Quality of Service (QoS) Provisioning

Roman Dunaytsev

The Bonch-Bruevich Saint-Petersburg
State University of Telecommunications

roman.dunaytsev@spbgut.ru

Lecture № 10

Outline

Outline

Introduction

- In recent years, the importance of **Quality of Service (QoS)** technologies for packet networks has increased rapidly
- In the beginning of telecommunications, there were in general 2 separate networks, one for voice and one for data
- Each network started with a simple goal of transporting a specific type of information (voice or data)

Introduction (cont'd)

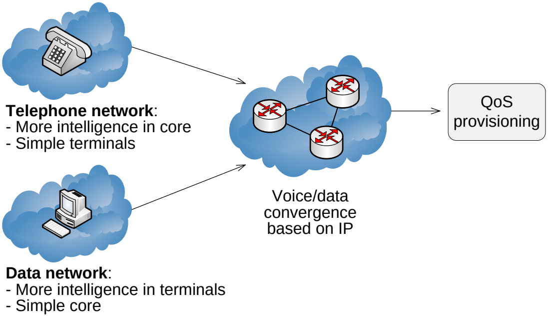
- In the early telephone network, there were 2 key measures of QoS:
  - **Probability of call blocking**
  - **Voice quality** (affected by circuit noise, echo, etc.)
- The original telephone network was designed with 2 main objectives:
  - To make sure that enough trunk circuits are provided to render call blocking probability reasonable (e.g., 0.1% or 2.0%)
  - To design the end-to-end path optimized for **voice** so that the network impairments such as noise, echo, and delay are reasonable

Introduction (cont'd)

- The early data network was a completely different type of network:
  - It was designed to carry **data**
  - Unlike voice, data delivery was (and still mostly is) a non-real-time service
- Since the information carried by the data network was different from that of the telephone network, the design philosophy was also different
- Because the early data network carried basically one type of information – non-real-time data – the network could be designed to operate in the '**best effort**' mode treating all packets equally
  - The network was designed to be as simple as possible
  - Most of intelligence was placed in the end systems

Introduction (cont'd)

- In the mid 1990's, the 2 separate networks started to merge
  - The idea is to create a single '**all-IP**' network to carry both voice and data
- With this convergence a new technical challenge has emerged
  - In the converged network, the 'best effort' operation of the earlier data network is no longer good enough to meet diverse performance requirements



Outline

Applications

- Internet traffic is produced by many different applications
- Depending on QoS requirements, applications can be classified as:
  - Elastic
  - Inelastic
- **Elastic applications** (aka **non-real-time**) – flexible in their bandwidth requirements and adapt their rates to the network conditions
  - No constraints for delivery as long as the packets reach their destination (loss-sensitive)
  - No specific demand on the delay bounds or bandwidth requirements
  - E.g., e-mail, P2P file sharing, Web browsing, FTP
  - Typically use TCP at the transport layer

Applications (cont'd)

- **Inelastic applications** (aka **real-time**) – less flexible in their bandwidth requirements and usually need a certain minimum bandwidth to work properly
  - E.g., VoIP, videoconferencing, multimedia streaming, network games
  - Typically use UDP at the transport layer
- Other ways to classify applications:
  - Delay or loss adaptive (aka **tolerant**) vs. nonadaptive (aka **intolerant**)
  - Mission-critical vs. non-mission-critical
  - Interactive vs. noninteractive

Applications (cont'd)

- **Real-time delay-tolerant applications**
  - Demand weak bounds on the maximum delay
  - Occasional packet losses are acceptable
  - E.g., Internet radio applications, which use buffering to hide delay variations from the user
- **Real-time delay-intolerant applications**
  - Demand tight bounds on the maximum delay and delay variation
  - E.g., VoIP applications, where excessive delay and jitter are hardly acceptable
- **Mission-critical applications** – reflect the importance of application usage, which determines the strictness of the QoS requirements; failing the mission may result in disastrous consequences:
  - Remote surgery: the surgeon performs an operation through remote surgical equipment
  - Telemedicine: the accuracy of medical images is extremely important

