5

SLA and Network Monitoring

5.1 Introduction

This chapter discusses the technologies and techniques available for SLA and network monitoring in QOS-enabled IP networks. There are two main approaches, which are generally used in concert to monitor the performance of a QOS-enabled network service in order to determine whether SLAs have been or can be met:

- *Passive network monitoring*. With passive network monitoring, network devices record statistics on network traffic, which can provide an indication of the status at a particular network element. Periodic polling is typically used to gather this data for reporting and analysis. This is a micromeasure which looks at each device in isolation; by looking at multiple network elements an aggregate view of the status of a network service may be deduced. Passive network monitoring does not require any additional traffic be used for measurement purposes.
- *Active network monitoring*. Unlike passive monitoring, active monitoring involves sending additional traffic into the network. Synthetic test streams comprising "probe" packets are sent across the network solely for the purpose of characterizing the network performance; analysis of the received streams is used for this characterization. Active monitoring provides a macromeasure of network SLAs in

This chapter has benefitted enormously from the input of Emmanuel Tychon, Technical Marketing Engineer for Cisco IOS IP Service Level Agreement (IP SLAS), whose contribution formed the basis of the active monitoring section.

that it reports the measured performance across a number of network elements as a system.

Passive and active network monitoring systems may be deployed for a number of reasons:

- For monitoring and reporting that the network service offered is achieving the committed SLA targets, this may include:
  - proactive network and SLA monitoring
  - long-term trending of the relative changes in network SLA performance over time.

  For network service providers (SPs), active and passive network monitoring provide potential value-added service opportunities as end customers look to outsource their end-to-end WAN-related capacity management. Hence, the SP may report enough information to the customer to let them assess their network usage and how well their SLAs were met.
- For monitoring that network performance is sufficient to meet the required application quality of experience (QOE) targets.
- As a feedback loop to network capacity planning processes, results from passive and active monitoring may provide heuristics, allowing capacity planning thresholds to be tuned based upon correlation between network or per-class load and SLA probing reports of delay, jitter, and loss. Capacity planning is discussed in detail in Chapter 6, Section 6.1.

Passive and active network monitoring are discussed in more detail in the following sections.

5.2 Passive Network Monitoring

From a QOS perspective, passive network monitoring involves polling the network devices for statistics which they maintain for QOS functions they perform, such as packet and byte counts, or queue depths, for example. This is typically performed using the Simple Network Management Protocol (SNMP) [RFC1157], to poll for information contained in management information bases or MIBs. The considerations on

polling and the types of statistics polled are described in the following sections.

5.2.1 How Often to Poll?

Any polling of network devices for statistics raises the question of how frequently to poll? In practice, this represents a balance between the polling capacity of the network management system (NMS), the number of devices that need to be polled, the load incurred on the polled devices, and the impact of the polling traffic on the network.

Many of the retrieved statistics will be in the form of packet and byte counts; these can be used to determine the average traffic demands over the previous sampling interval. Longer polling intervals implicitly have a larger sample size and may be acceptable for trending purposes; however, the polled data will implicitly be averaged over a longer time and hence issues may be hidden. Therefore, shorter intervals are preferred where measurements that are more granular are required, although this has to be balanced against the increased polling load.

For troubleshooting, proactive measurement and SLA reporting, within the bounds of the NMS and network constraints and capabilities, QOS statistics should be polled as often as possible to prevent visibility of SLA affecting network issues being lost due to the effects of averaging. If the polling is frequent, the data can always be averaged over longer timeframes.

For trending, it may be more appropriate to poll every hour. Longer duration measurements make the comparison between days, months, and years easier and more statistically relevant.

5.2.2 Per-link Statistics

Per-link QOS statistics can be used for different purposes, depending upon from where in the network they are recorded:

- *Access links*. Network access links can be both the boundary of a Diffserv domain and a customer/provider boundary. Hence, access link QOS statistics are used both for faultfinding and for reporting

statistics to customers of end-services such that they can provision their edge QOS classes adequately.

- *Core links*. On core links, per-link QOS statistics are used both for faultfinding and as an input to the core network capacity planning processes. Capacity planning is discussed in more detail in Chapter 6, Section 6.1.

Most vendors implement proprietary MIBs, which can be used to retrieve the relevant per-link statistics. They could also be retrieved from the Diffserv MIB [RFC3289], although this is not widely implemented by network equipment vendors. Where it is supported, the Diffserv MIB may be used for both monitoring and configuration of a router or switch that is capable of Differentiated Services functionality. As the Diffserv MIB is designed to be generic across vendors, vendor proprietary MIBs may provide information on QOS statistics that are specific to their implementation, and hence which are not available in RFC3289.

The following sections describe the most important per-link QOS statistics for monitoring Diffserv deployments in terms of the QOS functions and mechanisms that are applied. Consideration is also provided on how these statistics should be interpreted to assure the performance of a QOS-enabled network service. In some cases, it may not be necessary to monitor all of the statistics that are described; some of the statistics are interrelated and hence may be deduced from others without requiring explicit monitoring. This duplication can be useful in providing a means for cross-verifying the retrieved statistics.

5.2.2.1 Monitoring Classification

A router may classify a number of traffic streams into a single traffic class, to which actions may subsequently be applied. The following classification statistics are useful in understanding the offered traffic load in each class, and the constituents of that traffic class:

- *Per-classification rule*. If multiple rules are used to classify traffic streams into a single class, it may be useful to know the total number of packets and their cumulative byte count that have been classified per rule. For example, if traffic marked DSCP 18 (i.e. AF21)

and DCSP 20 (i.e. AF22) is to be classified into the same class, which is serviced with an AF PHB, then it may be useful to know how much AF21 traffic (which could, for example represent the "in-contract" traffic) and how much AF22 traffic (which could represent the "out-of-contract" traffic) there is within the class.

Further, by knowing both the number of packets and bytes classified into a class, it is possible to estimate the average packet size for the class. This information can be useful for ensuring that only small VoIP packets are being classified into a voice class, for example. Hence, in general for most QOS statistics polled, the results retrieved include both a packet and a byte count.

- *On aggregate*. Per-traffic class, it is also important to know the total number of packets and bytes that have been classified on aggregate (i.e. across all classification rules) into that particular class.

The main use for classification statistics is to verify that traffic is being correctly classified in the appropriate class. Classification statistics can also be used to verify or deduce other statistics; for example, the total number of packets dropped and transmitted by the other functions applied to a particular class after classification must equal the total number of packets classified into that class.

5.2.2.2 Monitoring Policing

Policers may be applied for a number of reasons as described in Chapter 2, Section 2.2.3. Which statistics are relevant when monitoring policers depends upon the way in which they are used.

- *Enforcing a maximum rate for a voice class*. The single rate three color marker (SR-TCM) defined in [RFC2697] is commonly applied to police the maximum rate of a voice class. This may be used both on core and access links. On core links policers are commonly applied to voice classes to ensure the voice class cannot starve other classes of bandwidth, as per the example in Chapter 3, Section 3.3.2.3.1. On access links policers are used both to prevent starvation of other

classes and to enforce a Diffserv edge traffic conditioning agreement (TCA), ensuring that only voice traffic which conforms to the voice class TCA is admitted into the Diffserv network.

In either case when the SR-TCM is used to police a voice class it would typically have a defined CIR and CBS, with EBS = 0, a violate (i.e. red) action of transmit and a conform (i.e. green) action of drop. Applied in this way the SR-TCM would enforce a maximum rate of CIR and a burst of CBS on the voice class and any traffic in violation of this would be dropped.

