CHAPTER 1

INTRODUCTION TO VoIP NETWORKS

Voice over Internet Protocol (VoIP) has exploded onto the technology scene in the past few years. VoIP is set as the technology that takes our current telephony system referred to as Public Switched Telephone Network (PSTN) to the next generation. Before delving into how VoIP stands to deliver on that promise, we take a brief look at telephony in the PSTN space. Our discussion of PSTN will be more conceptual rather than merely elaborating the components and protocols. The goal is to make the reader understand the philosophy that drove the design of the telephony network and also to lay a foundation to the type of services that would be expected of a full-fledged VoIP network.

1.1 PUBLIC SWITCHED TELEPHONE NETWORK (PSTN)

The era of telephone communication started in 1876, when Alexander Graham Bell enabled the transmission of voice over a wire connecting two phones. Fundamentally, the role of a telephone connection in completing a call is very simple – it needs to connect the microphone of the caller to the hearing piece of the receiver and *vice versa*. In the beginning of the telephony era, each pair of phones had to have a wire between them in order for them to communicate. There was no shared component between the devices, so while people wanted telephones to communicate, the system was not cost-effective.

VoIP: Wireless, P2P and New Enterprise Voice over IP Samrat Ganguly and Sudeept Bhatnagar

1.1.1 Switching

Perhaps the most important development that proved to be a huge step in making a large-scale telephone system viable was the concept of a *switch*. The insight that drove the design of a switch was that a dedicated wire between two telephones was essentially being used for a very small fraction of time (unless the parties at the two ends talked all day on the phone); so a way of using that line to serve some other connection while it was idle would serve to reduce the cost of deployment. In particular, the concept of *multiplexing* was used. The idea was to be able to share the line between multiple telephones on an on-demand basis. Of course, the trade-off was that if two pairs of phones were sharing a single line, only one pair of them could talk at a time. On the other hand, if most of the time only one pair of them intended to communicate, the telephone system could do with only one line rather than two.

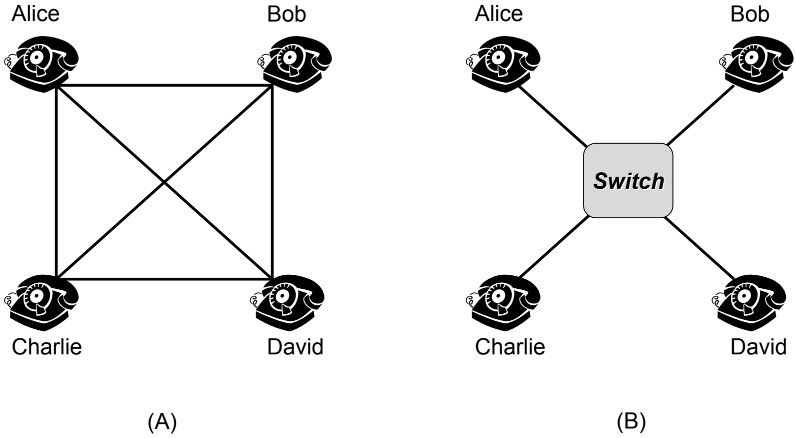

Figure 1.1 Basic functionality of a switch over a four-telephone network. (A) Without a switch six lines are required to connect the four telephones. (B) With a switch, only four lines are required to connect the four telephones to the switch.

In order to implement the concept of multiplexing, there was another problem to be solved. While sharing of a line was definitely a nice insight, the whole wire could not be shared end-to-end since its endpoints are two of the telephones residing at diverse locations. This led to the concept of segmenting the end-to-end wire into smaller pieces and applying the sharing logic onto these pieces. The device that connected these pieces together is a switch. Consider the example shown in Figure 1.1. There are four telephones that need to communicate with each other. In Figure 1.1(A), they are connected directly to each other requiring a total of six lines. In order to reduce the number of lines required, the lines are broken into smaller segments and connected to a switch. as shown in Figure 1.1(B). Now no two phones are directly connected to each other. When Alice wants to talk to Bob, it is now the switch's responsibility to connect the corresponding two segments together so that together they act as an end-to-end wire. Note that if Alice and Bob are talking, and Charlie wants to talk to Bob, then he cannot at that instant. This is because the line segment from the switch to Bob is already in use for the call from Alice. Thus, Bob's phone is 'busy'.