Applications (cont'd)

- **Interactive applications** – involve some form of interaction (action/reaction, request/response, or exchange of information) between 2 parties:
  - People-to-people: VoIP, videoconferencing
  - People-to-machine: multimedia streaming, network games
  - Machine-to-machine: automatic machine control
- The elapsed time between interactions is essential to the success of an interactive application

Applications (cont'd)

- Almost all of today's Internet traffic is generated by elastic applications (P2P file sharing, WWW, etc.) and most voice calls are still transported on dedicated infrastructure
- Nevertheless, inelastic applications gain in importance
  - E.g., Internet video (IPTV, videoconferencing, etc.)
  - As they have special QoS requirements and a high utility to the users, they need special attention of ISPs

Outline

QoS Parameters

- QoS – a measure of the ability of networks and computing systems to provide different levels of services to selected applications and associated data flows
- Since IP-based networks are expected to form the basis for all kinds of future communication services, while users expect at least the same quality for those services such as when delivered over dedicated networks, QoS support for IP networks is urgently required
- **Basic QoS parameters** (aka **QoS metrics**):
  - Bandwidth (aka throughput)
  - Transfer delay (aka latency)
  - Delay variation (aka delay jitter)
  - Loss or error rate
  - Etc. (fairness, availability, security, . . .)

QoS Parameters (cont'd)

- **Bandwidth**
  - Measured in bits per second
  - Considered to be the network resource that needs to be properly managed and allocated to applications
- The bandwidth required by an application depends on the application characteristics
- E.g., in a streaming video application, different video properties generate different data rate:
  - Frame size – a function of the number of pixels in each row and column and of the number of bits per pixel
  - Frame rate – the refreshing video frame rate (number of frames per second)
  - Color depth – the number of possible colors represented by a pixel (e.g., the 256-color video requires 8 bits of data per pixel, $2^8 = 256$)
  - Compression – reduction of the bandwidth consumption at the expense of image quality (e.g., MPEG1, MPEG2, and MPEG4)

QoS Parameters (cont'd)

- **2 types of applications**:
  - Constant Bit Rate (CBR)
  - Variable Bit Rate (VBR)
- **CBR applications** – generate data traffic with a constant data rate
  - E.g., digital telephony (which generates 64 kbit/s constant bit rate) and uncompressed digital video
- Most of the CBR applications are delay sensitive and require constant bandwidth allocation
  - Allocating bandwidth below the required bandwidth causes application failure
  - Allocating bandwidth above the requirement does not improve the user satisfaction

QoS Parameters (cont'd)

- **VBR applications** – generate data traffic with a variable data rate
  - E.g., compressed video and audio
  - The degree of bit rate variability depends on the application
- VBR applications require minimum bandwidth allocation in order to operate successfully
  - The more allocated bandwidth, the better the user-perceived quality
  - Bandwidth allocation beyond the maximum required bandwidth does not improve user satisfaction

QoS Parameters (cont'd)

- **Transfer delay**
  - Measured in seconds
  - Describes the average one-way delay that packets experience over a specific connection
  - Real-time applications require the delivery of information from the source to the destination within a certain period of time
- When data traffic is carried across a series of components in the communication system, each component introduces delay:
  - **Source-processing delay** – depends on the source host hardware configuration and its current load
  - **Transmission delay**
  - **Propagation delay**
  - **Protocol delay** – caused by communications protocols executed at different network components
  - **Queuing delay**
  - **Destination-processing delay** – depends on the destination host hardware configuration and its current load

QoS Parameters (cont'd)