Wherever a policer is applied to a voice class, the following statistics should be monitored per policer:

- *Number of packets and bytes conforming (i.e. green)*. This is the number of packets and bytes transmitted by the policer.
- *Number of packets and bytes violating (i.e. red)*. This is the number of packets and bytes dropped by the policer. Wherever a policer is used to enforce a maximum rate for a voice class, the policer is meant as a protective measure. If the policer actually drops voice packets there is an issue somewhere, which is affecting the service (assuming that the policer has been correctly configured that is) and voice call quality will be affected, hence ideally there should be no packets violating the SR-TCM policer definition. If there are, the resulting actions will depend upon where the policer is being used:
  - *Access links*. To resolve drops by a voice class policer on an access link, either the bandwidth provisioned for the voice class (and hence the policer rate) needs to be increased, or controls need to be put in place to limit the offered voice traffic load, e.g. using admission control.
  - *Core links*. Drops by a voice class policer are an indication of either a capacity planning failure, or a major network failure or a network attack. In either case, the occurrence of such drops should trigger further investigation to determine the cause of the drops and to prevent a reoccurrence.

- *Marking in- and out-of-contract*. Either the SR-TCM or the two rate three color marker (TR-TCM) defined in [RFC2698] are commonly

applied to AF classes to mark certain amounts of traffic in-contract and out-of-contract as described in Chapter 2, Section 2.2.3. When deployed in this way, which statistics are important depends upon whether the SR-TCM or TR-TCM is used:

- *SR-TCM*. The SR-TCM is commonly used for in-/out-of-contract marking with EBS = 0, a green action of {transmit + mark in-contract} and a red action of {transmit + mark out-of-contract}, as per the example in Chapter 3, Section 3.2.2.4.5. Applied in this way the SR-TCM would enforce a maximum rate of CIR and a burst of CBS on the traffic stream. Conforming traffic would be marked in-contract and any traffic in violation of this would be marked out-of-contract. When deployed in this way the important statistics are:
  - *Number of packets and bytes conforming (i.e. green)*. This is the number of packets marked in-contract by the policer, and their respective byte count.
  - *Number of packets and bytes violating (i.e. red)*. This is the number of packets marked out-of-contract by the policer, and their respective byte count.

  The purpose of marking certain amounts of traffic in-/out-of-contract is to be able to offer a committed SLA for a defined "in-contract" rate, and to allow traffic in excess of this rate to be transmitted but to mark it differently to indicate that it is "out-of-contract" such that it may potentially be given a less stringent SLA. Hence, when the SR-TCM is applied in this way, the main use for statistics of packets and bytes conforming and violating is for reporting to customers of end-services such that they can provision their edge QOS classes adequately, rather than for faultfinding.
- *TR-TCM*. The TR-TCM can be used to mark a certain amount of a traffic class as in-contract, and everything above that as out-of-contract, up to a maximum rate above which all traffic is dropped, by applying a green action of transmit, yellow action of {transmit + mark out-of-contract} and red action of drop. Applied in this way the TR-TCM would enforce a maximum rate of CIR and a burst of CBS on the traffic stream; any traffic in excess would then be marked out-of-contract up to a maximum

rate of PIR and a burst of PBS. When deployed in this way the important statistics are:

- *Number of packets and bytes conforming (i.e. green)*. This is the number of packets marked in-contract by the policer, and their respective byte count.
- *Number of packets and bytes exceeding (i.e. yellow)*. This is the number of packets marked out-of-contract by the policer, and their respective byte count.
- *Number of packets and bytes violating (i.e. red)*. This is the number of packets and bytes dropped by the policer.

Similarly to where the SR-TCM is used for in-/out-of-contract marking, where the TR-TCM is used for this purpose, the main use for statistics of packets and bytes conforming and exceeding is for reporting to customers of end-services. However, if there are a significant number of packets which are violating, i.e. dropped, relative to the number of packets transmitted, i.e. conforming + exceeding; this is an indication that the class load is exceeding the available capacity and the performance of all applications within that class may be affected. Hence, consideration should be given to increasing the PIR configured for that class or to reducing the traffic load within the class.

5.2.2.3 Monitoring Queuing and Dropping

For all queuing classes, it is normal to monitor the following statistics:

- *Number of packets and bytes transmitted*. This is the number of packets successfully transmitted from the queue by the scheduler, and their respective byte count.

- *Number of packets and bytes dropped*. This is the number of packets dropped by queue management functions acting on that queue, and their respective byte count. The statistics that matter with respect to dropping mechanisms depend upon the particular dropping mechanisms that are used.

5.2.2.3.1 Monitoring Tail Drop

If simple tail drop is used to enforce a queue limit (see Chapter 2, Section 2.2.4.2.1) then a count of the number of packets and bytes dropped per queue should be monitored.

If a queue limit is applied to a voice or video class queue, it is normal practice for the queue limit to be at least as great as the burst size for the policer configured for the class. In this case, the policer burst should constrain the class burst and there should be no tail drops experienced for that queue; if tail drops are experienced, this would be an indication of an issue. If the queue limit were set less than the policer burst and tail drops were experienced, then the same actions should be taken as if policer drops had occurred as described in Section 5.2.2.2.

If a queue limit is applied to a data class queue and the measured drop rate – that is, the ratio of packets and bytes dropped to packets and bytes transmitted – is high (where high is dependent upon the impact on application performance, as discussed in Chapter 1) then this indicates one of the following:

- either that the queue is operating in significant congestion and hence consideration should be given both to increasing the bandwidth assurance offered to that queue, and to reducing the traffic load within the queue
- or that the queue limit is set too low to accommodate the burst profile of the offered traffic load and hence the queue limit may need retuning.

5.2.2.3.2 Monitoring Weighted Tail Drop

Weighted tail drop is sometimes applied to AF class queues to discard a subset of the traffic within the queue preferentially if congestion is experienced within the queue. This can be used to differentiate between traffic that has been differentially marked as in- and out-of-contract (see Chapter 2, Section 2.2.4.2.2). Traffic that is marked out-of-contract is subjected to a lower queue limit and hence is discarded in preference to traffic that is marked in-contract and which is subject to a higher queue limit.

If weighted tail drop is used, then statistics of the number of packets and bytes dropped and transmitted per weighted tail drop profile should be monitored. If the intent of deploying weighted tail drop in this way is to ensure that in-contract traffic has a low loss rate, then the drop rate for the in-contract (i.e. higher) queue limit should be very low, where low is defined by the in-contract SLA for loss. If this is not the case then the indications and rectifying actions that should be taken with respect to the in-contract traffic are the same as for simple tail drop as described in Section 5.2.2.3.1.

When weighted tail drop is used, it would be expected that the drop rate for out-of-contract traffic would be higher than for in-contract traffic. It should be noted, however, that individual flows might have some packets marked as in-contract and others as out-of-contract. Therefore, if the drop rate for out-of-contract packets is too high, the performance of all applications using that queue may be affected and the indications and rectifying actions that should be taken with respect to the in-contract traffic are the same as for simple tail drop as described in Section 5.2.2.3.1.

5.2.2.3.3 Monitoring RED

Random early detection or RED is an active queue management mechanism, which was designed to improve overall throughput for TCP-based applications; RED is described in Chapter 2, Section 2.2.4.2.3. If RED is applied to a data class queue, then the following statistics should be monitored:

- *The number of packets and bytes enqueued*. This is the number of packets subjected to this RED profile that were successfully enqueued, and their respective byte count. Where only a single RED profile is active on the queue, this should be the same as the number of packets and bytes transmitted from the queue.
- *The number of packets and bytes random dropped*. "Random drops" are RED drops which occur when the measured average queue depth is between the configured minimum threshold and maximum threshold for that particular RED profile. If RED is configured

and working correctly then the majority of dropped packets should be random drops. If the drop rate for all RED drops is high relative to the number of packets transmitted then this indicates one of the following:

- either that the queue is operating in significant congestion and hence consideration should be given to increasing the bandwidth assurance offered to that queue, or to reducing the traffic load within the queue
- or the configured minimum and maximum thresholds, or exponential weighting constant for that queue are set too aggressively (i.e. too low) to accommodate the burst profile of the offered traffic load and hence may need retuning (see RED tuning in Chapter 3, Section 3.4).
- or there are applications in that queue, which are not responding to random drops and consideration should be given to whether these applications may be better serviced from a different class queue.

- *The number of packets and bytes force dropped*. Drops that occur when the measured average queue depth is above the configured maximum threshold are referred to as "forced drops." If RED is configured and operating correctly, then random drops should ensure that the average queue limit is below the configured maximum threshold and hence there should be very few forced drops. If there are a significant number of forced drops relative to the total number of RED drops then the possible causes and rectifying actions that should be taken are as described above for high RED drops.

