Research White Paper

WHP 183

June 2010

High performance file transfer over IP networks

Peter Brightwell

BRITISH BROADCASTING CORPORATION

BBC Research White Paper WHP 183

High performance file transfer over IP networks

Peter Brightwell

Abstract

This White Paper outlines how broadcasters will increasingly need to transfer large amounts of content as files over IP networks. It examines some of the performance issues that can arise in typical scenarios, as well as other important aspects such as interoperability, security and management. It outlines some of the tools (both commercial and open source) that are available to help file transfers, and discusses what broadcasters should consider when making technology choices.

This document was originally published in November 2009 in the EBU Technical Review (http://tech.ebu.ch/techreview).

Additional key words: acceleration, authentication, bandwidth, BDP, buffer, congestion, connectivity, CIFS, cloud, delivery, despatch, dispatch, encryption, exchange, facility, firewall, FTP, GridFTP, HTTPS, Internet, latency, MDP, NAT, optimisation, packet loss, performance, prioritisation, protocol, resource allocation, rushes, scheduling, SCP, web service, SFTP, tapeless, TCP, throughput, TOE, tuning, UDP, verification, WAN, window

White Papers are distributed freely on request.

Authorisation of the Head of Broadcast/FM Research is required for publication.

BBC Research White Paper WHP 183

High performance file transfer over IP networks

Peter Brightwell

1 The growing importance of file transfer

As broadcasters, production companies and other organisations in our industry modernise their technical infrastructure towards being completely computer-based, video and audio are increasingly treated as large data files. Ease of access to file-based content has already revolutionised workflows *within* post production facilities, news centres and playout areas. However, this is not always the case *between* facilities, as until relatively recently high-speed wide area networks were not a cost-effective way of moving large amounts of material over long distances, and the need to still support physical delivery of tapes has meant that often the advantages of file-based working have not been fully realised.

The situation is now changing, as the industry increasingly makes use of high performance private or shared WANs. For example, the BBC has access to a optically amplified network [1] that provides very high speed connectivity around the UK. EBU members can use Eurovision's fibre network (FiNE) [2], while Sohonet [3] provides connectivity to the post production community in London and elsewhere. Often such networks are used for multiple purposes, including video, telephony and streamed IP traffic.

In addition, the need to support an increasing number of different distribution platforms, and on-demand, non-real-time consumption of content make the adoption of file based delivery ever more relevant.

Figure 1 shows where files might be transferred in a future scenario where tape has been completely eliminated from the content creation process. Video and audio are recorded as files either directly onto hard-disk based recorders in the studio (or outside broadcast vehicle), or via removable storage in the camera. These are transferred into storage assigned to the production. This replaces video tapes on shelves in the production office, and allows the production team to view clips, add comments, etc. Content is sent to the post production facility and edited versions sent back to the production team for review. Finished programme material is sent to the departments dealing with playout, website and other delivery channels. During a production, content may be transferred both from and to archive storage, and might be sent to partner organisations such as co-producers. It may also be necessary to transcode and otherwise manipulate content by sending files to and from processing services that could be located elsewhere. In some cases these services could be hosted "in the cloud", in other words provided over the internet and resourced by a web service interface such as Amazon's EC2 [4].

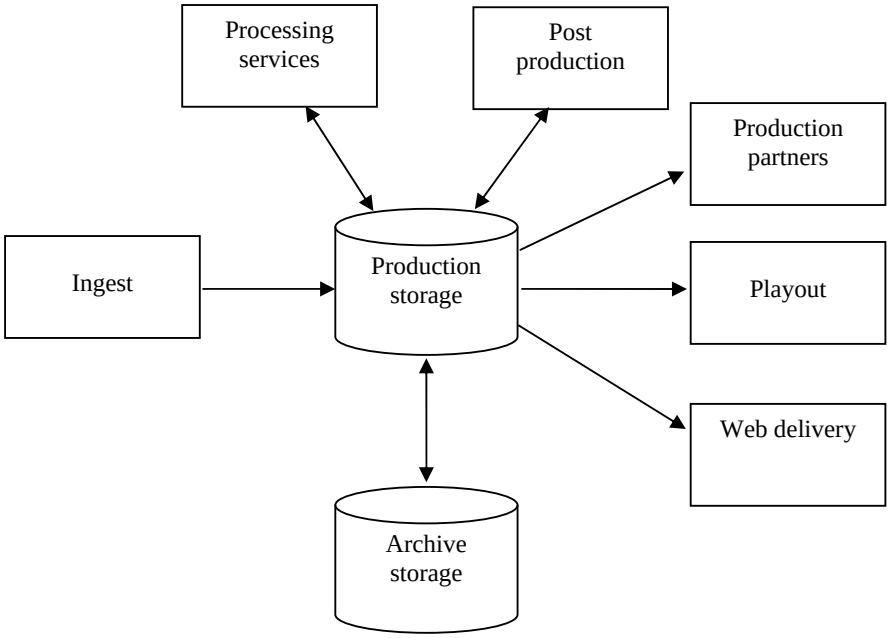

Figure 1 – File transfer in tapeless production

Other broadcaster activities will have other requirements, for example news departments will need to exchange files with other broadcasters, perhaps using the FiNE network as discussed in [5]. A wide range of networks might be used, including private networks, satellite links and the Internet.

