Chapter 1

Introduction

The Internet has had phenomenal success in the past 20 years, growing from a small research network to a global network that we use in a daily basis. The Internet is logically composed of end hosts interconnected by links and routers. When a host wants to communicate with other hosts, it uses the Internet Protocol (IP) to place information in packets, which are then sent to the nearest router. The router stores, then forwards, packets to the next hop, and through hop-by-hop routing, packets find their way to the desired destination. In other words, end hosts communicate through *packet switching*. With this communication technique, link bandwidth is shared among all information flows, and so these flows are statistically multiplexed on the link. The resulting service is best effort, in the sense that there are no deterministic guarantees.

Another switching technique that is widely used in communication networks, especially in the phone system, is *circuit switching*. When a terminal wants to communicate with another terminal, this technique creates a fixed-bandwidth channel, called a *circuit*, between the source and the destination. This circuit is reserved exclusively for a particular information flow, and no other flow can use it. Consequently, flows are isolated from each other, and thus their environment is well controlled. This Thesis studies how the Internet could benefit from more circuit switching than is prevalent today.

1.1 Motivation

The Internet has been very successful in part because its decentralized control has permitted the rapid development and deployment of applications and services. The success of the Internet is demonstrated by the enormous traffic growth that has already made the Internet carry more traffic than the phone network [24, 69, 100, 47].

If the Internet is based on packet switching, why would I want to use circuit switching? The answer is simple. There is a mismatch between the evolution rates of traffic and capacity of the Internet, and circuit switching can help bridge the gap between demand and supply.

The capacity of the network has to keep up with Internet traffic growth rates that are 10 times larger than that of voice traffic. Coffman and Odlyzko, among others, have been studying traffic growth in the Internet, and they have found that traffic has been doubling every year since 1997 [47, 135]. Studies by RHK [162] and Papagiannaki et al. [140] indicate similar growth rates. The capacity of the Internet should match these growth rates in order to avoid the collapse of the network. The next section studies the evolution trends of the underlying technologies in a router, and, as we will see, router technology is being outpaced by Internet demand.

If routers cannot keep up with demand, then one can only expand the network capacity by adding more nodes and links. This not only requires more equipment, but also more central offices to house them. It is an expensive proposition, and it also creates a more complex network, making network planning and maintenance more difficult. This Thesis takes a different approach; it focuses on how to improve the performance of the existing network by increasing the capacity of switches and links with the use of circuit switching in the core of the network.

1.2 Technology trends in routers and switches

In order to understand the technology trends to compare them to those of traffic, one has to know the functions that packet and circuit switches do, and the technology used to perform them. In the following, I will focus on the switching function (i.e., the forwarding of information) in a network node. Figure 1.1 shows the functional blocks of a packet switch, also called a *router*. When information arrives at the ingress linecard, the framing module extracts the incoming packet from the link-level frame. The packet then has to go through a route lookup to determine its next hop, and the egress port [82]. Right after the lookup, any required operations on the packet fields are performed, such as decrementing the Time-To-Live (TTL) field, updating the packet checksum, and processing any IP options. After these operations, the packet is sent to the egress port using the router's interconnect, which is rescheduled every packet time. Several packets destined to the same egress port could arrive at the same time. Thus, any conflicting packets have to be queued in the ingress port, the output port, or both. The router is called an input-queued switch, an output-queued switch, or a combined input/output-queued switch depending on where buffering takes place [123].

