



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

HGI-RD010-R3  
Home Gateway requirements  
for multiple session support

April 29, 2010

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29

PAGE LEFT INTENTIONALLY BLANK

## 1 Table of Contents

2	Table of Contents .....	3
3	1 Important notice, IPR statement, disclaimer and copyright.....	4
4	2 Acronyms.....	5
5	3 Definitions.....	6
6	4 Scope and purpose of this document.....	7
7	5 HG stability issues caused by P2P applications .....	8
8	5.1 Peer-to-peer networks and applications .....	8
9	5.2 HG stability issues .....	8
10	6 Sessions, flows and connections .....	10
11	6.1 An IP session (L3) .....	10
12	6.2 Higher-layer sessions, flows and connections .....	10
13	6.3 Transport layer sessions .....	12
14	6.3.1 TCP session.....	12
15	6.3.2 UDP session .....	12
16	7 Example peer-to-peer application .....	14
17	7.1 Description of the application .....	14
18	7.2 Application issues.....	16
19	7.2.1 Application imposed limits.....	16
20	7.2.2 Deep Packet Inspection (DPI).....	16
21	7.2.3 General session detection and counting.....	16
22	7.2.4 Releasing inactive transport layer sessions.....	17
23	7.2.5 Identification of UDP sessions .....	18
24	7.2.6 Supporting higher priority flows .....	18
25	8 Requirements .....	19
26	8.1 Generic HG requirement for supporting multiple transport layer sessions.....	19
27	8.2 Counting active transport layer sessions.....	19
28	8.3 Releasing inactive transport layer sessions .....	19
29	8.4 Setting the maximum number of transport layer sessions .....	19
30	8.5 Admittance of high-priority transport layer sessions.....	20
31	9 References .....	21
32		
33		

## 1 Important notice, IPR statement, disclaimer and 2 copyright

3 The Home Gateway Initiative (HGI) is a non-profit making organization created to define  
4 guidelines and specifications for broadband home gateways.

5 This document is the output of the Working Groups of the HGI and its members as of the  
6 date of release. Readers of this document must be aware that it can be revised, edited or have its  
7 status changed according to the HGI working procedures.

8 The HGI makes no representation or warranty on the contents, completeness, and accuracy  
9 of this publication.

10 This document, though formally approved by the HGI member companies, is not binding in  
11 any part on the HGI members.

12 IPRs essential or potentially essential to the present document may have been declared in  
13 conformance to the HGI IPR Policy and Statutes available at the HGI website  
14 [www.homegateway.org](http://www.homegateway.org).

15 Any parts of this document may be freely reproduced (for example in RFPs and ITTs) by  
16 HGI and non-HGI members subject only to the following:

- 17 • HGI Requirement numbers not being changed
- 18 • an acknowledgement to the HGI being given in the resulting document.

19 Trademarks and copyrights mentioned in this document are the property of their respective  
20 owners.

21  
22 The HGI membership list as of the date of the formal review of this document is: Alcatel-  
23 Lucent, Applied Micro, Arcadyan, Atheros, AVM, Belgacom, Broadcom, BT, Cisco, Comtrend,  
24 Deutsche Telekom, D-Link Corporation, DSP Group, Echelon EMEA, Ericsson AB, Fastweb SpA,  
25 France Telecom, Freescale Semiconductor, Gige Semiconductor, Huawei, Ikanos, Intel, JDSU,  
26 KDDI, KPN, LG-Nortel Co Ltd, Marvell Semiconductors, Mindspeed, Mitsubishi, NEC  
27 Corporation, Netgear, NTT, Philips, Ping Communication, Pirelli Broadband Solutions, Portugal  
28 Telecom, Sagem, Samsung, Sigma, SoftAtHome, Spidcom, Sumitomo, Swisscom AG,  
29 Technicolor, Telecom Italia, Telefonica, Telekom Malaysia, Telekom Slovenije, Telekomunikacija  
30 Polska, Telenor, TeliaSonera, Telstra, Tilgin AB, TNO, Vodafone, Vtech, Zarlink, ZTE, ZyXEL

31

## 12 Acronyms

2

ALG	Application Level Gateway
CPU	Central Processing Unit
DHT	Distributed Hash Tables
DPI	Deep Packet Inspection
FTP	File Transfer Protocol
GUI	Graphical User Interface
HG	Home Gateway
HGI	Home Gateway Initiative
HN	Home Network
HTTP	Hyper Text Transport Protocol
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IPR	Intellectual Property Rights
P2P	Peer to Peer
PPP	Point-to-Point Protocol
QoS	Quality of Service
RMS	Remote Management System
SDO	Standards Development Organization
SIP	Session Initiation Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator

3

## 13 Definitions

2**Application**: a function running at the application layer and using a number of sessions for  
3communication.

4**Application layer session**: A session that is maintained by applications. When it starts and stops  
5is determined at the application layer. An application layer session makes use of one or more  
6transport layer sessions.

7**BitTorrent technology**: technology using a peer-to-peer network to allow a large number of peer  
8entities to communicate with each other in order to obtain files, using a technique whereby each  
9peer sends a different piece of the file. At a peer, the received pieces of the file are merged  
10together until the file is complete. BitTorrent starts by tracking the remote peer entities that offer a  
11particular file requested by a peer end device.

