5

Applications

Internet traffic is produced by many different applications. Table 5.1 shows the amount of traffic volume produced by different application protocols. It was measured on a backbone link connecting ADSL customers, see Azzouna and Guillemin (2003). We notice that *web* and *peer-to-peer* applications are responsible for the majority of today's Internet traffic. These two application types are discussed in this chapter. Traffic patterns in the Internet will change when new applications become successful. Two applications that will be very important in the future are also discussed in this chapter: *network games* and *Voice over IP (VoIP)* applications. Both have special quality-of-service requirements and their traffic models differ significantly from the web and P2P traffic models.

Depending on the general Quality-of-service (QoS) requirements, we distinguish elastic and inelastic applications. *Elastic* applications are flexible in their bandwidth requirement and adapt their rate to the network conditions. They typically use TCP as transport protocol and therefore TCP's congestion control mechanisms to react to packet losses and delay. We described in Section 4.1.3 how to estimate the throughput of these applications. Typical elastic applications are file transfer applications such as web applications that were not identified in the study browsers, P2P, mail or File Transfer Protocol (FTP) clients.

Inelastic applications are less flexible in their bandwidth requirements and typically need a certain minimum bandwidth to work properly. A typical example is a voice call. Almost all of today's Internet traffic is generated by elastic applications and most voice calls are still transported on dedicated infrastructure. At the time of writing, the total amount of data traffic is roughly 10 times as large as all voice traffic (traditional telephony and VoIP) and growing faster. Nevertheless, inelastic applications such as VoIP, videoconferencing, video streaming and (some) network games gain in importance and as they have special quality-of-service requirements and a high utility to the users, they need the attention of ISPs.

In order to differentiate between applications with different requirements in a network, the application a traffic flow belongs to has to be identified in real time in the network. Traffic classification is therefore important. It is discussed towards the end of this chapter.

5.1 World Wide Web

The World Wide Web (WWW) is based on the Hypertext Transfer Protocol (HTTP) to transport web documents from a web server to the web browser of the end-user. Web documents are typically HTML files and the pictures referenced in these files.

The Competitive Internet Service Provider: Network Architecture, Interconnection, Traffic Engineering and Network Design
Oliver Heckmann © 2006 John Wiley & Sons, Ltd. ISBN: 0-470-01293-5

Table 5.1 Composition of Traffic by Application Type from Azzouna and Guillemin (2003)

Application type	Amount of traffic
World wide web (WWW)	14.6%
Peer-to-peer (P2P)	49.6%
File transfer protocol (FTP)	2.1%
Network news transfer protocol (NNTP)	1.9%
Other (it can safely be assumed that a large percentage is unidentified P2P traffic)	31.8%

5.1.1 QoS Requirements

Web browsing is an interactive application. The time from when the user clicks on a link until the web page is displayed in his web browser determines the perceived quality of service. This time largely depends on the throughput of the HTTP/TCP connection and this throughput again depends on the HTTP version and the network conditions (loss and delay) that determine the TCP throughput. The details are discussed in Section 4.1.3.

In Bhatti *et al*. (2000), user trials show that the tolerance of users for the time it takes until a web page is fully displayed in the browser depends on the task of the user, on whether a page is displayed progressively or not and on how long they have been interacting with the site. The study indicates that thresholds of acceptability change over time. Generally speaking, a web page should be fully displayed within very few seconds and the more interactive a user's task, the faster the transfer should be.

5.1.2 Traffic Model

A traffic model characterises traffic and is important for understanding how many resources are needed to support a certain traffic type, how to identify traffic by its behaviour and how to generate artificial yet realistic traffic, for example, for simulations.

The classic paper on modelling Internet traffic is Danzig and Jamin (1991). In this paper, a library of empirical traffic models for Internet applications that were common at the beginning of the 90s (FTP, SMTP, Telnet) is presented. Web traffic has not been included in it. Paxson (1994) later derived analytical models from traffic measurements by fitting probability distributions to the measured data. Today, there is a vast amount of work on traffic models.

For modelling *individual HTTP connections*, Mah (1997); Choi and Limb (1999) and Barford *et al*. (1999) are a good source, but also see the works cited therein.