In the early days, the function of switching was done manually. There was an operator who connected the lines together to provide connectivity. As technology advanced, these switches were automated and were able to switch several calls simultaneously. The switches today are electronic and very adept at their task while handling hundreds of calls simultaneously.

1.1.2 Routing

While the concept of switching was an important driver in making the telephony viable in a small geographic region, it was still not enough to spread to larger areas. This is because it was not feasible to connect all the phones in the whole area (state, country, world) to a single switch. This implies the need to have multiple switches corresponding to diverse geographic regions. Of course, this also means that if Alice's phone is connected to switch A and Bob's to switch B, then for Alice to call Bob, both switch A and B have to connect their respective segments (to Alice and Bob respectively) as well as to have a connecting segment between them. Conceptually, this requires that each pair of switches should now have a link between them to allow all pairs of telephones to be able to communicate with each other.

Again, the requirement for all switches to connect to all other switches is not scalable. For example, it may be reasonable to have a line connecting switches to two neighboring cities. However, having a line from each switch to every other switch in the world is infeasible.

The alternate strategy extends the concept described earlier. When two phones connected to two physically connected switches need to talk, we required three line segments to be connected together: Alice's phone to switch A, switch A to switch B, and switch B to Bob's phone. However, if switch A and switch B are not directly connected, they can still be able to connect through a chain of switches in between. Thus, the larger the distance between Alice and Bob, the larger the number of switches in the path between them. Conceptually, when Alice calls Bob, a whole set of segments and switches are connected in sequence to provide the feel of an end-to-end wire between the two of them. None of these segments can be used for any call while this call is in progress. Essentially, this results in building a dedicated *circuit* between Alice and Bob.

The above example uses links which can carry a single call at a time. In practice, the switch-to-switch links (also referred to as exchange-to-exchange links) are replaced by *Trunks* that can carry multiple calls simultaneously. This is achieved by methods such as Time Division Multiplexing (TDM) where frames from different calls (containing the encoded voice signals) are multiplexed into a TDM frame that runs over a higher bandwidth. This results in the perception of all the calls proceeding simultaneously. Although the number of calls carried in a trunk is much higher, the bandwidth limitations of the medium limit the number of simultaneous calls possible over a trunk as well. For example, in the USA, a TDM frame contains 24 voice frames implying at most 24 simultaneous calls over the corresponding trunk.

1.1.3 Connection hierarchy

With the help of call routing, any two telephones can communicate over a sequence of switches. However, how do we decide which switches are to be connected to each other? Consider a simple case, where there are three switches A, B, C. Physically, A and B are closer to each other (say in adjacent cities) and C is far away from both of them (another country). One possible connection could be to have $A \leftrightarrow C$ and $B \leftrightarrow C$ as the two links.

Now any call from a phone connected to A to a phone connected to B will have to be switched across the country to C from where it would be routed back to B. Clearly this type of long link should be avoided as much as possible.

This implies that the switches within each other's vicinity should be connected to each other rather than to those far apart. A natural extension of this philosophy implies that the switches within cities should be connected to form a network, a few access nodes from this network should connect to other networks in similar states, and the same philosophy extends to countries. Essentially, the political boundaries themselves serve as guidelines to forming networks of switches.

1.1.4 Telephone numbering

Once the hierarchical organization of switches and, in general, exchanges is decided, the final piece of the puzzle is to figure out where a particular phone is located in order to call and how the corresponding call should be routed. Across a large network spread across the globe, knowing all the routes to all the other switches and destinations is not feasible. Thus, each switch can know only a few neighboring switches.

The problem of routing in such a scenario is automatically solved using a proper telephone-numbering system (E.164) that we use today. For example, a telephone number consists of a code, an in-country zone code, and a number describing the local switch/exchange to which the phone is connected. Using the digits of the phone number, the switch at the caller's end would know to which of the neighboring switches the call should be routed. Following the same procedure end-to-end, a VoIP call is easily established.

1.1.5 Signaling

The call setup procedure described above requires some means of informing all the devices on the end-to-end path of the call to switch the call accordingly. This is achieved using signaling. The current telephony network is based on sophisticated signaling protocol called SS7. It is the most prominent set of protocols in use in the PSTN across the world. Its main use is in setting up and terminating telephone calls. SS7 uses an out-of-band signaling method to set up a call. The speech path of the call is separated from this signaling path to eliminate the chances of an end-user tampering with the setup protocol.