12**Connection**: the combined flows between a particular endpoint and a particular remote endpoint  
13for a particular application.

14**Flow**: A sequence of packets from a source to a destination.

15**Home Gateway**: device connecting the HN to the Internet and Service Platforms [1].

16**IP session**: the period of time during which an entity is able to communicate with an IP based  
17environment. It starts after a physical entity has requested and obtained an IP address, together  
18with IP configuration parameters, from a network service provider (usually after authentication), and  
19terminates when the IP address is released.

20**Peer-to-peer application (e.g. BitTorrent)**: an application (located at L7 of the OSI layer) running  
21on a peer (in the context of BitTorrent technology). There are many different programs which  
22implement BitTorrent technology.

23**Peer-to-peer (P2P) network**: network connecting end-devices (called peers) directly as opposed to  
24via a server. Peer-to-peer techniques are used for many purposes, but in particular for sharing  
25content and for real-time end-to-end services like telephony. There are many types of peer-to-peer  
26network.

27**Session**: an information exchange between devices that is established at a certain time and torn  
28down at a later time, generally in a controlled fashion. Examples are TCP sessions, UDP sessions  
29and application sessions such as Web session (HTTP sessions) and SIP sessions.

30**Transport layer session**: a TCP- or a UDP-session.

31

## 14 Scope and purpose of this document

2 This document discusses peer-to-peer traffic in general, and BitTorrent traffic as a particular  
3 example of this type of traffic, and the impact this has on an HG. Definitions of sessions,  
4 connections, and flows are given. It is shown that P2P BitTorrent technology can use an almost  
5 unlimited number of sessions. The document describes what happens in an HG when it is  
6 overloaded with P2P traffic and then moves on to make recommendations on how to manage this  
7 application type.

8 The document demonstrates that the unconstrained proliferation of P2P traffic must be  
9 avoided. Some solutions for achieving this are presented and examined. Finally, a number of  
10 requirements for the HG relating to this problem are formulated and presented.

11 It must be stressed that when using “BitTorrent” in this document, reference is always to the  
12 BitTorrent technology, and never to a particular implementation of that technology. The BitTorrent  
13 specification can be found in [2].

14

## 15 HG stability issues caused by P2P applications

### 25.1 Peer-to-peer networks and applications

3 Many applications are client-server based. This means that one end device requests a  
4service, the so-called "client," and a remote central server (which can be another end device or a  
5dedicated computer) responds to the request. Examples are E-mailing, Web browsing and FTP file  
6uploading/downloading.

7  
8 The Internet provides a means of connecting "any device to any device". By its very nature,  
9then, the Internet is a P2P network. In P2P, any peer can be a client or a server at any time, and  
10may be both simultaneously. A pure P2P communication network supports 'equal' peer devices that  
11simultaneously function as both "clients" and "servers".

12  
13 Note that there also exist hybrid P2P models, where a number of servers are using peer to  
14peer for server intercommunication, but which are also acting as central servers to clients.

15  
16 P2P networks can be classified according to their usage, for example:

- 17 • content delivery
- 18 • file sharing
- 19 • telephony
- 20 • chat

21  
22 In this document, we primarily focus on the use of peer-to-peer applications for file sharing.  
23However, the problems caused by peer-to-peer traffic in the HG apply to any form of peer-to-peer  
24traffic that increases the number of flows or sessions in an uncontrolled fashion.

### 265.2 HG stability issues

27 An HG is in general connected to multiple home devices, using a number of technologies.  
28The resulting HG traffic can vary from almost none (e.g. just keep-alive signals and management)  
29up to a large number of flows that can load the CPU and memory of the gateway to the point where  
30the HG system slows down, and may ultimately crash. Overload symptoms include slow web-  
31interface, pixelated IPTV, slow transfer of data (e.g. during browsing), not responding to user and  
32network requests, crashing or even rebooting.

33 The performance degradation depends on both the amount and nature of the incoming and  
34outgoing traffic, and the system dimensioning in terms of processing power and memory. Reaching  
35the point where the HG slows down significantly or crashes must of course be avoided.

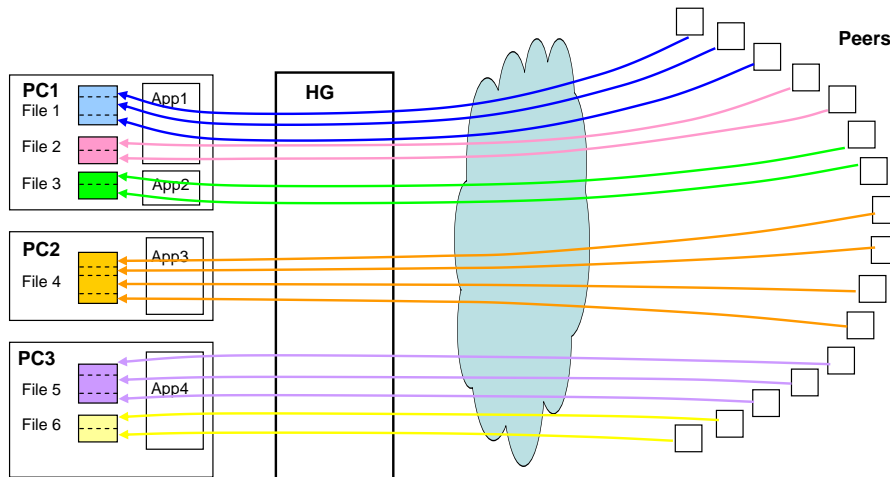
36 Peer-to-peer services are one of the main causes of session proliferation. These services  
37are used in both the residential and business environment. Specific examples are BitTorrent itself,  
38µTorrent, EMule, Azareus, but there are many others.

