1 Introduction

In recent years, the tremendous growth of the Internet and the increasing demand of user applications have resulted in a number of architectural changes to the Internet infrastructure. By its original, the Internet based on the packet-switched technology has been designed for delivering packets in a *best effort* fashion: the end systems do not need to inform the network prior to transmitting their IP packets, while routers simply perform routing and forwarding of these packets without distinguishing from each other. However, this design has been challenged due to the new requirements which have been dramatically different from over 30 years ago. For example, to realize the bandwidth and connectivity on demand for the service providers, a signaling protocol seems to be critical.

Signaling is not a new topic. In the telecommunication industry, signaling is common and can be dated back to when circuit switches first replaced human telephone operators. Even the modern Signaling System No. 7 (SS7) [187] began its development in the mid 1970's, based on the idea that relies on a separate control element (i.e., the SS7 signal switches) to signal to other control element to set up, manage and release voice trunk lines required to make a call. Based on the signaling standard for ISDN [8], ITU-T standardized a Q.2931 signaling protocol [9] which allows ATM nodes to exchange control of information, request the use of network resources, or negotiate for the use of circuit parameters, for instance, mapping between an input set and an output set of virtual circuit identifiers (VCIs) and virtual path identifiers (VPIs). Essentially, signaling protocols manages states in network nodes.They generally reflect some requirements of an end-to-end session/call to the traversed nodes. Thus, they need to be maintained properly, especially when network "conditions" change (e.g., some link or node fails, or the traversing route changes). The task of a signaling protocol involves establishing, maintaining and removing network control states, traversing from one end system to another through the network. Hence, the concept of signaling protocol discussed in this book mainly targets at network control state signaling.

However, it was until the early 1990s that people had begun to realize the importance of introducing flow-specific network control states and their signaling mechanism into the Internet, initially for creating Quality of Service (QoS) resource

reservation states in routers and hosts (as will be discussed in more details in Section 1.1.1 and Chapter 4). Since then, several architectural changes to the Internet concerning other middlebox functionalities for end-to-end communications have been suggested [39, 53, 54, 138].

The *network control state* concept will be covered throughout this book and hence it needs some explanation. Network control state refers to any control or configuration information maintained in the network nodes, which is associated to a given end-to-end (e2e) communication, or flow. On one hand, it differs from end-to-end protocol state (i.e., state stored at the application endpoints), such as those maintained by TCP, Stream Control Transmission Protocol (SCTP) [209], Session Initiation Protocol (SIP) [185], Hypertext Transfer Protocol (HTTP) [32,85], or the Real-Time Control Protocol (RTCP) [110], since it is mostly concerned with network nodes in the middle of the communication path, although the end hosts can also be involved. On the other hand, we should distinguish this flow-related network control state from routing state, such as the one created by the Routing Information Protocol (RIP) [148], Open Shortest Path First (OSPF) [162], Protocol Independent Multicast (PIM) [12,74], Border Gateway Protocol (BGP) [178], or Internet Group Management Protocol (IGMP) [81]. More specifically, although both are meant for communications, routing state is more static and (for a vast majority of routing protocols used until today,) just to establish state information for the purpose of routing IP packets between different subnetworks without being aware of individual end-to-end communications[1]. With this general notion, this book discusses in detail the recent development in the field of network control state signaling in IP-based networks, or in short "IP signaling". Sometimes, management of the network control state can be done through some means of configuration, for example explicit configuration of middlboxes such as firewall pinholes and NAT bindings.

In the following sections an overview of IP signaling scenarios and the major concerns for designing IP signaling protocols are presented.