The connectivity available will influence how broadcasters' workflows change; for example if a production team can be sure that their rushes will be available for use at a post production facility within a few hours of shooting (rather than waiting for tapes to be sent overnight, and then ingested into an editing system) then they may be able to shorten the overall time required for the production. As a result, deadlines are likely to increasingly depend on reliable transfer performance and management.

2 Performance

The speed of a file transfer depends on a number of factors:

- The raw bandwidth available over the link(s) used for the transfer.
- The network performance of the sending and receiving computers, which will depend on their network interface cards (NICs), drivers and protocol stack.
- The disk performance of the sending and receiving computers. For example to fill a gigabit link, it will typically be necessary to use RAID storage.
- The network latency. As discussed in the next section this can sometimes have a significant effect on TCP throughput.
- Bit errors, for example those caused by faulty cabling or a poor quality optical connection. The overhead required to correct these can cause significant performance degradation.
- Intermittent network connections. For example packets on a satellite link may be lost from time to time, or a network switch on a long distance link might fail during a transfer and packets may need to be re-routed.
- Network congestion. This is often a major cause poor performance, leading to packets being lost and increased latency and is discussed later in this article.
- Server congestion. If several users are trying to send files to the same recipient, performance will be reduced.

The type of network typically determines which of the above limit the performance obtained in practice. For example, when using a satellite link, latency and packet loss are likely to be most important, whereas when using a DSL connection to the Internet, bandwidth limitations and congestion are likely to be the determining factors.

Network performance is discussed in more detail in [5], which provides an introduction to file transfer over IP networks.

3 'Long Fat Networks'

The high-performance WANs mentioned previously are sometimes called "Long Fat Networks" (LFNs) as they provide high raw bandwidth but also a significant latency, so they have a large bandwidth-delay product (BDP), defined as the product of the link's maximum capacity in bits per section and its end-to-end delay in seconds. For example, a gigabit link from London to Manchester might have a BDP of 1 Gb/s * 8 ms = 8 * 10^6 bits.

Where the BDP is significantly greater than 10^5 bits, users may find that file transfer performance is less than expected. This is an effect of the TCP's "sliding window" flow control mechanism shown in **Figure 2** (TCP is used by transfer protocols such as FTP and HTTP to ensure reliable transmission of data). For each TCP connection, the receiver maintains a window in its circular buffer, determining how much more data it can accept. When the receiver acknowledges (ACKs) incoming packets it "slides" the window around the buffer, and tells the sender what is the current window size in the TCP header of the ACK packet. The sender must wait for data packets to be ACK'd before sending data that would exceed what is currently allowed by the window.

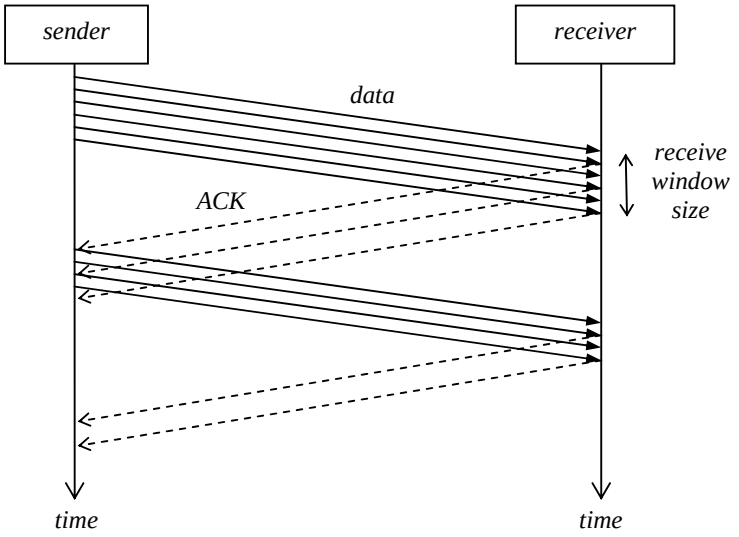

Figure 2 – TCP flow control

In the example shown, the window size is too small for the link latency, meaning that the available bandwidth is not fully used. In the basic TCP specification [7] the window size is limited to 64kB (because it is signalled using a 16 bit value in the TCP header), which is insufficient for today's LFNs. For example **Figure 3** shows the results of throughput measurements obtained by transferring large files between two standard Windows XP computers (2.4 GHz Intel Core2Duo, single SATA disk, Gigabit Ethernet card) using FTP and using CIFS (the protocol used for Windows shared folders). The latter performs particularly poorly as it is a "chatty" protocol, requiring many round-trip messages to occur during a transfer.