39 The number of flows through a given Gateway will depend on the number of:

- 40 • peer sources available from which to download a given file
- 41 • simultaneous downloads per single BitTorrent application
- 42 • simultaneous BitTorrent applications for a given user

- 1 • simultaneous BitTorrent users in a home
- 2 • files being sourced to requesting peers outside the home

3 All these lead to traffic and multiple sessions through the HG. The gateway can therefore  
 4 become a bottleneck for the incoming and outgoing traffic. The following figure shows the situation  
 5 from the HG viewpoint with regard to downloading. However, there are equivalent flows upstream  
 6 to the related peers. Note that the number of file pieces and hence peers can be much larger than  
 7 shown in this figure.



8  
 9 *Figure 1 Peer-to-peer flows through the HG*

10 The figure shows 3 PCs connected to an HG. PC1 is running 2 different BitTorrent  
 11 applications. The first application is downloading file1 from 3 peers, and file 2 from 2 peers. The  
 12 second application is downloading file 3 from 2 peers. Again note that there are also uploads  
 13 (possibly even uploading pieces of the same file that is being downloaded).  
 14

15  
 16

## 16 Sessions, flows and connections

2 The term 'session' may be interpreted in several ways. Therefore this section defines a  
3 number of terms that will be used when identifying the problems related to peer-to-peer  
4 applications. Definitions are given for session, connection, and flow, all in the context of typical  
5 client-server connections.

6

### 76.1 An IP session (L3)

8 Any entity in an IP network can only communicate if it has an IP address. An IP address is  
9 usually obtained during the device boot-up procedure. Common methods/protocols to get an IP  
10 address are the Point-to-point Protocol (PPP) and DHCP. The procedure is called IP configuration,  
11 and the entity not only obtains an IP address (private or public), but also a set of IP configuration  
12 parameters.

13

14 Once an entity has an IP address it can communicate with IP networks including the  
15 Internet. Once the entity is on an IP network, it is able to exchange traffic with other entities  
16 (servers, devices) attached to that IP network. While actively connected to an IP network, the entity  
17 has an **IP session**, and within this IP session a number of TCP sessions or application sessions  
18 can be started. The IP session remains until the assigned IP address is released.

19

20 An IP session is mainly related to control and configuration. Once the IP configuration is  
21 done and access to the IP based network is achieved, little if any additional traffic is generated  
22 relating to the IP session itself. As such, there is NO requirement with regard to limiting IP  
23 session(s). An IP session does not consume HG processing resources (except for occasional  
24 activities such as DHCP lease renewal). There are also some IP control flows (e.g. IGMP, ICMP)  
25 but their impact is also negligible.

26

### 276.2 Higher-layer sessions, flows and connections

28 To understand application traffic in terms of sessions at higher layers, flows and connections  
29 need to be considered. A number of application layer endpoints (e.g. a 'source' and a 'destination')  
30 are involved with applications that require communication over an IP network.. Each endpoint is  
31 connected to a remote endpoint. An information transfer, usually a sequence of packets, between  
32 an endpoint and a remote endpoint is called a **flow**. A flow is characterized by endpoints and  
33 direction. A **connection** consists of the combined flows between a particular local endpoint and a  
34 particular remote endpoint for a particular application. A connection may contain one or more flows,  
35 each flow in a connection being associated with the same application.

36 The possible flow types in a connection are:

- 37
- 38 • A flow from the local endpoint to the remote endpoint
  - 39 • A flow from the remote endpoint to the local endpoint
  - 40 • Flows in both directions between the endpoints, belonging to a single application- or  
TCP-session

