UNIT-IV

User Datagram Protocol (UDP)

The User Datagram Protocol (UDP)is a datagram transport which uses the underlying IP protocol for its network layer. It is used when there is a need to transmit short packets through a network where there is no stream of data to be sent as in TCP. It is consequently a much simpler protocol and therefore much easier to handle. It is also less reliable in that there are no sequence numbers and other error recovery techniques available. If errors and lost packets are important then the user of UDP must cater for these.

The format of a UDP header is shown in figure

source port

This field performs the same function as in TCP.

destination port

This field is also the same as in TCP. Note that the same port number can be used by TCP and UDP for different purposes.

length

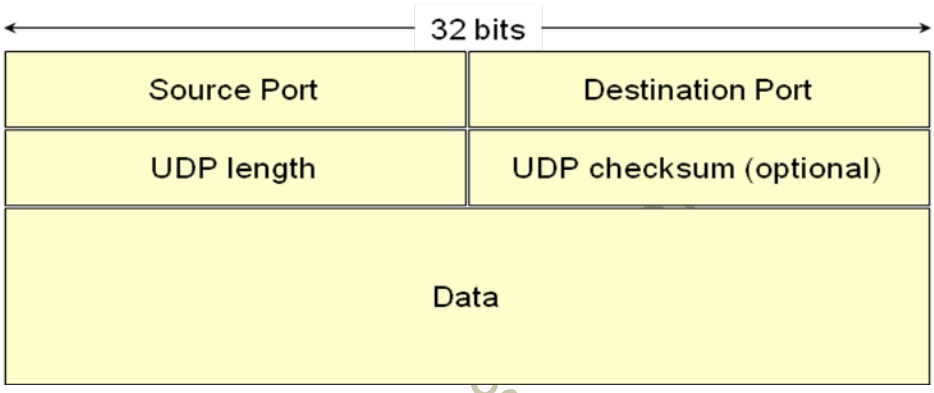
This field contains the length of the data field.

checksum

As in TCP, this field is the checksum of the header plus parts of the IP header.

Data This field is passed to the relevant program

UDP Frame Format



- No connection needs to be set up
- Throughput may be higher because UDP packets are easier to process, especially at the source
- The user doesn't care if the data is transmitted reliably
- The user wants to implement his or her own transport protocol

TCP

The Transmission Control Protocol (TCP) is one of the main transport layer protocols used with IP. It is a connection oriented protocol based on the connectionless IP protocol. Because it is the lowest layer which has end-to-end communication, it needs to handle things such as lost packets. In this respect it is similar to the data-link layer which must handle errors on an individual link.

Consequently many of its facilities are familiar from our discussion of the data-link layer.

The format of a TCP header is shown in figure 15.1.

Source port

All of the address fields in the lower layer protocols are only concerned with getting the packet to the correct host. Often, however, we want to have multiple connections between two hosts. The source port is simply the number of the outgoing connection from the source host.

Destination port

Similarly, this is the number of the incoming connection on the destination host. There must be a program on the destination host which has somehowtold the networking system on that host that it will accept packets destined for this port number. Standard system services such as SMTP, NNTP and NTP (all described in later chapters) have well known standard port numbers. Thus to connect to the SMTP port on a particular host (in order to transmit some email) a TCP connectionwould be set up with the correct destination port number (25). The
source port number does not matter except that it should be unique on the sending machine so that replies cvan be received correctly.

Sequence number

This is the sequence number of this packet. It differs from the usual data-link layer sequence number in that it is in fact the sequence number of the first byte of information and is incremented by the number of bytes in this packet for the next message. In other words, it counts the number of bytes transmitted rather than the number of packets.

Acknowledgement number

This is the sequence number of the last byte being acknowledged. This is a piggy-backed acknowledgement.

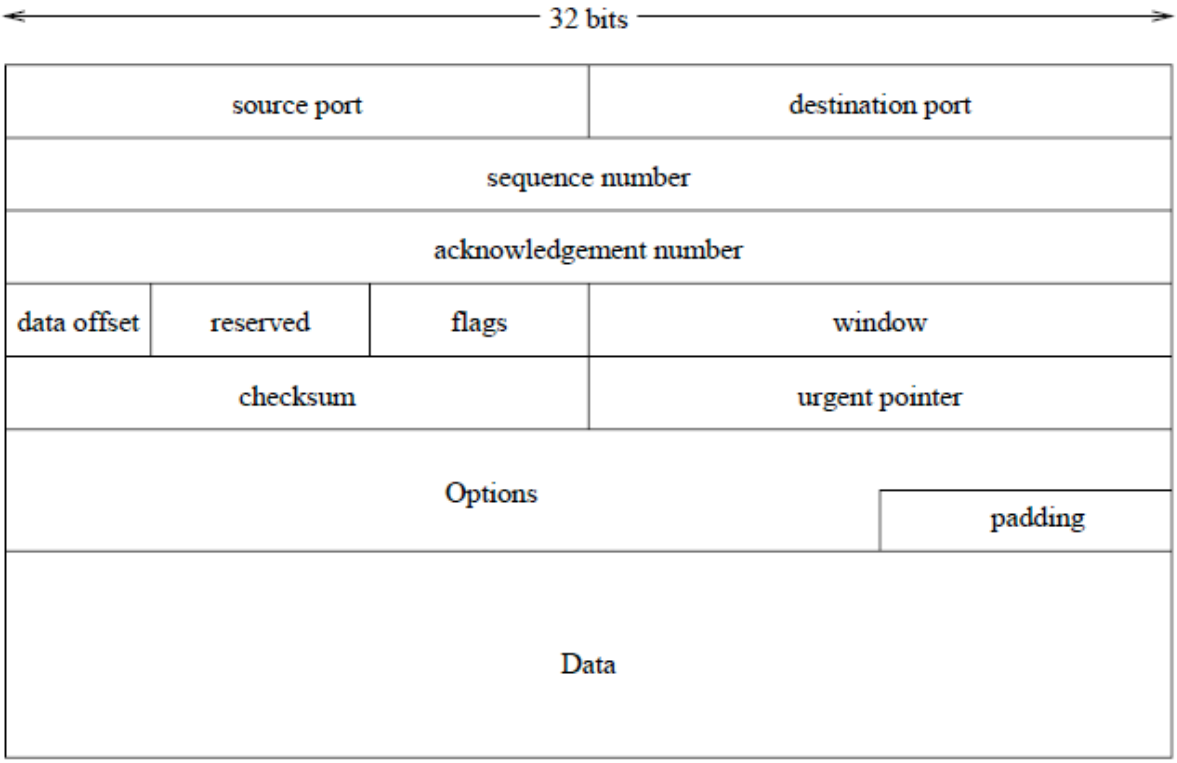


Figure 15.1. TCP header format

data offset

This field is the offset in the packet of the beginning of the data field. (In other words it is the length of the header.)

flags

This field contains several flags relating to the transfer of the packet. We will not consider them further here.

window

This field is used in conjunction with the acknowledgement number field. TCPuses a sliding window protocol with a variable window size (often depending on the amount of buffer space available. This field contains the number of bytes which the host is willing to accept from the remote host.

checksum

This field contains a checksum of the header. It actually uses a modified form of the header which includes some of the information from the IP header to detect some unusual types of errors.

urgent pointer

There is provision in TCP for some urgent data messages to be sent bypassing the normal

sequence number system. This field is used to indicate where such data is stored in the packet.

