

- ***Data flow -*** a sequence of packets having a common sender, a single set of recipients, and requiring the same level of QoS
- ***Flow specification (flowspec)***
  - Defines data flow, type of traffic, connection quality requirements
- ***Filter specification (filterspec)***
  - Resource allocation rules for flow processing
- RSVP works at **the session level** - multiple flows with the same set of recipients

RSVP Operation Example

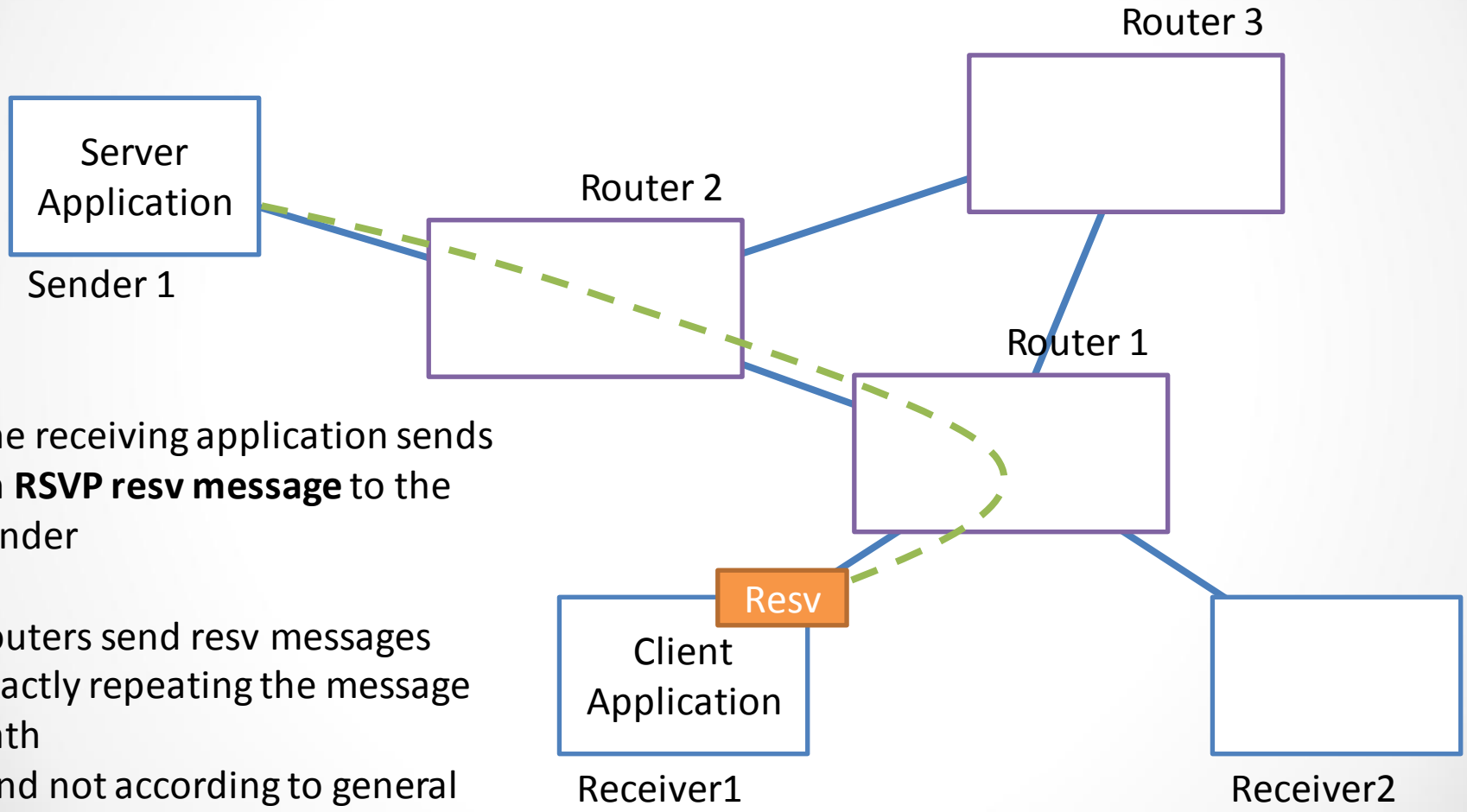


Sender application sends **RSVP path message** to recipients :

