

Deploying IP and MPLS QoS for Multiservice Networks

QoS Requirements and Service Level Agreements

LECTURE 3

Lecturer: Associate Professor A.S. Eremenko

VPN Hose and Pipe Models

Consider a network connecting four sites (#1, #2, #3 and #4); this could be a layer 2 virtual private network (VPN) offered by a service provider using a technology such as leased lines, Frame-relay or ATM, for example. Whichever underlying technology is used the sites could be interconnected in a hub and spoke arrangement with spoke sites connected back to a hub site using **leased lines** or **virtual circuits (VCs)**, or they could be interconnected with a **full mesh of leased lines or VCs**; both options are shown in **Fig. 1**.

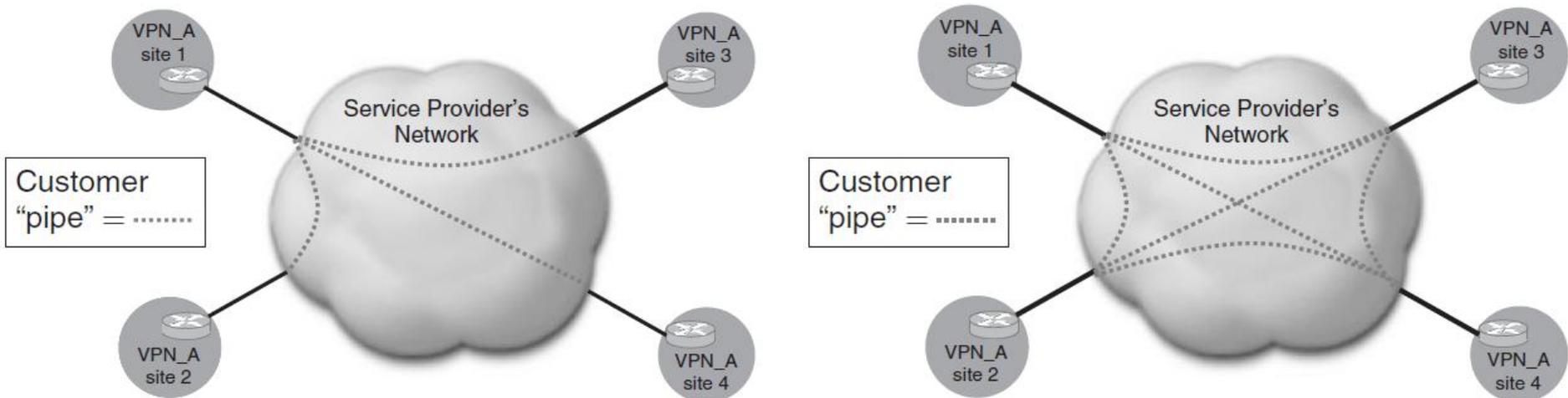


Figure 1 Hub and spoke (left) and full mesh (right) VPNs using the "pipe model"

VPN Hose and Pipe Models

In both cases shown in **Figure 1**, each leased line or VC may have a defined SLA commitment; **this type of point-to-point bandwidth commitment was termed a “pipe”**; these point-to-point commitments provide isolation between the performance of each “pipe.” **The use of the “pipe model” is obvious for point-to-point services such as leased lines, Frame relay, ATM or layer 2 “pseudo wires” as defined by the Pseudo Wire Emulation Edge-to-Edge Working Group within the IETF.** When layer 3 services are built on top of such point-to-point pipes, however, as the number of sites within a VPN increases, provisioning such point-to-point commitments can become cumbersome.

For example, a full mesh between n sites requires $n(n - 1)/2$ connections, e.g. **100 sites** would require $100 * 99/2=4950$ such **pipes**. In addition, such point-to-point commitments can be inefficient with respect to the use of provisioned capacity. For example, **site #1** may have **1 Mbps VCs** provisioned to each of **sites #2, #3, and #4** (i.e. **3 Mbps in total**), yet **if the VC to site #2 is not busy**, this unused capacity to/from site #2 **cannot be re-used for traffic between site #1 and sites #3 or #4**, i.e. up to 1 Mbps of capacity to/from site #1 would go idle.