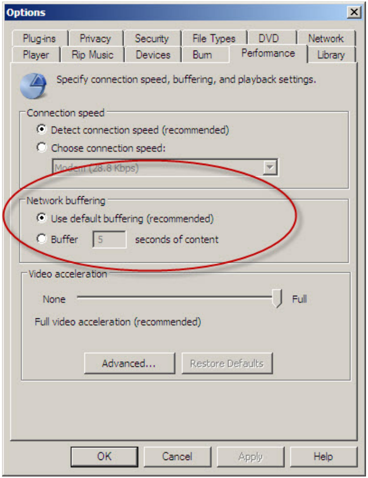
- **Delay variation**
  - Measured in seconds
  - Refers to the variation in the delay introduced by the components along the end-to-end path (RFC 3393)
  - Since the network conditions for each packet can be different, the delay varies
  - For data generated at a constant rate, the delay jitter distorts the time synchronization of the original traffic

QoS Parameters (cont'd)

- **Playing multimedia streams**:
- Playback as soon as packets arrive
  - The playback points are changed from the original timing reference
  - This introduces distortion in the playback signal
- Playback based on the original timing reference
  - The late packets that miss the playback point will be ignored
  - This also introduces distortion
- Use a de-jittered (aka play-out) **buffer**
  - All packets will be stored in the buffer and held for some time (aka **offset delay**) before they are retrieved by the receiver with the original timing reference
  - Large delay jitter requires a large buffer to hold the packets and smooth out the jitter
  - A large buffer also introduces large delays

QoS Parameters (cont'd)

- Network buffering in Windows Media Player
  - Buffering protects against the interruption of data flow



QoS Parameters (cont'd)

- **Packet loss rate** = Lost packets / Packets sent
- **Packet error rate** = Erroneous packets / Packets sent
- Packet losses and erroneous packets compromise the integrity of the data or disrupt the service
- At the network level, packet loss can be caused by network congestion, which results in dropped packets
- Another cause of loss is due to bit errors that occur as a result of a noisy communication channel
  - Such loss will most likely occur in a wireless channel
- **Techniques to cope with packet losses or errors**:
  - Retransmission (ARQ)
  - Forward Error Correction (FEC)
  - Codecs at the application layer that can compensate or conceal losses

QoS Parameters (cont'd)

- **Bandwidth** – mostly expressed as the average data rate
  - Uncompressed HDTV: 1.5. Gbit/s
  - ITU-T G.711: 64 kbit/s
  - For applications that generate VBR traffic, the average data rate does not properly capture the traffic characteristics
- **Transfer delay** and **delay jitter** – should be less than a certain value
  - In mission-critical or delay-intolerant applications, the network has to follow the delay requirement strictly
  - In delay-tolerant applications, the bound value is the average value (i.e., some packets can miss their deadline)
- **Loss rate** – should be less than a certain value
  - Not only the packet loss has impact on the voice quality, but also the pattern of the packet loss has impact
  - E.g., the loss of 1 packet may not be noticeable due to the sophisticated codecs that can conceal the loss, while 2 or more consecutive lost packets can cause voice quality deterioration

QoS Parameters (cont'd)

- Delay guidelines for VoIP

One-way delay	**Effect on perceived QoS**
< 100-150 ms	Excellent quality (undetectable delays)
150-250 ms	Acceptable quality (slight delays)
> 250-300 ms	Unacceptable quality

- Delay jitter guidelines for VoIP

Delay jitter	**Effect on perceived QoS**
< 40 ms	Excellent quality (undetectable jitter)
40-75 ms	Acceptable quality
> 75 ms	Unacceptable quality

Jitter

- '**Jitter**' – the variation in timing of some event against a clock
- In RFC 3393, the IETF defines packet jitter as the Instantaneous Packet Delay Variation and deprecates the use of the term 'jitter'
- **Instantaneous Packet Delay Variation (IPDV)** – the difference in one-way delay between successive packets, ignoring any lost packets
- **One-way delay** – the delay from the start of the packet being transmitted at the source address to the end of the packet being received at the destination
  - A sequence of N packets
  - Transmitted at times $t(1), \dots, t(N)$
  - Received at times $t'(1), \dots, t'(N)$
  - The sequence of delays is $d(1), \dots, d(N)$
  - Where $d(i) = t'(i) - t(i)$, $d(i) \geq 0$
  - IPDV is the sequence $d(2) - d(1), d(3) - d(2), \dots, d(N) - d(N-1)$
  - Maximum IPDV is $\max\{|d(2) - d(1)|, |d(3) - d(2)|, \dots, |d(N) - d(N-1)|\}$