In the PSTN, telephones constantly exchange signals with various network components such as dial tone and dialing a number. SS7 facilitates this type of signaling in the current PSTN. In general, SS7 forms the core of the current PSTN. Along with call establishment and termination, it provides the aforementioned functionalities such as call routing.

1.1.6 Summary

Our description of traditional telephony describes the most important concepts required in setting up a voice call across the network. Switching allows the telephone to multiplex over a limited number of links. With switches connected to each other indirectly, routing is required to set up a call over multiple hops. Using the concept of hierarchy, the E.164 numbering assigns a logical method to discover users and to set up end-to-end phone calls. In order to set up any end-to-end call, the devices on the path of that call need to take appropriate action to switch the call correctly so that it follows the set route. This entire setup is attained using signaling.

It is important to see that while these concepts are described as applicable to PSTN telephony network, in fact, any large-scale telephony network needs to provide these functions. Thus, enabling VoIP over the Internet (which is a large-scale network) also implies that these functions be provided in the Internet. We shall look at how these functions are provided in the Internet both in general and specifically for VoIP.

1.2 FUNDAMENTALS OF INTERNET TECHNOLOGY

What we described above gives the basic idea regarding any telephony system. To enable voice over an IP network such as the Internet,[1] the capabilities described above need to be provided in the Internet as well. In the following, we describe how these functionalities are provided in the Internet in general.

1.2.1 Packetization and packet-switching

The PSTN is based on the concept of *circuit-switching*. For any call to go through, a complete end-to-end path is set up comprising of intermediate switches and the dedicated links between them. This sets up a path that is specifically meant for the call prior to the user being able to communicate. Having such a dedicated *circuit* for a call means that the delay faced by each signal element (carrying the voice) is constant. The components used in the circuit are not available for use until the call terminates.

Each circuit has a capacity to carry some amount of information at each instant. In case of voice, this information is the signal containing the encoded speech. Dedicated circuit for a call results in a wastage of the capacity even if for a small time it is not being used to carry information. This wastage is more prominent in case there are other calls that are not able to connect for want of a segment of this underutilized circuit.

In order to overcome this capacity underutilization a new switching method was conceived. The switching method is called *packet-switching*. The idea is to *packetize* the information, i.e. break down the information to be transmitted into smaller chunks called packets and send each packet independently towards the destination. There is no dedicated end-to-end path setup prior to the communication. Each packet containing information to be delivered is sent towards the destination. Each switching element *router in the Internet world*, would look up the destination address in the packet and send it to the next switching element on the path to the destination. Essentially, different packets belonging to the same end-to-end communication session can take different paths in the network, since there is no circuit for them to follow.

The efficiency gain from circuit switching are from multiplexing at a fine level. Since there are no resources reserved for any end-to-end session at an intermediate router, the router treats all arriving packets as equals. The packets from different sessions are lined up in a queue inside the router which decides where to send the packets one by one. Thus, the router is being used by all calls simultaneously. In the case of VoIP, think of packets containing voice from two different calls sharing the router queue. Also, packet-switching does not lock the router (and a link) for a particular call, implying that packets from a second call can be switched if there are no packets from the first call using the router.

[1] While the Internet is an embodiment of an IP network, we shall use the two terms interchangeably throughout the book.

Packet-switching forms the backbone of the Internet. The computers (end-hosts) from the end points of communication are connected using routers. Two computers communicate with each other by packetizing the information to be sent out and then send each packet to the network. They are routed by the network to the destination without establishing an end-to-end connection *a priori*.

1.2.2 Addressing

In its present form, the most prevalent addressing scheme in the Internet is based on Internet Protocol version 4 (IPv4). The allocated addresses are called the IP addresses of the respective devices. Each IPv4 address is 32 bits (4 bytes) long. Each address can be written in the dotted decimal notation as A.B.C.D where each of A,B,C,D is a number between 0 and 255 (representing 8 bits).