- unicast
- multicast

Each router remembers where the message came from

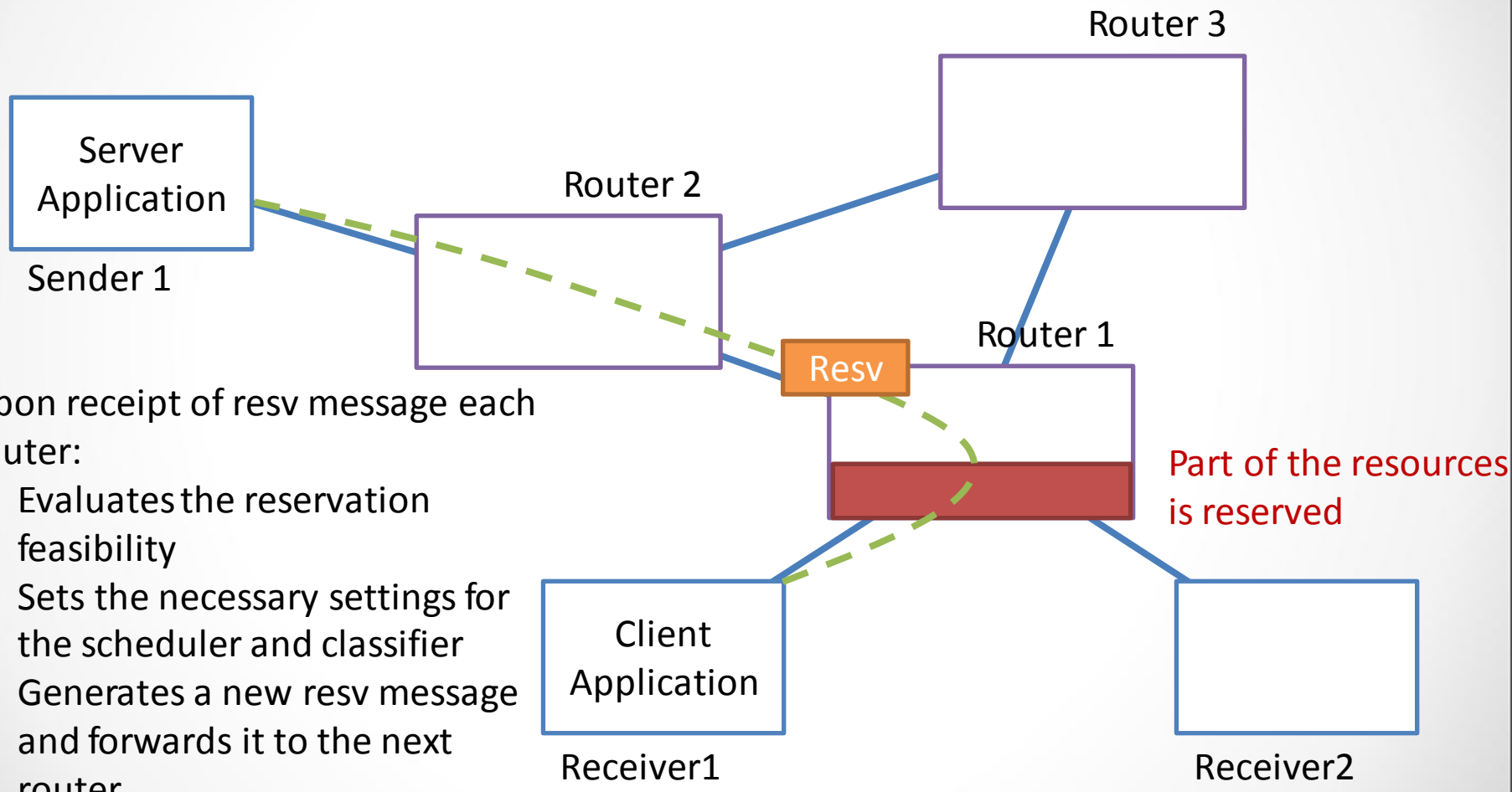
Route construction



The receiving application sends an **RSVP resv message** to the sender

Routers send resv messages exactly repeating the message path
(and not according to general packet routing rules)