- According to these studies, the average HTTP request from a client follows a lognormal distribution with a mean of 360 bytes per request.
- The request is answered by the server that sends back the requested web document. The size of web documents follows a heavy-tailed distribution. This means that most documents are relatively small. However, there is a small but significant chance that a random document is very large. To model the tail, typically a Pareto distribution

is used. A Pareto distribution has a heavy tail, while a Poisson distribution does not; this is visualised in Figure 5.1, where the PDF (Probability Distribution Function) of a Pareto distribution and a Poisson distribution are displayed. The probability of the Pareto distribution approaches zero much slower than the Poisson distribution for high-input values. This behaviour is called *heavy tailed*.
In Barford *et al*. (1999), a Pareto distribution for the tail is combined with a lognormal distribution of the body of the documents.

- A web document consists of one HTML page plus on average five to six objects. Most of these objects are pictures. According to empirical studies, a HTML page has an average size of 10 kB and the objects are around 8 kB.
- The viewing times of the user are around 40 s on average and can be modelled by a Weibull distribution. For more details, we refer to the cited papers.

If we assume that the complete transfer time for a web document including all pictures should not exceed 4 s, the minimum throughput for an average web document should be at least 100 kbps. In reality, owing to the TCP congestion control, a significantly higher amount of bandwidth should be calculated for web browsing.

As a side remark, the popularity of *individual* web documents in the Internet can be described by a Zipf distribution, see for example, Hubermann *et al*. (1998). This means that there will be a few documents that are extremely popular and are requested very often while most of the documents are not.

A widely used tool for generating web traffic is SURGE, see Barford and Crovella (1998). SURGE imitates a stream of HTTP requests from an assumed population of WWW users. Users follow an on–off process. If a user is on, it downloads web documents according to the aforementioned traffic models. Liu *et al*. (2001) describe a similar tool that follows a slightly higher-level traffic characterisation.

Aggregate web traffic shows self-similar behaviour. Self-similarity is a phenomenon observed often in the real world. Coastlines, for example, are statistically self-similar as parts of them show the same statistical properties at different scales. Typical network traffic has self-similar properties. This means that the traffic shows bursts not only on a small

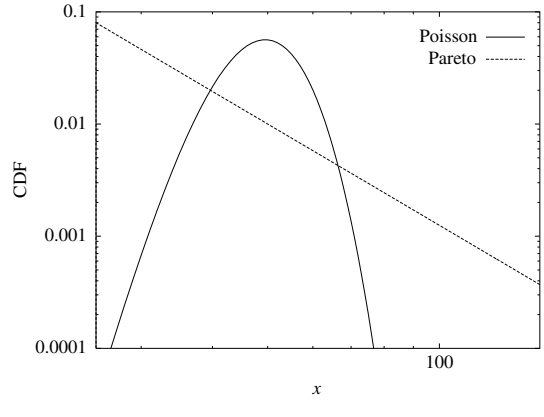

Figure 5.1 Pareto and Poisson Distributions (Logarithmic Scale)

timescale but also on a larger timescale (long-range dependency). An important conclusion from this is that simple traffic models using a Poisson distribution for packet arrival[1] are inaccurate, as for Poisson models the bursts disappear more quickly on larger timescales. This is visualised in Figure 5.2. Networks designed without considering self-similarity are likely to not have enough buffer space and to not work as expected.

Self-similarity has been shown for LAN traffic by Leland *et al*. (1994), for WAN traffic by Paxson and Floyd (1995) and for web traffic during busy hours by Crovella and Bestavros (1997). Self-similarity in web traffic can be explained by heavy-tailed file size distributions (see the preceding text) and by user reading times, see for example, Crovella and Bestavros (1997).