- *Average queue depth*. Polling for the measured RED average queue depth is not essential but provides additional data, which can be used to supplement the RED other statistics. If the measured average queue depth is frequently close to or above the configured RED maximum threshold then this is also an indication that either the queue is operating in significant congestion or the RED configuration is set too aggressively and rectifying actions that should be taken are as for described above for high RED drops.

5.2.2.3.4 Monitoring WRED

Weighted RED (WRED) is commonly applied to AF queues to differentiate between in- and out-of-contract traffic (see Chapter 2, Section 2.2.4.2.4). To achieve this two RED profiles are applied to the same queue and traffic marked out-of-contract is subjected to the more aggressive RED profile (i.e. with lower minimum threshold and maximum threshold) and hence in congestion is discarded in preference to traffic which is marked in-contract and which is subject to a RED profile with higher minimum and maximum thresholds.

Where WRED is used, then the number of packets and bytes dropped and transmitted per RED profile is required. The sum of the packets successfully enqueued across all RED profiles should be the same as the number of packets and bytes transmitted from the queue.

As for weighted tail drop, the intent of deploying WRED in this way is to ensure that in-contract traffic has a low loss rate, then the drop rate for the in-contract RED profile should be very low, where low is determined by the in-contract SLA for loss. If this is not the case then the indications and rectifying actions that should be taken with respect to the in-contract traffic are the same as for RED as described in Section 5.2.2.3.3.

As for weighted tail drop, if the drop rate for out-of-contract packets is too high the performance of all applications using that queue may be affected and the indications and rectifying actions that should be taken with respect to the in-contract traffic are the same as for RED as described in Section 5.2.2.3.3.

5.2.3 System Monitoring

Ideally, all packet drops within a router are handled intelligently by the QOS functions configured on that router, which may be applied outbound on each interface, for example. In practice, however, depending upon how a particular router is architected and implemented, there may be cases where drops can occur on other parts of the system, due to system constraints. If, in the part of the system where these drops occur, there is no understanding of the class of the traffic being dropped, then traffic may be dropped indiscriminately of traffic class.

Clearly, systems should be designed to try to minimize the occurrence of such indiscriminate traffic drops; however, in cases where they can occur it is essential to monitor for them because they can provide an indication of serious system issues that can potentially affect the SLAs across all traffic classes.

The system drops that can occur will depend upon the implementation of a particular device; however, some of the most common types of system drops are as described below:

- *No buffer drops*. In Chapter 2, Section 2.2.4.2 we explained the difference between buffers and queues. Where buffer memory is shared between queues in a system, there may be cases where a packet arrives and there is insufficient packet buffer memory available to store the packet, in which case there is no alternative but to drop the packet. Such "no buffer drops" should be an exception in any well-designed system, rather than the norm; however, the occurrence of "no buffer drops" can be exacerbated in a heavily congested system if RED and queue limit settings are excessively high.
- *Input drops/ignores*. Input drops, which are also known as ignores, occur when there are insufficient packet buffers to store a packet even before a routing or switching decision can be made. Input drops are a symptom of an oversubscribed system, e.g. where the packets per second forwarding performance of the system or component is being exceeded.

System drops such as no buffer drops and input drops will generally need to be monitored using vendor-specific MIBs, as system specific statistics are not available from the Diffserv MIB. Due to the impact they can have on the SLAs of all traffic classes, the occurrence of any such system drops should trigger further investigation to determine the cause of the drops and to prevent a re-occurrence.

5.2.4 Core Traffic Matrix

The core traffic demand matrix is the matrix of ingress to egress traffic demands across the core network. Traffic matrices can be measured

or estimated from statistics gathered using passive monitoring techniques. The main benefit of the core traffic matrix is for core network capacity planning, in that it can be used to predict the impact that demand growths can have, and in the simulation of "what-if" scenarios, to predict the impact that the failure of core network elements can have on the utilization of the rest of the network. There are a number of techniques for gathering the core traffic matrix; the application of these techniques and their use in capacity planning is discussed in more detail in Chapter 6.

5.3 Active Network Monitoring

Ideally, it would be possible to measure the delay, jitter, loss, and throughput that actual traffic experiences as it traverses a network. In some cases, it may be possible to retrieve this information from the application end-systems. Where the real-time protocol (RTP) [RFC 3550] is used, for example, the timestamp and sequence number information in the RTP header could be used to determine the delay, jitter, and loss of the received stream at the receiving end-system. This is not generally possible in practice, however, due to the following reasons: many applications do not use RTP; retrieving such statistics from all application end-systems would be unscalable; the end-systems may not be under the same administrative responsibility as the network elements. Further, to provide this information at the network level would require the network elements to identify uniquely a packet at every single hop and to timestamp it very accurately, which is not possible in practice.

Network level active network monitoring is an alternative approach, which is more generally applicable. Active monitoring uses specially tailored synthetic traffic test streams comprising "probe" packets – that aim to emulate actual network traffic – which are sent between active monitoring devices in order to characterize network performance and thereby infer the performance experienced by the emulated traffic. In Diffserv deployments, active monitoring can be used to measure the performance of all classes of traffic.

Active network monitoring requires the deployment of an active SLA probing system, supporting capabilities such as those defined by the IP performance metrics (IPPM) working group [IPPM] within the IETF. In such a system, active monitoring agents are deployed (potentially on existing network elements) and test streams are sent between the agents. The agents measure the received streams and typically keep a statistical analysis of the measured results, which can then be retrieved periodically from the active measuring devices, via SNMP for example. In addition, the active monitoring devices may proactively issue traps, if defined thresholds for the measured performance of the test streams are exceeded.

In deploying an active monitoring system, consideration should be given to the following questions, which are addressed in the proceeding sections:

- What test streams should be used?
- How often should testing be undertaken and for how long?
- What metrics should be measured for the received streams?
- Where should active monitoring devices be deployed and what paths should the active monitoring streams monitor?

To avoid confusion, we differentiate between the active monitoring traffic, i.e. the active measurement probes, and the monitored traffic, the performance of which the active monitoring traffic is trying to estimate.

5.3.1 Test Stream Parameters

The characteristics of the test stream will affect the characteristics of the network that the test stream will measure. These measured test stream results are only useful if they are in some way representative of the performance experienced by the monitored application or traffic class. This gives rise to the question of what test stream parameters are required to ensure that the measured characteristics of the active

measurement stream accurately reflects the characteristics (e.g. delay, jitter, loss, packet re-ordering, and availability) of the traffic from the monitored application or traffic class? The answer to this question is still the subject of further study; however, the following sections consider the key parameters to define for an active measurement stream. It is noted that the term "accurately" in this context does not mean that the difference between measured test stream characteristics and the characteristics of the traffic must be small, but is does mean that the two results must be highly correlated, such that it is possible to predict the measured traffic performance from test stream measurements with high fidelity.

5.3.1.1 Packet Size

There are two general approaches to the setting of packet sizes for active monitoring probes:

- *Same size as monitored traffic*. One approach is to use probe packets that are the same size as the packets of the monitored traffic. There are two justifications for this approach:
  - As discussed in Chapter 3, Section 3.2.2.4.1, packet size has a more significant impact on serialization delay with lower-speed links, hence using packets the same size as the packets of the monitored traffic will potentially provide a more accurate measurement of delay. It is noted, however, that if the link speeds on the path are known, adjustments can be made to take differences in serialization delay between monitoring and monitored traffic into account.
  - Packets larger or smaller than the packets of the monitored traffic may experience a different loss than the monitored traffic itself; if congestion occurs in part of the network, as the queue depth increases a smaller packet is more likely to be enqueued than a larger one.
- *Small sized packets*. An alternative approach is to use small sized packets, for two reasons:
  - In environments where there are very low speed links, such as in some mobile environments where the bandwidth is scarce

and expensive, the smallest possible sized packets are used for bandwidth economy.