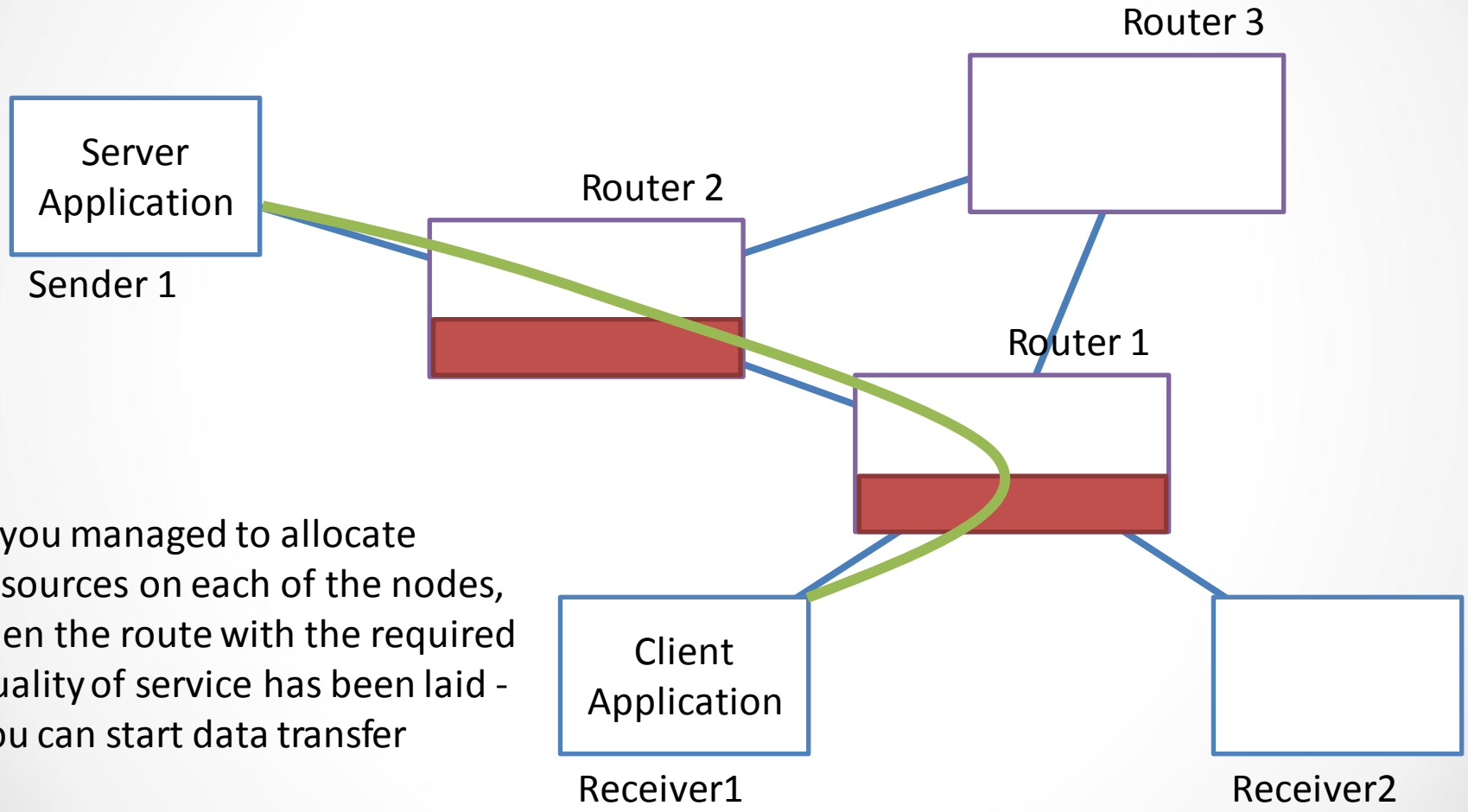
Resource reservation



Upon receipt of resv message each router:

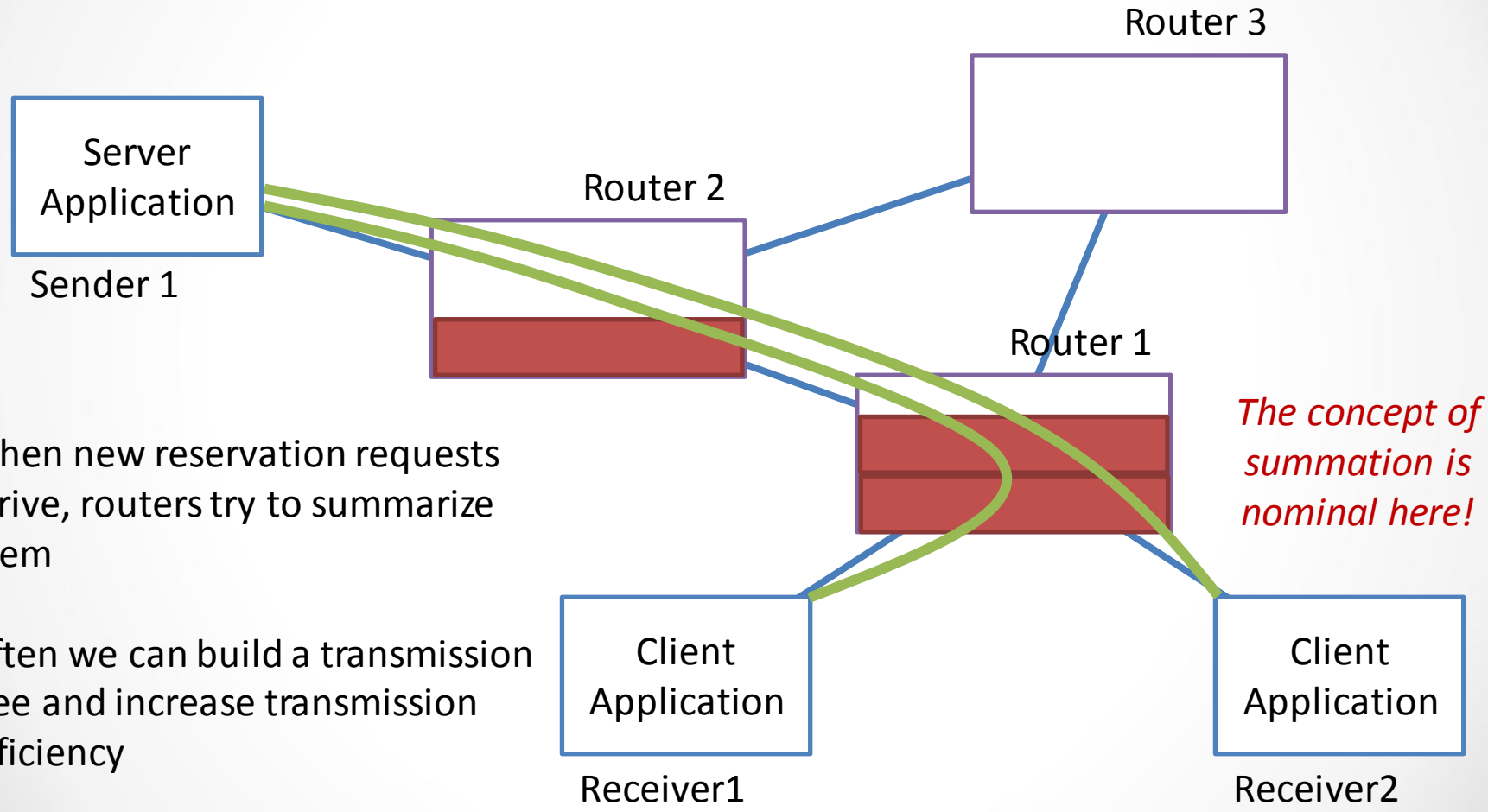
- Evaluates the reservation feasibility
- Sets the necessary settings for the scheduler and classifier
- Generates a new resv message and forwards it to the next router

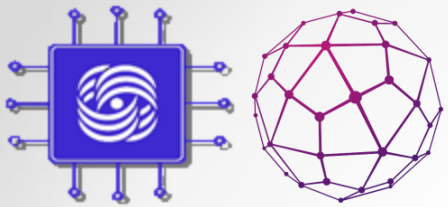
Transfer start



If you managed to allocate resources on each of the nodes, then the route with the required quality of service has been laid - you can start data transfer

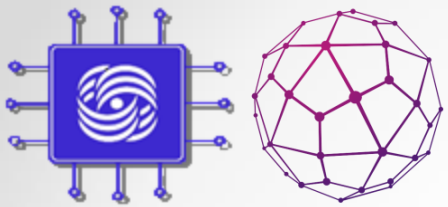
Filterspec usage





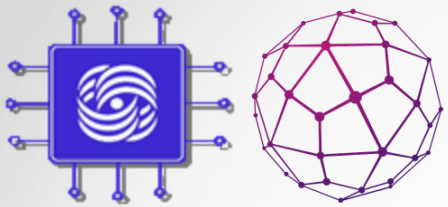
FilterSpec reservation styles

- Fixed Filter
  - Resources are allocated to the sender for individual use
  - Video streaming
- Wildcarded Filter
  - Resources are shared between a group of senders according to a given predicate
  - During an audio conference simultaneous data transfer is unlikely
- Shared-Explicit
  - Resources are shared between a group of senders
  - Group members may change over time



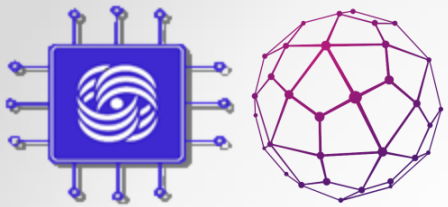
RSVP features

- RSVP - **signaling protocol**: provides only reservation along the route: primary route selection is a care of routing protocols
- The protocol economically consumes resources with partial route match
- Receiver initiates a reservation
- The same resources can be used by several senders at once
- Reservation is constantly updated by the sender and/or reciever (***Soft State***)