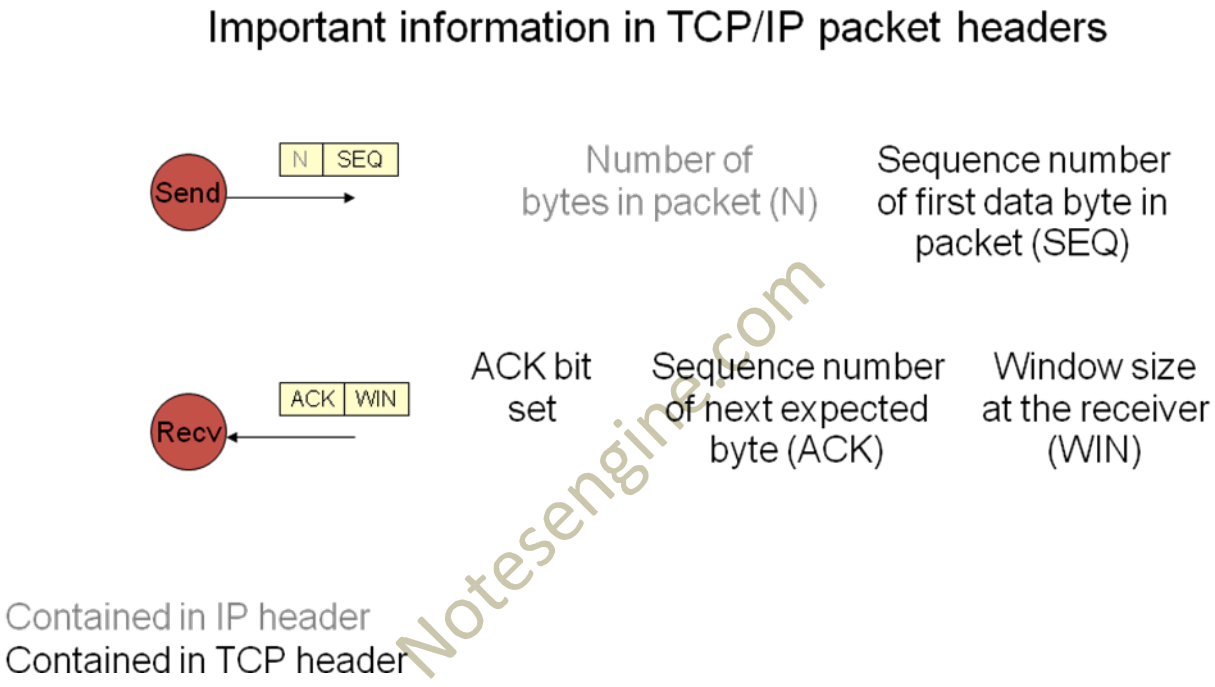
Adaptive Flow Control

F**low control** is the process of managing the rate of data transmission between two nodes to prevent a fast sender from outrunning a slow receiver. It provides a mechanism for the receiver to control the transmission speed, so that the receiving node is not overwhelmed with data from transmitting node. Flow control should be distinguished from congestion control, which is used for controlling the flow of data when congestion has actually occurred . Flow control mechanisms can be classified by whether or not the receiving node sends feedback to the sending node.

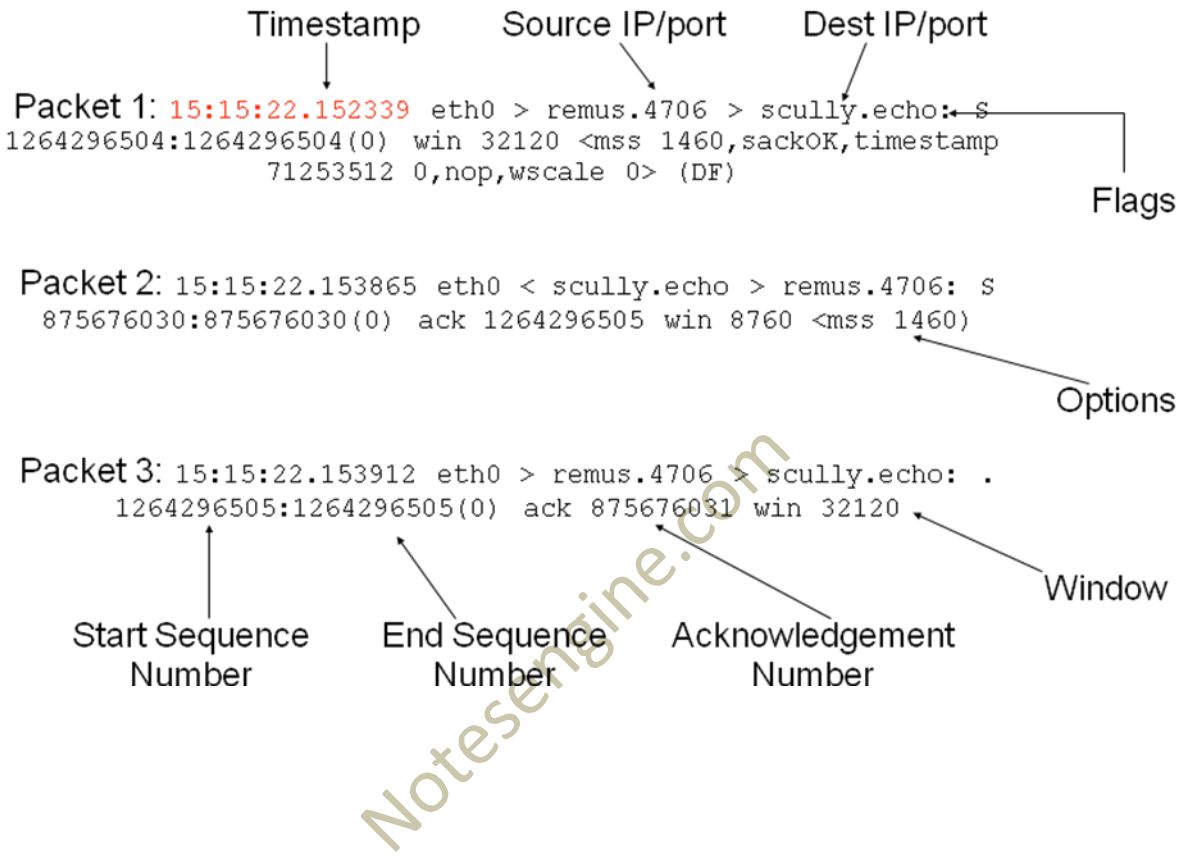
Flow control is important because it is possible for a sending computer to transmit information at a faster rate than the destination computer can receive and process them. This can happen if the receiving computers have a heavy traffic load in

comparison to the sending computer, or if the receiving computer has less processing power than the sending computer.