- Where a high rate of test packets is needed to achieve measurement accuracy, the use of larger packets may have a significant impact on the traffic being measured. In this case, small sized packets are used to minimize the potential impact.

There is no industry consensus on which approach is best; however, we note the following conclusions from research in this area:

- From simulations studying the effectiveness of active SLA monitoring on a 2 Mbps link, [HILL] concludes that: *"The accuracy of the probes is not really affected by probe size. Both sizes [41-bytes and 850-bytes] show equally good correlation coefficients for delay and loss."* He also concludes that larger sized probes have significantly greater impact on the delay and jitter of the traffic whose performance the test stream is trying to estimate. Therefore, he recommends that probes should be small such that the active monitoring traffic has less impact on the other traffic.
- [SOLANGE] also found no evidence that packet size affected the measurements of packet loss.

In practice, however, most deployments use the same packet size for test streams that are used by the applications they are emulating. It is further noted that on higher speed links, where the impact of serialization delay is less, and the traffic is more highly aggregated, the impact of probe packet sizing is likely to be less significant.

5.3.1.2 Sampling Strategy

The probe sampling strategy determines the distribution of the delay separating consecutive test packets. There are three general probe sampling strategies that may be used:

- *Periodic sampling*. Periodic sampling consists of sending probes at equally spaced intervals, i.e. every n seconds. Opponents of this approach argue that one cannot fully characterize the network

behavior by "sampling" at regular intervals. There might be some cases where unforeseen synchronization between the sending of probe packets, or possibly other network events, could potentially lead to inaccuracies. This kind of phenomenon, although theoretically possible, is rarely seen in practice. [RFC 3432] describes a methodology for network performance measurement with periodic streams.

- *Random sampling*. Random sampling consists of sending a probe at random intervals, where the interval is regulated by a probability density function. Most commonly, a Poisson process is used to distribute the probe packets, meaning that the interarrivals between probing packets should be independent and exponentially distributed with the same mean. This approach provides an unbiased estimate of the desired time average, which is a property referred to as "*Poisson Arrivals See Time Average*" or PASTA [WOLFF]. This approach is suggested by the IETF, where [RFC 2679] and [RFC 2680] standardize metrics based on Poisson sampling processes. Consequently, the IPPM working group has made the support of Poisson streams mandatory for their one-way active measurement protocol (OWAMP) [RFC4656]. The counterpoint to the use of a variable inter-packet delay is based upon the fact that most of the real world applications, which require tightly bounded delay and jitter and hence which are often a focus of active monitoring, do not have a Poisson distributed interpacket delay. Voice and video applications, for instance, commonly have streams with a constant interpacket delay and so why attempt to measure the performance of these applications on the network with something other than a stream that emulates the application?

  A variation on random sampling is to divide the total sampling period into fixed time intervals and then to send a probe within each interval with a random offset from the start of the interval, where the offset is regulated by a probability density function. The benefit of this approach is that the sample size within a defined number of intervals is known. This approach is referred to as stratified random sampling, where each interval represents a stratum.

- *Batch sampling*. With batch sampling, rather than sending individual probe packets, probes are sent in bursts, where the spacing between bursts may be periodic or random.

These different sampling regimes are illustrated in Figure 5.1.

Several works have attempted to compare both approaches to find if there is a tangible difference between the methods:

- In "*Poisson versus Periodic Path Probing (or, Does PASTA Matter?*" [TARIQ], the authors conclude that: "*The experimental results in this paper indicate that there may not be a significant difference between Poisson and Periodic probing, at least in the context of real Internet measurements*."
- [SOLANGE] conclude that "*... for similar probing rates and coloring, a periodic pattern leads to a slightly better [delay] and [loss] match than Poisson patterns*."
- [HILL] concludes that both random and periodic sampling provide acceptable accuracy for measuring delay and loss for VoIP and TCP, but also concludes that neither approach provides acceptable accuracy for measuring jitter.
- [HILL] suggests that batch sampling be considered to improve jitter measurement (Section 5.3.2.2), while [SOLANGE], who investigated the use of batch sampling, conclude that: "*... when compared [to Poisson] with similar rates, [batch sampling] produced better estimates of delay, jitter and loss*."

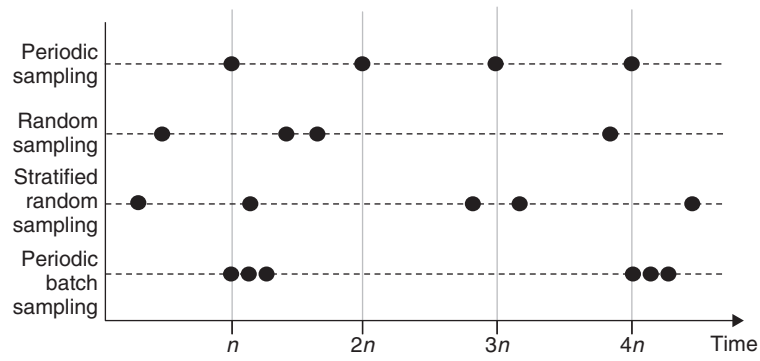

Figure 5.1 Active monitoring sampling strategies

In practice, however, periodic test streams with a constant inter-packet delay are most commonly used because this approach is easier to implement and interpret and because it most closely emulates the applications that the active monitoring is targeting. In recognition of this, [RFC 3432] states: "*Poisson sampling produces an unbiased sample for the various IP performance metrics, yet there are situations where alternative sampling methods are advantageous.... Predictability and some forms of synchronization can be mitigated through the use of random start times and limited stream duration over a test interval*."

5.3.1.3 Test Rate

The test rate determines the amount of packets sent within the test duration, and consequently, it affects the perturbation introduced by the measurement stream on the actual network traffic. For instance, sending a large amount of test traffic over a path with small bandwidth may potentially interfere with the delivery of the actual measured traffic stream that the active monitoring is trying to monitor. Such an effect would clearly invalidate the measured results. Conversely, if the test rate is too low, the measured characteristics of the test stream may not reflect the characteristics of the measured traffic stream itself. Therefore, determining an appropriate test rate is a balance between testing with a high enough rate that the measured result is an accurate reflection of the measured traffic stream, while ensuring that the measuring stream does not interfere with the measured traffic stream significantly, such that it affects the very characteristics it is trying to measure.

There is no general answer to the question of what test rate to use, but rather it depends upon the characteristics of the application or class being monitored.

- Based upon simulations of a 2 Mbps bottleneck link, using both periodic and random sampling, [HILL] concludes that "*both delay and loss can be measured accurately (taking into account the systematic [underestimation]) at around a probe rate of 10 probes per second." He also notes that "... higher probe rates report more accurate traffic results*

for [delay and loss]," but suggests using the following rates in order to measure delay and loss (for jitter, see Section 5.3.2.2):

- "*For TCP traffic the optimum strategy was to send ... [probes] at a mean rate of ten probes per second*," i.e. approximately 0.2% of the link rate with probe 41-byte packets.
- "*For VoIP traffic, the optimum strategy was to send in probes at a rate of 5 probes per second*," i.e. approximately 0.1% of the link rate with probe 41-byte packets.

- [SOLANGE] conclude that for EF traffic, using Poisson random sampling, as low as 2 probes per second (pps) are effective for measuring delay and jitter, based upon simulations of a 34 Mbps bottleneck link, i.e. approximately 0.01% of the link rate with 100-byte probe packets.

  However, they report that for AF classes subject to loss and more significant jitter, at these probing rates using random sampling there is significant overestimation of jitter (see Section 5.3.2.2) and some loss events are missed altogether. They also note that while higher probing rates improve the measurement accuracy of delay and jitter, even at probing rates of 192 pps, some loss events are missed. Hence, although higher probing rates provide an improvement, due to the overhead incurred and the failure of loss estimation at higher probe rates, they suggest that the use of an alternative sampling scheme, such as batch sampling, should be considered instead.

5.3.1.4 Test Duration and Frequency

The test duration defines how long an active measurement test case will run. The test frequency determines how many times the test will repeat within a specified time window.