41 Such flows occur within an IP session, and are characterized by:

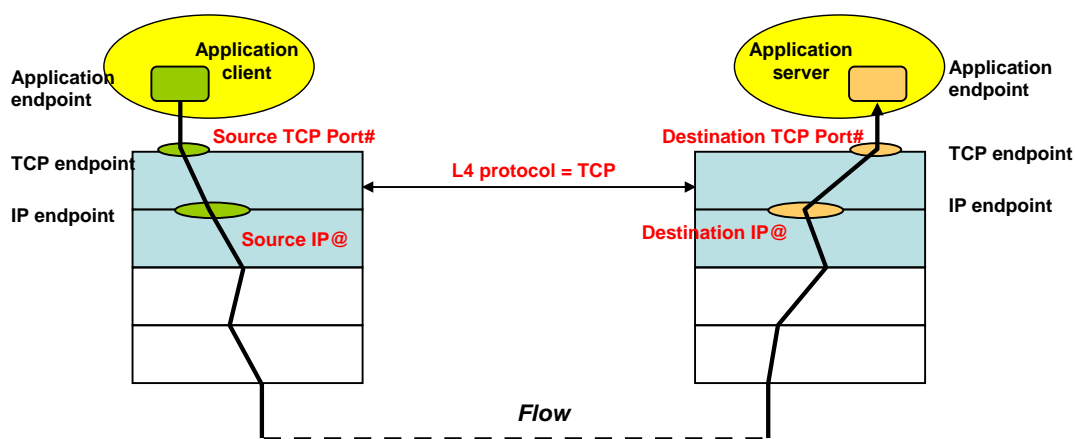
- 1           • The destination and source IP address
- 2           • The transport layer (L4) protocol used: in general either TCP or UDP
- 3           • The transport layer port number: a number of such ports have a default value for
- 4           commonly used protocols (HTTP, FTP etc.)

5           These 5 parameters (source IP-address, destination IP-address, transport layer protocol,  
6source transport layer port, and destination transport layer port) uniquely identify a flow. This set of  
7parameters is called the 5-tuple. Figure 2 shows a flow communication and its 5-tuple (in red). This  
8flow, together with the reverse flow, forms a bidirectional connection between the local and the  
9remote endpoint.

10

11           An application session may or may not be based on TCP sessions, and may include one or  
12more connections.

13



14

15

16

Figure 2 The 5-tuple identifying a flow

17

18           The FTP application is shown as an example in Figure 3. The application session consists of  
19both data endpoints and control endpoints in the ftp-client and the ftp server. These connections  
20use the TCP protocol. The FTP server has been assigned some default port numbers, 20 for the  
21data and 21 for the control. The application uses 2 TCP sessions, each session containing a flow in  
22both directions.

23

24           Note that an application may contain both TCP and UDP sessions, e.g. a SIP application  
25may use TCP for SIP signaling and UDP for voice/video transmission.

26

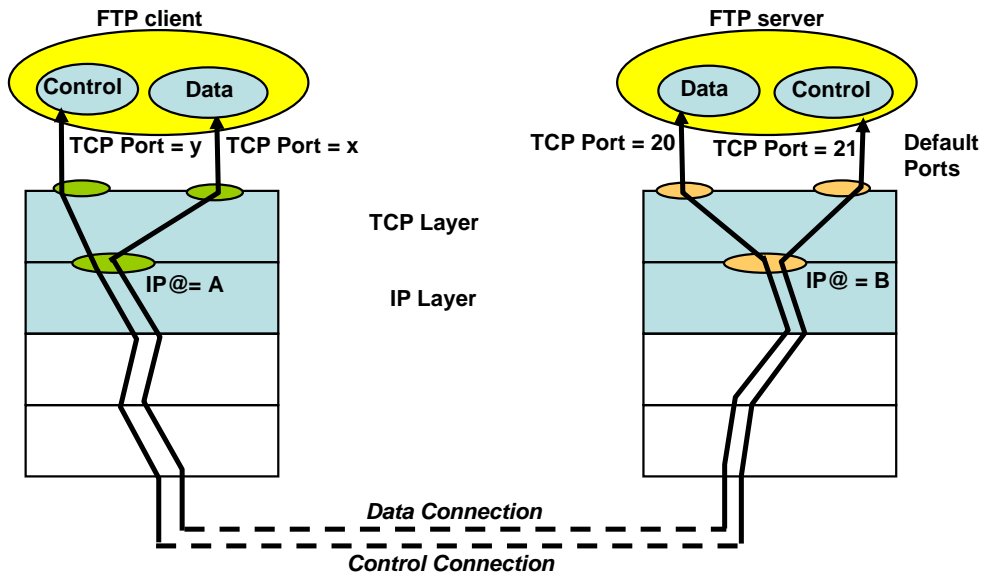


Figure 3 The FTP application session and its 2 TCP sessions

1  
2  
3

### 46.3 Transport layer sessions

#### 56.3.1 TCP session

6 A TCP session may be set up and released by the application to which it belongs. It is  
7 identified by a 5-tuple where the transport layer protocol is TCP. The setup and release are shown  
8 in Figure 4.