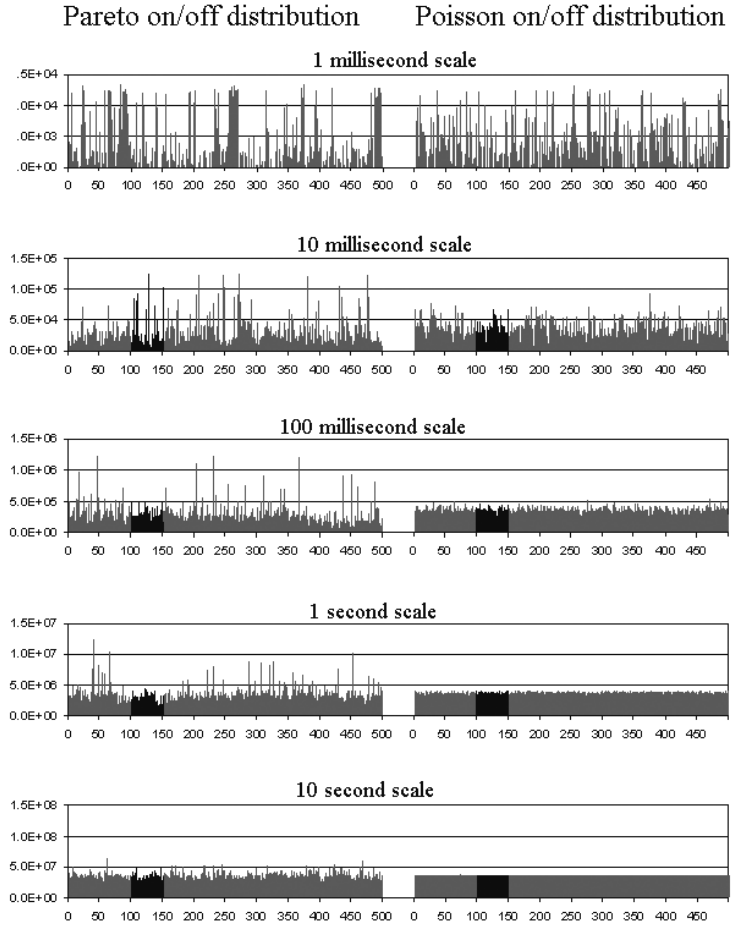

Figure 5.2 Self-similarity and Long-range Dependency (from Kramer (2004)). (Reproduced by permission of Glen Kramer)

[1] This does not necessarily imply that session arrivals cannot be Poisson distributed.

Aggregate web traffic can be well modelled as a superposition of a large number of individual on/off sources with heavy-tailed on/off period lengths. See Kramer (2004) for a simple traffic generator. For measuring self-similarity, tools such as SELFIS can be used, see Karagiannis and Faloutsos (2002).

For further information, we also recommend Beran (1994), Park and Willinger (2000), Mannersalo and Norros (2002), and Addie *et al.* (2002).

5.2 Peer-to-Peer Applications

As shown in Table 5.1, the bulk of today's Internet traffic is caused by P2P applications; see also Azzouna and Guillemin (2003); Fraleigh *et al.* (2003); Sandvine Incorporated (2003).

A *P2P system* is a self-organising system consisting of end systems (called 'peers') that form an overlay network. Peers offer and consume services and resources and have significant autonomy. The participating peers exchange services. Long-term connectivity of individual peers cannot be assumed in a P2P system. This means that a P2P system has to explicitly deal with dial-up users, variable IP addresses, firewalls, Network Address Translation (NAT) and that the system typically operates outside the domain name system. For general literature on P2P systems, see Oram (2001) and Steinmetz and Wehrle (2005).

Almost all of today's popular P2P applications are file sharing applications. They are used to exchange files between end-users. The large majority of the shared files are movies and music files, see for example, Heckmann *et al.* (2004). In 2005, despite increasing counter-measures of the music and movie industry, file sharing makes, to a large extent, illegitimate use of copyrighted material.

5.2.1 QoS Requirements

General file sharing applications are bulk transfer applications and have no real-time constraints and few requirements with respect to loss, delay or jitter. User satisfaction mainly depends on the duration a complete file transfer takes, which is a function of the long-term throughput. P2P traffic is typically treated as low priority or background traffic in most networks, if the network supports the differentiation of different traffic types. In order to do so, however, P2P traffic must be correctly identified in real time in a network. This is not trivial as port-based classification fails for a large part of the P2P traffic. This problem is discussed in Section 5.5.