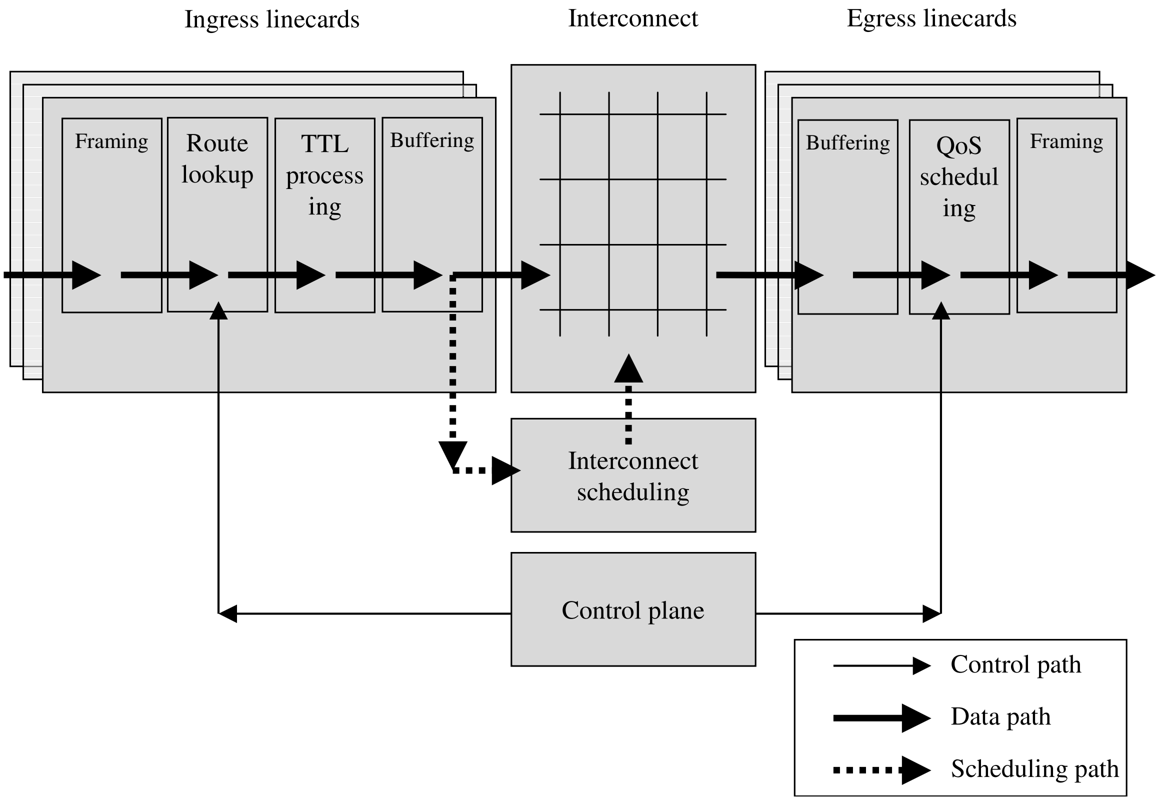

Figure 1.1: Functionality of a packet switch.

In the output linecard, some routers perform additional scheduling that is used to police or shape traffic, so that quality of service (QoS) guarantees are not violated. Finally, the packet is placed in a link frame and sent to the next hop.

In addition to the data path, routers have a control path that is used to populate the routing table, to set up the parameters in the QoS scheduler, and to manage the router in general. The signaling of the control channel is in-band, using packets just as in the data channel. The control plane might obtain the signaling information through a special port attached to the interconnect.

The main distinction between a router and a circuit switch is when information may arrive to the switch. In packet switching, packets may come at any time, and so routers resolve any conflicts among the packets by buffering them. In contrast, in circuit switching information belonging to a flow can only arrive in a pre-determined channel, which is reserved exclusively for that particular flow. No conflicts or unscheduled arrivals occur, which allows circuit switches to do away with buffering, the on-line scheduling of the interconnect, and most of the data-path processing. Figure 1.2 shows the equivalent functions in a circuit switch. As one can see, the data path is much simpler.

In contrast, the control plane becomes more complex: it requires new signaling for the management of circuits, state associated with the circuits, and the off-line scheduling of the arrivals based on the free slots in the interconnect. Usually there is a tradeoff between the signaling/state overhead and the control that we desire over traffic; the tighter the control, the more signaling and state that will be needed. However, in circuit switching, as in packet switching, a slowdown in the control plane does not directly affect the data plane, as all on-going information transmissions can continue at full speed. In general, its data path determines the capacity of the switch.

Another important difference between a router and a circuit switch is the time scale in which similar functions need to be performed. For example, in both types of switches the interconnect needs to be scheduled. A packet switch needs to do it for every packet slot, while a circuit switch only does it when new flows arrive. In general, a flow carries the same amount of information as several packets, and thus packet scheduling needs to be faster than circuit scheduling.

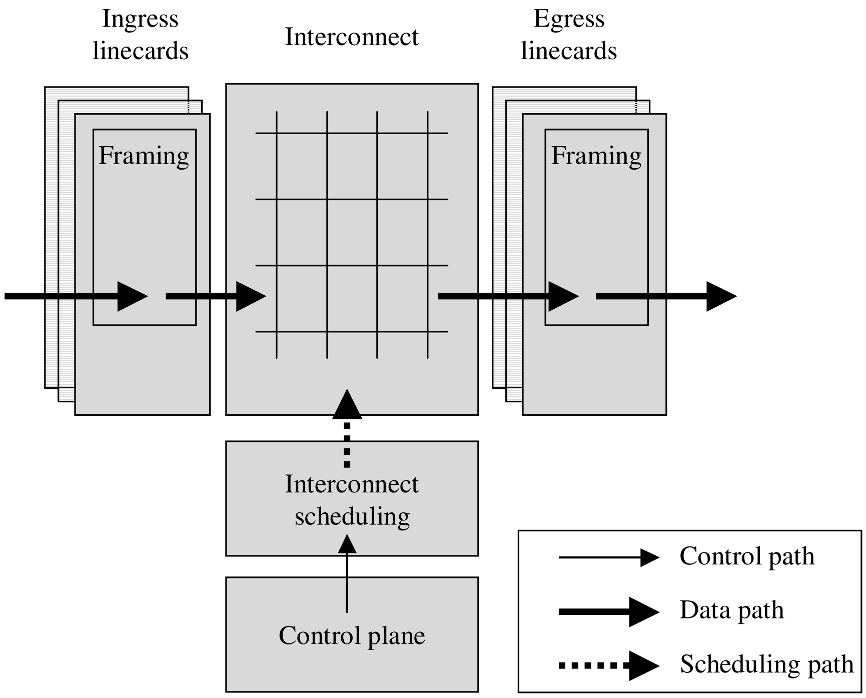

Figure 1.2: Functionality of a circuit switch.