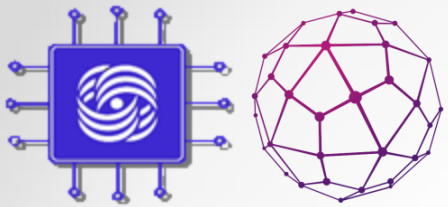
RSVP disadvantages

- It is necessary to conduct routing at the flow level - for each flow its state must be stored
- Individual allocation of resources is too demanding to the hardware
- Poor scalability of the solution
- ***Internal Resource Fragmentation*** with Static Reservation - Low Hardware Utilization



RSVP heritage

- RSVP's ability to generate stable routes formed the basis for **flow (circuit) switching** design
- IFMP (1996, Ipsilon Networks) – ***tag switching*** in ATM networks
- MPLS (1997, Cisco Systems) – ***label switching*** in IP networks
- Ethane project & OpenFlow protocol (2008)

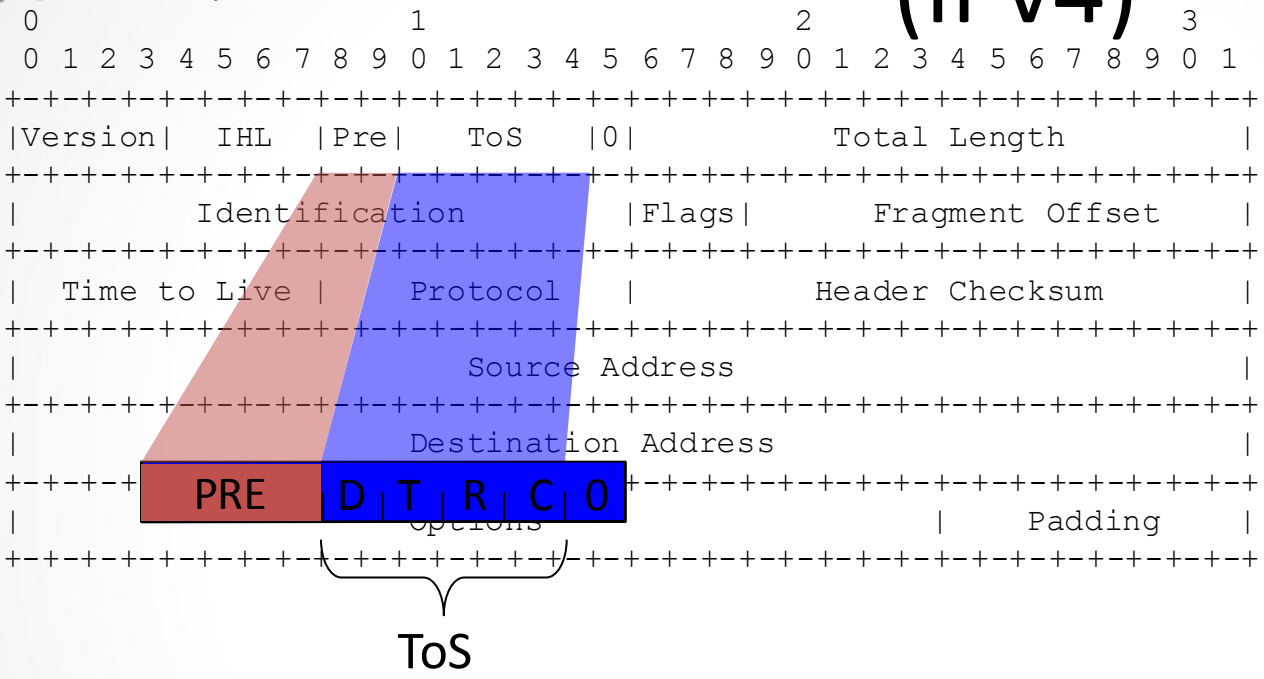


Differentiated Services (DiffServ)

- Each router has several predefined classes of service
- Border routers determine the class of the flow, label its packets with **DSCP labels** and conduct **traffic conditioning** – use the policing & shaping tools to set the desired traffic profile
- On internal router packets with a higher priority receive a larger share of resources, and vice versa

Internet Protocol version 4 (IPv4)

[RFC1349]