1.1 IP Signaling Scenarios

As described above, the task of signaling – in the context of this book – is to deliver flow related information to various network elements involved in the data packet handling, and accordingly manipulate network control state. In other words, signaling is to establish, maintain and release control states in network elements. The

[1] QoS-based routing [63] is an exception. It relies on QoS information to modify the path of data packets through the network.

state establishment procedure aims to establish state at network elements that are or will be hopefully traversed by the data traffic. There is a non-subtle effect for evolving the Internet architecture into a generic infrastructure which accommodates a variety of usage scenarios and to meet new requirements. By moving this component from being limited to one or two specific applications to a broader applicability approach, we can simply and easily allow vendors and service providers to build various value-added services. It is expected that such a generic approach would allow the IETF and other standardization organizations the ability to easily extend the existing protocol by defining new signaling applications, new QoS signaling models and many other signaling features. In the following subsections we give a few examples of how individual demanding signaling scenarios are motivating our work on this field.

1.1.1 Resource Reservation for QoS Provisioning

As discussed in previous section, the Internet has been evolving from a packet-switched network with certain fundamentally architectural changes. One among these changes has been motivated by the strong need for providing Quality of Service (QoS) with the emerging multimedia and other real time applications, such as voice over IP. QoS provisioning usually refers to the mechanisms for meeting application-perceived network performance requirements, such as delay, bandwidth and jitter. Due to an ever-increasing capacity offered by some link layer technologies and therefore also by some providers, over-provisioning has been regarded, at least sometimes, by some backbone operators or regions with rich bandwidths, as a solution to avoid QoS signaling. However, high bandwidth does not necessarily imply a guaranteed QoS such as delay assurance [163], even if bandwidth would be increasing more quickly than the amount of network traffic. In order to provide service differentiation, the introduction of new components for (control plane) resource reservation and (forwarding plane) traffic control, etc., is still necessary. Therefore, since the last decade, QoS has become a hot topic for the Internet and the research community. The Resource Reservation Protocol (RSVP) [44, 255] has been recognized as the most famous protocol for resource reservation since the early 1990s.

In order to support the increasing demand of multimedia and other mission-critical network applications, people have identified a number of architectural components to be added in the Internet architecture, including admission control and resource reservation, packet classification, marking, scheduling and policing. They are known as QoS provisioning techniques and the IETF has standardized two QoS

architectures, namely the finer-grained Integrated Services (IntServ)) [43] and the coarser-grained Differentiated Services (DiffServ) [38]. To provide dynamic QoS guarantees for end-to-end applications, a signaling protocol is required, that allows applications state their traffic characteristics and reservation requirements, such as bandwidth and token bucket, to routers along the path from a sending host to a receiving host. A detailed description of the best-known signaling protocol for resource reservation, the Resource Reservation Protocol (RSVP) [44], is given in Chapter 2.

To deal with resource reservation in various scenarios, including mobile environments, in addition to serve for other signaling purposes such as firewall and NAT configuration as described in the following subsections, either RSVP needed to be extended or a new protocol framework was needed. To this end we have extensively studied the existing extensions of RSVP and developed proposals and solutions for mobility and more generic signaling purposes, which will be addressed throughout this book.

The IETF has formed a new working group, namely the Next Steps in Signaling (NSIS) working group [4] in November 2001, to develop the architecture and protocols. QoS signaling and firewall/NAT signaling are the first signaling applications of the NSIS working group targets to address.

Note there are other ways for providing QoS in the Internet, including:

Stateless explicit IP signaling: In contrast to signaling for maintaining reservation state in network nodes, Ion Stoica at Carnegie Mellon proposed in his Ph.D. thesis (2000) [212] a *Dynamic Packet State* (DPS) concept, which approxiates QoS achieved with per-flow reservation mechanisms without maintaining per-flow state (more details about DPS will be explained in Section 2.2.2).

Explicit Congestion Notification: The *Explicit Congestion Notification* (ECN) proposed by Floyd *et al.* [177] is a well-known mechanism to allow TCP endpoints to react to congestions by marking one ECN bit of the traversed TCP data traffic in ECN-aware routers upon congestion. This essentially is another form of stateless signaling. ECN has been standardized by the IETF but has not been widely explored.