VPN Hose and Pipe Models

VPNs built using IP or MPLS technology, e.g. BGP MPLS VPNs as per [RFC4364], can implicitly provide “any-to-any” connectivity between the sites within the VPN; however, this gives rise to the question of *how to define SLAs between sites within VPNs that provide multipoint-to-multipoint connectivity, when you do not have a corresponding pipe (or its SLA assurances) between those sites?*

It can be defined the “**hose model**” for multipoint-to-multipoint VPN services. With the hose model, rather than defining SLAs on a point-to-point basis between pairs of sites, the SLAs are defined in terms of a “hose” from each site to and from the VPN provider network. From a capacity perspective, the “hose” for each site is defined in terms of the ingress committed rate (ICR) to the provider and the egress committed rate (ECR) from the provider, as shown in Figure 2.

Traffic between two sites that is within the ICR contract at the source site, and within the ECR contract at the destination site is assured end-to-end. ICR/ECR could be defined with a single class per site, or in the context of a Diffserv enabled service, could be offered on a per class per site basis. Hose model SLAs can provide the benefits of statistical multiplexing, where pipe model SLAs cannot.



VPN Hose and Pipe Models

For example, if **site #1** has an **ICR and ECR of 3 Mbps**, it could use that capacity to communicate with any of **sites #2, #3, and #4**, i.e. if there were no traffic to **site #2**, the **unused capacity to/from site #2 could potentially be re-used for traffic between site #1 and sites #3 or #4**. A consequence of this is that hose model SLAs also need to make provision for mediation between the ICR and ECR between different sites.

For example, the **ICR for site #1 may be 3 Mbps**; however, the **attainable capacity to site #4 will be limited by the ECR of site #4, which is 1 Mbps**. In addition, hose model SLAs need to take into account cases where the loss of attainable throughput is due to customer-based traffic aggregation. For example, if **sites #2, #3 and #4 all attempt to send traffic at their full ICRs (which totals 4 Mbps) to site #1**, their aggregate attainable capacity will be limited by the **ECR of site #1, which is only 3 Mbps**.

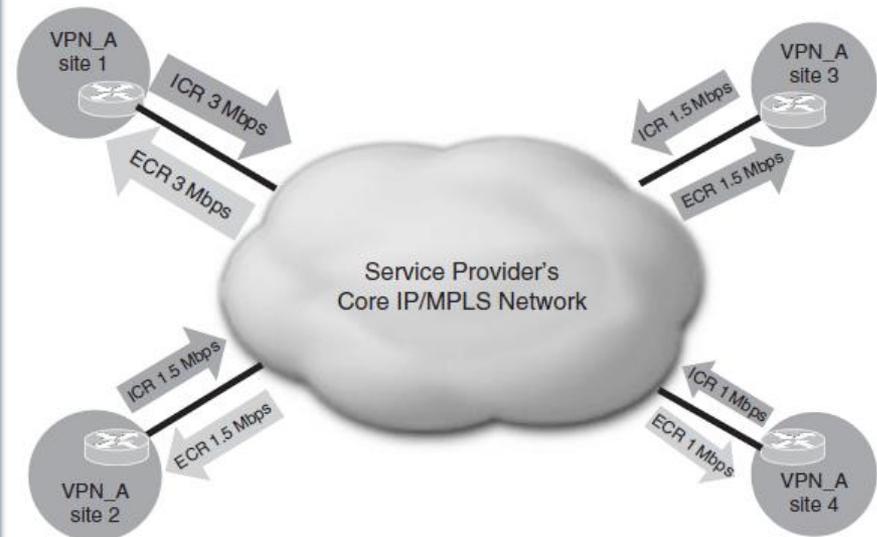


Figure 2 Any-to-any VPNs using the "hose model"

Per Flow Sequence Preservation

IP does not guarantee that packets are delivered in the order in which they were sent. As defined in the IETF by [RFC4737], if the packets in a flow were numbered sequentially in the order in which they were sent, a packet that arrived with a sequence number smaller than that of their predecessor would be defined as out-of-order, or re-ordered.