ToS

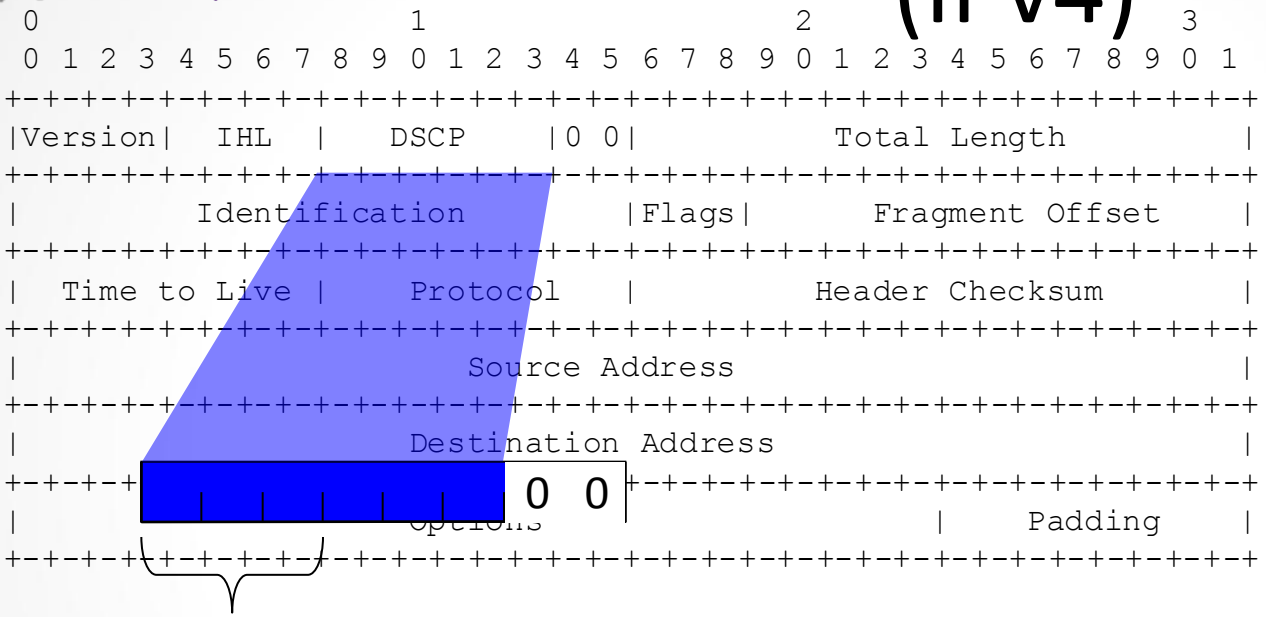
- Type of Service
  - D – minimize delay
  - T – maximize throughput
  - R – maximize reliability
  - C – minimize cost

PRE

- Precedence Field
  - Priority of the packet

Internet Protocol version 4 (IPv4)

[RFC2474]