9

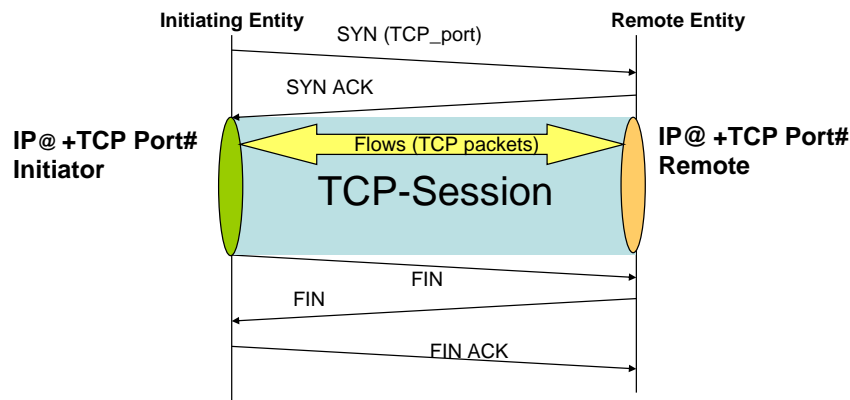


Figure 4 TCP session setup and release

10  
11  
12

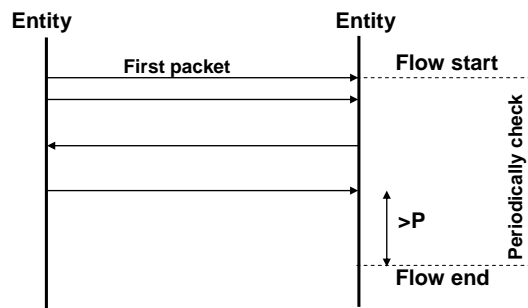
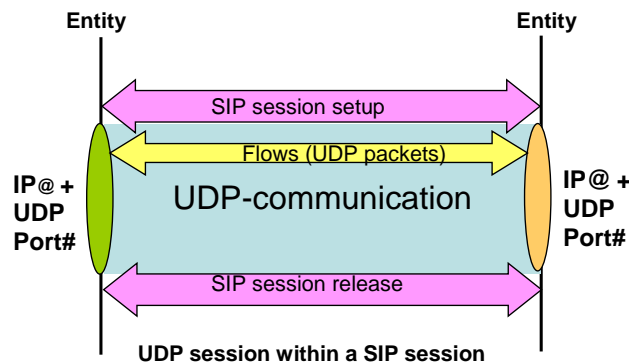
#### 136.3.2 UDP session

14 A UDP session is also identified by a 5-tuple but here the transport layer protocol is UDP.

1 However, unlike a TCP session, a UDP session is not set up or released using session control  
 2 messages. So another method is needed to identify a UDP session; two possibilities are as follows:

- 3
- 4 1. A flow of UDP packets will occur within an application session. One can recognize a UDP  
 5 session after the application session has successfully negotiated a UDP connection. The  
 6 UDP session can be deemed to start when the application session sends the first packet  
 7 identified by the 5-tuple, and the UDP session ends when the application session ends and  
 8 packet transfer stops. An example of such an application session is a SIP session. The SIP  
 9 protocol can negotiate UDP connections using SDP. Typically, voice connections use UDP  
 10 for streaming the voice packets while TCP is used to set up and tear down the call (and  
 11 terminate the UDP connection/session).
  - 12 2. The HG system detects the first packet of **any** new UDP flow (within an application session  
 13 or not). The HG decides that it is a new session based on the new 5-tuple. A mechanism is  
 14 also needed for an HG to conclude that a UDP session has ended. This is based on  
 15 inactivity detection which uses a timer. Note that this type of mechanism may also be used  
 16 for a TCP session. Indeed, one should not assume that all TCP sessions are terminated  
 17 gracefully using appropriate session release messages.

18  
 19  
 20 These two methods of UDP session identification are shown in Figure 5. A UDP session  
 21 only has a single UDP flow.



UDP session identified by first packet and inactivity check

22  
 23 *Figure 5 Two methods of UDP session start and end: application controlled and inactivity-*  
 24 *timer based.*

## 17 Example peer-to-peer application

2 This section describes BitTorrent technology in more detail, the problems that can be  
3 caused in a home gateway when BitTorrent applications are used, and ways to avoid these  
4 problems. The following description of BitTorrent technology uses the definitions of  
5 session/flow/connection established in the previous section.

6  
7 The context is as was shown in Figure 1, focusing on the HG dealing with a number of  
8 flows/sessions from attached home devices, usually PCs, running multiple BitTorrent applications  
9 and performing multiple simultaneous downloads/uploads which all have to be transferred through  
10 the HG.

11

### 12 7.1 Description of the application

13 BitTorrent is a peer-to-peer file sharing protocol used to distribute data files between  
14 participants. The initial distributors of the complete file, as well as subsequent distributors, are  
15 known as 'seeders'. Each peer that downloads data can also upload data to other peers. This is a  
16 sharing service that has the advantage that peers are not dependent upon the presence of the  
17 initial distributor. The disadvantage is however the increasing, and potentially large, number of  
18 flows. As long as a client does not have the complete file, and is still in the process of downloading,  
19 this peer is not called a seeder, but a leecher. In this document both seeders and leechers are  
20 referred to as 'peers'.

21

22 A BitTorrent client is a program that implements the BitTorrent protocol, and presents a user  
23 interface. Each client is capable of preparing, requesting, and transmitting any type of file over the  
24 IP network. A peer is any computer running an instance of a BitTorrent client. To share a file or  
25 group of files, a client first creates a small file called a "torrent." A torrent contains the URL of the  
26 tracker, the computer that coordinates the file distribution. It also contains metadata about the file(s)  
27 to be shared. A BitTorrent client that wants to download the file first obtains a related torrent file,  
28 and connects to the torrent-specified tracker, in order to get information about other peers offering  
29 the file. A tracker is a peer acting as a server that maintains a list of available files and the peers  
30 that have these files

31

32 A BitTorrent download is based on many small data requests over different TCP sockets,  
33 each request getting just a piece of the file. All the pieces are merged in the right order at the client.  
34 This differs from a sequential file download from one single location, such as is done via web  
35 browsers or FTP. Peer-to-peer downloads can take time to rise to full speed because of the time for  
36 multiple peer connections to be established.