The addresses are allocated to organizations in sets defined by the common prefix shared by the addresses in the set. In the initial era, the addresses were categorized into classes and allocations were at the granularity of classes. The classes to be allocated for unicast addresses were called Classes A, B and C. Class D defined multicast addresses and Class E addresses were reserved for future use. Each Class A group of addresses was identified by its first 8-bit prefix and hence contained 2^{24} distinct addresses. Similarly Classes B and C had 16- and 24-bit prefixes resulting in address spaces of 2^{16} and 2^{8}, respectively. Assigning address spaces at this granularity had an adverse impact as the available address space started depleting very quickly.

To address this concern, a new proposal called Classless Inter-Domain Routing (CIDR) was introduced where the IP addresses were allocated in chunks and identified by their prefixes rather than classes. Thus, a large chunk of addresses that contain all addresses starting with the first 9 bits being 100 001 011 is written as 133.128.0.0/23 where the 23 represents the number of bits that can vary with the prefix 9 bits fixed to 100 001 011 (133.128 represents the decimal value of 1 000 010 110 000 000).

In CIDR addressing, a large chunk of addresses is now given to allocation authorities that can create smaller chunks out of it to allocate to the organizations. For example, a country can be allocated a chunk 133.0.0.0/24. From this chunk, organizations can be given smaller chunks which may depend on their location in the country. This will automatically construct a hierarchy of addresses. The major benefit of such a hierarchy is seen in the routing efficiency.

1.2.3 Routing and forwarding

Routing refers to the process of computing the routes between any two hosts. In a router, the routing process fills out a *routing table* (or forwarding table), that contains information about which interface the router should forward a packet to so that the packet reaches closer to its destination.

There are two types of routing protocol in the Internet: intra-domain and inter-domain. The intra-domain routing protocols (such as RIP, OSPF) operate in a single domain under the control of one administration. In a domain, the messages contain information about the connectivity information of all the nodes in the domain. After applying the correct routing algorithm such as Dijkstra's shortest path algorithm, the routes are computed and the routing table of each router is populated. Inter-domain protocols (such as BGP) operate on a coarser granularity. The border router of a domain provides a list of prefixes to which it can route. Based on the routing policy, the routers will decide the routes for these prefixes. An interesting observation here is that there is implicit hierarchy here. Inside a domain each router knows every other router and would also know the IP addresses of all hosts that are directly connected to this network. However, the external network appears as a single entity in the form of a prefix advertised by a neighbor.

The routing table of each router is computed using the intra-domain and inter-domain routing protocol. Since the Internet is a packet-switched network, the goal is to be able to route any packet to its destination. The routing table is the core that allows this. It contains information about where a router should send a packet, based on its destination IP address. An interesting thing to note is that if the routing table contains an individual entry for each destination IP address in the Internet, there will be 2^{32} entries in the routing table. It is very difficult to manage this number of entries in a router. The CIDR-based scheme allows routing tables to be compacted. In this case, the adjacent prefixes could merge into single entry if the corresponding outgoing interfaces for both sets of prefixes is the same. The exact packet header matching algorithm is called *Longest Prefix Match*. If there are multiple entries in the routing table that match the destination address of a packet, then the entry which has the maximum number of prefix bits common with the destination IP address is considered the valid matching entry and the packet is forwarded accordingly. For example, for a packet with destination address 133.193.20.24, if there are two entries in the routing table 133.192.0.0/24 and 133.193.0.0/16 (with corresponding forwarding interface), both will match with the packet's destination address. However, we will use the entry with the latter prefix as it has more prefix bits in common with the destination IP address and forward the packet to the interface corresponding to that entry.

When a packet arrives at a router, the following functions are performed in order:

- *Routing Lookup:* At the incoming interface, the router needs to determine the output interface for the packet. The router uses the longest prefix match to find the most specific entry in the routing table corresponding to the packet's destination. A lookup on that entry gives the output interface to which to send the packet. Using the router's switching fabric, this packet is sent to the corresponding output interface.