1.2.1 Technology trends

In order to study how the capacity of links and switches will scale in the future, one needs to understand the evolution trends of the underlying technologies used in routers and circuit switches. This enables one to foresee where bottlenecks might occur.

Below, I will focus on the data path of a router, since the data path of a circuit switch is just a subset of it. In general, a router has to:

- **Send and receive packets**: A router receives data through its ingress port and sends it shortly afterwards through the appropriate egress port. Information is sent either through fiber optics for the long haul and high speeds, or through copper cables for the short haul and mid-to-low speeds.

- **Buffer packets**: Packets contend for resources, such as an output port. Conflicts are resolved by deferring the transmission of all but one of the conflicting packets until some later time when the contention has been cleared. As a rule of thumb routers, usually need a *Link Rate* $\times$ *Round Trip Time* worth of buffers[1] because of the way the flow control mechanisms of TCP work [182]. For example, for an OC-768 link of 40 Gbit/s and a typical round trip time (RTT) of 250 ms, a linecard needs 1.2 GBytes of memory. Dynamic RAM (DRAM) is thus used to meet this capacity requirement. In addition, some fast Static RAM (SRAM) is needed to cope with the fast arrival rate of packets. The minimum packet size is 40 Bytes, so packets could arrive with a separation of only 8 ns. Designers find it challenging to build packet buffers for a 10-Gbps line card, and it is even more difficult to achieve 40 Gbps, particularly when power consumption is an issue. Most router capacity is limited by memory availability.

- **Process and forward packets**: Routers need to look up the destination address in a routing table to decide where to send a packet next, or in which queue it should be buffered. Packets also need to be scheduled to use the internal interconnect, so that they go from the ingress port to the egress port without contention. Additionally, other fields in the packet header, such as the TTL or the checksum, have to be updated. Currently, this processing and forwarding is done electronically using specialized ASICs, FPGAs or network processors.

To study the performance trends, I will focus on the core of the network, where traffic aggregation stresses network performance the most. The core also uses the state of the art in technology because costs are spread among more users. The backbone of the Internet is built around three basic technologies: silicon CMOS logic, DRAM memory, and fiber optics.

As was mentioned before, Internet traffic has been doubling every year since 1997.[2] In contrast, according to Moore's law, the number of functions per chip and the number of instructions per second of microprocessors have historically doubled every 1.5 to 2 years[3]

[1] In practice, routers are built to handle congestion for a much lesser period of time, needing fewer buffers, but increasing the packet loss probability during overload.

[2] This trend could be broken by the sudden adoption of a new bandwidth-intensive application, such as video streaming, similar to the period of 1995-6 when the massive adoption of the web made traffic double every 3-4 months.

[3] Experts believe that this trend will slow down as microprocessor and ASIC technologies gradually move from the current two-year cycle to a three-year node cycle after 2004.

[3, 144]. Historically, router capacity has increased slightly faster than Moore's law, multiplying by 2.2 every 1.5 to 2 years. This has been due to advances in router architecture [123] and packet processing [82].

DRAM capacity has quadrupled on average every three years, but its frequency for consecutive accesses has been increasing less than 10% a year [144, 143], equivalent to doubling every 7 to 10 years. Modern advanced DRAM techniques, such as Synchronous Dynamic RAM (SDRAM) and Rambus Dynamic RAM (RDRAM), are attacking the problem with I/O bandwidth across pins of the chip, but not the latency problem [58]. These techniques increase the bandwidth by writing and reading bigger blocks of data at a time, but they cannot speed up the time it takes to reference a new memory location.

Finally, the capacity of fiber optics has been doubling every 7 to 8 months since the advent of DWDM in 1996. However, the growth rate is expected to decrease to doubling every year as we start approaching the maximum capacity per fiber of 100 Tbit/s [124]. Despite this future growth slowdown of DWDM, the long-term growth rate of link capacity will still be above that of Internet traffic at least past the year 2007 [116].

Figure 1.3 shows the mismatch in the evolution rates of optical forwarding, traffic demand, electronic processing, and electronic DRAM memories. We can see how link capacity will outpace demand, but how electronic processing and buffering clearly drag behind demand. Link bandwidth will not be a scarce resource, but the information processing and buffering will be. Instead of optimizing the bandwidth utilization, we should be streamlining the data path.