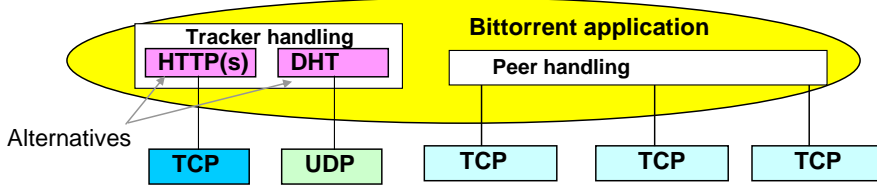
37

38 Figure 6 shows the structure of a BitTorrent application, as well as the BitTorrent session as  
39 seen by a particular BitTorrent client. The application gets (via HTTP) lists of peers from the  
40 tracker(s) identified in the torrent information. TCP sessions are set up to each peer using their  
41 various URLs. A particular piece of the file is downloaded from each peer, and the BitTorrent  
42 application puts the pieces together in the correct order. While downloading, some of the pieces  
43 may also be uploaded to other peers. Therefore, the communication in the TCP sessions can be  
44 bidirectional. Note that there exists an alternative method for tracking, using a method called DHT  
45 (Distributed Hash Tables), which does not use TCP, but rather UDP sessions. DHT mode sends  
46 thousands of UDP packets that can quickly overflow the table keeping track of connections (called  
47 ip\_contrack-table in Linux routers). Each connection requires about 300 bytes of memory.

1  
2  
3

The layered communication setup for the download/upload is as in Figure 7.

**Bittorrent application structure**



**Bittorrent session**

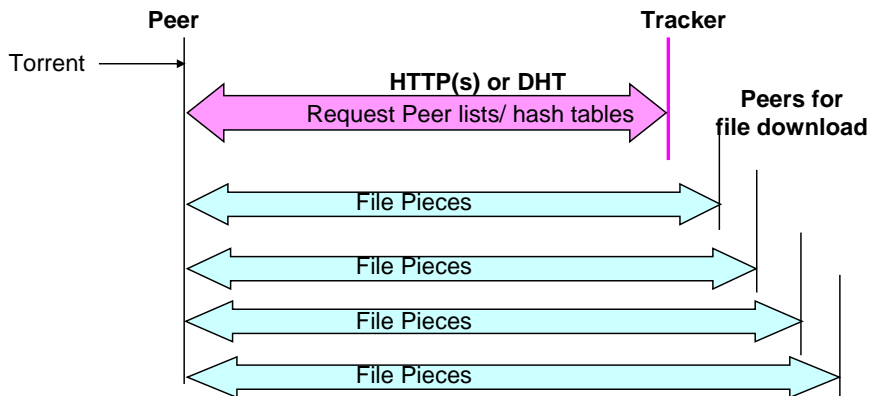


Figure 6 A BitTorrent application structure and session

4  
5  
6  
7

Ports reserved for BitTorrent are typically 6881-6889

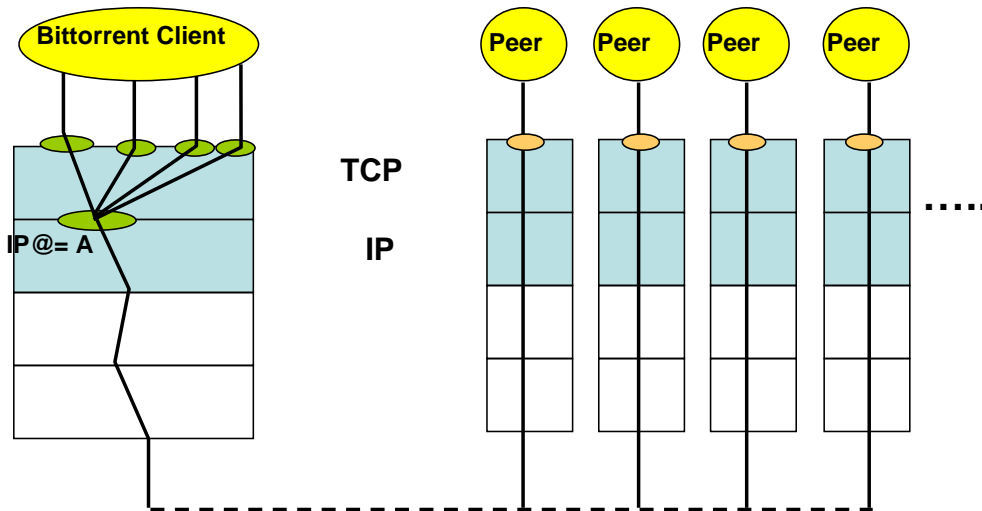


Figure 7 BitTorrent requires a high number of transport layer sessions

8  
9  
10

## 17.2 Application issues

2 While most applications only use a limited number of transport layer sessions, BitTorrent  
3 application(s) may use a much larger number, to the point that it can cause the HG problems.