Jitter (cont'd)

- **Packet Delay Variation (PDV)**
  - Defined by the IETF as the difference in one-way delay between selected packets, ignoring any lost packets
- The IETF does not define what the selection criteria is for the PDV
- However, it could be randomly selected packets in a sliding window or the packets which give the maximum and minimum delay in a sequence
- E.g., PDV $= \max\{d(1), d(2), \ldots, d(N)\} - \min\{d(1), d(2), \ldots, d(N)\}$, PDV ≥ 0

Fairness

- Flows with different QoS requirements should receive appropriate part of network resources and should not be blocked by other flows pretending for better QoS
- Providing fairness is extremely challenging
  - If the total load is low, all flows normally receive appropriate amount of network resources
  - If the total load is high, the probability that packets from a single flow will occupy certain resources rises, while packets from other flows may get blocked
- **Fairness** – a characteristic exhibited by networks that allow connections to evenly share available bandwidth

Fairness (cont'd)

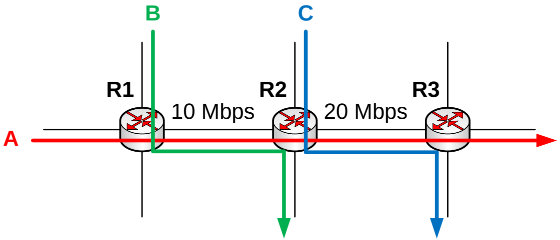
- The simplest way to express fairness is to divide up the resources equally among all users
- Fairness can quantified using **Jain's fairness index**
- If the system allocates rates to N contending users, such that the i-th user receives a rate allocation x_i, the fairness index $f(x)$ is defined as:

$$f(x) = \frac{(\sum_{i=1}^{N} x_i)^2}{N * \sum_{i=1}^{N}(x_i)^2}$$

- This is a good measure for fairness because $f(x) = 1$ if all allocations x_i are perfectly equal and immediately becomes less than 1 upon the slightest deviation:
    - $x_1 = 0.5$ and $x_2 = 0.5$, then $f(x) = 1$
    - $x_1 = 0.6$ and $x_2 = 0.4$, then $f(x) = 0.96$

Fairness (cont'd)

- However, simply dividing single resources is not good enough, as it can lead to a rate allocation where several users could increase the rate even further without degrading the throughput of others
- Example:
  - Link R1-R2: flow A gets 5 Mbit/s and flow B gets 5 Mbit/s
  - Link R2-R3: flow A gets 10 Mbit/s and flow C gets 10 Mbit/s
  - But flow A is already limited by 5 Mbit/s at link R1-R2
  - Thus, it would be better to divide the capacity of link R2-R3 between flows A and C as 5 Mbit/s and 15 Mbit/s



Fairness (cont'd)

- **Max-min fairness**
  - At the beginning, all rates are 0
  - All rates grow at the same pace, until one or several link capacity limits are hit
  - All rates except for the rates of flows that have already hit a limit are increased further
  - This procedure continues until it is not possible to increase
- A feasible allocation of rates is 'max-min fair' if and only if an increase of any rate within the domain of feasible allocations must be at the cost of a decrease of some already smaller or equal rate
  - A 'feasible' rate allocation is a rate allocation that does not exceed the total network capacities and is greater than 0
  - A max-min fair rate allocation is unique

Outline

Standardization Bodies

- **Telecommunication Standardization Sector (ITU-T)** – coordinates standards for telecommunications on behalf of the International Telecommunication Union (ITU)
- **Recommendation Y.1540** 'Internet protocol data communication service – IP packet transfer and availability performance parameters'
  - Defines parameters that may be used in specifying and assessing the performance of speed, accuracy, and availability of IP data communication service
  - The defined parameters apply to end-to-end IP service and to the network portions that provide or contribute to the provision of such service
- **Recommendation Y.1541** 'Network performance objectives for IP-based services'
  - Defines classes of network QoS with objectives for IP network performance parameters
  - These classes are intended to be the basis for agreements among network providers, and between end users and their network providers