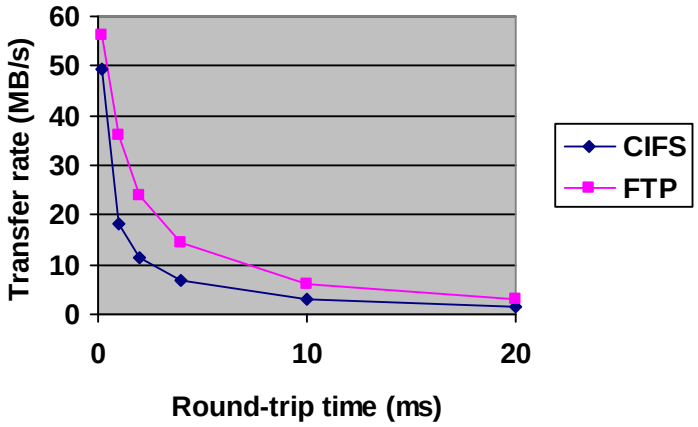

Figure 3 – Effect of latency on throughput

The example shown above was for an uncongested link. If there is significant traffic on the link, latency times may increase, and packet loss may occur, requiring retransmission of TCP packets, leading to further loss of performance (**Figure 4**).

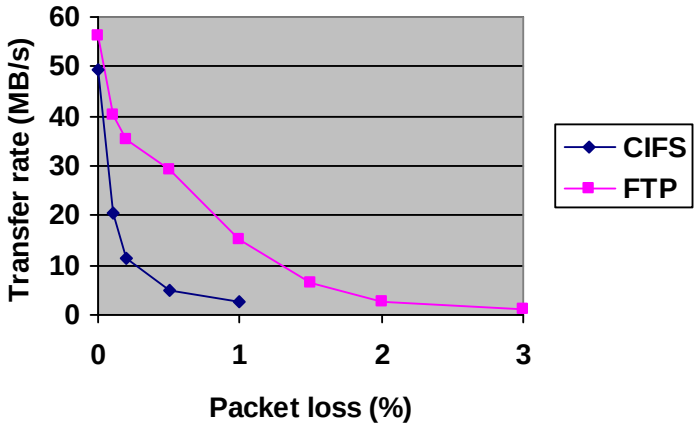

Figure 4 – Effect of packet loss on throughput (low latency)

4 Approaches to increasing performance

4.1 Tuning TCP

One method of increasing throughput in high latency conditions is to make use of the window scaling extension to TCP [8]. Where both ends support this, window sizes of up to a gigabyte can be used. The send and receive buffer sizes will also need to be increased accordingly.

For older operating systems (such as Windows XP), doing this requires some expertise, for example of adjusting Windows Registry settings, or using a specialist tuning tool. Fortunately, newer versions, including Windows Vista and recent Linux kernels, support automatic tuning of some TCP parameters and the task is now simpler, although for the longest or fattest links some tuning may still be required.

By making the TCP buffer and window sizes large enough, the effect of latency can be eliminated over a WAN link. This may also require increased buffer sizes in the application layer; for example the default size for Windows XP's command line FTP client is only 4 kB (this can be increased using the "-w" option).

Figure 5 shows an example of where the tuning approach has been used successfully to deliver BBC production rushes from a studio about 30 km west of London to a post production facility in Manchester, making use of several types of network. The maximum capacity of the link was about 400 Mb/s (limited by the performance of VPN hardware) and the round-trip latency about 18 ms. A throughput of near 50 MB/s was reliably achieved, showing that latency had no significant effect.

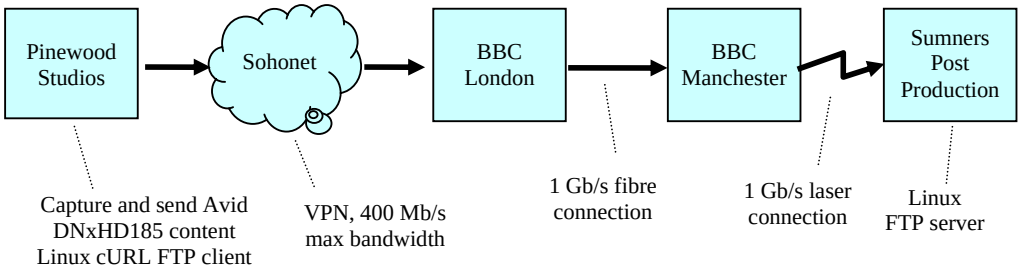

Figure 5 – Case study for WAN delivery of rushes

A potential disadvantage of this approach is that optimising the buffer and window sizes for large file transfer can cause reduced responsiveness for other activity over the link, such as web browsing, or accessing small files.