Assuming a given test traffic rate, the test duration and frequency need to be high enough that the measured result is an accurate reflection of the measured traffic stream. The lower the (*duration x frequency*) in any given time window, the greater the probability that significant events will be missed, as illustrated in Figure 5.2.

If the active monitoring devices do not keep the raw data of the individual probes, but rather keep a statistical representation of the

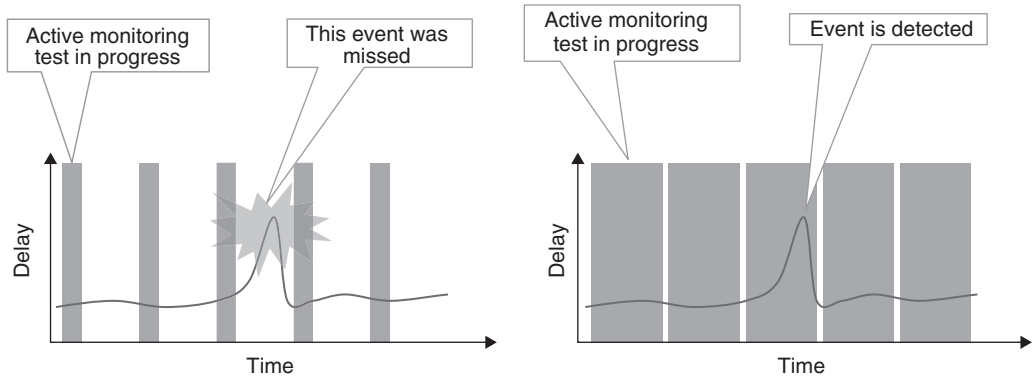

Figure 5.2 Impact of test (*frequency* × *duration*)

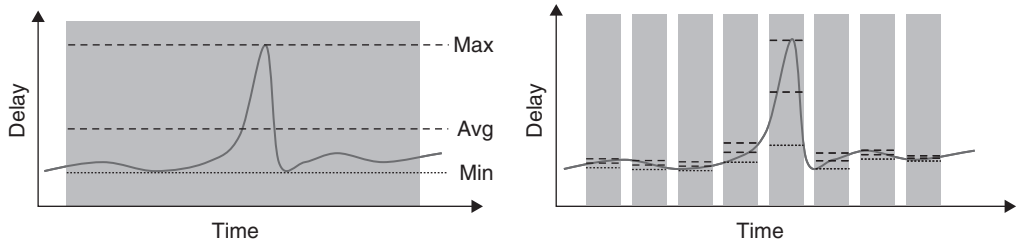

Figure 5.3 Impact of test duration

results over the test duration, as is commonly the case, then assuming a given test traffic rate the test duration will implicitly impact the measured statistics, as shown in Figure 5.3.

Similarly to the discussion on the passive monitoring polling interval in Section 5.2.1, longer active monitoring test durations may be acceptable for trending purposes; however, shorter durations are preferred where more granular measurements are required, although this has to be balanced against the increased polling load. A possible polling scheme could be as follows:

- For troubleshooting, proactive measurement and SLA reporting, a network segment could be measured constantly with a test duration of 2 minutes.

- For trending, it may be more appropriate to measure for one hour every day, during the peak hour previously determined by the more granular measurements. Longer duration measurements make the comparison between days, months and years easier and more statistically relevant.

5.3.1.5 Protocols, Ports, and Applications

In order to ensure that the network characteristics determined by a measuring traffic stream are representative of the traffic stream they are measuring, it is important that the measuring stream is classified the same as the target stream along the end-to-end network path. If Diffserv is deployed the network performance experienced by applications will depend upon how the traffic is classified within the network; if measurement probes are classified differently than the emulated stream in any part of the network, they may experience different delay, jitter, and loss, and hence will not provide representative results.

Where simple classification is used, the probe packets should share the same marking (be it DSCP, IP precedence or even 802.1p based) as the target stream, but need not necessarily share the same IP addressing or protocol as the target stream.

Where complex classification is used (see Chapter 2, Section 2.2.1.2), the criteria used for complex classification should produce the same results for the measuring test stream as for the measured application. If, for example, VoIP traffic is classified by a combination of identifying UDP packets, with even UDP port numbers (e.g. representing RTP data) and from a specific source IP address, then headers of the probe packets should be such that they also match these criteria. If the target traffic stream is TCP-based and complex classification is used, the IP protocol number of the probe packets may also need to be set to 6 to indicate that the packets are TCP.

Where Diffserv is deployed with AF classes supporting the concept of in- and out-of-contract as described in Chapter 2, Section 2.3.4.2.2, the in-contract traffic has a lower probability of packet loss than the traffic. Hence, if monitoring of the in-contract SLA is required, it is

important that any policers used to mark traffic as in- or out-of-contract do not re-mark the in-contract probes, else they may be wrongly classified and may not be report the in-contract SLA correctly.

Some probing systems may attempt to characterize application as well as network performance. For example, a probe may record the response time of a DNS query to a particular DNS server or an HTTP GET of a specific web page. In these cases, the results will capture multiple components such as session establishment, end-system processing, sending, and receiving multiple packets between the client and the server, and closing the connection. This kind of application-oriented operation may be useful to measure the user experience, but gives no visibility of the performance of the individual components that make up the measured response.

5.3.2 Active Measurement Metrics

The SLA metrics that are important for defining IP service performance are described in detail in Chapter 1, Section 1.2. Once the appropriate test stream for your particular application has been identified, consideration needs to be given to which metrics to measure, how they are measured and to how the resultant measurements should be interpreted. Multiple metrics can be determined from a single test stream.

5.3.2.1 Delay

Delay can be quantified either as one-way delay, or as round trip delay (round trip time or RTT). Measurement of RTT requires that probes are sent from a sending active monitoring agent to a responder and then back to the sender. In this case the RTT can be determined if the sender timestamps the probes when it sends them (the timestamp is carried in the data of the probe packet) and subtracts this value from the corresponding timestamp when it receives the probe response. Measurement of one-way delay requires that the sender and receiver's local time clocks are synchronized such that the one-way delay can be

determined at the receiver, if the receiver also timestamps the probe packets on receipt; the difference between the sending timestamp and receiving timestamp is the one-way delay. Ensuring synchronization between sender and receiver with acceptable accuracy poses challenges; this is discussed in more detail in Section 5.3.3.4. RTT is easier to implement and measure than one-way delay, and may provide sufficiently measurement utility for many applications.

For applications such as VoIP or interactive video conferencing, the important delay metric when considering the engineering of the network is the one-way end-to-end delay in each direction from end-system to end-system. From a monitoring perspective, however, it may be acceptable to monitor the RTT between the end-systems as from a service perspective, it may not matter in which direction excess delay is experienced; if excess is experienced at all, then the service will be impacted. If SLA violations for delay occur, however, RTT hides the detail of in which direction the issue causing the violation occurred. Hence, measurement of one-way delay may be more useful for network troubleshooting.

Delay can provide a number for important indicators of network performance. Most active monitoring end-systems will analyze the received probes and present statistics on the resulting data set, but which statistics are important with respect to delay measurement?

- *Minimum delay*. The minimum network delay is the network delay "baseline," providing an indication of the delay that traffic will experience when the path from source to destination is lightly loaded. This will largely be composed of propagation delay, switching delay, and serialization delay. Delay values above the minimum provide an indication of the congestion experienced along the path. Considering the percentile delay for a low percentile (e.g. 0.1 percentile), will provide an indication of the minimum delay experienced while discounting outliers, e.g. spuriously low results due to measurement system glitches.
- *High percentile delay*. The maximum delay across a network may not be interesting if it is caused by on a very small percentage of

outliers; considering the percentile delay for a high percentile (e.g. 99.9 percentile), will provide an indication of the maximum delay experienced while discounting outliers.

- *Threshold exceeded count*. For applications which have a stringent requirement on delay, it may be useful to count the number of probe packets out of the total which experienced a delay in excess of a defined threshold, set to indicate when a packet arrived too late to be useful.
- *Average delay*. The average delay may be interesting for trending purposes, but for purposes of comparison should be recorded together with the standard deviation of the sample; higher than normal standard deviations may be indicative of spurious issues rather than of a trend.