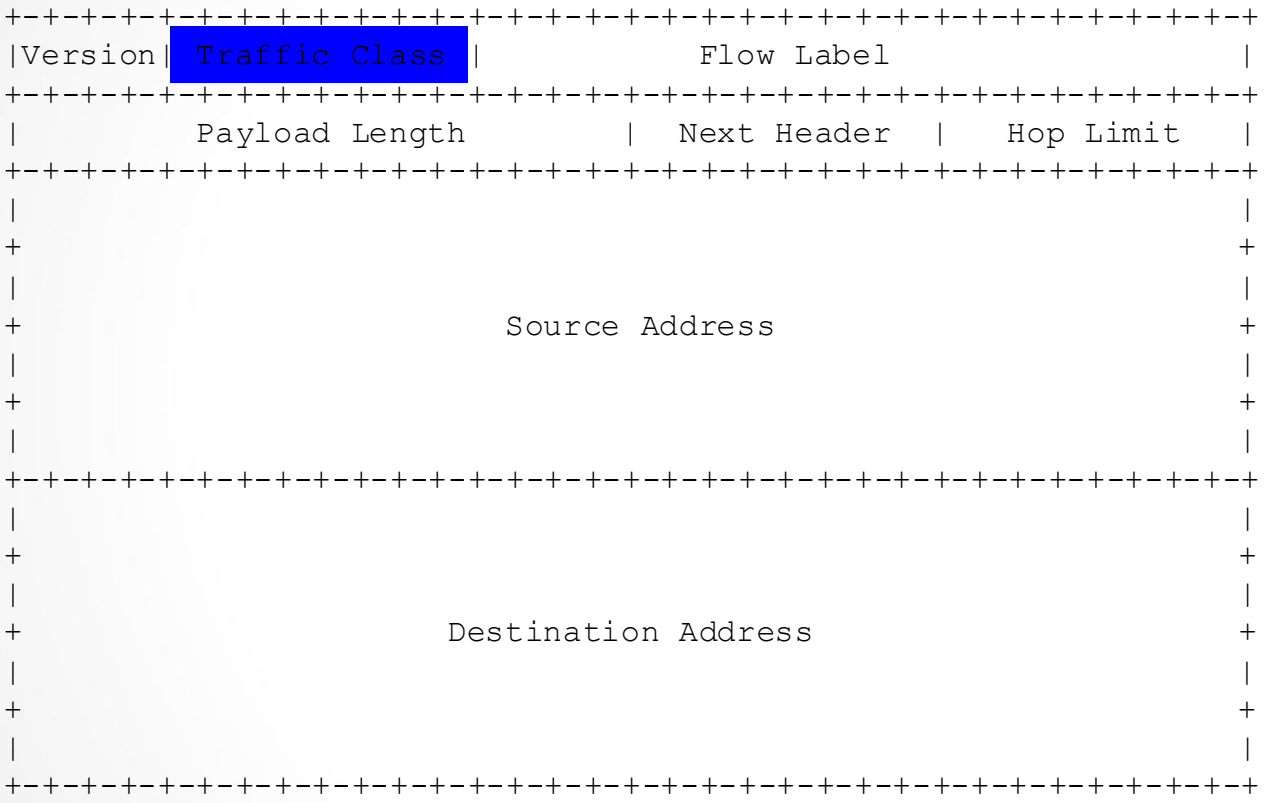


Class selector codepoints of the form xxx000

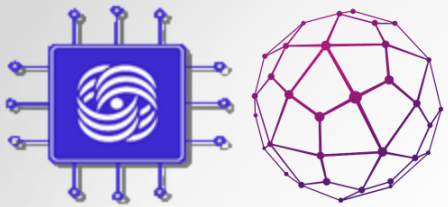
DSCP

- Differentiated Services Codepoint
  - xxxxx0 reserved for standardization
  - xxxx11 reserved for local use
  - xxxx01 open for local use, may be standardized later

Internet Protocol version 6 (IPv6)

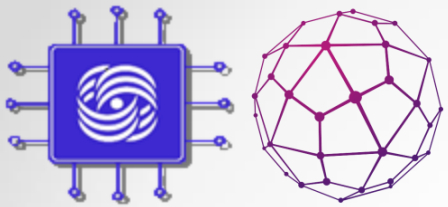


- Traffic class
  - Interpret like IPv4's DS field



DiffServ Standard Service Classes

- Default Forwarding (DF)
  - Usually serviced by best-effort.
- Expedient Forwarding (EF)
  - Passing through the highest priority queue
  - Small delay, jitter & loss
- Assured Forwarding (AF)
  - Passing through a lower priority queue
  - Covers multiple classes with different drop policies when the queue is full



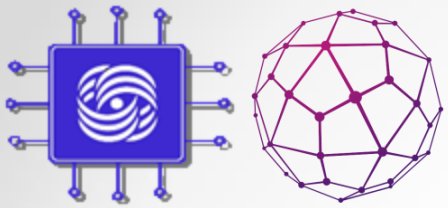
DiffServ model

Advantages

- Lack of internal fragmentation
- High degree of equipment utilization
- Easy to implement in hardware
- Good scalability

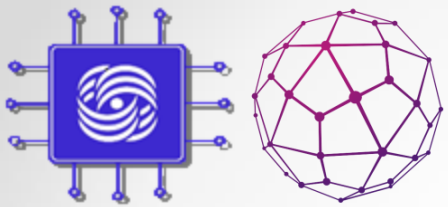
Disadvantages

- Does not provide quality guarantees
- Quality metrics are not calculated explicitly
- Limited number of quality classes



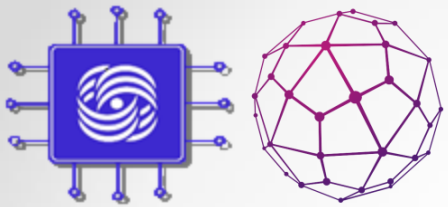
QoS Routing

- Generates individual routes for flows, taking into account their QoS requirements
  - Routes between two points in a network can be generated in different ways if flow quality requirements do not match
- There is an opportunity to involve resources located outside the main data transmission routes
- The method is compatible with both resource management models



QoS Routing Issues

- Granular control of switching devices is needed
  - Flow level management
- Centralized management is needed
  - Routing Algorithms Will Not Converge
- We should be able to get QoS requirements from applications
- Routing is a difficult task



The problem of the best resource allocation

You can manage QoS requirements compliance and the network utilization level using a special route generation algorithm

Optimization criteria:

- Increased resilience to peak loads - uniform distribution
- Transmitting with fewer devices - you can turn off some equipment
- Maximizing network reliability