Although the window/buffer size usually has the greatest effect on performance, there are other TCP settings that can be tuned, and a number of utilities and websites are available to help with this [9].

For transfer over congested links with high packet loss, performance can be further improved by changing to a more suitable TCP congestion control algorithm. This determines how the sender determines whether the network is overloaded, and how it sets another window (the "congestion window"). Until recently, most implementations were based on the Reno algorithm, but modern operating systems offer more advanced algorithms that are more suited to congested LFNs. For example modern Linux implementations offer the CUBIC algorithm [10] and Windows Vista uses Compound TCP [11].

4.2 Parallel connections

Another approach to improving performance is to break a large file up into multiple parts sent in parallel using multiple TCP connections. This can make better use of both the available bandwidth, and each individual connection will be less affected by latency. This technique is supported by a number of modern file transfer protocols, both proprietary and open. Of particular note is GridFTP [12], which also supports striped transfers in which the parts of the file can be simultaneously sent from (or to) multiple computers. GridFTP was originally developed by the grid computing community to move content between distributed services, and has been used for television applications [13].

4.3 UDP

An alternative way of avoiding the problems associated with TCP is to use UDP instead [14]. More typically used for applications such as VoIP, UDP is simpler than TCP and in itself cannot guarantee that the file data has been sent successfully. Transfer protocols built upon UDP therefore implement their own mechanisms for acknowledging and resending data; these can be optimised for the expected network conditions. Either UDP or TCP can be used to convey this back-channel information. In addition such protocols might make use of forward error correction (FEC) techniques to help reduce the effect of packet loss.

Figure 6 shows some measurements made using a proprietary UDP-based protocol; even with a very high latency throughput is still good.

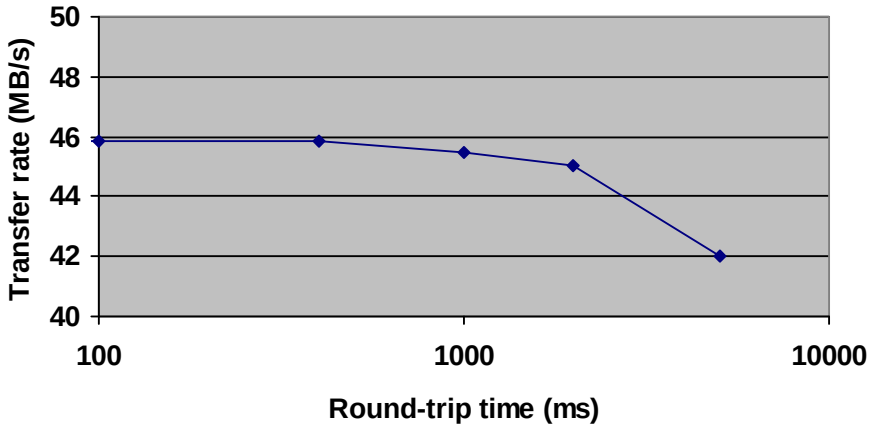

Figure 6 – Effect of latency with UDP-based transfer

4.4 Jumbo frames and TOE cards

Jumbo frames are often mentioned in respect to increasing network throughput. They work at a lower layer (data link) than the techniques described previously, and allow compliant Gigabit Ethernet infrastructure to go beyond the normal 1500-byte payload (MTU) sizes for frames, normally to 9000 bytes. They offer several potential benefits: reduced load on the CPU, reduced header overhead, and better throughput as packet loss rates increase [15].

Unfortunately, jumbo frames are not yet standardised, and many implementations do not support them. Mixing them with regular frames on the same network is known to give poor performance, so they are only really suitable for dedicated LANs or other specialist deployments. In particular much of the infrastructure of the Internet does not support jumbo frames, which is unfortunate as this is where the packet loss benefits would be most helpful.

Furthermore, the CPU benefits are now less important than when jumbo frames were first introduced a decade ago; a modern multi-core PC might require less than 20% of its CPU power to saturate a gigabit TCP/IP connection. However, 10 Gb/s Ethernet links could still benefit.

This last argument also applies to network cards that include TCP Offload Engines (TOE), which free up processing resources by performing the TCP/IP stack in hardware. The benefits for gigabit use are now marginal, and TOE vendors are now concentrating 10 Gb/s Ethernet cards. A further drawback of using such hardware is that is more difficult to update and tune the TCP implementation.