Figure 1.3 shows how there is an increasing performance gap that could cause bottlenecks in the future. The first potential bottleneck is the memory system. Routers may be able to avoid it by using techniques that hide the increasingly high access times of DRAMs [91], similarly to what modern computers do. With these techniques access times come close to those of SRAM, which follows Moore's law. However, they make buffering more complex, with deeper pipelining, longer design cycles and higher power consumption.

The second potential bottleneck is information processing. The trend would argue for the simplification of the data path. However, there is a lot of pressure from carriers to add more features in the routers, such as intelligent buffering, quality-of-service scheduling, and traffic accounting.

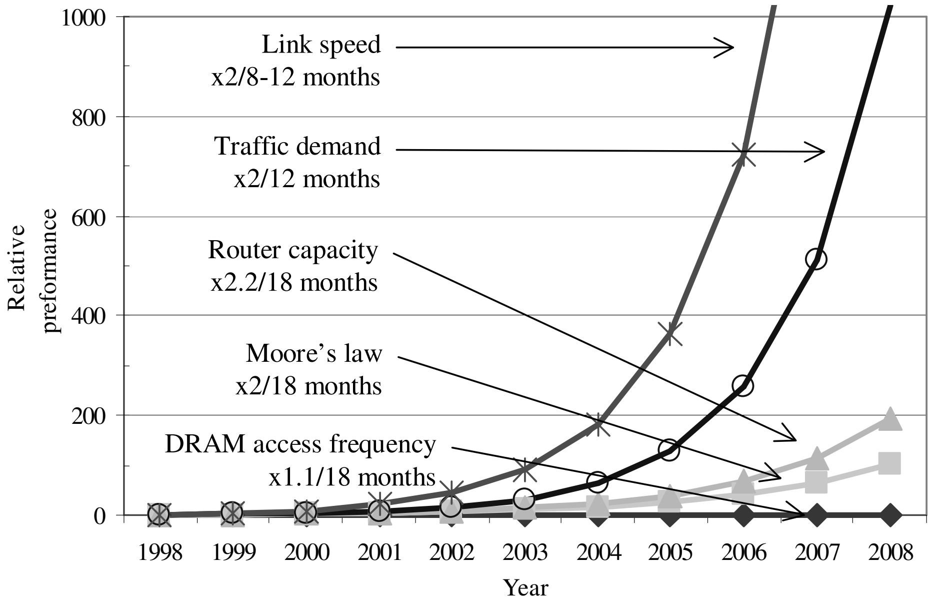

Figure 1.3: Trends of traffic demand and the underlying technologies in the Internet [1998 = 100%]. Trends for Silicon processing and router forwarding capacity are kept at the same value as today, despite talks of a slow down after 2004.

If we keep the number of operations per packet constant, in ten years time, the same number of routers that we currently have will be able to process 200 times as much traffic as today. In contrast, traffic will have grown 1000 times by then. This means that we will have a five-fold performance gap. In ten years time we will need five times more routers as today. These routers will consume five times more power, and will occupy five times more space.[4] This means building over five times as many central offices and points of presence to house them, which is a very heavy financial burden for the already deeply indebted network carriers. To make matters worse, a network with more than five times as many routers will be more complex and more difficult to manage and evolve. The economical and logistical cost of simply adding more nodes is prohibitive, so we need to be creative, and think out of the box, trying to find a more effective solution that solves the mismatch between traffic demand and router capacity, even if it represents a paradigm shift.

[4]Actually they will occupy more than five times the space, as many of the routers linecards will be used to connect to other routers within the same central office, rather than to routers in other locations.

1.2.2 Optical switching technology

One possible solution is to use optical switching elements. Optics is already a very appealing technology for the core because of their long reach and high capacity transmission. Additionally, recent advances in MEMS [15, 83], integrated waveguides [85], optical gratings [101, 27], tunable lasers [187], and holography [149] have made possible very high capacity switch fabrics. For example, optical switches based on MEMS mirrors have shown to be almost line-rate independent, as opposed to CMOS transistors, which saturate before reaching 100 GHz [3, 130]. Ideally, we would like to build an all-optical packet switch that rides on the technology curve of optics. However, building such a switch is not feasible today because packet switching requires buffers in order to resolve packet contention, and we still do not know how to buffer photons while providing random access. Current efforts in high-speed optical storage [178, 109, 151] are still too crude and complex. In current approaches, information degrades fairly rapidly (the longest holding times are around 1 ms), and they can only be tuned for specific wavelengths. It is hard to see how they could achieve, in an economical manner, the high integration and speed that provides 1.2 Gbytes of buffers to a 40 Gbit/s linecard.[5]

Another problem with all-optical routers is that processing of information in optics is also difficult and costly, so most of the time information is processed electronically, and only the transmission, and, potentially, the switching is done in optics. Current optical gates are all electrically controlled, and they are either mechanical (slow and wear rather quickly), liquid crystals (inherently slow), or poled $LiNbO_3$ structures (potentially fast, but requiring tens of kV per mm, making them slow to charge/discharge). Switching in optics at packet times, which can be as small as 8 ns for a 40 Gbit/s link, is very challenging, and, thus, there have been proposals to switch higher data units [188], called optical bursts. Rather than requiring end hosts to send data in larger packets, these approaches have gateways at the ingress of the optical core that aggregate regular IP packets into mega-packets. These gateways perform all the buffering that otherwise would be performed in the optical core, so the buffering problem is not eliminated, but rather pushed towards the edges.