5.2.2 Traffic Model

In the last years, the Internet has seen many different P2P file sharing applications emerging, becoming successful and then vanishing into insignificance for a variety of reasons. It is therefore hard to derive a general traffic model for P2P traffic. However, certain properties can be assumed:

- P2P applications are bandwidth greedy.

- Compared to WWW applications, they generate more long-lived and therefore reactive TCP connections over which the dominating part of traffic is exchanged. To support this claim, we did some measurements in the eDonkey network[2]:
  Our measurements in Heckmann *et al.* (2004) show that at the time of the study, an average eDonkey user was sharing 57.8 files with an average size of 217 MB, a large proportion of those files being movies. An average active TCP connection between two clients has a duration of almost 30 minutes, definitely long lived. During this time, on average 4 MB are transferred; this volume is mostly limited by the ADSL upload capacity that is typically almost fully used by the P2P application[3]. Few (around 1%), but extremely popular, files account for a very large part (>50%) of the generated traffic; this is also confirmed by Leibowitz *et al.* (2003) for the Kazaa file sharing network. In Kazaa as well as in eDonkey, files are either of medium (few megabytes) or of very large size (>600 MB). This is explained by most files being songs or movies. Measurements in Tutschku and Tran-Gia (2005) show that the flow size of eDonkey can be approximated well by a lognormal distribution. It seems that the heavy tail of the flow size distribution is reduced because eDonkey – like many other P2P file sharing applications – splits large files into smaller chunks.

If we look at the aggregate traffic, P2P traffic has some nice characteristics for ISPs, see Hasslinger (2005). It shows relatively little variability over time as the aggregate peak-to-mean rate over a day is usually smaller than 1.5. Web browsing and other applications typically have a factor of two and more, because of the fact that they are used mainly in the busy hours and less at night.

For web traffic, the popularity of Internet servers can change abruptly, for example, when a new service pack comes out that is downloaded by many systems within a short time or when a web page with previously little attention receives a great deal of attention within short notice, because it is referenced in a popular news magazine. The latter is also called the *Slashdot effect*. In P2P networks, new content quickly becomes more or less uniformly distributed in the network. Therefore, P2P applications lead to a more uniform distribution of traffic sources over the network, independent of sudden changes in the popularity. This makes it easier for ISPs to plan their capacity. P2P traffic is mostly symmetric traffic. Following the argument of Hasslinger (2005), aggregate P2P traffic approaches a Gaussian distribution.

5.2.3 The Future of P2P

The P2P communication paradigm is a powerful communication paradigm and is slowly adapted to other applications as well, because it promises scalability, cost savings, rapid deployment and more. Emerging P2P applications are the VoIP telephony application Skype (www.skype.com), groupware Groove (www.groove.net) or the P2P webcam network Camnet (Liebau *et al.* (2005)); for more applications see Steinmetz and Wehrle

[2] eDonkey was selected because according to Sandvine Incorporated (2003), the eDonkey/eMule network was with 52% of the generated file sharing traffic the most successful P2P file sharing network in Germany at the time of the studies.

[3] Keep in mind that a single client has multiple parallel TCP data transfers in progress at almost all times.

(2005). Therefore, future P2P applications can be expected to show much more variety than today's file sharing applications and their traffic can no longer be assumed 'low priority' or 'unwanted'.

5.3 Online Games

5.3.1 Computer Game Market

The computer game market and especially the online game market is a fast growing market with a tremendous amount of opportunities:

- According to ESA (2005), the computer and video game software sales reached 7.3 billion dollars in the United States of America and roughly 24.4 billion dollar worldwide in 2004.
- In 2005, IDC (www.idc.com) predicted an increase in turnover of 50% per year in the United States. For Asia, the turnover was 761 million dollars in 2003 with a prognosis of 1.84 billion dollars in 2008.
- Jupiter Research (www.jupiterresearch.com) forecasts a growth of the *online game* market in Europe from 96 million EUR in 2003 to 589 million EUR in 2007. Gamers are predicted to pay 79 EUR per month for games.