- *Queue Management:* Each output interface has a buffer where it queues all packets forwarded to it by all incoming interfaces. The buffering is required because the output link capacity might not be sufficient to handle the combined traffic from all interfaces. Since the buffer size is finite, the buffer could be full when a new packet arrives. The basic task of the queue management strategy is to determine *which* packet to drop in such a case. Traditionally, the routers follow a *drop-tail* policy where the incoming packet is dropped if the buffer is full. Note that this is also an implication of packet-switching as the buffer is being shared by packets from diverse connections. Of course, if the buffer is not full, typically the packet will be enqueued at the back of the queue of packets that already reside in the buffer. This is not always the case because there are certain *Active Queue Management* mechanisms (such as Random Early Detect – RED) where an incoming packet may be dropped even if the buffer is not full. This is done to indicate to the host that congestion is imminent and it should slow down its traffic rate.

- *Scheduling:* The scheduler resides on the output interface of a router. Its task is to select a packet from the queue to transmit. In the current IP routers, the predominant scheduling policy is *First In First Out (FIFO)*. Thus, the scheduler picks the first packet in the queue and sends it out on the link. However, from the perspective of VoIP, it may be beneficial to send voice packets, which may be at the back of the queue, prior to the other non-real-time packets such as those belonging to an FTP session.

1.2.4 DNS

Domain Name System (DNS) provides the name to which address the mapping service in the Internet. It is one of the most important services in the Internet. DNS provides the service equivalent of directory lookup.

A DNS query takes a Fully Qualified Domain Name (FQDN) such as a URL and the response contains the current IP address associated with the given FQDN. One of the most important benefits of DNS is that it allows users to remember easy-to-memorize strings rather than IP addresses. For example, it is much easier to remember the website for Wiley as `www.wiley.com` rather than remembering a set of four numbers representing its IP address.

While from the perspective of the user this FQDN to IP address translation is the single advantage that DNS provides, it has several features from the perspective of the service providers. It allows the servers handling different services in an organization to be identified. We shall see more details of one such usage in Chapter 16. Furthermore, it allows load balancing across servers by returning different mirror server addresses to different user queries. This provides a simple load-balancing solution. As a further extension, the same applies to the use of DNS to provide fault tolerance. If one of the servers providing a service fails, DNS can be used to provide the address of another server seamlessly.

In order to provide this basic service, DNS essentially serves like a distributed database. The Internet namespace is divided into *zones* with the responsibility of managing the namespace in each zone being delegated to a particular authority. Thus, a zone is essentially a unit of delegation. For example, the authority of the `.com` zone is delegated to a single authority and that of the `wiley.com` zone is delegated to another authority. Each zone can have one or more DNS servers which maintain the local namespace database. For example, the name to IP address mapping information for `www.wiley.com` would rest with the DNS server for `wiley.com` zone.

A DNS request can originate from any host in the Internet. In the simplest case, the DNS query would process the text of the FQDN from right to left. Thus, to query for the IP address of `www.wiley.com`, a host would first go to the *root domain server*. That server will redirect it to one of the top-domain servers for the `.com` domain. The `.com` domain server would inform the user to query the `wiley.com` domain's server which will have the answer to the query. In practice, this process is augmented with caching. When a query is issued (say by the web browser visiting a web site), it calls a *resolver* software in the local machine. The resolver usually caches some popular FQDN-to-IP matches that some prior DNS lookup had returned. If the current query is satisfied by a cached entry, the resolver returns that address. If not, it forwards the query to the preconfigured DNS server that the host's ISP has provided. Again that DNS server maintains a cache of frequently resolved FQDN-to-DNS mappings. If the query is not answered from its cache, it follows the aforementioned procedure as a client would, and returns the result to the host.

1.3 PERFORMANCE ISSUES IN THE INTERNET

While the Internet provides all the features that are required of a telephony network, there are significant other problems that it introduces. It may be tempting to think that with switching, routing, addressing and lookups being provided, VoIP would have no special concerns in the Internet. However, this is not correct. In fact, while the cost of deploying VoIP over the Internet is considerably less (as it is using a shared network), there are significant performance issues that need to be addressed. For VoIP to be a viable alternative to the PSTN, not only should it be cheaper and easier to deploy and maintain, it should provide similar or better call quality so as to motivate an end-user to move to VoIP.

The performance issues that the Internet faces stem from its packet-switching nature. Packets from several flows share the queue at the output interface of a router. The bandwidth that the link connected to that interface is limited. Thus, the resources of the router are shared, resulting in several performance glitches.