TCP Flow Control *(cont'd)*



Example TCP session



Types of flow control

Network congestion

A prevention mechanism that provides control over the quantity of data transmission that enters a device.

Windowing Flow control

mechanism used with TCP.

data buffer

A prevention control mechanism that provides storage to contain data- bursts from other network devices, compensating for the variation of data transmission speeds.

Transmit flow control

Transmit flow control may occur

- between data "on and off" terminal equipment (DTE) and a switching center, via data circuit-terminating equipment (DCE), the opposite types interconnected straightforwardly,
- or between two devices of the same type (two DTEs, or two DCEs), interconnected by a crossover cable.

The transmission rate may be controlled because of network or DTE requirements. Transmit flow control can occur independently in the two directions of data transfer, thus permitting the transfer rates in one direction to be different from the transfer rates in the other direction. Transmit flow control can be

- either stop-and-go,
- or use a sliding window.

Flow control can be done

- either by control signal lines in a data communication interface (see serial port and RS 232),
- or by reserving in-band control characters to signal flow start and stop (such as the ASCII codes for XON/XOFF).

Hardware flow control

In common RS 232 there are pairs of control lines:

- **RTS flow control**, RTS (Request To Send)/CTS (Clear To Send) and
- **DTR flow control**, DTR (Data Terminal Ready)/DSR (Data Set Ready),

which are usually referred to as **hardware flow control**.

Hardware flow control is typically handled by the DTE or "master end", as it is first raising or asserting its line to command the other side:

- In case of **RTS control flow**, DTE sets its RTS, which signals the opposite end (the slave end such as a DCE) to begin monitoring its data input line. When ready for data, the slave end will raise its complementary line, CTS in this example, which signals the master to start sending data, and for the master to begin monitoring the slave's data output line. If either end needs to stop the data, it lowers its respective "data readyness" line.
- For PC-to-modem and similar links, the case of **DTR flow control**, DTR/DSR are raised for the entire modem session (say a dialup internet call), and RTS/CTS are raised for each block of data.

Software flow control

Oppositely, XON/XOFF is usually referred to as **software flow control**. In the old mainframe days, modems were called "data sets" hence the survival of the term.

Open-loop flow control

The open-loop flow control mechanism is characterized by having no feedback between the receiver and the transmitter. This simple means of control is widely used. The allocation of resources must be a “prior reservation” or “hop-to-hop” type. The Open Loop flow control has inherent problems with maximizing the utilization of network resources. Resource allocation is made at connection setup using a CAC (Connection Admission Control) and this allocation is made using information that is already “old news” during the lifetime of the connection. Often there is an over-allocation of resources. Open-Loop flow control is used by ATM in its CBR, VBR and UBR services (see traffic contract and congestion control)

Closed-loop flow control

The Closed Loop flow control mechanism is characterized by the ability of the network to report pending network congestion back to the transmitter. This information is then used by the transmitter in various ways to adapt its activity to

existing network conditions. Closed Loop flow control is used by ABR (see traffic contract and congestion control) Transmit Flow Control described above is a form of Closed-loop flow control.