5.3.2 Classification of Computer Games

Figure 5.3 shows a classification of computer games by the type of game, the device the game is running on, the number of players, the interactivity between games and the network connectivity needed. The aspects of ISPs especially important with respect to QoS requirements are marked in grey: Online real-time games.

Online games can be persistent: If a player logs off for a while and logs on again later, he continues more or less from the previous state (e.g. with his previous character in a role-playing game), while for non-persistent games he typically starts a new gaming session, although certain information like the gamer's previous high scores might be kept.

The most important online games today are Massive Multiplayer Online Role Playing Games (MMORPG) such as Ultima Online, EverQuest and World of Warcraft; Real-time Strategy (RTS) games such as Starcraft and First Person Shooters (FPS, also called *ego shooters*) such as Counterstrike. For MMORPGs and some other online games, customers often pay a monthly subscription fee to be allowed to play online with/against other players. For World of Warcraft, the monthly fee at the time of writing was 14.99 dollars. MMORPGs can have multi-million subscribers.

5.3.3 Online Game Architectures

Some online computer games are played purely peer to peer with communication directly and exclusively between the participating parties; this is typical for most computer-based card and board games. However, most games use a client-server architecture where servers are used to distribute the information, and information exchange directly between the players is uncommon. Servers simplify the synchronisation between a larger number of

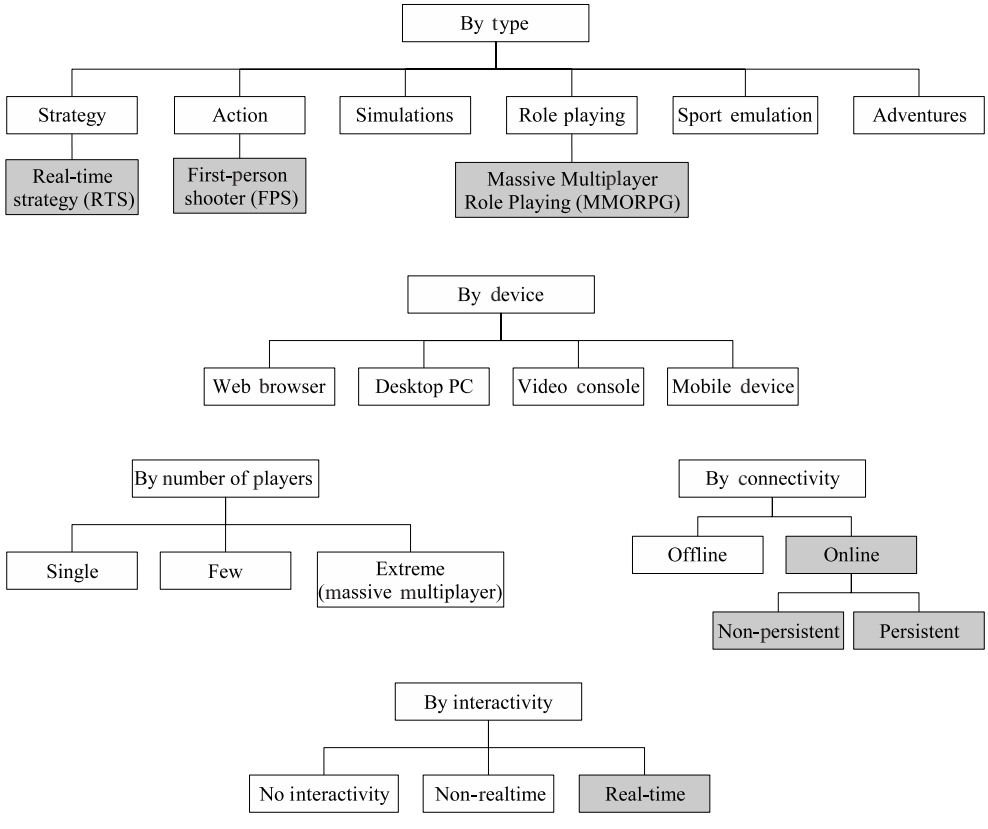

Figure 5.3 Classification of Computer Games

players and can be used to store game information persistently when players go offline. Servers do not necessarily have to be hosted by the producer of the game; in many games the server functionality is included in the game, allowing one of the players to start a session with his computer acting as a server for the duration of the session (Figure 5.4). For MMORPGs, large and widely distributed networks of servers are used to host the game and improve the quality of service by hosting the games with a server close to the gamers.