Standardization Bodies (cont'd)

- ITU-T guidance for IP QoS classes
  - VTC = videoteleconferencing

Class	Applications (examples)	Node mechanisms	Network techniques
0	Real-time, jitter sensitive, high interaction (VoIP,VTC)	Separate queue with preferential servicing, traffic grooming	Constrained routing and distance
1	Real-time, jitter sensitive, interactive (VoIP, VTC)		Less constrained routing and distance
2	Transaction data, highly interactive (signaling)	Separate queue, drop priority	Constrained routing and distance
3	Transaction data, highly interactive		Less constrained routing and distance
4	Low loss only (short transactions, bulk data, video streaming)	Long queue, drop priority	Any route/path
5	Traditional applications of default IP networks	Separate queue (lowest priority)	Any route/path

Standardization Bodies (cont'd)

- **3rd Generation Partnership Project (3GPP)** – a collaboration project between groups of telecommunications associations
  - 3GPP specifications are based on evolved Global System for Mobile Communications (GSM) specifications
  - 3GPP standardization encompasses Radio, Core Network and Service architecture
- **Technical Specification 3GPP TS 23.107 V9.0.0** 'Quality of Service (QoS) concept and architecture'
  - Specifies the list of attributes applicable to the UMTS Bearer Service and the Radio Access Bearer Service, as well as describe the QoS architecture to be used in the 3GPP system (GSM, UMTS, LTE)

Standardization Bodies (cont'd)

- UMTS QoS classes:
  - Conversational (e.g., telephony speech, VoIP, videoconferencing)
  - Streaming (e.g., streaming multimedia)
  - Interactive (e.g., Web browsing, data base access)
  - Background (e.g., e-mail, SMS)
- Conversational class is meant for traffic which is very delay-sensitive
- Background class is the most delay-insensitive traffic class

Standardization Bodies (cont'd)

- 3GPP guidance for QoS classes

Traffic class	Fundamental characteristics	Example of the application
Conversational	- Preserve time relation (variation) between information entities of the stream - Conversational pattern (stringent and low delay)	Voice
Streaming	- Preserve time relation (variation) between information entities of the stream	Streaming video
Interactive	- Request/response pattern - Preserve payload content	Web browsing
Background	- Destination is not expecting the data within a certain time - Preserve payload content	Background download of emails

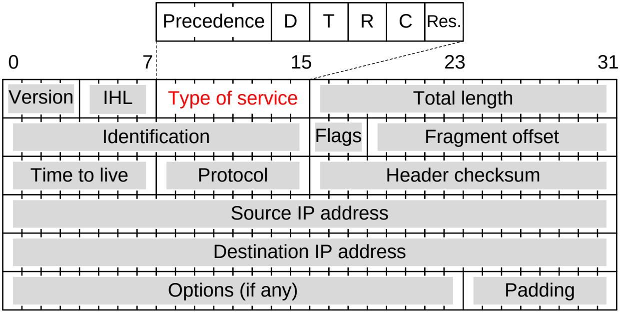
Standardization Bodies (cont'd)

- **Internet Engineering Task Force (IETF)** – has been working to define Internet QoS models for many years
- The task has not been easy since packets must cross many networks, and providers must agree not only how QoS will be managed, but also how it is paid for
- QoS architectures developed by the IETF:
  - ToS (Type of Service)
  - IntServ (Integrated Services)
  - RSVP (Resource Reservation Protocol)
  - DiffServ (Differentiated Services)
  - MPLS (Multiprotocol Label Switching) and Forwarding Equivalence Class (FEC)

Outline

QoS Architectures

- **Type of Service (ToS)**
  - Defined in RFC 791, RFC 1122, RFC 1349
- ToS in the IP header specifies abstract parameters of the desired QoS
  - 3 bits for priority levels (called 'Precedence')
  - 4 bits for specific requirements (Delay, Throughput, Reliability, Cost)