4

5 BitTorrent application flows are in general categorized as best-effort traffic, which means  
6 that the BitTorrent traffic will slow down if flows of higher priority are present, **and** some priority  
7 scheme is being used. However the BitTorrent application still consumes HG resources by  
8 occupying ports, memory, and CPU time. Even when sessions are maintained that do not have  
9 active flows (for instance when a transport layer session is not properly terminated by the  
10 application), HG resources are consumed.

11

12 There are a number of ways of avoiding the unconstrained proliferation of transport layer  
13 sessions, which are discussed in the following sections.

14

### 157.2.1 Application imposed limits

16 BitTorrent applications themselves have settings that can limit not just the number of flows  
17 but also the downstream and upstream bandwidth. This is done on a per application, and therefore  
18 per PC, basis. This may help, but requires the user to tune each application. The HG may still  
19 suffer if there are a large number of connected PCs/applications, and the HG cannot control or  
20 impose any limits on the attached devices.

21

22 The BitTorrent application specification [2] defines a default peer list limit of 50 entries, and a  
23 maximum of 30 TCP sessions per file download. However, these limits are still too high to avoid HG  
24 problems, especially when there are multiple downloads and/or application instances.

25

### 267.2.2 Deep Packet Inspection (DPI)

27 A maximum could be set on the number of BitTorrent flows through the HG. However, this requires  
28 DPI which is a form of packet filtering that examines the data and header of a packet. BitTorrent  
29 packets could therefore be identified, and so subjected to some kind of rate limitation. However DPI  
30 is an expensive feature which is not available in current HGs.

31

### 327.2.3 General session detection and counting

33 There are 2 types of IP routers, stateless or stateful. Routers with static IP filters and port  
34 forwarding tables are stateless. They perform filtering and forwarding from the external network to  
35 hosts on the internal network on a packet by packet basis, based upon criteria contained within  
36 each packet and a configuration set by remote or local management. A stateless router filters and  
37 forwards packets, but does not keep track of flows.

38

39 More intelligent and secure routers with dynamic IP address and packet based filtering are  
40 known as stateful. Stateful routers can also have application proxies (such as ALGs). Stateful  
41 routers have more per packet processing overhead. Most routers in current HGs are stateful. Note  
42 that being stateful is tightly linked to the firewall used in conjunction with the router. A router with a  
43 stateful firewall is a stateful router. A stateful router keeps track of all the active transport layer

1 sessions by detecting and identifying related flows. It therefore knows how many transport layer  
2 sessions are active at a particular time.

3  
4 A stateful router will monitor TCP signalling exchanges (client requests and server  
5 responses) to determine the current state of a session. Releasing an inactive session will free up  
6 router resources and memory associated with that session. As a result, newly arriving packets  
7 arriving after session release will be dropped by the router.

8  
9 UDP does not use signalling to establish sessions, but the router may observe outgoing  
10 (from the LAN to the WAN) UDP messages and establish state in the router that will permit  
11 incoming messages to be forwarded to the LAN. Releasing an inactive UDP session thus has a  
12 similar effect to releasing an inactive TCP session.

13  
14 Limiting the total number of transport layer sessions in an HG regardless of application type  
15 will automatically provide a way of controlling BitTorrent traffic. BitTorrent applications will not get  
16 any more session requests granted when the maximum number of sessions in the HG is reached.  
17 The maximum number of sessions may be based on the available CPU power and memory. The  
18 processing capability is different from product to product, and also dependent on the number of  
19 interface ports and services provided by the HG. As such, a 'general' safe maximum cannot be  
20 defined for all HGs. The limit is specific to sessions, and should be embedded in the software. It  
21 should not be user configurable, as there is a real danger that the user sets the value too high.

22  
23 This solution in an HG with a stateful firewall/router only requires a counter of transport layer  
24 sessions and a threshold value.

25

#### 26 **7.2.4 Releasing inactive transport layer sessions**

27 It would be possible to release transport layer sessions containing flows with very little, but  
28 non-zero, traffic on the grounds that system resources might be better employed on more active  
29 sessions.

30  
31 Although this removal of low rate flows is a possible way of dealing with excessive  
32 flow/connection demands, it presumes that the low rate flows are indeed less important, which may  
33 not be the case, and there is a risk is that the HG will enter a loop situation where sessions are  
34 released and immediately re-requested. Therefore it is safer to only release sessions which are  
35 completely inactive.

36  
37 Because of the different nature of applications based on TCP and UDP, different dynamics  
38 for session release should be observed. TCP sessions may be valid (though not active) for up to 24  
39 hours, whereas it is commonly assumed that UDP sessions which do not show activity for more  
40 than typically 2 hours can be safely removed. We therefore specify the release of TCP sessions  
41 after 24 hours of inactivity (which is already done by some gateways) and UDP sessions after a  
42 configurable time between 10 minutes and 2 hours.