1.3.1 Latency

Latency is the total delay that a packet faces while it travels from its source to its destination. There are multiple contributors to the latency. The foremost of these contributors is the physical limit imposed by the speed of light (or electromagnetic wave, depending on the carrier). For example, if a packet (or a signal) has to travel 3000 km over a link, then at speed of light (300 000 km/s), it will take 10 ms to travel. In practice, the signal travelling speed is lower than the speed of light. This delay has to be faced independently of the underlying signal-carrying technology. The second contributor to the latency is the *queueing delay*. This is the delay that a packet faces at a router when it is stuck behind other packets waiting for its turn to be transmitted. Note that this delay is not present in the circuit-switched network where there is a dedicated circuit present for the signals for a call. The last source of latency is the *transmission delay*. This delay is due to the limited bandwidth of the link on which the packet will be transmitted. Transmission delay calculates the time between the first and last bits of the packet being put on the wire. For example, a 500-byte (4 Kb) packet on a 1 Mbps link will incur a transmission delay of 4 ms because of constraints imposed by the bandwidth of the link.

Delay is an additive quantity. All types of delay incurred at all components add up. Thus, the longer the path, the more the number of routers that a packet will pass through, and the more delayed it is. Furthermore, if traffic at some other source increased so that the packet concerned sees a large queue, it will be delayed further.

1.3.2 Packet loss

There is no concept of loss in the circuit-switched networks. If a connection is established, then until the connection is terminated by the involved parties, all information communicated over the circuit will follow the established circuits and reach the other end. There will be no information loss.

In case of packet-switching, there is a possibility of packet loss. As discussed earlier, this happens in the extreme case where the buffer on a router's output interface is full and a new packet arrives. There is no space for the packet in the queue and hence it has to be dropped. Second, there is no notion of a circuit, so there is no notification to the involved parties that their packet was dropped. In fact, for reliable transmission of information in packet-switched networks such as the Internet, special protocols such as TCP have to be designed that identify a packet as being lost (somewhere along the path) and retransmit the packet so that the receiver obtains its content. While increasing packet delays serve as an indication that the queues in some routers are building up, the Internet protocols such as TCP react more drastically to a drop-in packet so as to reduce the load, thereby ameliorating congestion.

While a router's output interface queue becomes overloaded due to a surge in the traffic from some (potentially other) source , the impact of that surge is seen by our packet under consideration. This type of cross-interaction is possible due to packet switching.

1.3.3 Jitter

Jitter represents the variance in delay seen over a bunch of packets belonging to the same end-to-end connection. Simply put, over the life of a connection, several packets will be exchanged between the source and the destination. It is highly unlikely that each of these packets will face exactly the same queueing delay at all the routers along the path. In fact, given the Internet routing model, it is not guaranteed that all the packets will follow the same path and encounter the same routers.

This variability in the latencies of different packets of a connection is referred to as jitter. There is no jitter in a circuit-switched network. This is because once the end-to-end circuit is set up, there is no one contending with the corresponding connection to grab a share of that circuit.

From the perspective of VoIP, each packet carries some data corresponding to what was spoken. With a large jitter, the words that a packet contains would seem either too cluttered or too spread apart if the packets are played out as and when they arrive. To smooth out this effect, a *jitter buffer* is used to hold the packets for a while and release them at a smooth rate to the application to play.

1.4 QUALITY OF SERVICE (QoS) GUARANTEES

We have seen that in its native form, the Internet suffers from several problems that can have a significant impact on the performance of real-time applications such as VoIP. In order to counter these scenarios, the Internet Engineering Task Force (IETF), proposed mechanisms where the flows with such real-time requirements would be segregated from the other flows, even while they use the same router equipment. In essence, an application could request a certain amount of network resources along its entire path and its packets can receive preferential treatment from the network. Thus, the packets would be admitted in a special queue to conceal them from the effects of other traffic, and be scheduled for transmission with a higher priority. These requirements have been formalized by two standard mechanisms: Integrated Services and Differentiated Services.

In order to realize both these QoS models, the current Internet architecture needs to be altered, the service model changed, and the router functionalities modified to support the new services. We look first at the architectural changes that are required for both QoS services.