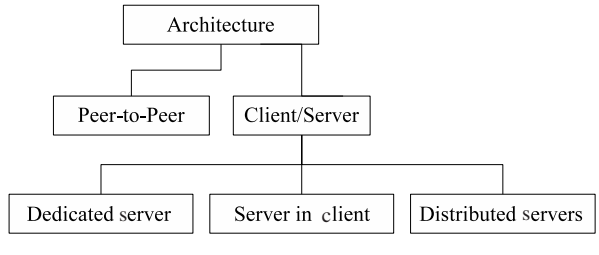

Figure 5.4 Computer Game Architectures

5.3.4 QoS Requirements

The QoS requirements of network games depend strongly on whether they are real-time games or not. Non-real-time games do not have any special QoS requirements. On the other hand, the QoS requirements of real-time computer games depend on the exact type of the game. Table 5.2 lists the results for the recommended upper limit of loss respectively latency from several different studies. The results are consistent with other studies of interactive applications such as those presented in Bailey (1989) that indicate an upper round-trip time of 200 ms for real-time interaction, MacKenzie and Ware (1993) that recommends less than 225 ms latency for interaction in virtual realities or Institute of Electrical and Electronic Engineers (1996) that recommends an upper bound of 300 ms on latency for military simulations; see also Henderson and Bhatti (2003).

As can be seen from Table 5.2, network games are generally more sensitive to delay than loss. The cited studies also show that better players are more affected by the delay in their performance and are generally more aware of QoS degradation. The most sensitive games are action games, especially first person shooters. Increased latency has the highest effect on shooting with precision weapons and only very little effect on actions like moving the game character, see Beigbeder *et al*. (2004). RTS games are relatively insensitive to loss and delay and online role playing games, even more.

5.3.5 Traffic Model

Internet real-time games with little tolerance for latency do not use TCP as transport protocol but use UDP instead to avoid the congestion control behaviour and the delay from retransmissions. This can be seen by comparing the transport protocol in Table 5.3 with the maximum latency in Table 5.2; only the very latency-tolerant MMORGPs use TCP. In addition to more control over the latency, using UDP gives the application full control of retransmitting a lost packet or not.

Table 5.3 lists traffic models for some common network games. Zander and Armitage (2004) list a number of references with advanced traffic models. As can be seen from the table, most of today's network games are designed to operate over dial-up Internet connections and therefore have a throughput of approximately 40–64 kbps. The traffic from the server to the client has significantly larger packet sizes than in the opposite way and has a packet interarrival time of 50 ms for most games.

To reduce the bandwidth requirements of online real-time computer games, game designers use mechanisms like dead reckoning and to reduce the effect of latency, they use mechanisms like buffering and artificial delays for actions on the local machine, time distortion and client predictions.

5.4 Voice over IP

5.4.1 QoS Requirements

Voice over IP (VoIP) applications use the standardised Session Initiation Protocol (SIP) or H.323 signalling protocols or proprietary protocols (like Skype). At the time of writing, SIP seems to be the protocol of choice although Skype also has a very large user base.

Table 5.2 QoS Requirements of Real-time Network Games

Game	Type	Source	Maximum Loss	Maximum Latency
Car racing	Action	Pantel and Wolf (2002)		100 ms
XBlast shooting game	Action	Schaefer *et al.* (2002)		140 ms
Quake 3	Action, FPS	Armitage (2001); Zander and Armitage (2004)	over 10%	150–180 ms
Half-life	Action, FPS	Henderson (2001)		225–250 ms
Halo	Action, FPS	Zander and Armitage (2004)	4%	200 ms
Unreal tournament 2003	Action, FPS	Beigbeder *et al.* (2004)	3%	150 ms
Madden NFL football	Sport emulation	Nichols and Claypool (2003)		500 ms
Warcraft III	Strategy, RTS	Sheldon *et al.* (2003)		800 ms
Everquest II	MMORPG	Fritsch *et al.* (2005)		1250 ms

The ITU G.114 standard recommends a one-way transmission delay up to 150 ms for voice communication, although a one-way delay of 400 ms is still considered acceptable (see International Telecommunication Union (2000)). Callers typically notice the delay if it is 250 ms round trip. The amount of tolerable jitter depends on the buffering strategy on the receiver side; if the jitter is high, more buffering is necessary, which adds to playback latency.