5.3.2.2 Delay-jitter

Delay-jitter (which is also known as jitter), as described in Chapter 1, is generally considered to be the variation of the one-way delay for two consecutive packets. Measurement of one-way delay requires time-stamping at both sending and receiving devices, which requires synchronization between sender and receiver; this is difficult for the reasons discussed in Section 5.3.3.4. Fortunately, to calculate jitter there is no need to know the individual one-way delays: instead, this can be calculated from the difference between timestamps taken on single devices. No operation need be performed between timestamps on two different devices, which makes measurement of one-way delay-jitter simpler than measurement of one-way delay. Consider that $T_s[n]$ is the time when the packet n was sent, and $T_r[n]$ is the time when the packet n was received; the one-way delay of this packet is denoted as D[n]. Then, the jitter J between packets n and $n + 1$ can therefore be calculated as:

$$\begin{aligned} J\,[n, n+1] &= D[n+1] - D[n] \\ &= (T_r[n+1] - T_s[n+1]) - (T_r[n] - T_s[n]) \\ &= (T_r[n+1] - T_r[n]) - (T_s[n+1] - T_s[n]) \end{aligned}$$

The most important statistics to report with respect to jitter are high percentile jitter, threshold exceeded count, and average jitter. It is noted that the higher the rate of the traffic stream, the lower will be the measured jitter, as illustrated in Figures 5.4 and 5.5, which show

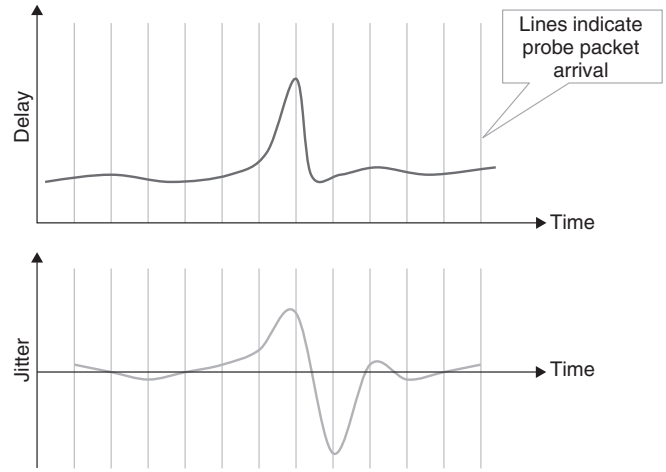

Figure 5.4 Lower rate – higher measured jitter

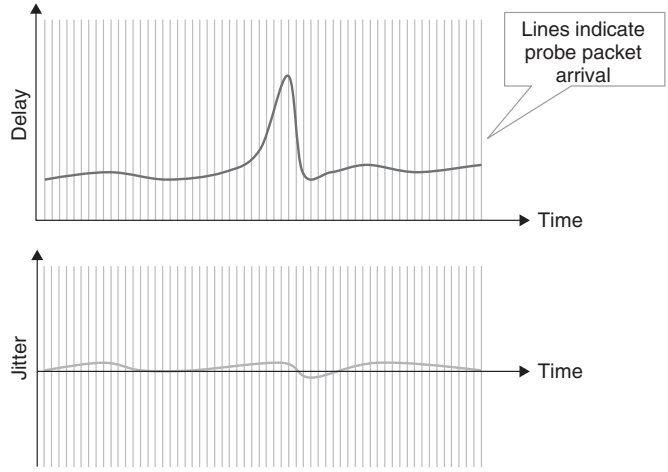

Figure 5.5 Higher rate – lower measured jitter

the variation in queuing delay within a queue, and the resulting jitter measured by probes within that queue, for different probe rates.

Hence, measurements streams at rates below that of the measured traffic will likely report higher jitter than that actually experienced by the traffic itself. This is supported by the findings in [HILL], who suggests a batch sampling strategy to overcome this problem, and [SOLANGE], who concluded that batch sampling produced a better estimate of jitter than random Poisson sampling.

5.3.2.3 Packet Loss

In order to determine packet loss there needs to be a way to distinguish between a lost packet and a packet with a large but finite delay. In practice, depending upon application and end-system implementations, packets delayed beyond a certain threshold will be of no use and hence can be considered lost; acceptable delay thresholds for different applications are discussed in Chapter 1. The loss of an individual packet is a binary measure, however, SLAs for loss are generally defined statistically and hence loss commitments need to be provided over a defined time interval.

The measure of the percentage of packets dropped may be useful for trending purposes; however, it does not say anything about how those packets were dropped. Hence, it is not possible to understand the potential impact on applications from this measure alone. [RFC 3357] introduces some additional metrics, which describe loss patterns and can be used to analyze the possible impact on applications:

- *Loss period*. The loss period defines the frequency and length (loss burst) of loss once it starts
- *Loss distance*. The loss distance defines the spacing between the loss periods.

It is therefore recommended that the loss period and loss distance are measured and compared against application-specific thresholds indicating where the measured loss will unacceptably affect application performance. The impact of packet loss on different applications is discussed in Chapter 1.

5.3.2.4 Bandwidth and Throughput

Application throughput is dependent upon many factors, which can vary widely depending upon end-system implementations and traffic profiles. Hence, active monitoring systems generally do not attempt to characterize application throughput explicitly. Rather, application throughput is generally inferred. Considering TCP for example, TCP performance can be inferred from the measured network RTT and packet loss rate, as discussed in Chapter 1, Sections 1.3.3.1.5–1.3.3.1.7. Active monitoring systems may send packets which appear to be TCP packets (i.e. use IP protocol 6), but they need not – and commonly do not – implement a TCP stack, i.e. the transmission of the packets is not controlled by TCP's flow and congestion control mechanisms.

5.3.2.5 Re-ordering

IP does not guarantee that packets are delivered in the order in which they were sent; as discussed in Chapter 1, packet re-ordering can have an adverse impact on the performance of many applications.

Re-ordering within an active monitoring test stream is determined by adding sequence numbers to the packets transmitted in the stream and then comparing the sequence numbers of the received packets with the order in which they are received. If a packet arrives with a sequence number smaller than its predecessor's then that packet 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].

5.3.2.6 Availability

Availability for IP services is generally defined either as network availability or as service availability, as described in Chapter 1, Section 1.2.6.

- *Network availability*. Bidirectional network availability or connectivity between two active monitoring devices can be determined using probes sent from a sender to a responder and then back to

the sender; for each response successfully received the network is considered available and for each not received the network is considered unavailable. As with packet loss, a delay threshold needs to be defined after which a response is considered "lost."

- *Service availability*. Service availability is a compound metric defining when a service is available between a specified ingress point and a specified egress point within the bounds of the committed SLA metrics for the service, e.g. delay, jitter, and loss. This is discussed in more detail in Chapter 1, Section 1.2.6.2.

5.3.2.7 Quality of Experience

Active monitoring end-systems do not normally implement the full end-system behavior for the applications they are trying to measure. Some active monitoring devices, however, will interpret the metrics of a received stream in order to provide an objective measure of the quality of the application performance that will be experienced from the perspective of the end-users, which is also known as the user "quality of experience" or QOE. The most common QOE measure is the "mean opinion score" or MOS, which provides a subjective numeric measure of the QOE of a voice call. ITU standard [G.107] uses a number of measured network parameters to determine a "rating factor," which can be transformed to give estimates of the MOS for calls, which use that network service. QOE is discussed in more detail in Chapter 1, Section 1.2.7.

5.3.3 Deployment Considerations

5.3.3.1 External versus Embedded Agents

An active measurement system uses active monitoring agents to send and receive probe packets. These agents may be implemented in dedicated active monitoring devices or alternatively may be embedded into existing network devices:

- *External agents*. External agents are implemented in dedicated active monitoring devices, which may either use specialized hardware or

dedicated but off the shelf computers running active monitoring software. This approach decouples the forwarding path (routers and switches) from the measurement devices; the dedicated active monitoring devices appear as customers connected to the network and hence this approach may provide the closest view to the end-customer experience. The use of dedicated devices, however, requires addition network equipment, which incurs additional cost in terms of capital expenditure, accommodation, power, management, and maintenance. Hence, for end-user or small branch office locations the use of dedicated active monitoring devices is generally not viable.