4.5 Other WAN acceleration techniques

Techniques such as data compression and data caching are frequently used to increase the performance over WAN links. These have been shown to work well for many types of application; for example during the 2008 Beijing Olympics the BBC used hardware acceleration devices to provide its production staff in China with more responsive access to documents and web pages hosted in London [16]. However they are less appropriate for transferring video files that are already compressed, and where the files are only transferred once to a particular location.

5 Bandwidth and resource allocation

Unlike the case presented in **Figure 5**, file transfers often will have to share the network capacity with other traffic. This could be real-time (e.g. from a VoIP call) or have some time constraints (for

example, users typically do not like to wait more than ten seconds for a web page to load its main content). There is a danger that unconstrained high performance file transfers will seriously degrade such activities. This is especially the case with UDP-based file transfers. **Table 1** shows the measured performance of video-over-IP and web traffic on a 100 Mb/s link with 40 ms round-trip time; when a UDP-based transfer was started this dropped to an unacceptable level.

	20 x H.263 CIF video pairs			*50 x HTTP web clients*	
	Throughput	Delay	Packet loss	Throughput	Response time
Before	1.45 Mb/s	17 ms	0%	128 kb/s	80 ms
After	1.0 Mb/s	100 ms	25%	<5 kb/s	1000 ms

Table 1 – effect of unconstrained UDP-based file transfer (proprietary algorithm)

A number of approaches are available to avoid such problems. Ideally file transfers will be kept separate from other traffic, for example by using a physically separate network (although this may often be inconvenient to deploy), or by making use of separate wavelengths on an optical link. Generally large files will be transferred between dedicated servers, allowing IP packets to be easily routed onto the appropriate path. In more complex cases technologies such as MPLS [17] allow particular packets to be labelled and routed based on their intended usage.

Where sharing with other traffic is unavoidable, it is desirable to limit the maximum rate of each transfer; typically UDP-based algorithms support a target and “ceiling” rate, as do some FTP implementations. Experienced system administrators can make use of traffic shaping tools, such as Linux’s “tc” utility, to achieve fine-grained control of the bandwidth available to particular types of network traffic.

Even where transfers are given their own bandwidth, they will often have to share this with each other. In addition, multiple transfers might require access to the same file servers, so possible disk bottlenecks also need to be considered. As workflows become increasingly reliant on timely file delivery, it will become more important to manage how the available resources are allocated. For instance an urgent news item or sports programme might be given a high priority, whereas rushes from a production that will take weeks to post-produce might be given a lower priority or scheduled for a quiet period, for example overnight. Users of file delivery might be offered “gold”, “silver” and “bronze” services (at different costs) as best fits their needs.

Transfers may also be prioritised according to the type of content that is delivered. For example, a lower resolution proxy could be sent at higher priority so that users can start working with it while the full quality files are still transferring. As broadcasters gain experience with file-based working, they will learn how best to schedule and manage their delivery capacity.

6 Security

Programme makers have always had to trust the people involved in delivering video tapes to their destination, and in generally they are happy to cope with the risks involved. Being able to physically hold the content provides users with a degree of (possibly false) security; file based workflows take this away from them, and naturally they may be suspicious of trusting “the net”. It is therefore vital that best security practice is followed to avoid incidents that could increase this distrust.

So if content is sent over a shared network (especially the Internet) all access must be authenticated and encryption should be considered. Traditional FTP is particularly poorly suited to such use as passwords are sent in the clear.

More suitable protocols include:

- SFTP and SCP, using SSH certificates rather than passwords.
- GridFTP, using X.509 public-key authentication [18]
- HTTPS, using both server and client certificates
- Some proprietary protocols can optionally use an TLS/SSL connection

In addition the recipient should check that the content has not been corrupted during transfer, for instance using a secure hash such as SHA-1 (or better still one of its more secure variants such as SHA-512).

Authentication, encryption and verification all have an overhead on CPU usage. For example, the throughput of a GridFTP transfer between two quad-core PCs fell from about 60 MB/s to 20 MB/s when all security features were enabled. Where this is likely to cause problems, co-processor cards are available to offload TLS/SSL or other algorithms. However, as with TOE network interface cards, these are likely to be less flexible and future-proof than software implementations.

Firewalls pose potential complications for secure transfer. Many protocols require ports to be opened (e.g. port 22 for SFTP and SCP), which will increase the “attack surface” that is exposed. Furthermore, many organisations use proxy firewalls to provide greater protection of private networks; here simply opening a port is not a possibility.

Often a firewall will use network address translation (NAT) to hide the organisation’s private address range; this can add additional complication as the firewall may need to route an incoming transfer request to the appropriate local computer, without exposing any details of the local computer to the outside world.