Signaling for obtaining precise congestion information: The *eXplicit Control Protocol* (XCP) proposed by Katabi *et al. [136]* and *Performance Transparency Protocol* (PTP) proposed by Wezl [238] extend the ECN approach as follows. Data packets in XCP carry a congestion header in which the sender

requests a desired throughput, while XCP-aware routers make a fair per-flow bandwidth allocation (and the sender is eventually notified with the bottleneck throughput) without maintaining any per-flow state. PTP follows a similar way but its signaling is done by out-of-band signaling messages. In simulations, both XCP and PTP are found to be scalable, and better than TCP (without ECN) in terms of end-to-end performance.

Pre-Congestion Notification: The IETF PCN working group developed a *Pre-Congestion Notification* (PCN) [45, 58, 72] concept, which also builds on the ECN concept. The essential means in PCN is to introduce flow admission control and pre-emption for admitted flows in a DiffServ region. In normal circumstances admission control should protect the QoS of previously admitted flows. When experiencing heavy congestion pre-emption of admitted flows could preserve the QoS of remaining flows. Interior routers use bulk PCN packet marking to give early warning of their own congestion. Exterior routers convert measurements of this packet-level marking into admission control and pre-emption functions at flow granularity. This way, interior routers do not require flow state or signaling while an end-to-end controlled load service can be achieved.

1.1.2 Configuration of Firewall Pinholes and NAT Bindings

Firewalls prevent using certain applications between hosts situated in different administrative domains. Applications are those which use control signaling channel and separate data exchange streams are particularly affected, since they cannot be identified by a pre-determined matching filter expression. Some of such applications are now becoming widely used, for example, IP Telephony or Instant Messaging which use dynamic ephemeral ports.

Network Address Translators (NATs) break all applications that carry IP addresses and ports embedded within their messages, as well as applications that do not have keep-alive capabilities to prevent a NAT binding from being deleted during an idle period. IP telephony (see [17,111,126,185]), Video over IP and Internet gaming have started to demonstrate the architectural impact that NATs have imposed on the Internet. The most frequently deployed NAT is the traditional NAT [205] – or *outbound NAT* – which creates a binding for outbound flows (i.e., flows destined to hosts external to a private address realm). There are two types of traditional NATs:

1. Basic NAT:

   Only the source IP address of the IP packet is replaced by a public address when the IP packet is forwarded to the Internet. Meanwhile, the public IP address is replaced by the private one when the packet returns from the host on the Internet.

2. Network Address and Port Translator (NAPT):

   The source IP address, as well as the source port, is replaced by public ones when the packet is sent from the private address realm.

In this book the term NAT refers to both basic NAT and NAPT. In case the sender is behind a NAT, signaling is necessary for a data sender to obtain its public IP address and port number as seen by the data receiver. If firewall traversal is desired, properly configured packet filters are necessary to avoid blocking applications. Since information about application endpoints is usually negotiated as part of the application signaling session (e.g., SIP), a static configuration is insufficient and hence a protocol is required to establish, maintain and remove state at the firewall.

There have been previous attempts to develop signaling protocols to interact with these middleboxes. UPnP [234] and RSIP [40] are recent examples. UPnP uses a link local multicast discovery protocol to find middleboxes (within the same LAN segment) and a separate protocol to signal them. RSIP, which was simply designed for requesting NAT bindings, requires the usage of a tunnel between the end host and the NAT. However, neither UPnP nor RSIP could be considered as well-secured protocols and their use would be inappropriate when stringent security requirements are mandated in a network environment.

More recent attempts to signal firewalls and NATs were developed in the IETF MIDdlebox COMunication (MIDCOM) [125] and NSIS working groups. MIDCOM aims to develop a protocol that fits the usage scenario for application servers to interact with the middleboxes. These application servers are assumed to have knowledge about the network topology and the middlebox that has to be contacted for a particular data flow. For a simple network topology with a single firewall, this is a trival task. However, for more complex topologies, the task of middlebox discovery becomes a problem. Therefore, a different approach that reuses previous work on RSVP was proposed. This work was moved to the IETF NSIS working group. The NSIS framework allows signaling of various network services and is not specific to QoS. The NAT and Firewall NSIS Signaling Layer Protocol (NAT/FW NSLP) [211] is specifically designed to handle NAT and Firewall aspects in parallel with other NSIS NSLP protocols, such as Quality of Service signaling.