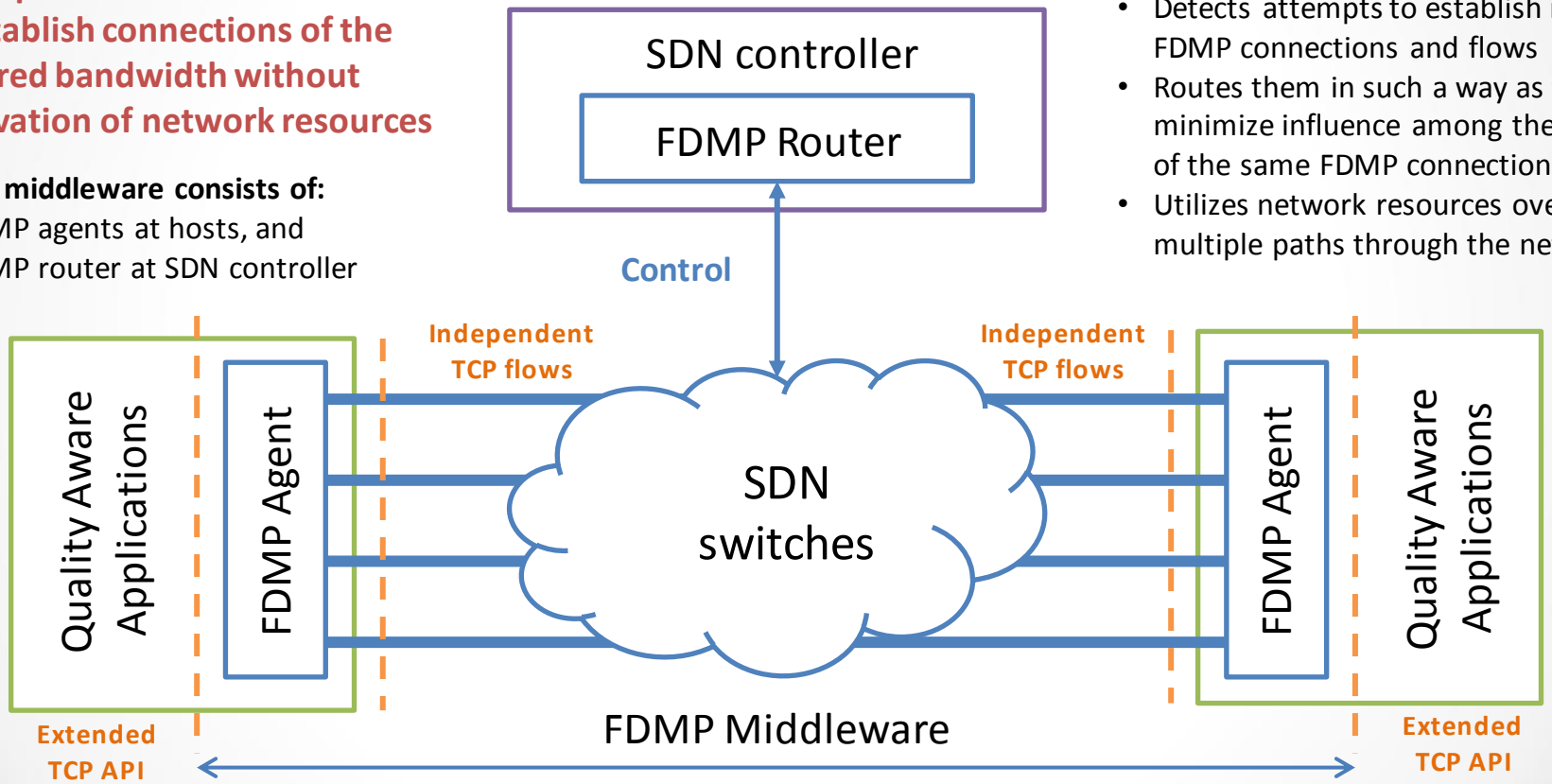
Flow DeMultiplexing Protocol

Managing Quality of Service with Flow (De) Multiplexing Protocol

FDMP provides a middleware to establish connections of the required bandwidth without reservation of network resources

FDMP middleware consists of:
- FDMP agents at hosts, and
- FDMP router at SDN controller

FDMP Router:
- Detects attempts to establish new FDMP connections and flows
- Routes them in such a way as to minimize influence among the flows of the same FDMP connection
- Utilizes network resources over multiple paths through the network



FDMP agent at the sender host:
Splits byte stream shipped by sender application into a number TCP flows and sends them independently

The number of TCP flows changes dynamically in accordance to current state of the network and immediate requirements of the application

FDMP agent at the receiver host:
Reassembles packets received over the TCP flows send by sender agent back into an original byte stream