[5] Needless to say, this vision of the future could completely change if a breakthrough in technology made fast, high-density optical memories possible.

In summary, all-optical routers are still far from being feasible. On the contrary, all-optical circuit switches are already a reality [15, 111, 112, 174, 54, 32]. This should not be a surprise, since circuit switching presents a data path that requires no buffering and little processing. For example, Lucent has demonstrated an all-optical circuit switch, using MEMS mirrors, with switching capacities of up to 2 Pbit/s [15]; this is 6000 times faster than the fastest electronic router [94].

Even when we consider electronic circuit switches and routers, the data path of circuit switches is much simpler than that of electronic routers, as shown in Figure 1.1 and Figure 1.2. This simple data path of circuit switches allows them to scale to higher capacity than equivalent electronic routers. This is confirmed by looking at the fastest switches and routers that are commercially available in the market at the time of writing; one can see that circuit switches have a capacity that is 2 to 12 times bigger than that of the fastest routers, as shown in Table 1.1. The simple data path of circuit switches comes at the cost of having a more complex control path. However, it is the data path that determines the switching capacity, not the control path; every packet traverses the data path, whereas the control path is taken less often, only when a circuit needs to be created or destroyed.

Product	Type of switch	Bidirectional switching capacity
Cisco 12016	router	160 Gbit/s
Juniper T640	router	320 Gbit/s
Lucent LambdaUnite	circuit switch	320 Gbit/s
Ciena CoreDirector	circuit switch	640 Gbit/s
Tellium Aurora 128	circuit switch	1.28 Tbit/s
Nortel OPTera HDX	circuit switch	3.84 Tbit/s

Table 1.1: Bidirectional switching capacities of commercial switches [42, 94, 40, 174, 132]. While I have tried to be careful in the comparison, comparing product specifications from different vendors is not necessarily comparing ”apples with apples”, and should be taken only as a rough indication of their relative capacities.

In this Thesis, I argue that we could close the evolution gap between Internet traffic and switching capacity by using more circuit switching in the core, both in optical and electronic forms. I also explore different ways of how one could integrate these two techniques.

1.3 Circuit and packet switching

If one had to give a very succinct description about how the Internet works, one would say it as being composed of end hosts and routers interconnected by links, as shown in Figure 1.4. In more detail, the Internet is a packet-switched, store-and-forward network that uses hop-by-hop routing and provides a best effort network service. This technology was chosen because it enabled a robust network that made an efficient use of the network resources [11, 43, 172, 14].

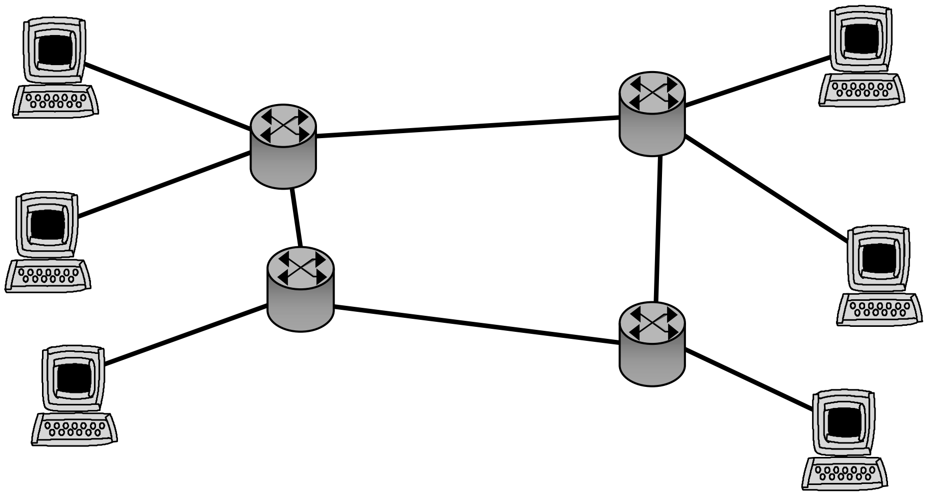

Figure 1.4: Simple architecture of the public Internet as described in textbooks.

However, the real Internet is more complex than this, and if we look closely, we will find that there is plenty of circuit switching in the Internet, as shown in Figure 1.5. We have circuit switches both in the access networks (leased lines, DSL and phone modems), and in the core of the network (SONET/SDH and DWDM). This figure also shows the market sizes in the year 2001, and it shows how the market sizes for the segments that use circuit switching are significant.