The QoS architectures could be classified broadly into two categories: *stateful* and *stateless*. The stateful architectures require per-flow states at all routers; the stateless architectures do not have such a requirement. In practice, the stateless architectures are actually core-stateless where the edge routers of a domain maintain per-flow state and the core routers do not. The major benefit of the stateless architectures comes from eliminating the costly packet classification operation at the core routers. The Integrated Services architecture is stateful whereas Differentiated Services is stateless in this terminology. In both architectures, various router functionalities are altered, and in turn they provide different levels of guarantee as well as having a different level of impact on scalability.

1.4.1 Integrated services

The Integrated Services (Intserv) framework intends to provide strong QoS guarantees to flows. Intserv requires that all routers have a per-flow state. Each router has a separate queue in the output buffer for each flow. A packet is added to the tail of its flow's queue in the output buffer. This adds another problem for the queue management component: If it has to drop a packet, then it has to take an additional decision about which queue's packet should be dropped. The scheduler is no longer FIFO because it has to select a queue from which to send the next packet. Since the number of queues is the same as the number of flows, the time complexity of the scheduler depends on the number of flows. The choice of a particular scheduler depends on the type of service provided under the Intserv purview. There are two key services defined under the Intserv framework: *Guaranteed and Controlled-load Services*.

- *Guaranteed Services:* Guaranteed service semantics intend to provide per-flow bandwidth and delay guarantees [1]. The routers have to ensure that its packets are never dropped as long as they are compliant with its traffic specification. Additionally, the scheduler employed has to schedule the packets of the flow based on its deadline and rate requirements. However, the complexity of these schedulers is significant and limits the scalability of the framework.

- *Controlled-load Services:* The controlled-load service intends to isolate a flow from the impact of other flows. The key specification of the controlled-load semantics is to provide an uncongested network view to a flow. The controlled-load service intends to provide a service similar to best-effort service when the routers are unloaded [2]. This type of service is suitable for adaptive real-time applications. VoIP is well-suited for this type of service.

In summary, the Intserv model has strong per-flow service semantics. However, it requires maintenance of per-flow states which renders it unscalable in the number of flows.

1.4.2 Differentiated services

As an architecture that does not mandate the per-flow state at all routers, Differentiated Services (Diffserv) is more scalable than Intserv. Diffserv classifies the routers as edge and core routers. Under Diffserv, only the edge routers maintain a per-flow state. On receiving a packet, an edge router classifies it to find the class it belongs to, and marks the type of service in the packet's header using a Differentiated Services Code Point (DSCP) that represents its class. The core routers only maintain a small number of queues corresponding to the number of classes and implement different *per-hop-behaviors* to service different DSCPs. They do not distinguish between individual flows and serve the packets having the same

DSCP in an identical fashion irrespective of the flow to which they belong. Since the core routers only have a fixed number of service classes (defined by the number of DSCPs), their scalability becomes independent of the number of flows.

The Diffserv framework has two types of defined service: *Assured Service and Premium Service.*

- *Assured Service:* Assured service aims at providing a lightly loaded network view by giving better-than-best effort drop rates to flows [3]. This is attained by implementing preferential dropping where a customer's *out-of-profile* traffic is dropped before his *in-profile* traffic. At the ingress router the user's packets are marked as in-profile or out-of-profile using a meter (or the user could indicate his preference). If a router becomes congested, it will drop the out-of-profile packets first. Thus, the user is assured of a fixed bandwidth (given by its in-profile rate).
- *Premium Service:* Premium service provides a virtual wire of a desired bandwidth from an ingress point to an egress point [4]. It is implemented using priority queuing to forward the premium packets at the earliest. Note that premium service can provide a bandwidth guarantee to the entire aggregate but since it does not distinguish between packets of individual flows, the delay bounds of individual flows cannot be distinguished, i.e. all flows in an aggregate have the same delay bound irrespective of their requirements.

Thus, the Diffserv framework alleviates the scalability problem of Intserv but can only provide weaker service semantics.

1.4.3 Other modifications

The QoS architectures require changes in the functionalities depending on the type of service model. They also require some additional operations which help in providing these services efficiently. We discuss these operations briefly.

1.4.3.1 Route pinning One of the most fundamental changes required to support the QoS services is the ability to pin a flow to a fixed route. The conventional IP routing allows the routes to change at any time and different packets from the same session can take different paths. These changes could occur based on the underlying traffic changes, for load-balancing or due to topology alterations. To provide bandwidth or delay guarantees to a flow, the network has to be sure that the flow's path has sufficient resources to meet its requirements. If a flow's path changes during its lifetime, the new path might not have the desired resources.