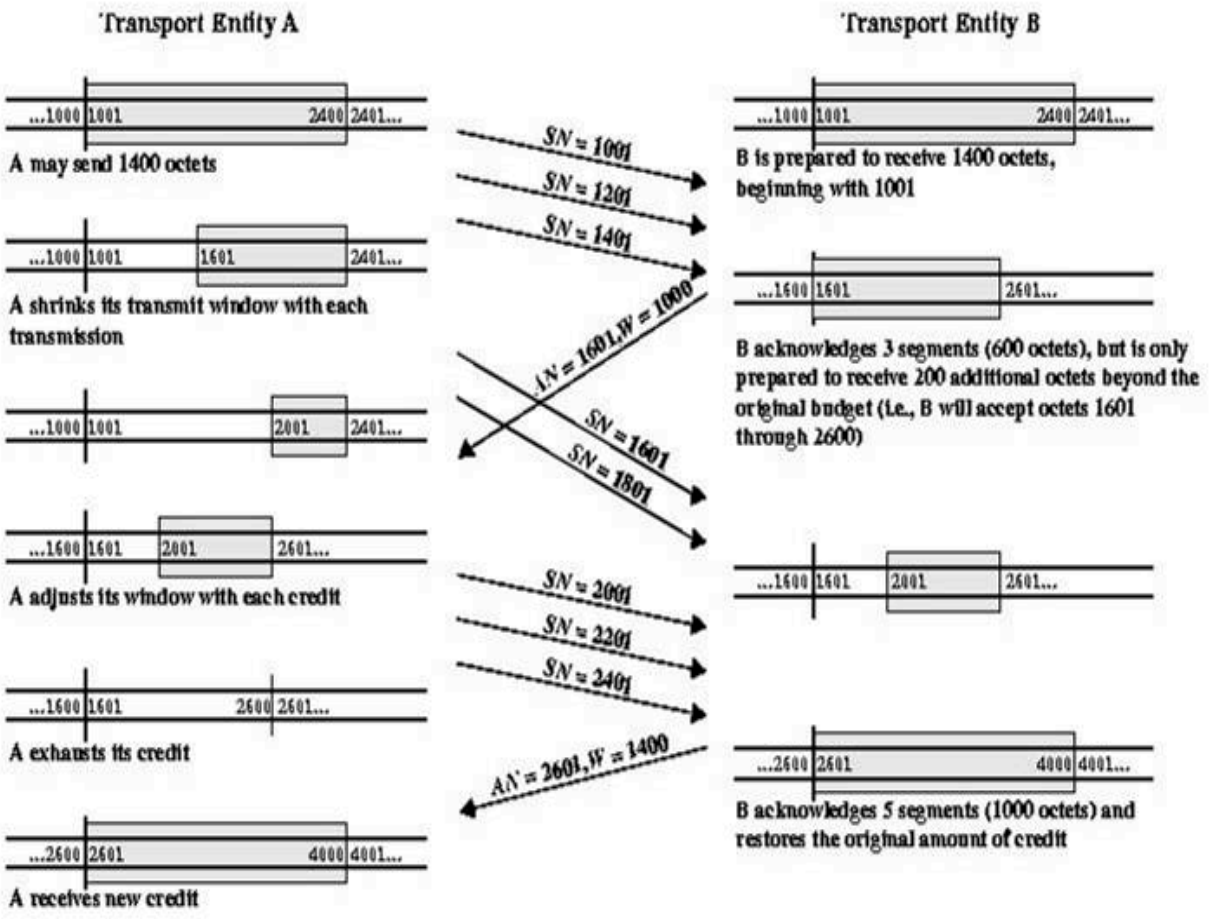


Figure Example of TCP Credit Allocation Mechanism

Example TCP session

```
(1)remus:$ tcpdump -S  host  scully
     Kernel filter, protocol ALL, datagram packet socket
     tcpdump: listening on all devices

15:15:22.152339 eth0 >  remus.4706 >  scully.echo: S 1264296504:1264296504(0) win 32120 <mss
     1460,sack OK,timestamp 71253512 0,nop,wscale 0>
     15:15:22.153865 eth0 <  scully.echo >  remus.4706: S 875676030:875676030(0) ack 1264296505 win
     8760 <mss 1460>15:15:22.153912 eth0 >  remus.4706 >  scully.echo: . 1264296505:1264296505(0) ack
     875676031 win 32120
```

remus: telnet scully 7

A <return>

A

Adaptive Retransmission Timer

As network or internet conditions change, a static retransmission timer is likely to be either too long or long short. All TCP implementations attempt to take average of observed round-trip times over number of segments. If average accurately predicts future delays, resulting retransmission timer will yield good performance.

Average Round-Trip Time (ARTT)

$$ARTT(K+1) = \frac{1}{K+1} \sum_{i=1}^{K+1} RTT(i)$$

$$= \frac{K}{K+1} ART(K) + \frac{1}{K+1} RTT(K+1)$$

The second formula can be used to avoid recalculating sum every time. RTT(i) is the round-trip time observed for the ith transmitted segment and ARTT (K) is the average round-trip time for the first K segments. Each term in the summation is given equal weightage, but we may like to give greater weight to more recent instances because they are more likely to reflect future behavior. A common technique for predicting the next value on the basis on time series of past values as specified in RFC 793 which is exponential averaging, we can calculate SRTT(K), the smoothed round-trip time estimate.

$$SRTT(K+1) = \alpha \times SRTT(K) + (1 - \alpha) \times RTT(K+1)$$

By using a constant value of α ($0 < \alpha < 1$), independent of the number of past observations we have a circumstance in which all past values are considered, but the more distant ones have less weight.

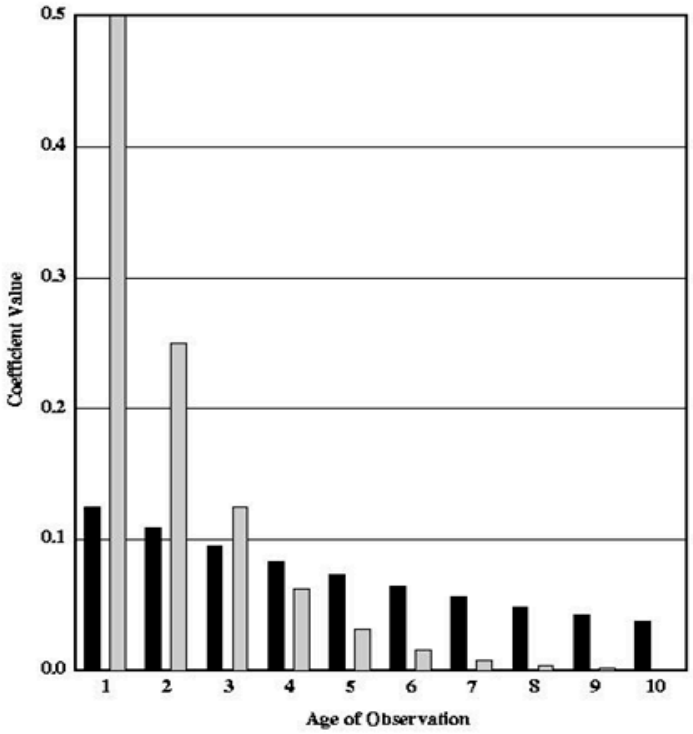
$$SRTT(K+1) = (1 - \alpha) \times RTT(K+1) + \alpha (1 - \alpha) \times RTT(K) + \alpha^2 (1 - \alpha) \times SRTT(K - 1) +$$

$$\ldots\ldots + \alpha^K (1 - \alpha) \times RTT(1)$$

Since both α and $(1 - \alpha)$ are less than one, each successive terms in the preceding equation is smaller. Example, if $\alpha = 0.8$, the expansion is,

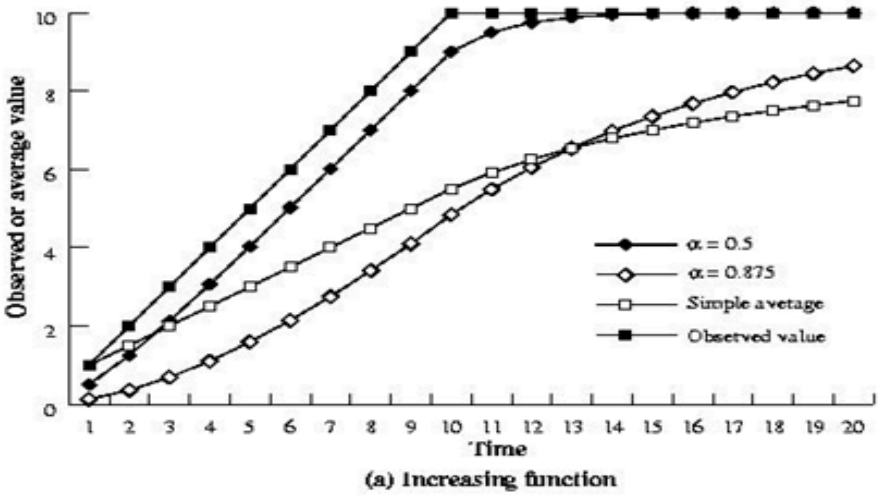
$$SRTT(K+1) = 0.2 \times RTT(K+1) + 0.16 \times RTT(K) + 0.128 \times SRTT(K - 1) + \ldots.$$

Figure shows the size of the coefficient as a function to its position in the expansion.