VoIP is not tolerant of packet loss for most codecs. For the 'standard' G.711 codec or the G.729 codec, 1% packet loss significantly degrades a call. Other more compressing codecs are even less robust. Packet Loss Concealment (PLC) or Packet Loss Recovery (PLR) algorithms can increase the acceptable packet loss rate to about 5%.

5.4.2 Traffic Model

The amount of required bandwidth depends on the used codec. Some of the standard voice codecs are G.711, G.729, G.726, G.723.1 and G.728. Bandwidth requirements and packet sizes depend on the codec and the configuration. Typically, a VoIP call will consume 25–100 kbps of bandwidth with 22–100 packets/second and a packet size of 60–200 bytes. G.711 has a 64 kbps voice bandwidth and if sampled every 20 ms the payload of each packet is 160 bytes. With 40 bytes IP/UDP/RTP header this leads to uniformly distributed constant bit-rate (CBR) flow with a bandwidth requirement of 80 kbps on IP layer.

Table 5.3 Traffic Models of Real-time Network Games

Game	Type	Source	Traffic Model
Half-life	Action FPS	Henderson (2001)	UDP, server to client 60–300 byte packet length (depends on the map used in the game) with interarrival times of 50 ms, client to server 60–90 bytes with regular interarrival times between 33 and 50 ms
Unreal tournament 2003	Action FPS	Beigbeder *et al.* (2004)	UDP, 63–70 kbps with a std. dev. of about 10 kbps. Median packet size around 70 bytes. Packet interarrival time server to client 50 ms, irregular in opposite direction (depends on user action)
Madden NFL football	Sport emulation	Nichols and Claypool (2003)	UDP, < 20 kbps/player, < 90 byte packet size (median 77 bytes). For high latencys, packets are aggregated and packet size increases accordingly
Counterstrike	Action FPS	Feng *et al.* (2005); Claypool *et al.* (2003)	UDP, 15–24 kbps/player, client to server average 40 byte packets, server to client average 130 bytes, large periodic bursts every 50 ms
Starcraft	Strategy RTS	Claypool *et al.* (2003)	UDP, 5.2–6 kbps/player, 120 byte median packet size, only small deviation from average packet size, packets sent uniformly over a range of 10–300 ms
Warcraft III	Strategy RTS	Sheldon *et al.* (2003)	UDP, mostly 46 or 49 byte packet size, interarrival rate 200 ms
Lineage II	MMORPG	Kim *et al.* (2005)	TCP, client to server average 59 bytes packet size, server to client average 358 bytes packet size
Everquest II	MMORPG	Fritsch *et al.* (2005)	TCP, client to server average 0.4 kbps (maximum 4.7 kbps), server to client 0.9 kbps (maximum 4.2 kbps)
ShenZhou online	MMORPG	Chen *et al.* (2005)	TCP, 7 kbps/player, 98% of packets smaller than 71 bytes, 30% are TCP acknowledgement

To calculate the necessary bandwidth for aggregate VoIP traffic, different traffic models are suited. Erlang B, extended Erlang B and Erlang C are the most commonly used ones; other models include Poisson and Neal–Wilkerson, see for example, Freeman (2004).

5.5 Traffic Classification

5.5.1 Port-based Traffic Classification

The standard way of identifying which application a data packet belongs to is by looking at the ports in the TCP/UDP header. The TCP/UDP ports can be distinguished into the so-called well-known ports from 0 to 1023, the registered ports from 1024 to 49151, and the dynamic/private ports from 49152 to 65535. A list of assigned ports is available at http://www.iana.org/assignments/port-numbers. The default ports of some important applications are listed in Table 5.4.