The current mix of packet and circuit switching in the Internet is due to historical reasons. In the early days of the Internet, when two Internet Service Providers (ISPs) in different and distant locations wanted to interconnect with each other, they leased a connection from the only companies that had a continent-wide network, that is the long-distance telephone companies, and these companies have always based their service on pure circuit switching. Similarly, the circuits in the edges were one of the few options for an ISP to get to its customers, namely, by using the existing infrastructure of the local telephone company.

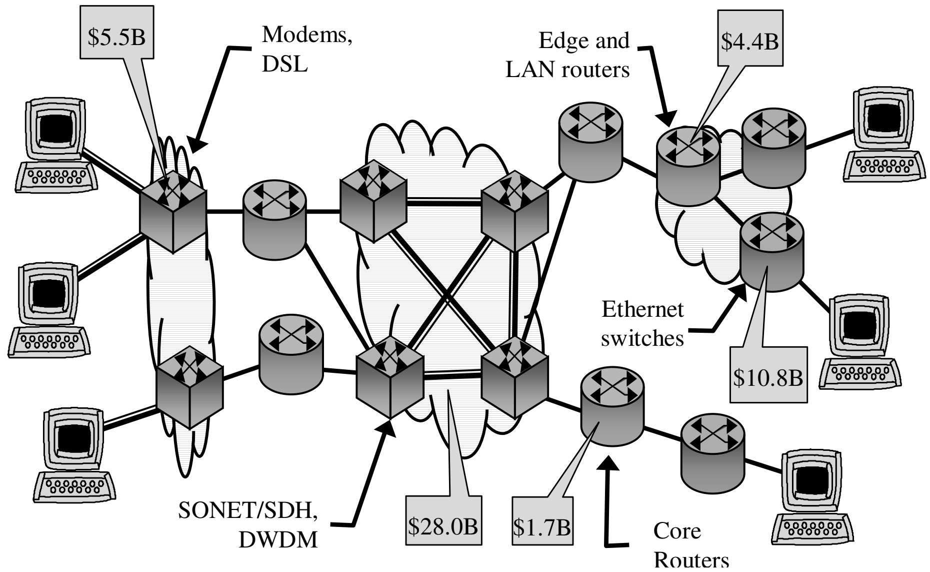

Figure 1.5: Architecture of the public Internet as found in the real world. The figures in the boxes represent the world market sizes in the year 2001. [161, 158, 157, 60, 61]

Given the current situation, one may ask two related questions. First, is this hybrid architecture the right network architecture? If we were to rebuild the Internet from scratch and with unlimited funds, would we choose a solution based on only packet switching, only circuit switching, or a mix of the two? Second, given that it would be too costly to build a brand new network, how can the current legacy Internet evolve in the future? Will the network still follow a hybrid model as today, or will it change? These two questions are the focus of the first part of the Thesis. I will conclude that it makes more sense to use circuit switching in the core and packet switching in the edges of the network.

Currently, the circuits that we find in the Internet are considered by IP as static, layer-2, point-to-point links. In other words, the circuit and packet switched parts of the network are completely decoupled, and changes in IP traffic patterns do not prompt an automatic reconfiguration of the circuits over which IP travels. It is usually the case that circuits are manually provisioned by either the network operator (circuits in the core) or the end user (circuits as access lines). This also means that the time scale in which circuits operate is much larger than that of packets.

We would make a more efficient use of the network resources if we could integrate the world of circuits with that of packets in such a way that circuits follow in real time the fluctuations of the packet switched traffic. In this Thesis, I make two proposals of how to integrate these two technologies in an evolutionary manner, without changing existing end hosts and routers. One approach uses fine-grain, lightweight circuits; the other uses coarse-grain, heavyweight circuits (such as optical wavelengths).

1.3.1 Virtual circuits

There is a third family of networks, which uses virtual circuits, such as ATM or MPLS. This family attempts to get the best of two worlds: on one hand, it takes the statistical multiplexing of packet switching. On the other hand, the traffic management and quality of service of circuit switching. Despite their name, virtual circuits are essentially a connection-oriented version of packet switching; it forwards information as packets (sometimes called cells), but it keeps connection state associated with each flow. In contrast, IP is based on the connectionless switching of packets, where no per-flow state is kept. Switches using virtual circuits are hard to design; they have the scalability issues of both the data path of packet switching and the state management and signaling of circuit switching. Therefore, virtual circuits will not be studied any further in this Thesis.

In the early 90's, there was a race between IP and ATM to dominate data networks. In the end, IP routers prevailed over ATM switches partly because the former were simpler and thus faster to hit the market and easier to configure. In contrast, MPLS works just below IP, rather than competing with it, and it is an attempt to do simple traffic engineering in the Internet. Only recently has MPLS started getting sizable deployment with some backbone carriers [34].