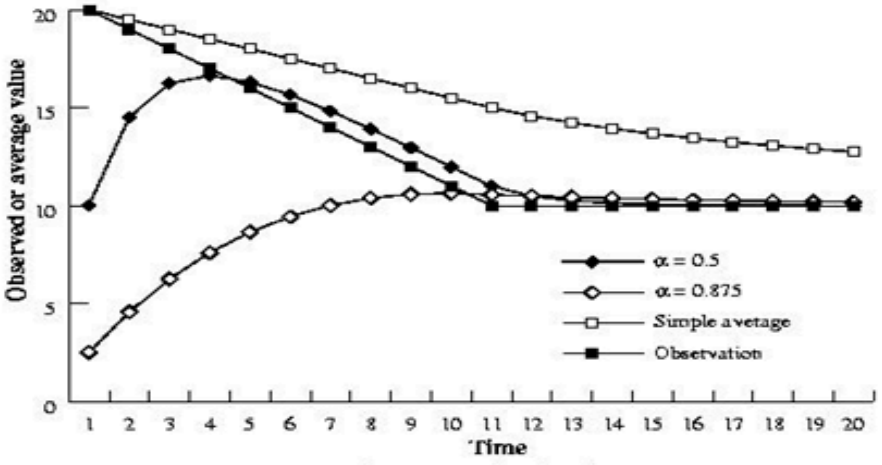


Exponential Smoothing Coefficients

The next Figure compares the simple averaging with exponential averaging for different values of α. In (a) the observed value begins at 1, grows gradually to 10 and then stays there. In (b) the observed value begins at 20 declines gradually to 10 and then stays there. In both the cases SRTT (0) = 0.



(a) Increasing function



(b) Decreasing function

Figure : Use of Exponential Averaging

Usually, the retransmission timer should be set at a value somewhat greater than the estimated round-trip time. The possibility is to use a constant value,

RTO (K + 1) = SRTT (K + 1) + Δ where TO is the retransmission timer(or retransmission timeout) and Δ is a constant. For large values of SRTT, Δ is relatively small and fluctuations in the actual RTT will result in unnecessary retransmission. For small values of SRTT, Δ is relatively large and causes

unnecessary delays in retransmitting lost segments. Thus a range can be considered,

RTO (K + 1) = MIN (UBOUND, MAX (LBOUND , β × SRTT(K + 1)))

where UBOUND and LBOUND are prechosen fixed upper and lower bounds on the timer value and β is a constant. Example values for α, β: $0.8 < \alpha < 0.9$ $1.3 < \beta < 2.0$

Implementation Policy Options

The TCP standard provides a precise specification of the protocol to be used between TCP entities. The design areas for which options are specified are the following:

- ✓ Send Policy – TCP may construct a segment for each batch of data provided by its user or it may wait until a certain amount of data accumulates before constructing and sending a segment, depending on performance considerations.
- ✓ Deliver Policy – same as in the send policy.
- ✓ Accept Policy
  - In-order – accept only segments that arrive in order, if not discarded.
  - In-window – accept all segments that are within the receive window.
- ✓ Retransmit Policy
  - First-only – maintain on retransmission timer for the entire queue. If an acknowledgement is received, remove the appropriate segment or segments from the queue and reset the timer. If the timer expires, retransmit the segment at the front of the queue and restart the timer.
  - Batch - maintain on retransmission timer for the entire queue. If an acknowledgement is received, remove the appropriate segment or segments from the queue and reset the timer. If the timer expires, retransmit all the segment in the queue and restart the timer.
  - Individual - maintain on retransmission timer for the entire queue. If an acknowledgement is received, remove the appropriate segment or segments

from the queue and reset the timer. If the timer expires, retransmit the corresponding segment individually and restart the timer.

- ✓ Acknowledge Policy
  - Immediate – when data are accepted, immediately transmit an empty (no data) segment containing appropriate acknowledgement number.
  - Cumulative – when data are accepted, record the need for acknowledgement but wait for an outbound segment with data on which to piggyback the acknowledgement. To avoid long delay, set a window timer. If the timer expires before an acknowledgement is sent, transmit an empty segment containing the appropriate acknowledgement number.

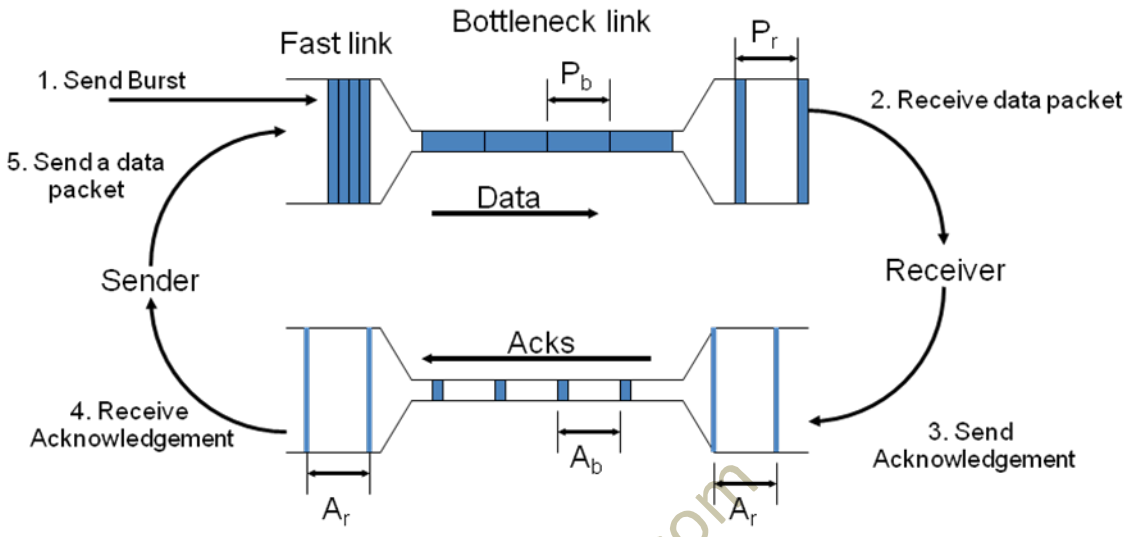
TCP Congestion Control

Congestion in a network creates obvious problems. In PSN or Frame Relay network dynamic routing can alleviate congestion by spreading load more evenly. But this is only effective for unbalanced loads and brief surges in traffic. **Congestion can only be controlled by limiting total amount of data entering network.** ICMP source Quench message is a crude method and not effective. RSVP may help but it is not widely implemented. In TCP congestion control is difficult because,

- IP is connectionless and stateless, with no provision for detecting or controlling congestion
- TCP only provides end-to-end flow control.
- There is no cooperative, distributed algorithm available to bind together various TCP entities

In TCP, the rate at which a TCP entity can transmit is determined by the rate of incoming ACKs to previous segments with new credit. The rate of ACK arrival is determined by the round-trip path between the source and destination. The bottleneck may be either in the destination or internet. The sender cannot tell which of these is true. Thus only the internet bottleneck can be due to congestion.