Some modern firewalls provide additional capabilities that can make them more suitable. By inspecting incoming traffic at the application level, the firewall can determine who is asking for the transfer, what type of files are involved and to what programme they correspond, allowing “smarter” acceptance or rejection and routing. This can be combined with stateful packet inspection (SPI) techniques that check that the data packets correspond to the correct connection, a better approach than just opening ports.

In practice broadcasters might choose to combine firewalls with an electronic version of the traditional “tape dispatch” department (Figure 7). This holds outgoing and incoming files delivered to or from other organisations, and could include format compliance checking or virus scanning. Such an approach can provide a good level of security, though the additional delay caused might not be suitable for workflows that require a fast response.

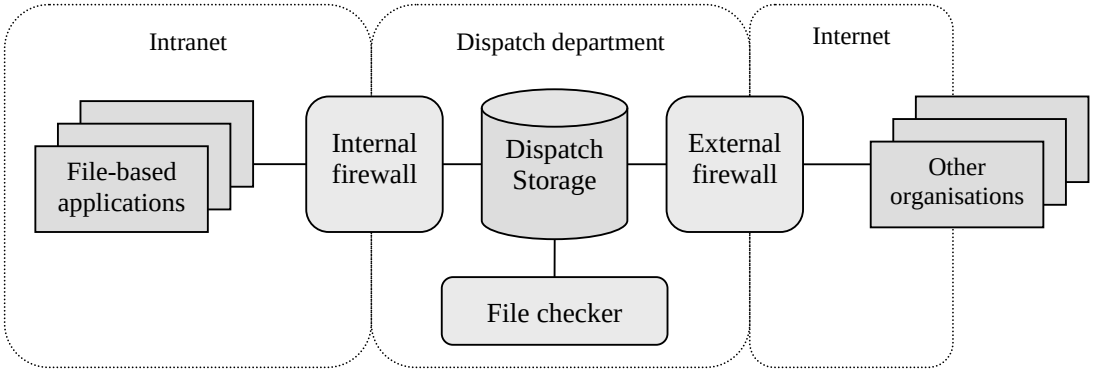

Figure 7 – Electronic dispatch department

Longer term, IPv6 offers the possibility of increased security by authenticating and encrypting at the network layer using IPsec, and simplification by providing enough addresses to remove the

need to use NAT. However IPv6 traffic currently only forms a tiny proportion of the traffic on the Internet [19] and it is likely to be several years before this happens.

One thing that is happening, however, is that broadcasters are involved in increasingly complex relationships with external partners, which will lead to more complex trust relationships. With these will come new technical security problems, for example how to manage keys and certificates securely, how to specify policies for accessing content and services, and how to cope with distributed denial of service attacks.

7 File delivery tools

This article has described a number of techniques that can be used to improve the performance and usefulness of file delivery. Putting these all into practice does however require a degree of technical resources that may not always be available. Fortunately a number of commercial tools are available that are aimed at providing a turnkey solution. Some providers of delivery tools used by the broadcast industry include (in alphabetical order): Aspera, Digital Rapids, FileCatalyst, Kencast, Radiance, RocketStream, Signiant, SmartJog. In addition, many other systems (such as video servers and editing systems) provide use file delivery functionality.

Typically such tools require “agent” software to be installed on each machine that will be sending or receiving files. Each agent knows the details of the other relevant agents, either directly, or under the control of a transfer management system (see **Figure 8**). Often a web version of the software may also be available, in which the agent runs within the browser (e.g. using Java, ActiveX or Flash); this is useful for supporting “ad hoc” transfers, for example where a journalist needs to send content from an internet café and cannot install specialist software.

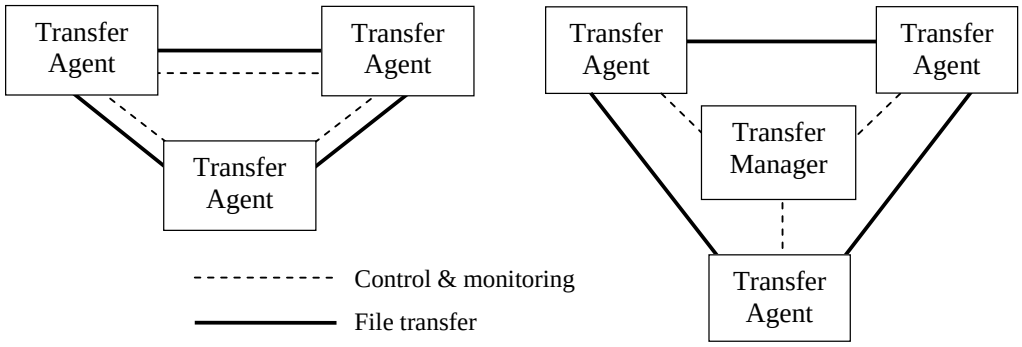

Figure 8 – Transfer agents: point-to-point and centrally managed

Such tools implement a useful range of functionality, for example:

- Accelerated transfers, generally using a proprietary UDP protocol, and/or multiple TCP connections.
- Control of bandwidth, scheduling and prioritisation of multiple transfers, allowing the concepts discussed above to be implemented. Of course, broadcasters will still have to consider their likely usage patterns and architect their networks accordingly.
- User interfaces allowing an operator to monitor and manually control transfers. For example, **Figure 9** shows some parts of Signiant’s transfer management interface.
- Ability to use different transfer configurations based on the type of content and details of the sender and recipient.
- Graceful handling of network problems. For example if there is a loss of connectivity part way through a transfer the tool may wait for a time then attempt to restart from where it left off.
- Data compression (although this is less useful for already-compressed video files).

- Implementation of security features as discussed above. Many tools include their own access control, requiring a user to log into the system; the user interfaces and control options presented to the user will depend on policy settings, as will which transfers they are allowed access to.
- Automatic notification (for example through email) of when a transfer has completed successfully, or has failed.
- Sending the same file to multiple recipients.
- Ability to build automated “delivery workflows” by combining transfer operations and other operations. For example: the tool monitors an export folder from a video editing system, and when some new content appears in the folder it automatically sends this to an external transcoder to make a web version for review. When this is completed it delivers a copy to a number of recipients.
- Integration features, for example a SOAP-based API to help implement such workflows. Pre-prepared solutions for integration with well-known asset management, editing systems and video servers may be provided.

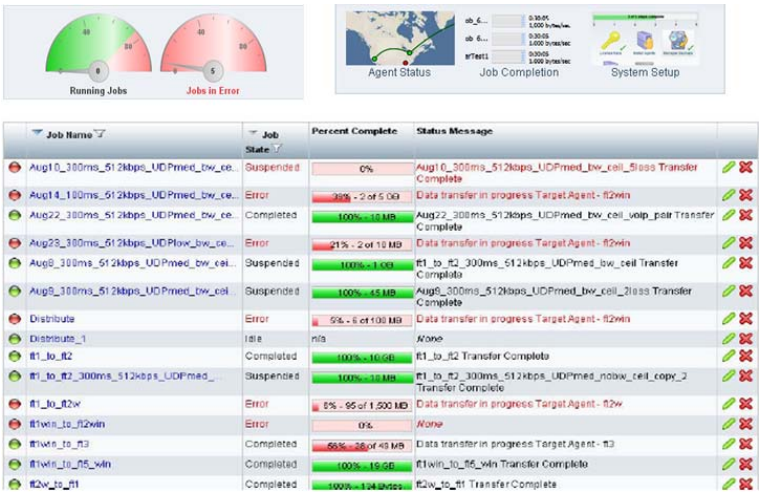

Figure 9 – A commercial file transfer tool

Clearly such products offer significant convenience and may make it possible to implement a sophisticated delivery solution relatively quickly. Also, in some cases a solution may already be in place; for example the Limelight content delivery network offers its customers the option to use Aspera to upload content.

However, where broadcasters have a choice, they should consider possible longer-term implications. Different vendors use their own proprietary protocols both for the file transfers themselves, and for the information *about* the transfers such as what files will be sent, what URLs should be used, when they will be sent, whether they succeeded. Deploying a single solution within an organisation can lead to vendor lock-in, while requiring other organisations to use the same tool can lead to increased costs due to the need to maintain multiple different systems. Another possible disadvantage is that users are reliant on the vendor to keep the software updated to prevent future security breaches.

Recent history suggests that open solutions are likely to have a long-term future (for example, IP-based LAN technologies have now almost totally replaced proprietary ones such as IPX/SPX and NetBEUI/NetBIOS, and Apache is the leading web server). So as well as understanding how to optimise use of their networks and adopting security practice, broadcasters should ask their suppliers about their use of open technology. Protocols such as GridFTP or SFTP can be used to provide secure and robust transfer, while the Media Dispatch Protocol (MDP) [20] provides an SMPTE-standardised means of sending information to coordinate large media file transfers.

8 Conclusions

Broadcasters migrating to tapeless workflows will increasingly need to address how to transfer large content files effectively, especially over wide area networks. They will need to appreciate how latency and congestion affect performance, and how to manage the bandwidth used by file transfers if they cannot be separated from more time-critical traffic. Fortunately modern computers and operating systems are much better suited to these tasks than was the case a decade ago, and there are many resources on the web to help users.

A number of suppliers provide turnkey file delivery solutions that are suited to broadcasters' requirements. These typically offer a number of additional useful features that make them suitable to be deployed in a corporate environment. However, such products often use proprietary protocols and so do not interoperate with their competitors; users will need to be aware of possible longer-term implications of this as they start exchanging more files with other parties.