Route-pinning techniques make sure that a flow follows its assigned path during its entire lifetime. The most prominent of these strategies are IP source routing, Multi-Protocol Label Switching (MPLS) [5], and Virtual Circuit Switching in ATM networks. In recent years, MPLS has become increasingly popular as a route-pinning and efficient packet-forwarding technology.

1.4.3.2 Packet classification The forwarding mechanism in the Internet is based on the longest prefix match algorithm which takes the destination IP address of a packet as an input. The second change common to both QoS architectures is an ability to classify packets based on fields in their header other than the IP destination address. This is a must because the routers need to identify *which* packets belong to a flow with QoS guarantees and *how* these packets need to be treated. This requires a special packet classification functionality which essentially supersedes the routing lookup operation. Packet classification could be done based on multiple fields in the packet header, e.g. the source and destination addresses, protocol number and type of service field.

1.4.4 Admission control

Admission control refers to the process of limiting the number of QoS flows in the system so that their respective QoS guarantees are not violated. Specifically, on receiving a new request for some QoS guarantee, it is the responsibility of the admission control component to test whether there are sufficient resources in terms of bandwidth and buffer at the routers to support the new flow without violating the guarantees of the existing flows.

Of all the services listed above, only assured service could remain meaningful in the absence of admission control. All other services require some sort of admission control. However, the specific type of admission control they require varies, based on their respective characteristics. The admission control methods in the literature could be broadly classified as *deterministic* or *statistical* in nature. The deterministic admission control methods take as input the request parameters such as delay and bandwidth and determine whether or not the request could be supported after taking into account the existing reservations and using the knowledge of the scheduler characteristics and buffer availability. Statistical admission control methods on the other hand try to estimate whether or not the request could receive its desired service with some probability.

1.4.5 Status

The two QoS architectures for the Internet have been explicitly defined. However, they still await deployment in the Internet. This can be attributed to a lack of business model for these services along with the fact that any flow's path will traverse through multiple Internet Service Providers (ISPs) from source to destination. For a QoS guarantee to be given, all these traversed domains have to cooperate at the fine-granularity of per-flow or per-packet. This limitation is another reason for the lack of deployment of these services. Nonetheless, knowledge of these architectures is important as the techniques developed in this context are relevant in VoIP applications.

1.5 SUMMARY

Switching is a core concept that enables a telephony network to scale to a large user population. The traditional PSTN network uses a circuit-switching model where an end-to-end path is established prior to the communication starting. The circuit is dedicated to the call for its entire lifetime. The Internet uses a packet-switching model where the information to be communicated is packed into destination-labelled packets. Individual packets belonging to a single end-to-end session are switched independently. In both cases, routing is used to determine the end-to-end path – in PSTN the complete path is set up *a priori* whereas in the Internet, the next hop for a packet is determined when the packet arrives at a router. The addressing schemes in both types of network enable aggregation of addresses into more compact representations and thereby, reduction in the routing information.

The packet switching used in the Internet provides efficient resource utilization and scalability, but it results in additional performance problems in terms of packets facing excessive delays that could vary significantly from packet to packet and in the worst case the packet could be lost. In order to provide a certain quality of service guarantee to tackle these problems, two major standards have been defined that allow flows preferential treatment at the routers along their path. In order to ensure that the network has sufficient resources to support an additional request, admission control is used to determine whether there are sufficient resources remaining to address a request's demands.

REFERENCES

1. Shenker, S., Partridge, C. and Guerin, R. Specification of guaranteed quality of service. *IETF Request for Comments RFC 2211* (1997).
2. Wroclawski, J. Specification of the controlled-load network element service. *IETF Request for Comments RFC 2211* (1997).
3. Heinanen, J., Baker, F., Weiss, W. and Wroclawski, J. Assured forwarding PHB group. *IETF Request for Comments RFC 2597* (1999).
4. Jacobson, V., Nichols, K. and Poduri, K. An expedited forwarding PHB. *IETF Request for Comments RFC 2598* (1999).
5. Rosen, E., Viswanathan A. and Callon, R. Multiprotocol label switching architecture. *IETF Request for Comments RFC 2597* (2001).