- *Embedded agents*. Some network hardware vendors implement software active monitoring agents embedded in products, which may be network devices such as routers or switches or could be end-systems such as IP phones. The use of embedded agents in devices which are already on the data switching path allows the installed base of network equipment to be leveraged, enabling the rapid roll-out of an active SLA monitoring system without requiring the deployment of new network equipment.

5.3.3.2 Active Monitoring Topologies

When deploying an active monitoring system, a key question is where to deploy the active monitoring devices, be they external or embedded agents. In general, the measurements from active monitoring should represent the application's experience, and hence the active monitoring devices should be as close to the application end-system as possible. In all deployments, however, there are constraints, which limit the location of such devices; there may be parts of the network that are not under the control of the measuring organization, for example. In large deployments, scalability of the active monitoring system is an additional consideration.

The selection of the active monitoring topology depends upon these constraints. Consider the example physical network topology shown in Figure 5.6. A number of different active SLA monitoring topologies – where the active SLA monitoring topology is defined by

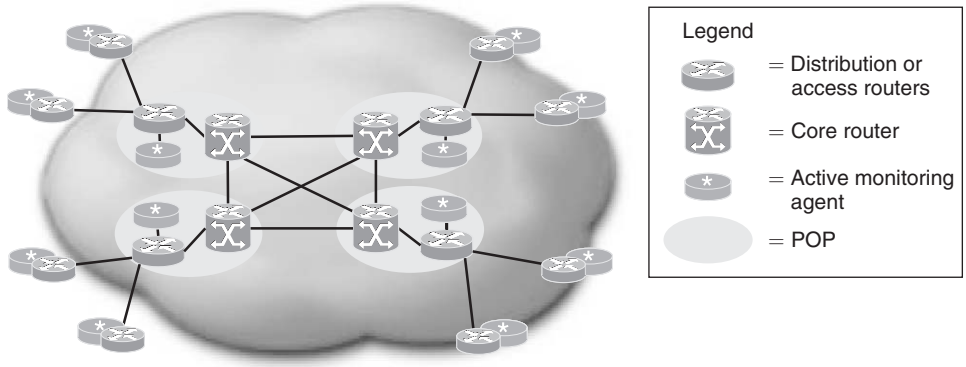

Figure 5.6 Example physical network topology

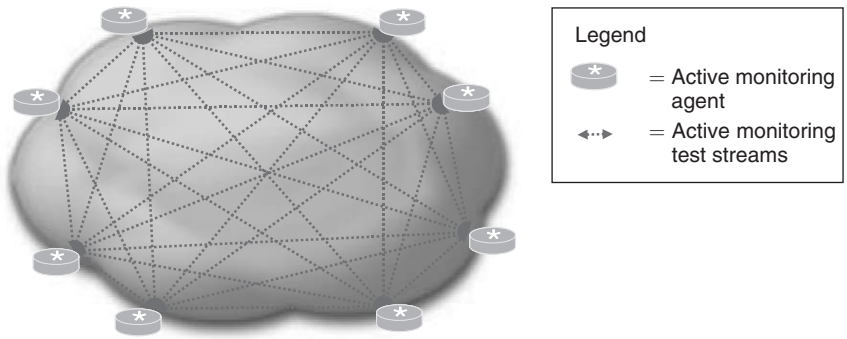

Figure 5.7 Full mesh active monitoring topology

the sources and destinations of the active monitoring test streams – can be overlaid on this physical topology:

- *Full mesh*. A full mesh requires probes from every active monitoring location to every other active monitoring location, as shown in Figure 5.7. This approach is the most accurate because it measures end-to-end paths between all locations and gives full network coverage.

  In practice, however, the full mesh approach does not scale well; as the number of active monitoring nodes (n) increases, the

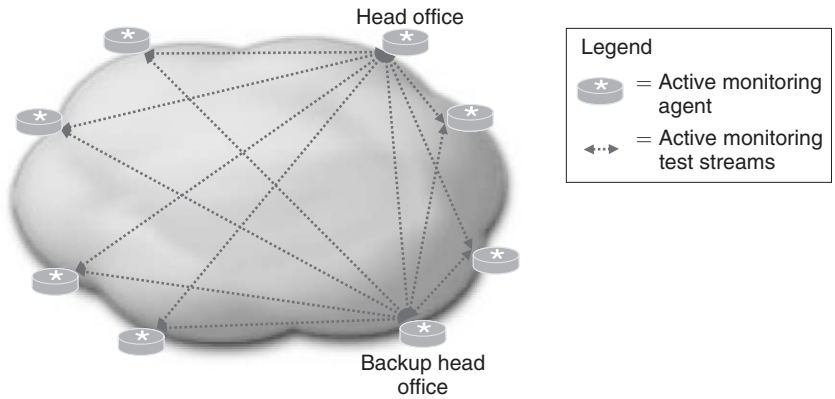

Figure 5.8 Partial mesh active monitoring topology

number of bidirectional active monitoring test streams required to interconnect them is $n * (n - 1)/2$, which increases more than linearly with the number of nodes. Beyond a few nodes, the full mesh approach may result in a configuration burden, the test streams may use a significant amount of bandwidth and the retrieval of the measurement data from all nodes may incur significant management system overhead. For these reasons, a full mesh active monitoring topology is only used where there are a limited number of sites to be monitored.

- *Partial mesh*. A partial mesh involves running a mesh of test streams on a subset of the topology. For example, this could be a hub and spoke active monitoring topology in networks where remote sites (the spokes) only communicate with the head offices (the hubs), as shown in Figure 5.8.

  A partial mesh reduces the number of test streams required and provides end-to-end monitoring between a subset of locations. In a hub and spoke topology, if round-trip active monitoring is used, the hub sites may be configured as the active monitoring probe senders, with the spoke sites acting as responders; in this case, the active monitoring measurement data need only be retrieved from the hub sites.

- *Hierarchical mesh*. In networks with any-to-any communication between sites, a full mesh may be unscalable, while a partial mesh may not provide sufficient network coverage. In these cases, a hierarchical mesh may be used; with a hierarchical mesh, the active monitoring is segmented. In a typical deployment, centralized active measurement devices are located in each point of presence (POP) and test streams are run from each POP to their connected remote sites in a hub and spoke active monitoring topology. Test streams are then run in a full mesh from each POP to every other POP, as shown in Figure 5.9.

  This approach facilitates the scaling of a network-wide active monitoring system and hence it is commonly used in practice; it significantly reduces the number of test streams required compared to a full mesh, while providing full network coverage and being relatively easy to manage. If the POP active monitoring devices are configured as senders for round-trip probes, with their respective remote sites monitoring devices acting as responders, then the active monitoring measurement data need only be retrieved from the central sites and there is no need to access the remote sites.