1.4 Performance metrics for core IP routers

To study what network architecture is better, we need to have some performance metrics to compare the different options. In the network, there are two main stakeholders: the end users and the network carriers. Evidently, they have different concerns and views of the network. The most common use of the network is to request and download pieces of information[6] (be a web page, an image, a song, a video or a record in a database). After *reachability*, the end user is mainly interested in a fast *response time*, defined as the time since we request the object until the last byte arrives. Another important set of user applications (e.g., voice or video conferencing and streaming, or gaming) requires *quality of service (QoS)* guarantees, such as bounds on bandwidth, maximum delay, delay jitter or loss. These network guarantees are often expressed as a service level agreement between the network user and the ISP.

Network carries have a very different set of requirements. Tables 1.2 and 1.3 show a survey by BTexact of several IP backbone carriers about their current concerns and the features they will need in the next 2 years.

1	equipment reliability and stability
2	scalability
3	performance
4	feature support
5	management
6	total cost of ownership
7	environmental considerations (power, size)

Table 1.2: Concerns of carriers for network equipment in decreasing order of importance [25].

After interconnectivity, which is always taken as a given, router *reliability* and *stability* are the greatest concern to carriers today. Significant improvements are required in these

[6]today, over 65% of the traffic is web browsing and peer-to-peer file sharing [31].

1	denial of service attack mitigation
2	wire-rate performance of interfaces
3	system access security
4	port density improvements
5	quality of service support

Table 1.3: New features required by carriers for network equipment in decreasing order of importance [25].

areas, particularly in the area of software reliability. It is highly undesirable to have equipment that fails, needs continuous attention, ties up valuable human resources and spoils the reputation of the carrier. Following in importance are *scalability* and *performance*. Even though the total cost of ownership comes last in the survey, the economic problems that numerous carriers currently face have probably increased its relevance; a good network has to come at a reasonable cost.

In terms of new features that carriers desire, the mitigation of Denial of Service (DoS) attacks ranks first, as such attacks can make the network connection unusable, damaging the reputation of the carrier. Improvements in performance come second, followed by better authentication and access security. Quality of service is last; however, it is more relevant for an operator that wants an integrated network that carries both low-margin data traffic and high-margin voice traffic.

In summary, end users care about low response time and quality of service, whereas network operators desire reliability, scalability and performance. As we will see end users will see no difference when using circuit switching or packet switching in the core of the network, whereas network carriers will clearly benefit from getting a network of higher capacity and reliability.

1.5 Understanding Internet traffic and failures

Before we can choose a solution for the network, we need to understand what types of workloads are currently being injected into the Internet and how reliable different elements in the network are. Different workloads will stress the system in different ways and will have different performance requirements and notions of quality of service. For example, a workload with traffic sent periodically in fixed-sized bursts will behave quite differently than one that has a lot of variation in terms of interarrival times, flow durations and flow rates. The first workload would be best served by a slotted network, while the second one would not.

This Thesis provides a short analysis and discussion of the traffic and failures that we see in real networks, especially in or near the backbone. Some of these results are based on my own analysis of traffic traces [131, 170], other results have been reported elsewhere [31, 30, 102, 107, 106, 105]. In general, one is interested in knowing both the type of application (to prioritize the performance metrics) and the distributions and correlations of:

- interarrival times (of flows and packets)
- sizes or durations (of flows and packets)
- transmission rates of flows
- failures of network elements

Based on those observations, we can make some assumptions of the system workload. The most fundamental one is that flow durations in the Internet have long and heavy tails [146, 56, 183, 23], as shown in Figure 1.6. It shows how fewer than 10% of the flows in a backbone link carry over 90% of the bytes transported in the link. There are, thus, two types of user flows: most flows are short; and then a few are very, very long and carry most of the bytes. These long flows may hog the system for extended periods of time and degrade the overall system performance significantly. All these flow characteristics will be incorporated in the performance study of packet and circuit switching in the core of the network.

1.6 Organization of the Thesis

Often one can find in the technical press and literature predictions about how IP routers will eventually replace all circuit switches [186, 150, 21, 99, 35, 37, 142, 87, 156, 128, 110].