The simplest metric by which to measure the magnitude of re-ordering is as a re-ordering ratio, which is the ratio of re-ordered packets that arrived, relative to the total number of packets received. A number of other metrics for quantifying the magnitude of re-ordering are defined in [RFC4737].



Due to the adverse impact that packet re-ordering can have on the performance of some applications, it is accepted **best practice in IP network design to prevent packet re-ordering within a flow.**

There are **two key design best practices** in order to prevent packet re-ordering within a flow.



Per Flow Sequence Preservation

- 1) It is important that any IP load balancing across multiple paths within the network is performed on a per flow level rather than on a per packet level such that all packets within a flow follow the same path. This load balancing is performed by Equal Cost Multipath (ECMP) algorithms, where there are multiple IGP paths with the same cost. ECMP algorithms commonly perform a hash function to determine which of the paths a packet will take; where the hash function uses the 5-tuple of IP protocol, source IP address, destination IP address, source UDP/TCP port, and destination UDP/TCP as inputs, and results in packets within a single flow consistently hashing to the same path.
- 2) QOS designs and scheduling algorithms must ensure that packets from the same flow are always serviced in the same order and from the same queue; this is a fundamental principle of both the Integrated Services and Differentiated Services QOS architectures.



Availability. Network Availability

Availability for IP services is generally defined in one of two ways: either as **network availability** or as **service availability**.

Network availability (sometimes referred to as **connectivity**) is defined as the fraction of time that network connectivity is available between a specified network ingress point and a specified network egress point.

Availability can be **unidirectional** or **bidirectional**; bidirectional connectivity is what matters to the vast majority of IP applications, i.e. that a source can send a packet to a destination that elicits a response which is received by the source.

A metric for measuring connectivity has been defined by [**RFC2678**] in the IETF.

The availability of the network can be estimated by calculating the availability of each individual network element and then combining the availabilities in series or in parallel as appropriate using the following formulae.



Network Availability

Component availability

The availability (**A**) of an individual component is the proportion of the time for which the device is working:

$$A = \frac{\text{time_working}}{\text{total_time}} = \frac{MTBF}{MTBF + MTTR}$$

Where:

MTBF – mean time between failures;

MTTR – mean time to restore.

Component unavailability

The unavailability (**U**) of an individual component is the proportion of the time for which the device is not working:

$$U = \frac{\text{time_not_working}}{\text{total_time}} = \frac{MTTR}{MTBF + MTTR} = 1 - A$$



Network Availability

Availability of components in series

The availability of components (a, b, c, \dots) in series (A_s) is given by:

$$A_s = [A(a) \times A(b) \times A(c) \times \dots]$$

Availability of components in parallel

The availability of components (a, b, c, \dots) in parallel (A_p) is given by:

$$A_p = [1 - (U(a) \times U(b) \times U(c) \times \dots)]$$

For most applications, however, simply having connectivity is not enough. For VoIP for example, it is of little practical use if there is connectivity between two VoIP end-systems but the VoIP packets arrive so delayed that speech between the two calling parties becomes unintelligible, hence service availability is often a more meaningful metric.



Service Availability

Service availability is defined as the fraction of time the service is available between a specified ingress point and a specified egress point within the bounds of the other defined SLA metrics for the service, e.g. delay, jitter, and loss.

Service availability can be defined in one of **two ways**:

1. either it can be defined **independently of network availability**, in which case the service availability cannot exceed the network availability,
2. or it can be defined as being applicable only when the network is considered available.