  This approach gives segmented measurements for the access links and across the core network and maps well to the concept of

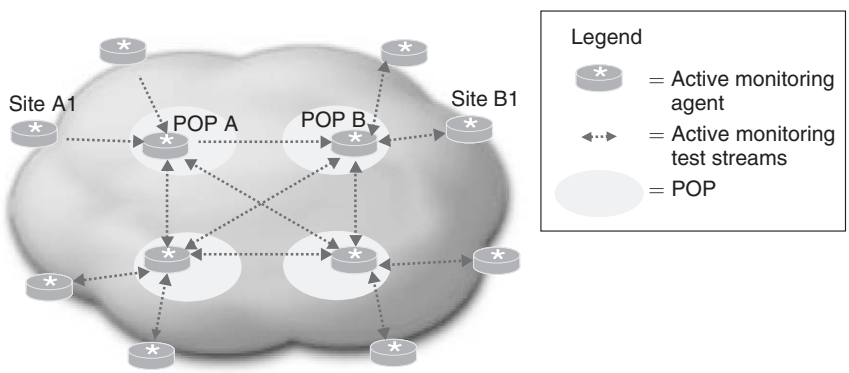

Figure 5.9 Hierarchical mesh active monitoring topology

a segmented SLA discussed in Chapter 1, Section 1.4.1. The disadvantage of this approach is that it does not provide end-to-end monitoring. Hence, if measurements between two sites A1 and B1 were required, they would need to be statistically estimated by combining, where possible, the measured results for each segment in the end-to-end path, i.e. from site A1 to POP A, from POP A to POP B, and from POP B to site B1. For example, it is possible to estimate the average (or a specific percentile) end-to-end delay by summing the average (or specific percentile) measured delay for each segment. To estimate the end-to-end packet loss probability, if the probability of packet loss on segment x is given by P_x, then the end-to-end packet loss probability (P) across n segments is:

$$P = 1 - [(1 - P_1) \times (1 - P_2) \times ... \times (1 - P_n)]$$

It is not, however, possible to estimate end-to-end jitter from the measured jitter of the segments on the end-to-end path because the measured jitter in IP networks is not statistically additive in practice (see Chapter 6, Section 6.1.3). Where a measure of end-to-end jitter is required, end-to-end monitoring should be selectively deployed.

5.3.3.3 Measuring Equal Cost Multiple Paths

Many networks have multiple paths between different parts of the network, for reasons of both resilience and capacity provision. Interior Gateway routing Protocols (IGPs) such as OSPF [RFC 2328] and ISIS [RFC1142] determine which paths will be used between any two points in the network by choosing whichever path has the least total cost, where the path cost is calculated by summing the individual metrics (which express the preference of a link) of the links along the path. If there is more than one least cost path, then the routing protocol will potentially distribute the traffic between the two points across all of those paths. The algorithms that balance the load across the paths are generally referred to as equal cost multi-path (ECMP) algorithms. ECMP algorithms are generally proprietary to each vendor. Different vendors will use different criteria to determine which

path will be used for a particular packet, although a common implementation is to perform a hash function using inputs including fields within the packet header, such as source IP address, destination IP address, protocol number, source UDP/TCP port, and destination UDP/TCP port.

ECMP poses a significant issue for active monitoring for which there is no ideal answer; a single measurement can only use one of the many possible paths and not all of them. There are a number of potential resolutions to this issue; however, none of them is a panacea that will provide a solution in all circumstances. It may be possible to vary the source and destination IP addresses and UDP/TCP port numbers of sent probes in order to try to use more than one of the paths. In practice, however, ECMP algorithms can be difficult to predict (some also use a random seed as an input to the hash), hence it may not be possible to guarantee that all paths are being tested. Alternatively, if the test is run from the load-balancing router itself, then it may be possible to force probe packets via each of the load-balancing interfaces in turn; however, this will not guarantee that response probe packets use all return paths also.

5.3.3.4 Clock Synchronization

To achieve highly accurate one-way delay measurements, the clocks on all the network elements participating in the test must be synchronized; any synchronization error will result in an error in the measured one-way delay. Network devices maintain local time using on board clocks, which provide time to the device operating system. There are a number of potential ways that the local clocks on network devices can be synchronized.

The most accurate way to synchronize clocks on network devices is to synchronize each device with an accurate "stratum 1"[1] external clock source such as a GPS clock or radio clock. This is, however, an expensive approach and while it may be viable for devices within the core of the network, it would not be viable for end-user or small branch office locations.

An alternative approach is to distribute stratum 1 time using a protocol, such as the network time protocol (NTP) [RFC 1305]. NTP

synchronizes clocks between network devices by exchanging time-stamped messages between a server and its clients. NTP seeks long-term accuracy at the expense of the short-term accuracy; it will, for instance, slow or accelerate the internal clock (or add/subtract time quanta) to adjust the local clock progressively to what it believes is the true time. If measurements are taking place during those adjustments, strange results like negative delay might be observed. NTP can usually maintain time to within 10 ms in wide area networks; this does not generally provide a sufficient level of accuracy for those applications with tight delay bound requirements, which require one-way delay monitoring such as VoIP and video streaming. In local area networks, under good conditions, NTP can usually maintain time to 1 ms or better, which may be sufficient for active monitoring purposes.

Due to the constraints and costs of interdevice clock synchronization, a common deployment model is to distribute time from a stratum 1 clock source to all the devices within a point of presence (POP) using a separate network (commonly the management network), to ensure synchronization via NTP to within 1 ms or better. This enables the measurement of one-way delay between POPs. Synchronization of access routers via NTP is generally not accurate enough and the use of stratum 1 clock sources in these locations is generally not viable, hence SLA reporting of the access links from POP to access router is commonly reported as RTT rather than one-way delay.

References[2]

[G.107] ITU-T Recommendation G.107, The E-model, a computational model for use in transmission planning, International Telecommunication Union, Geneva, Switzerland, Feb. 2003

[HILL] J. Hill, Assessing the Accuracy of Active Probes for Determining Network Delay, Jitter and Loss, MSc Thesis in High Performance Computing, The University of Edinburgh, 2002. Available at: http://www.epcc.ed.ac.uk/msc/dissertations/dissertations-0102/jhill.pdf

[IPPM] http://www.ietf.org/html.charters/ippm-charter.html

[RFC1142] D. Oran, Ed., OSI IS-IS Intra-domain Routing Protocol, *RFC 1142*, February 1999 [republication of ISO DP 10589]

[RFC1157] J. Case et al., A Simple Network Management Protocol (SNMP), *RFC 1157*, May 1990

[RFC1305] D. Mills, Network Time Protocol (Version 3) Specification, Implementation and Analysis, *RFC 1305*, March 1992

[RFC2328] J. Moy, OSPF Version 2, *RFC 2328*, April 1998

[RFC2679] G. Almes, S. Kalidindi, and M. Zekauskas, A One-way Delay Metric for IPPM, *RFC 2679*, September 1999

[RFC2680] G. Almes, S. Kalidindi, and M. Zekauskas, A One-way Packet Loss Metric for IPPM, *RFC 2680*, September 1999

[RFC2697] J. Heinanen, R. Guerin, A Single Rate Three Color Marker, *RFC 2697*, September 1999

[RFC2698] J. Heinanen, R. Guerin, A Two Rate Three Color Marker, *RFC 2698*, September 1999

[RFC3289] Baker, K. Chan, A. Smith, Management Information Base for the Differentiated Services Architecture, *RFC 3289*, May 2002

[RFC3357] R. Koodli, R. Ravikanth, One-way Loss Pattern Sample Metrics, *RFC 3357*, August 2002

[RFC3432] G. Grotefeld, A. Morton, Network performance measurement with periodic streams, *RFC 3432*, November 2002

[RFC3550] H. Schulzrinne, RTP: A Transport Protocol for Real-Time Applications, *RFC 3550*, July 2003

[RFC4656] S. Shalunov et al. A One-way Active Measurement Protocol (OWAMP), *RFC 4656*, September 2006

[RFC4737] A. Morton et al. Packet Re-ordering Metric for IPPM, *RFC4737*, November 2006

[SOLANGE] Solange R. Lima, Paulo M. Carvalho, and Vasco L. Freitas, Measuring QoS in class-based IP networks using multipurpose

colored probing patterns, *Proceedings of SPIE*, Volume 5598, September 2004, pp. 171–182

[TARIQ] Muhammad Mukarram Bin Tariq et al., Poisson versus periodic path probing (or, does PASTA matter?), pp. 119–124 of the *Proceedings of the IMC '05*, 2005 Internet Measurement Conference, October 2005

[WOLFF] Ronald W. Wolff, Poisson Arrivals See Time Averages, *Operations Research*, Vol. 30, No. 2, March–April 1982

Notes

1. NTP refers to clock sources by their strata, where stratum 1 sources are considered most accurate; stratum 2 sources derive their time from stratum 1 sources, and so on.
2. The nature of the networking industry and community means that some of the sources referred to in this book exist only on the World Wide Web. All Universal Resource Locators (URLs) have been checked and were correct at the time of going to press, but their longevity cannot be guaranteed.