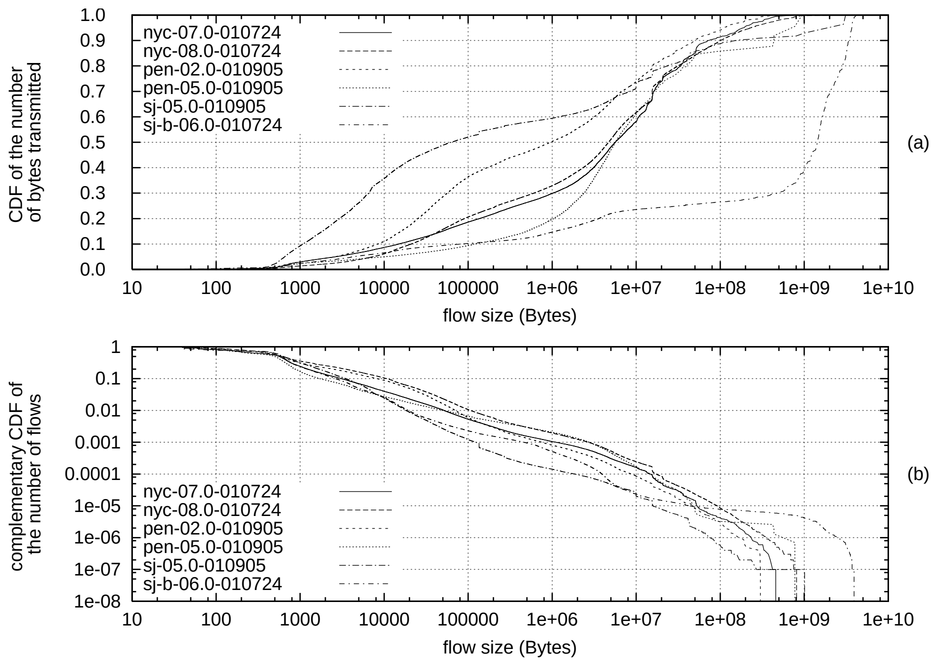

Figure 1.6: Heavy-tailed traffic. The figure shows (a) the empirical cumulative distribution function of the number of bytes transported, and (b) the empirical complementary cumulative distribution function of the flow size frequency with respect to the flow size in bytes. Each of the six backbone traces has between 5 and 40 million flows of different OC-48 links, and they expand over a period of more than 24 hours [170]. They include all types of flows (TCP, UDP and ICMP)

These articles extend the original arguments for adopting packet switching in the early days of the Internet (namely, efficiency and robustness), by adding the simplicity, cost advantage, and ability to provide QoS of IP. These are some of the *sacred cows* of IP, and in Chapter 2, I evaluate them one by one to demystify the ones that do not hold up to scrutiny and to identify the ones that really apply.

One key claim of packet switching is left for Chapter 3; namely that the statistical multiplexing of packet switching consistently delivers a lower response time than circuit switching while downloading information. This is indeed the most relevant performance metric for end users, and thus it gets its own chapter.

The conclusion of these two chapters is that packet switching is very attractive in Local Area Networks (LANs) and access networks, because of the poor end-user response time of dynamic circuits. On the contrary, circuit switching is more attractive in the core of the network because of its higher capacity, its perfect QoS, and a response time that is similar to that of packet switching. In the future, one can expect a dominant role of IP in the edge of the network, whereas various forms of circuit switching will dominate the core of the network. This partially validates the hybrid network architecture that we currently have and that is shown in Figure 1.5.

However, in the current Internet these two distinct parts are completely decoupled; the edges switch packets independently of the circuits used in the core. Chapter 4 presents a network architecture (TCP Switching) that allows the integration of circuit switching in the core of a packet-switched Internet in an evolutionary way. The chapter starts with a description of what a typical application-level flow in the Internet is, as observed on access points to the Internet of several universities and research institutions. A key observation is that despite the connectionless nature of IP, our use of the network is very connection oriented, and this fits well with the use of circuits. TCP Switching is based on the idea of IP Switching [129], and it maps each application flow to a lightweight circuit. This proposal encompasses a family of solutions, with several design choices. Also in this chapter, I choose one solution based on what constitutes a typical flow in the Internet.

One potential problem with such fine-grain circuits in the core, as thin as 56 Kbit/s, is that they might not fit well with many circuit switch designs. Most core circuit switches have interconnects that only use channels of at least 51 Mbit/s. In addition, optical switches only forward wavelengths carrying channels of over 2.5 Gbit/s. The signaling of these switches might be heavyweight because of the slow reconfiguration of the switch fabrics or because of a signaling mechanism that requires circuit creation confirmation. In Chapter 5, I present another technique for controlling the coarse-grain, heavyweight circuits in the core by monitoring user flows rather than tracking packets or queue lengths. I show the requirements for different circuit setup times. These results could be used in Generalized Multi-Protocol Label Switching (GMPLS), a technique that uses heavyweight circuits to adapt the network capacity dynamically between edge routers.

In Chapter 6, I describe some of the related work in the area of high speed switching in the core of the network. Some proposals include the use of circuit switching in the core (GMPLS [7], OIF [13], ODSI [53], Zing [181]), while others attempt to extend packet switching to all-optical switches (Optical Packet Switching – OPS [186] –, and Optical Burst Switching - OBS [188]). Some emphasis is placed on the comparison of TCP Switching with OPS and OBS. Two metrics are used for the comparison: the loss and blocking probabilities for a given network load, and the complexity of the overall network.

Chapter 7 concludes the Thesis, restating how we would benefit from more circuit switching in the core of the network, and how we could integrate this circuit switched core with the rest of the network in an evolutionary way.