Self-Clocking Model



A Given: $P_b = P_r = A_r = A_b = A_r$ (in units of time)

s : Sending a packet on each ACK keeps the bottleneck link busy

illustrated in the Figure, the returning ACKs function as pacing signals. In the steady state, after an initial burst, the sender's segment rate will match the arrival rate of the ACKs. Thus the sender's segment rate is equal to that of the slowest link on the path. In this way TCP automatically senses the network bottleneck and regulates its flow accordingly. This is referred to as **TCP's self-clocking behavior.**

TCP Congestion Avoidance:

- TCP's strategy
  - control congestion once it happens
  - repeatedly increase load in an effort to find the point at which congestion occurs, and then back off
- Alternative strategy

  - predict when congestion is about to happen
  - reduce rate before packets start being discarded
  - call this congestion *avoidance*, instead of congestion *control*
- Two possibilities
  - router-centric: DECbit and RED Gateways
  - host-centric: TCP Vegas

End Host

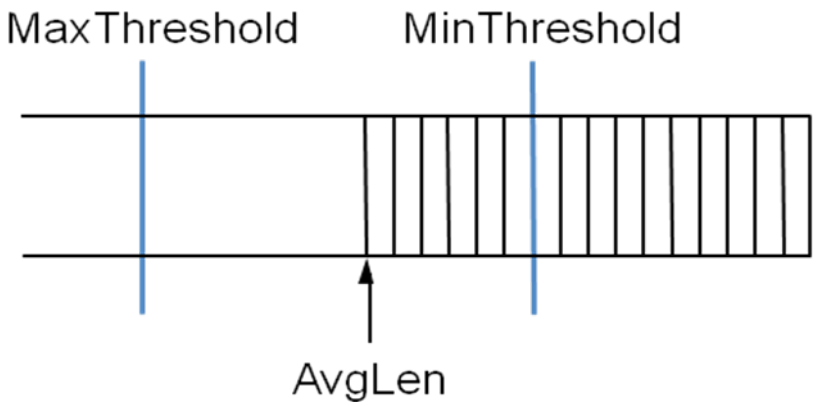
- Destination echoes bit back to source
- Source records how many packets resulted in set bit
- If less than 50% of last window's worth had bit set
  - increase **CongestionWindow** by 1 packet
- If 50% or more of last window's worth had bit set
  - decrease **CongestionWindow** by 0.875 times
- Notification is implicit
  - just drop the packet (TCP will timeout)
  - could make explicit by marking the packet
- Early random drop
  - rather than wait for queue to become full, drop each arriving packet with some *drop probability* whenever the queue length exceeds some *drop level*
- Compute average queue length

  AvgLen = (1 - Weight) * AvgLen +

  Weight * SampleLen

  0 < **Weight** < 1 (usually 0.002)

  SampleLen is queue length each time a packet arrives



Quality of service(QoS): In the field of computer networking and other packet-switched telecommunication networks, the traffic engineering term **quality of service** (QoS) refers to resource reservation control mechanisms rather than the achieved service quality. Quality of service is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. For example, a required bit rate, delay, jitter, packet dropping probability and/or bit error rate may be guaranteed. Quality of service guarantees are important if the network capacity is insufficient, especially for real-time streaming multimedia applications such as voice over IP, online games and IP-TV, since these often require fixed bit rate and are delay sensitive, and in networks where the capacity is a limited resource, for example in cellular data communication.

- A network or protocol that supports QoS may agree on a traffic contract with the application software and reserve capacity in the network nodes, for example during a session establishment phase. During the session it may monitor the achieved level of performance, for example the data rate and delay, and dynamically control scheduling priorities in the network nodes. It may release the reserved capacity during a tear down phase.
- A best-effort network or service does not support quality of service. An alternative to complex QoS control mechanisms is to provide high quality communication over a best-effort network by over-provisioning the capacity so

that it is sufficient for the expected peak traffic load. The resulting absence of network congestion eliminates the need for QoS mechanisms.

- In the field of telephony, **quality of service** was defined in the ITU standard X.902 as "A set of quality requirements on the collective behavior of one or more objects". Quality of Service comprises requirements on all the aspects of a connection, such as service response time, loss, signal-to-noise ratio, cross-talk, echo, interrupts, frequency response, loudness levels, and so on. A subset of telephony QoS is Grade of Service (GOS) requirements, which comprises aspects of a connection relating to capacity and coverage of a network, for example guaranteed maximum blocking probability and outage probability.

- QoS is sometimes used as a quality measure, with many alternative definitions, rather than referring to the ability to reserve resources. Quality of service sometimes refers to the level of quality of service, i.e. the guaranteed service quality. High QoS is often confused with a high level of performance or achieved service quality, for example high bit rate, low latency and low bit error probability.
- An alternative and disputable definition of QoS, used especially in application layer services such as telephony and streaming video, is requirements on a metric that reflects or predicts the subjectively experienced quality. In this context, QoS is the acceptable cumulative effect on subscriber satisfaction of all imperfections affecting the service. Other terms with similar meaning are the Quality of Experience (QoE) subjective business concept, the requried "user perceived performance" [2], the required "degree of satisfaction of the user" or the targeted "number of happy customers". Examples of measures and measurement methods are Mean Opinion Score (MOS), Perceptual Speech

Quality Measure (PSQM) and Perceptual Evaluation of Video Quality (PEVQ). See also subjective video quality.

- Conventional Internet routers and LAN switches lack the ability to provide Quality of Service guarantees. This made Internet equipment cheaper, faster and thus more popular than competing more complex technologies that provided QoS mechanisms, for example X.25. Internet traditionally therefore runs at default QoS level, or "best effort". There were four "Type of Service" bits and three "Precedence" bits provided in each IP packet, but they were ignored. These bits were later re-defined as DiffServ Code Points (DSCP) and are sometimes honored in peered links on the modern Internet.

With the advent of IP-TV and IP-telephony, QoS mechanisms to the end user have eventually become common, but not necessarily based on layer 3 IP routing, but on layer 2 technologies.

A number of attempts for layer 2 technologies that add QoS tags to the data have gained popularity during the years, but then lost attention. Examples are Frame relay and ATM. Recently, MPLS (a technique between layer 2 and 3) have gained some attention. However, today Ethernet may offer QoS and is the by far most popular layer 2 technology.

In Ethernet, Virtual LANs (VLAN) may be used to separate different QoS levels. For example in fibre-to-the-home switches typically offer several Ethernet ports connected to different VLAN:s. One VLAN may be used for Internet access (low priority), one for IP-TV (higher priority) and one for IP telephony (highest priority). Different Internet providers may use the different VLAN:s.