This article has only covered some aspects of file transfer in our industry. As workflows change, broadcasters will need to address problems such as how best to identify content items across organisations, how to decide which content really needs to be transferred when this is not dictated by the use of videotape, and how technologies such as MXF are best used in these situations.

9 References

1. EBU Networks 2005 Report. www.ebu.ch/CMSimages/en/NMC2005report_FINAL_tcm6-40551.pdf
2. FiNE – Fiber Network Eurovision. www.netinsight.net/pdf/040823_Casestudy_EBU_2.pdf *(is there an EBU reference?)*
3. Sohonet. www.sohonet.co.uk
4. Amazon Elastic Compute Cloud. aws.amazon.com/ec2
5. Debellemaniere, File-based News Exchange Project, Networked Media Exchange 2009. tech.ebu.ch/events/networks09
6. Berg et al, EBU Tech 3318: N/FT-AV File transfer guidelines. www.ebu.ch/CMSimages/en/tec_doc_t3318-2008_tcm6-60595.pdf
7. IETF RFC793: Transmission Control Protocol. www.ietf.org/rfc/rfc793.txt
8. Jacobson et al, IETF RFC1323: TCP Extensions for High Performance. www.ietf.org/rfc/rfc1323.txt
9. ESnet TCP Tuning Guide: fasterdata.es.net/TCP-tuning/
10. Rhee & Xu. CUBIC: A New TCP-Friendly High-Speed TCP Variant. www4.ncsu.edu/~rhee/export/bitcp/cubic-paper.pdf
11. Sridharan et al. IETF Internet Draft: Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks. http://tools.ietf.org/html/draft-sridharan-tcpm-ctcp-02
12. GridFTP. www.globus.org/grid_software/data/gridftp.php
13. PRISM project. www.bbc.co.uk/rd/projects/prism/
14. Postel, IETF RFC768: User Datagram Protocol. www.ietf.org/rfc/rfc768.txt
15. www.psc.edu/networking/papers/model_abstract.html
16. Zheng, BBC's Technology Innovations at the Beijing Olympics. EBU Networked Media Exchange 2009. tech.ebu.ch/events/networks09
17. Rosen et al, IETF RFC3031. Multiprotocol Label Switching Architecture. www.ietf.org/rfc/rfc3031.txt
18. ITU Recommendation X.509 www.itu.int/rec/T-REC-X.509-200508-I
19. Huston and Michaelson, Measuring IPv6 Deployment, www.ripe.net/ripe/meetings/ripe-56/presentations/Huston-Measuring_IPv6_Deployment.pdf
20. SMPTE 2032, Media Dispatch Protocol, store.smpte.org/category-s/22.htm

Abbreviations

ACK – ACKnowledgement
API – Application Programming Interface
BDP – Bandwidth-Delay Product
CIFS – Common Internet File System
CPU – Central Processing Unit
CUBIC – *this is a variant of BIC="Binary Increase Congestion" that a cubic window function, so the name is a sort of play on words.*
DSL – Digital Subscriber Line
EC2 – Elastic Compute Cloud
FEC – Forward Error Correction
FiNE – Fiber Network Eurovision
FTP – File Transfer Protocol
HTTP – HTTP Secure
HTTP – Hypertext Transfer Protocol
IP – Internet Protocol
IPsec – Internet Protocol security
IPv6 – Internet Protocol version 6
IPX – Internetwork Packet Exchange
LAN – Local Area Network
LFN – Long Fat Network
MDP – Media Dispatch Protocol
MPLS – MultiProtocol Label Switching
MTU – Maximum Transmission Unit
MXF – Material eXchange Format
NAT – Network Address Translation
NetBEUI – NetBIOS Extended User Interface
NetBIOS – Network Basic Input/Output System
NIC – Network Interface Cards
RAID – Redundant Array of Independent (or Inexpensive) Disks
SATA – Serial Advanced Technology Attachment
SCP – Secure CoPy
SFTP – SSH File Transfer Protocl
SHA – Secure Hash Algorithm
SMPTE – Society of Motion Picture and Television Engineers
SOAP – *originally stood for Simple Object Access Protocol, now doesn't stand for anything, see* *http://www.w3.org/TR/soap12-part1/#intro*
SPI – Stateful Packet Inspection
SPX – Sequenced Packet Exchange
SSH – Secure Shell
SSL – Secure Sockets Layer
TCP – Transmission Control Protocol
TLS – Transfer Layer Security
TOE – TCP Offload Engine
UDP – User Datagram Protocol
URL – Uniform Resource Locator
VoIP – Voice over Internet Protocol
VPN – Virtual Private Network
WAN – Wide Area Network