Service availability may encompass application performance as well as network performance.

For instance, the service availability may comprise hostname resolution (DNS server) and transaction time, thereby depending on network delay and web server performance.

There may be overlap between the definition of network or service availability and the definition of other SLA parameters.



Quality of Experience

In addition to the metrics, which define the characteristics of the network, there are additional **metrics**, which aim to **quantify the performance experienced by the applications using the network**. These metrics define the perception of application performance, experienced from the perspective of the end-users, which is also known as the user **“quality of experience” (QOE)**.

For IP-based voice and video applications the QOE is a compound metric dependent upon the quality of the encoder used, the quality of the service delivered by the IP network, and the quality of the decoder used. As such, QOE targets do not directly define the delay, jitter, loss etc., characteristics that a network should provide, but rather for a specified application, using a defined encoder/decoder, the network characteristics may be implied given a particular QOE target.

QOE metrics can be measured subjectively or objectively:

1. Subjective measures rely upon end-user feedback of their perception of the quality of the service.
2. Objective measures use measurements of characteristics of the received stream, and possibly also of the transmitted stream, in order to infer the subjective quality that would be experienced by the end-user.

There are QOE metrics defined for **voice**, **video** and **on-line gaming applications**.



Voice

Subjective measures

The Mean Opinion Score (MOS) is a well established scheme, which provides a numeric measure of the quality of a voice call at the destination.

MOS is a formally tested subjective measure, which is defined by the **ITU [P.800]** and is determined using a number of human listeners participating in a set of standard tests, subjectively scoring the quality of test sentences read aloud over the communications medium being tested using the scale:

excellent (5), good (4), fair (3), poor (2) and bad (1).

The MOS for the medium under test is calculated by taking the arithmetic mean of all the individual scores.

A typical Public Switched Telephony (PSTN) voice service has MOS of 4.3, while mobile telephone services typically have a MOS of between 2.9 and 4.1.



Voice

Objective measures

There are several **recommendations** provided by the **ITU**, which provide methods for **objective voice quality monitoring, and that can also be used to estimate the MOS**. These schemes rely on characteristics of the received stream only.

- ❑ **ITU P.862** defines the **Perceptual Evaluation of Speech Quality (PESQ, pronounced “pesk”)**, and is a full reference (where full information about both the transmitted and received audio signals are available when the audio quality is determined) objective method for predicting the subjective MOS quality of telephony services.
- ❑ **ITU G.107** defines the so-called **“E model”** which uses a number of transmission level parameters to rate the quality of a transmission system, in order to assess the effects on telephony services. The primary output from the **E model** is the **“Rating Factor,” R**, which can be transformed to give estimates of the MOS (an objective MOS score) for calls which use that transmission service.



Video

- ❑ **Subjective measures.** The main concepts behind the subjective measurement of video quality are the same as for MOS for voice. The most established video subjective testing scheme in the broadcasting world is the **Double Stimulus Continuous Quality Scale (DSCQS)** method defined in **ITU specification [BT.500]**. An alternative methodology that **The European Broadcasting Union (EBU)** has defined is called the **SAMVIQ Subjective Assessment Methodology for Video Quality**.
- ❑ **Objective measures.** The most established objective measure of video quality is defined in **ITU-T standard J.144**. This provides guidelines on perceptual video quality measurement for use in digital cable television applications when the full reference video signal is available, i.e. both the transmitted and received video signals are available for comparison when the video quality is determined. It is still a subject of study as to whether reduced reference (where only partial information about transmitted video signal and full information about the received video signal is available when the video quality is determined) or no reference (where only the received video signal is available when the video quality is determined) can be used to accurately infer subjective video quality, i.e. to correctly suggest that the video quality is bad when, and only when, the end viewer also thinks it is bad.

Also defined a MOS metric to classify the player's perceived game quality in **On-line Gaming applications** [NetGames'05, Hawthorne (NY), U.S.A.].