Key qualities of traffic

When looking at packet-switched networks, Quality of Service is affected by various factors, which can be divided into "human" and "technical" factors. Human

factors include: stability of service, availability of service, delays, user information. Technical factors include: reliability, scalability, effectiveness, maintainability, Grade of Service, etc.[3]

Many things can happen to packets as they travel from origin to destination, resulting in the following problems as seen from the point of view of the sender and receiver:

- **Throughput**

  Due to varying load from other users sharing the same network resources, the bit-rate (the maximum throughput) that can be provided to a certain data stream may be too low for realtime multimedia services if all data streams get the same scheduling priority.

- **Dropped packets**

  The routers might fail to deliver (*drop*) some packets if they arrive when their buffers are already full. Some, none, or all of the packets might be dropped, depending on the state of the network, and it is impossible to determine what will happen in advance. The receiving application may ask for this information to be retransmitted, possibly causing severe delays in the overall transmission.

- **Delay**

  It might take a long time for each packet to reach its destination, because it gets held up in long queues, or takes a less direct route to avoid congestion. This is different from throughput, as the delay can build up over time, even if the throughput is almost normal. In some cases, excessive delay can render an application such as VoIP or online gaming unusable.

- **Jitter**

  Packets from the source will reach the destination with different delays. A packet's delay varies with its position in the queues of the routers along the path between source and destination and this position can vary unpredictably. This

variation in delay is known as jitter and can seriously affect the quality of streaming audio and/or video.

- **Out-of-order delivery**

  When a collection of related packets is routed through the Internet, different packets may take different routes, each resulting in a different delay. The result is that the packets arrive in a different order than they were sent. This problem requires special additional protocols responsible for rearranging out-of-order packets to an isochronous state once they reach their destination. This is especially important for video and VoIP streams where quality is dramatically affected by both latency and lack of isochronicity.

- **Error**

  Sometimes packets are misdirected, or combined together, or corrupted, while *en route*. The receiver has to detect this and, just as if the packet was dropped, ask the sender to repeat itself.

  The Internet2 QoS Working Group concluded that increasing bandwidth is probably more practical than implementing QoS.[1][2] However, this group is focused on next-generation Internet rather than QoS in private and converged networks, where QoS is essential.

Applications requiring QoS

A defined Quality of Service may be required for certain types of network traffic, for example:

- streaming multimedia may require guaranteed throughput to ensure that a minimum level of quality is maintained.
- IPTV offered as a service from a service provider such as AT&T's U-verse
- IP telephony or Voice over IP (VOIP) may require strict limits on jitter and delay
- Video Teleconferencing (VTC) requires low jitter and latency
- Alarm signalling (e.g., Burglar alarm)

- dedicated link emulation requires both guaranteed throughput and imposes limits on maximum delay and jitter
- a safety-critical application, such as remote surgery may require a guaranteed level of availability (this is also called *hard QoS*).
- a remote system administrator may want to prioritize variable, and usually small, amounts of SSH traffic to ensure a responsive session even over a heavily-laden link.
- online games, such as fast paced real time simulations with multiple players. Lack of QoS may produce 'lag'.
- Industrial Ethernet protocols such as Ethernet/IP which are used for real-time control of machinery

These types of service are called *inelastic*, meaning that they require a certain minimum level of bandwidth and a certain maximum latency to function.
By contrast, *elastic* applications can take advantage of however much or little bandwidth is available. Bulk file transfer applications that rely on TCP are generally elastic.

Obtaining QoS

- Per call
- In call
- In advance: When the expense of mechanisms to provide QoS is justified, network customers and providers typically enter into a contractual agreement termed a service level agreement (SLA) which specifies guarantees for the ability of a network/protocol to give guaranteed performance/throughput/latency bounds based on mutually agreed measures, usually by prioritizing traffic.
- Reserving resources: Resources are reserved at each step on the network for the call as it is set up. An example is RSVP, Resource Reservation Protocol.

QoS mechanisms

An alternative to complex QoS control mechanisms is to provide high quality communication by generously over-provisioning a network so that capacity is based on peak traffic load estimates. This approach is simple and economical for networks with predictable and light traffic loads. The performance is reasonable for many applications. This might include demanding applications that can compensate for variations in bandwidth and delay with large receive buffers, which is often possible for example in video streaming.

Commercial VoIP services are often competitive with traditional telephone service in terms of call quality even though QoS mechanisms are usually not in use on the user's connection to his ISP and the VoIP provider's connection to a different ISP. Under high load conditions, however, VoIP quality degrades to cell-phone quality or worse. The mathematics of packet traffic indicate that a network with QoS can handle four times as many calls with tight jitter requirements as one without QoSYuksel et al. have determined 60% required extra capacity by simulating IP traffic under conservative assumptions [.

The amount of over-provisioning in interior links required to replace QoS depends on the number of users and their traffic demands. As the Internet now services close to a billion users, there is little possibility that over-provisioning can eliminate the need for QoS when VoIP becomes more commonplace.

For narrowband networks more typical of enterprises and local governments, however, the costs of bandwidth can be substantial and over provisioning is hard to justifyIn these situations, two distinctly different philosophies were developed to engineer preferential treatment for packets which require it.

Early work used the "IntServ" philosophy of reserving network resources. In this model, applications used the Resource reservation protocol (RSVP) to request and reserve resources through a network. While IntServ mechanisms do work, it was realized that in a broadband network typical of a larger service provider, Core routers would be required to accept, maintain, and tear down thousands or possibly tens of

thousands of reservations. It was believed that this approach would not scale with the growth of the Internet, and in any event was antithetical to the notion of designing networks so that Core routers do little more than simply switch packets at the highest possible rates.

The second and currently accepted approach is "DiffServ" or differentiated services. In the DiffServ model, packets are marked according to the type of service they need. In response to these markings, routers and switches use various queuing strategies to tailor performance to requirements. (At the IP layer, differentiated services code point (DSCP) markings use the 6 bits in the IP packet header. At the MAC layer, VLAN IEEE 802.1Q and IEEE 802.1p can be used to carry essentially the same information) Routers supporting DiffServ use multiple queues for packets awaiting transmission from bandwidth constrained (e.g., wide area) interfaces. Router vendors provide different capabilities for configuring this behavior, to include the number of queues supported, the relative priorities of queues, and bandwidth reserved for each queue. In practice, when a packet must be forwarded from an interface with queuing, packets requiring low jitter (e.g., VoIP or VTC) are given priority over packets in other queues. Typically, some bandwidth is allocated by default to network control packets (e.g., ICMP and routing protocols), while best effort traffic might simply be given whatever bandwidth is left over.