Compared to other traffic classification mechanisms, port-based classification is relatively cheap on a high bandwidth link in real time. However, there are ambiguities in the port registration; many applications are not listed in the port directories. In addition, nothing forces an application to use the assigned ports. In fact, many P2P applications allow their user to change the standard port or use random ports straightaway to avoid detection. In addition, many applications besides the web are tunnelled through HTTP (e.g. chat, streaming, P2P).

5.5.2 Advanced Mechanisms

The widespread usage of P2P file sharing applications and the problem of reliably identifying their traffic lead to the works on more advanced traffic classifications as discussed in the next section.

Table 5.4 Standard Ports of Some Applications

Application Protocol	(Main) Transport Protocol	(Main) Standard Ports
HTTP, HTTPS	TCP	80, 443
FTP	TCP	20, 21
Telnet	TCP	23
SSH	TCP	22
SMTP	TCP	25
POP, POPS	TCP	110, 995
IMAP, IMAPS	TCP	143, 993
DNS	UDP/TCP	53
Skype VoIP	TCP/UDP	Random and 80, 443
eDonkey P2P	TCP/UDP	4661–4665
Kazaa P2P	TCP/UDP	1214
BitTorrent P2P	TCP	6881–6889
Gnutella P2P	TCP/UDP	6346–6347

5.5.2.1 Signature Detection

Signature-based detection techniques are used in the context of network security and intrusion detection. Signatures can also be used for traffic classification. Sen *et al*. (2004), for example, present a traffic classification mechanism for P2P applications that uses application signatures. The signatures are application-specific bit patterns that occur in the payload of packets. A flow is classified depending on which signatures are identified in its packets.

Payload inspection has the drawback that it involves looking into the payload. This is costly and might involve legal considerations in some countries. Unknown applications cannot be classified and if the payload is encrypted, the method fails. Furthermore, some P2P protocols and other applications use HTTP requests and responses and can therefore not be distinguished from normal WWW traffic with this method. Despite all these drawbacks, payload inspection is the most common of the advanced techniques.

5.5.2.2 Traffic Statistics

Roughan *et al*. (2004) presents a statistical supervised learning approach for general traffic classification. It does not aim at identifying the exact application protocol; instead, it aims at identifying whether the application is interactive, a bulk transfer, streaming or transactional. The same application – for example a web browser – can be used for interactive transfers of web pages as well as for the bulk transfer of, for example, the latest Linux distribution CD image. In both cases, the same protocol (HTTP over TCP) is used; however, the QoS requirements differ. The approach of Roughan *et al*. (2004) promises to identify the QoS requirements of the flow more or less independent of the application protocol. To do so, traffic statistics such as the packet size, flow duration, bytes per flow, packets per flow, and so on, are used to classify a new flow into predetermined categories.

Moore and Zuev (2005) propose a Bayesian analysis that requires hand-classified network data as input. Zander *et al*. (2005) use machine learning techniques for self-learning traffic classification mechanisms.

Karagiannis *et al*. (2004) use two heuristics to identify P2P traffic in traffic traces. The first uses the fact that many P2P applications use TCP and UDP at the same time while few non-P2P applications do so. The second heuristic looks at the source/destination IP/port pairs. Web traffic has a higher ratio of the number of distinct ports versus the number of distinct IP addresses than P2P traffic. The mechanism is good for offline traffic characterisation. An extension of this concept is discussed in Karagiannis *et al*. (2005).

5.6 Summary and Conclusions

In this chapter, four important traffic types were discussed. Web traffic is based on the HTTP protocol and is mostly interactive traffic. P2P traffic is mainly caused by file sharing applications. It is mostly bulk transfer traffic and makes up the largest amount of traffic in today's Internet. As online games are becoming more and more important and as the

important online games have special quality-of-service requirements, this traffic type will need increased attention of ISPs in the future. Voice over IP traffic is currently exploding and becoming one of the major traffic sources. It also has special QoS requirements. At the end of this chapter, methods for determining the application or application type of a traffic flow in real time were discussed. The drawbacks of port-based classification, which is mainly used today, were pointed out and advanced concepts were discussed.