QoS Architectures (cont'd)

- The default TOS field value is all '0's
- The original intention was for a sending host to specify a preference for how the packet would be handled as it made its way through an internetwork
  - E.g., one could set the TOS value to prefer low delay, while another might prefer high reliability and normal delay
- In practice, the TOS field has not been widely implemented
- Recently, the TOS field has been redefined as the Differentiated Services Code Point (DSCP) and the Explicit Congestion Notification (ECN)

QoS Architectures (cont'd)

- **Integrated Services (IntServ)**
  - Defined in RFC 1633
- It was designed to provide per-flow QoS guarantees to individual application sessions
- With IntServ, when an application needs a specific QoS level, it uses the RSVP signaling protocol to reserve the required network resources at each node on the entire path
  - Such a framework implies the storage of state information for every flow across their routes.
  - Since Internet core routers forward 1000s of flows, maintaining and managing their associated state information would consume an excessive amount of resources

QoS Architectures (cont'd)

- **Resource Reservation Protocol (RSVP)**
  - Defined in RFC 2205
- It is a network layer signaling protocol for allowing an application to dynamically reserve network bandwidth
- Hosts and routers use RSVP to deliver QoS requests to the routers along the paths of the data stream and to maintain router and host state to provide the requested service, usually bandwidth and latency
- RSVP is used by:
  - **Transmitting applications** – to describe their data traffic characteristics
  - **Receiving applications** – to describe their QoS requirements
  - **Routers** – to deliver QoS requests to other routers along the path of a flow

QoS Architectures (cont'd)

- **Differentiated Services (DiffServ)**
  - Defined in RFC 2474, RFC 2475, etc.
- In contrast to the fine-grained IntServ model, DiffServ provides different levels of network service without maintaining per-flow state and signaling at every hop
- The mechanisms proposed for DiffServ are derived from a model that considers aggregate traffic streams instead of individual flows
- This architecture applies forwarding behaviors to aggregated traffic which has been appropriately marked at network boundaries and, therefore, per-flow state need not be maintained at core nodes
- DiffServ Code Points (DSCPs) are similar to labels in the Frame Relay, ATM, or MPLS technologies
  - DSCP is 6 bits wide, allowing coding for up to 64 different forwarding behaviors

Outline

Queue Management

- The way that routers handle an overflow of arriving traffic is to use a queue management mechanism to sort the traffic, and then to use some method of prioritizing it onto an output link
- **2 elements of queue management**:
  - **Queuing algorithms** – which packets to send
  - **Drop algorithms** – which packets to drop in case of congestion
- **Queuing algorithms** (aka **scheduling algorithms**):
  - First-In, First-Out (FIFO) queuing
  - Priority Queuing (PQ)
  - Weighted Fair Queuing (WFQ)
  - etc.
- Each queuing algorithm was designed to solve a specific network traffic problem and has a particular effect on network performance

Queue Management (cont'd)

- **FIFO queuing**
  - There is only 1 queue
  - Every packet exits the queue in the order it arrived
  - As a result, FIFO uses no priorities
  - It is the simple mechanism, and it is useful for high capacity lines where there is no congestion
  - But it performs badly when there is congestion, or when bursty applications dominate the queue and other applications' packets are rejected
  - FIFO also has the potential problem of delaying short frames behind longer frames

Queue Management (cont'd)

- **Priority Queuing (PQ)**
  - PQ allows different priorities and can handle multiple queues
  - One queue has strict priority and is always preferentially served
  - Packets are inserted in the proper queue depending on their classification
  - The priority queuing mechanism checks the queues sequentially by starting from the highest priority queue, until a non-empty queue is found
  - The first packet from that queue is then transmitted, and the procedure starts over
  - Depending on the incoming rate for high priority packets, other queues might be served very slowly or not at all
  - The latter might occur if high priority traffic arrives at a rate close to or higher than the link capacity
  - PQ can greatly decrease latency for high-priority packets, but it still has FIFO-style problems of delaying short packets behind long ones