Additional bandwidth management mechanisms may be used to further engineer performance, to include:

- Traffic shaping (rate limiting):
  - Token bucket
  - Leaky bucket
  - TCP rate control—artificially adjusting TCP window size as well as controlling the rate of ACKs being returned to the sender[*citation needed*]
- Scheduling algorithms:
  - Weighted fair queuing (WFQ)

    - Class based weighted fair queuing
    - Weighted round robin (WRR)
    - Deficit weighted round robin (DWRR)
    - Hierarchical Fair Service Curve (HFSC)
- Congestion avoidance:
    - RED, WRED - Lessens the possibility of port queue buffer tail-drops and this lowers the likelihood of TCP global synchronization
    - Policing (marking/dropping the packet in excess of the committed traffic rate and burst size)
    - Explicit congestion notification
    - Buffer tuning

As mentioned, while DiffServ is used in many sophisticated enterprise networks, it has not been widely deployed in the Internet. Internet peering arrangements are already complex, and there appears to be no enthusiasm among providers for supporting QoS across peering connections, or agreement about what policies should be supported in order to do so.

One compelling example of the need for QoS on the Internet relates to this issue of congestion collapse. The Internet relies on congestion avoidance protocols, as built into TCP, to reduce traffic load under conditions that would otherwise lead to Internet Meltdown. QoS applications such as VoIP and IPTV, because they require largely constant bitrates and low latency cannot use TCP, and cannot otherwise reduce their traffic rate to help prevent meltdown either. QoS contracts limit traffic that can be offered to the Internet and thereby enforce traffic shaping that can prevent it from becoming overloaded, hence they're an indispensable part of the Internet's ability to handle a mix of real-time and non-real-time traffic without meltdown.

Asynchronous Transfer Mode (ATM) network protocol has an elaborate framework to plug in QoS mechanisms of choice. Shorter data units and built-in QoS were some of

the unique selling points of ATM in the telecommunications applications such as video on demand, voice over IP.

QoS priority levels

Priority Level	Traffic Type
0 (lowest)	Best Effort
1	Background
2	Standard (Spare)
3	Excellent Load (Business Critical)
4	Controlled Load (Streaming Multimedia)
5	Voice and Video (Interactive Media and Voice) [Less than 100ms latency and jitter]
6	Layer 3 Network Control Reserved Traffic [Less than 10ms latency and jitter]
7 (highest)	Layer 2 Network Control Reserved Traffic [Lowest latency and jitter]

Protocols that provide quality of service

- The Type of Service (TOS) field in the IP header (now superseded by Diffserv)
- IP Differentiated services (DiffServ)
- IP Integrated services (IntServ)

- Resource reSerVation Protocol (RSVP)
- Multiprotocol Label Switching (MPLS) provides eight QoS classes
- RSVP-TE
- Frame relay
- X.25
- Some ADSL modems
- Asynchronous Transfer Mode (ATM)
- IEEE 802.1p
- IEEE 802.1Q
- IEEE 802.11e
- HomePNA Home networking over coax and phone wires
- The ITU-T G.hn standard provides QoS by means of "Contention-Free Transmission Opportunities" (CFTXOPs) which are allocated to flows which require QoS and which have negotiated a "contract" with the network controller. G.hn also supports non-QoS operation by means of "Contention-based Time Slots".

QoS Solutions

The research project MUSE defined a QoS concept in Phase I which was further worked out in another research project PLANETS. The new idea of this solution is to agree on a discrete jitter value per QoS class which is imposed on network nodes. Including best effort, four QoS classes were defined, two elastic and two inelastic. The solution has several benefits:

- End-to-end delay and packet loss rate can be predicted
- It is easy to implement with simple scheduler and queue length given in PLANETS
- Nodes can be easily verified for compliance
- End users do notice the difference in quality

The MUSE project finally elaborated its own QoS solution which is primarily based in:

- The usage of traffic classes
- Selective CAC concept
- Appropriate network dimensioning

Quality of service procedures

Unlike the Internet 2 Abilene Network, the Internet is actually a series of exchange points interconnecting private networks and not a network in its own right.[5] Hence the Internet's core is owned and managed by a number of different Network Service Providers, not a single entity. Its behavior is much more stochastic or unpredictable. Therefore, research continues on QoS procedures that are deployable in large, diverse networks.

There are two principal approaches to QoS in modern packet-switched networks, a parameterized system based on an exchange of application requirements with the network, and a prioritized system where each packet identifies a desired service level to the network. On the Internet, Integrated services ("IntServ") implements the parameterized approach. In this model, applications use the Resource Reservation Protocol (RSVP) to request and reserve resources through a network. Differentiated services ("DiffServ") implements the prioritized model. DiffServ marks packets according to the type of service they need. In response to these markings, routers and switches use various queueing strategies to tailor performance to requirements. (At the IP layer, differentiated services code point (DSCP) markings use the first 6 bits in the TOS field of the IP packet header. At the MAC layer, VLAN IEEE 802.1q and IEEE 802.1p can be used to carry essentially the same information.) Cisco IOS NetFlow and the Cisco Class Based QoS (CBQoS)Management Information Base (MIB) can both be leveraged within a Cisco network device to obtain visibility into QoS policies and their effectiveness on network traffic. [6]

Non-IP protocols, specially those intended for voice transmission, such as ATM or GSM, have already implemented QoS in the core protocol and don't need additional procedures to achieve it.

End-to-end Quality of Service

End-to-end Quality of Service usually requires a method of coordinating resource allocation between one automonous system and another. Research consortia such as EuQoS [3] and fora such as IPsphere [4] have developed mechanisms for handshaking QoS invocation from one domain to the next. IPsphere defined the SSS signaling bus (Service Structuring Stratum) in order to setup, invoke and assure network services. EuQoS conducted experiments to integrate SIP, NSIS and IPsphere's SSS.

The Internet Engineering Task Force (IEFT) defined the RSVP protocol for bandwidth reservation. RSVP is an end to end bandwidth reservation protocol that is also useful to end to end QoS.RSVP:Resource reservation protocol. The traffic engineering version, RSVP-TE, is used in many networks today to establish traffic-engineered MPLS label-switched paths.

The IEFT also defined, NSIS (Next Steps in Signalling) with QoS signalling as a target. NSIS is a development and simplification of RSVP. NSIS [5]

Quality of service circumvention

Strong cryptography network protocols such as Secure Sockets Layer, I2P, and virtual private networks obscure the data transferred using them. As all electronic commerce on the Internet requires the use of such strong cryptography protocols, unilaterally downgrading the performance of encrypted traffic creates an unacceptable hazard for customers. Yet, encrypted traffic is otherwise unable to undergo deep packet inspection for QoS.