43  
44 This mechanism is meant to release sessions that are truly inactive, and which have not  
45 been properly closed by the underlying application. Unfortunately the mechanism does not prevent  
46 an apparently inactive session being terminated when the underlying application may still be  
47 working properly. Operators should therefore configure the UDP session cleanup timer carefully so  
48 as to minimize the number of improperly interrupted applications. An alternative would have been to  
49 reserve a small number of sessions for IPTV, VoIP etc., and leave them outside the monitoring

1 mechanism. This approach was not adopted to avoid needing a session timer per application, and  
2 avoid requiring higher-layer mechanisms to determine the application behind each flow.

3

#### 47.2.5 Identification of UDP sessions

5 In the absence of application layer UDP session control, a session is taken to start when the  
6 first packet is sent to a remote endpoint. When no activity is detected during a pre-configured time,  
7 the session is released.

8

9 Any new UDP packet using a particular 5-tuple, must be counted as one flow, and therefore  
10 as a single UDP session, either until a related inactivity check is positive, or the application to which  
11 it belongs terminates it. This method for identification of UDP sessions by identification of UDP  
12 flows is needed in order to be able to count the total number of transport layer sessions (TCP +  
13 UDP) in the correct way.

14

#### 157.2.6 Supporting higher priority flows

16 It is good practice to not allow all flows to be BitTorrent flows even when the maximum  
17 number of transport layer sessions is limited to protect the HG.

18

19 There is a need to ensure there is 'always' capacity for flows that have a higher priority.  
20 Some applications are more important than others e.g. video applications are generally more  
21 important to the user than the BitTorrent application, while voice may have an even higher  
22 importance. It is therefore sensible to reserve a number of sessions for higher priority services.

23

24 Figure 8 shows the mechanism devised by the HGI to achieve this. The number of active  
25 sessions is counted and results in a value  $J$ .  $J$  is increased when sessions are added, and  
26 decreased when sessions are released. When  $J$  reaches a certain number ( $M-N$ ), additional  
27 sessions are only admitted if, according to some classifier, they are identified as high-priority. This  
28 can continue as long as the number of sessions does not reach an absolute maximum  $M$ . At that  
29 point no more sessions are allowed. When the number of sessions decreases below  $M-N$ , sessions  
30 can be allowed again without using the high-priority classification check.

31



32  
33  
34  
*Figure 8 HGI mechanism for limitation of sessions*

## 18 Requirements

2 Based on the preceding analysis, the HG requirements for multiple session control are given  
3 below. They apply to transport layer sessions and provide a mechanism for the control of peer-to-  
4 peer traffic in the HG, so that HGs are more stable. The names of the parameters are illustrative at  
5 the time of publication.

6

### 7 8.1 Generic HG requirement for supporting multiple 8 transport layer sessions

N°	Requirement
R1.	The HG MUST be able to support a minimum of 2000 concurrent transport layer sessions, any or all of which may be active.

9

### 10 8.2 Counting active transport layer sessions

N°	Requirement
R2.	The HG MUST keep a count, $J$ , of the total number of currently active transport layer sessions.
R3.	Any packet to a new unique 5-tuple MUST be counted as 1 new transport layer session.
R4.	When the HG releases a transport layer session, all the state in the HG associated with that session MUST be removed and $J$ MUST be decremented by 1.
R5.	$J$ MUST be able to be read by the RMS as well as by the LM Remote UI of the HG.

11

### 12 8.3 Releasing inactive transport layer sessions

N°	Requirement
R6.	When no traffic has been sent in a given TCP session during a period of 24 hours, the HG MUST release that session.
R7.	When no traffic has been sent in a UDP session during a period $P$ , the HG MUST release that session.
R8.	$P$ MUST be configurable by the RMS, in the range of 10 minutes to 120 minutes.

13

### 14 8.4 Setting the maximum number of transport layer 15 sessions

N°	Requirement
R9.	The HG MUST be able to limit the maximum number of transport layer sessions to $M$ .
R10.	$M$ MUST be configurable and readable by the RMS only.

N°	Requirement
R11.	<i>M</i> MUST be pre-configured by the HG vendor to the maximum number of transport layer sessions that the HG supports under normal operating conditions.
R12.	<i>M</i> MUST be configurable from 10 up to the maximum number of transport layer sessions that the HG is able to support.

1

## 28.5 Admittance of high-priority transport layer sessions

3

N°	Requirement
R13.	The HG MUST be able to classify a new transport layer session as High_Priority or Low_Priority on the basis of the packet classification as defined in sections 8.10.1.1, 8.10.1.3 and 8.10.1.6 of HGI-RD001-R2.
R14.	<i>N</i> is the maximum number of transport layer sessions that may additionally be admitted when $J = M - N$ . <i>N</i> MUST be configurable and readable by the RMS only.
R15.	When $J < M - N$ , the HG MUST admit any new transport layer session.
R16.	When $M - N \leq J < M$ , the HG MUST only admit new transport layer sessions which are High_Priority.

4

## 19 References

- 2[1] *HGI-RD001-R2*: Home Gateway Technical Requirements: Residential Profile V1.01
- 3[2] The BitTorrent specification v1.0 <http://wiki.theory.org/BitTorrentSpecification>
- 4