Queue Management (cont'd)

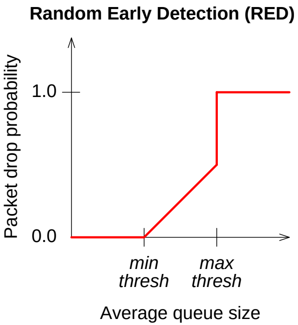
- **Weighted Fair Queuing (WFQ)**
  - WFQ assigns weights to each queue
  - These weights effectively control the percentage of the links bandwidth each flow will get
  - In case some queues are empty, the excess bandwidth that would be used by these queues is shared among the rest of the queues according to their weights
  - WFQ is the main Cisco's queuing algorithm
- **Drop algorithms**:
  - Drop Tail (aka Tail Drop)
  - Random Early Detection (RED)
  - Weighted Random Early Detection (WRED)
  - etc.

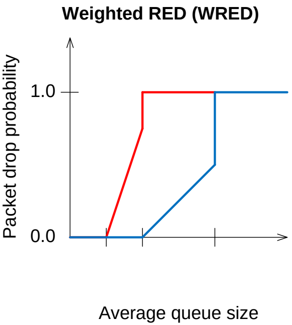
Queue Management (cont'd)

- **Drop Tail**
  - When the buffer overflows, the packet at the tail of the queue is dropped
- **Random Early Detection (RED)**
  - The simple Drop Tail strategy causes global synchronization of TCP flows
  - I.e., many TCP sources will detect packet loss and slow down at the same time
  - RED eliminates the synchronization phenomenon and thereby achieves better utilization and smaller queues
  - Instead of waiting for the buffer to overflow, RED drops a packet randomly from the queue so that multiple TCP flows will lose packets at different times
  - Therefore, the TCP flows slow down and ramp up at different times
  - Since the flows are out of phase, their aggregate rate is smoother than when the flows are synchronized

Queue Management (cont'd)

- **Weighted Random Early Detection (WRED)**
  - WRED extends RED to provide service differentiation
  - Packets are dropped selectively based on their priorities
  - Packets with a higher priority are less likely to be dropped than packets with a lower priority





Outline

Traffic Shaping

- The idea behind traffic shaping is to smooth out a traffic flow and reduce traffic clumping
  - Almost all traffic is bursty to some degree
  - Shaping is used to control the impact of bursts on the network
- Traffic shaping attempts to adjust the transmission rate of packets that match a certain criteria
  - By holding packets in a buffer and releasing them at a preconfigured rate
- Typically, traffic shaping is used either at the traffic source (as near to it as possible) or in network devices on the periphery of the network
- **Traffic shaping algorithms**:
  - Leaky bucket
  - Token bucket

Traffic Shaping (cont'd)

- **Leaky bucket**
  - If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket
  - The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty
  - The input rate can vary, but the output rate remains constant
  - Similarly, in networking, a technique called 'leaky bucket' can smooth out bursty traffic
  - Data bursts are stored in the buffer and sent out at an average rate

Traffic Shaping (cont'd)

- **Token bucket**
  - The leaky bucket algorithm is very restrictive
  - I.e., it does not credit an idle host
  - E.g., if a host is not sending for a while, its bucket becomes empty
  - Now if the host has bursty data, the leaky bucket allows only an average rate; the time when the host was idle is not taken into account
  - The token bucket algorithm allows idle hosts to accumulate credit for the future in the form of tokens
  - For each tick of the clock, the system sends N tokens to the bucket
  - The system removes 1 token for every packet (or byte) of data sent
  - E.g., if $N = 100$ and the host is idle for 100 ticks, the bucket collects 10,000 tokens
  - Now the host can consume all these tokens in 1 tick with 10,000 packets, or the host takes 1000 ticks with 10 packets per tick, etc.
  - Thus, the host can send bursty data as long as the bucket is not empty