1.1.3 Label Distribution for MPLS Networks

Traffic engineering is the process of optimization of the network to maximize performance and efficiency. Multi-Protocol Label Switching (MPLS) is a tool for network traffic engineering and hence is becoming a technology of choice for Internet backbone.

MPLS introduces a new forwarding paradigm for IP networks [184]. The idea is similar to the approach taken with the asynchronous transfer mode (ATM) and frame relay networks. A label switching path (LSP) is first established using a signaling protocol; then a label in the packet header, rather than an IP destination address, is used for making forwarding decisions in the network. This way, MPLS introduces the notion of "virtual-circuit"-oriented forwarding in an IP network. MPLS thus offers a solution for traffic engineering, establishing a path and sending traffic along that path.

LSPs are basically a concatenation of one or more label switched routers (LSRs). A signaling protocol installs and maintains control state in these LSRs that allows each LSR swaps the incoming label for the outgoing label assigned to the next LSR for that data stream. Upon MPLS signaling, an LSP is established when each MPLS node along the path between the initial (ingress) MPLS node and the final (egress) MPLS node has a binding between an incoming label and an outgoing label. The most widely used MPLS signaling protocols are the traffic engineering extension for RSVP (RSVP-TE) [23] and the Label Distribution Protocol (LDP) [16] (including its constraint-based routing extension, CR-LDP [21]).

1.1.4 Code Distribution for Active Networks

Since the notion of *active networking* was presented by Tennenhous *et al.* [218], it is being viewed as one of the solutions for the fast, flexible and dynamic deployment of new telecommunications network services.

Active networking can be simplistically regarded as a set of *active nodes* which perform customized operations along the data path. End users, operators, and/or service providers need to distribute service-specific application code into networks, in order to obtain required network support for new services. Thus, active networking involves programmable network node operations, platforms, and security, as well as dynamic services provisioning. One important component among them is active code distribution, or dissemination of active packets.

One proposal for performing active code distribution is ANTS [244], a toolkit designed specifically for active networks. Another approach is the Active Network

Encapsulation Protocol (ANEP) [14, 194]. Nonetheless, none of them has been successful enough to gain wide acceptance. If general active networking support for the Internet is desired, a signaling protocol for active code distribution will be necessary.

1.1.5 Path Status Diagnostics

Monitoring, metering and accounting of packets is an important functionality in many networks today. The IETF has developed different protocols to collect and report usage data for resource consumption in a network by a particular entity. For example, the IPFIX WG defines a protocol to collect such data. The PSAMP WG defines a standard to sample subsets of packets by statistical and other methods. Starting with network access authentication, authorization and accounting for dial-up users, RADIUS [180, 181] and DIAMETER [50, 113] have been developed. In addition to user authentication and authorization funcationality, these protocols offer the ability to report consumed resources for charging. Furthermore, they are used also in other areas which require authentication and authorization for application layer services, since reusing the existing deployment is highly desired. Another approach is to use SNMP [119] where the Meter MIB [46] can be used to collect flow information.

Many of the above-mentioned protocols assume that the nodes running these protocols are configured to perform a specific task or assume a certain network deployment (e.g., simply targeting at the edge of a network). The configuration of nodes along the path in order to collect path status statistics is subject to ongoing work in [70].

1.2 IP Signaling Protocol Design Issues

Having discussed the use scenarios of IP signaling protocols, in this section the key design aspects are identified, which allows identifying and characterizing the requirements and properties of signaling protocols. They also help to justify the tradeoffs produced by different design decisions and assess the suitability of various design principles for different signaling purposes. The analysis of the protocol properties and the design tradeoffs yield guidelines for the review of the existing protocols (detailed in Chapter 2) and the development of the new signaling architecture and protocols in later chapters of this book.