BBC

Research White Paper

WHP 158

December 2007

Standardising media delivery in a file-based world

Peter Brightwell

BRITISH BROADCASTING CORPORATION

BBC Research
White Paper WHP 158

Standardising media delivery in a file-based world

Peter Brightwell

Abstract

Delivery of broadcast content between media organisations using file transfer is becoming an increasingly attractive alternative to the physical movement of tapes, films or disks. However, widespread adoption of file-based delivery, especially over public networks such as the Internet, requires the adoption of secure, reliable and interoperable solutions. This white paper outlines the work of the Pro-MPEG Forum in developing codes of practice in this area, including an overview of the Media Dispatch Protocol (MDP): a standards-based means of automating and managing deliveries. Case studies of the implementation and use of MDP are presented, together with an investigation of transfer performance and a discussion of security best practice.

This white paper is a revised version of a paper from the International Broadcasting Convention in 2006 (original title ‘File-based delivery using the Media Dispatch Protocol’). This version includes information about relevant activity in the following year, including the standardisation of MDP, further research studies, and a consideration of how this work relates to the BBC’s current file delivery agenda.

Additional key words: tapeless, FTP, HTTP, XML, open source, manifest, B2B, SOA, SMPTE, despatch, production gateway, latency, packet loss, architecture.

White Papers are distributed freely on request.

Authorisation of the Head of Research is required for publication.

© BBC 2007. All rights reserved. Except as provided below, no part of this document may be reproduced in any material form (including photocopying or storing it in any medium by electronic means) without the prior written permission of BBC Future Media & Technology except in accordance with the provisions of the (UK) Copyright, Designs and Patents Act 1988.

The BBC grants permission to individuals and organisations to make copies of the entire document (including this copyright notice) for their own internal use. No copies of this document may be published, distributed or made available to third parties whether by paper, electronic or other means without the BBC's prior written permission. Where necessary, third parties should be directed to the relevant page on BBC's website at http://www.bbc.co.uk/rd/pubs/whp for a copy of this document.

BBC Research
White Paper WHP 158

Standardising media delivery in a file-based world

Peter Brightwell

1 Introduction

As the media industry continues its progress towards being completely computer-based, all forms of media are increasingly represented as data files, regardless of the physical medium they are stored on. As a result, tapeless working has revolutionised working practices within the industry. However, exchange between organisations still usually involves the physical movement of media, and many of the advantages of tapeless working are lost when work is transferred between organisations.

1.1 Why FTP is not good enough

While file transfer is a trivial problem that can be addressed by using FTP, in practice exchanging files between organisations is much more complicated than just moving the data. Simple FTP-based working loses many of the advantages of the long-established working practices built up around moving tapes. Sending a tape is a much more complex operation than putting it on a bike. It must be carefully labelled and its movement logged and controlled. It may require checking for technical quality or editorial content. The tape may have considerable commercial value, and as its delivery often marks the start or end of a commercial transaction, security is important.

Simple protocols such as FTP just move the bits from one place to another; they may not provide sufficient security, and do not provide a standard way of initiating, supervising, auditing and retrying transfers. In addition, the very simplicity of just moving a file from one place to another allows almost unlimited possible variations in setup and conventions: even simple differences in setup between companies can result in interoperability problems, and there are many more ways to do file transfers than there are tape formats.

1.2 The need for open standards

Without standardisation, file transfer can become more of a problem than a solution. Organisations need to install and manage multiple file transfer systems, potentially a different one for each other organisation they deal with. Proprietary managed file transfer services can provide a partial solution to this problem, but the overheads of subscribing to multiple services (in addition to ad-hoc setups), and the cost of paying intermediaries to carry data that companies could move for themselves, seem to outweigh their advantages.

1.3 Pro-MPEG and the manifest document

In 2003, the Pro-MPEG Forum [1] set up a working group to develop codes of practice for file-based delivery of content. The group identified a need for a mechanism to allow organisations to agree on the details of a delivery, manage its lifecycle, and exchange information about its progress. From this requirement, the concept of the **manifest document** was developed [3]. This is an XML document encapsulating the details of a delivery, for example:

- Which organisations are involved, who requested the delivery and when.
- Which files may be transferred, how large they are, etc.
- What transfer protocols and security mechanisms may be used
- How each individual file transfer is progressing

2 Media Dispatch Protocol (MDP)

More recently, the group has extended this work to specify how the manifest documents should be exchanged, and has defined the semantic rules for a delivery. This has led to the development of the **Media Dispatch Protocol** (MDP) for orchestrating and controlling file transfer of media between organisations.

Because MDP is an open protocol, anyone can implement systems using it. Just as with SMTP, the protocol that makes email possible, MDP is designed to let systems work together without prescribing how those systems are implemented. MDP provides a common way for different systems to plug together.

Figure 1 shows MDP in context. Organisations that wish to deliver or receive media make use of **MDP agents**. These are pieces of software that implement the functionality of the protocol, and manage the file transfers. An agent will provide an API to allow applications to request deliveries and check on how they are progressing. The applications accessing the API could be just a simple user interface, or a complex workflow automation system. Often an agent will run on a separate computer with direct access to the Internet.

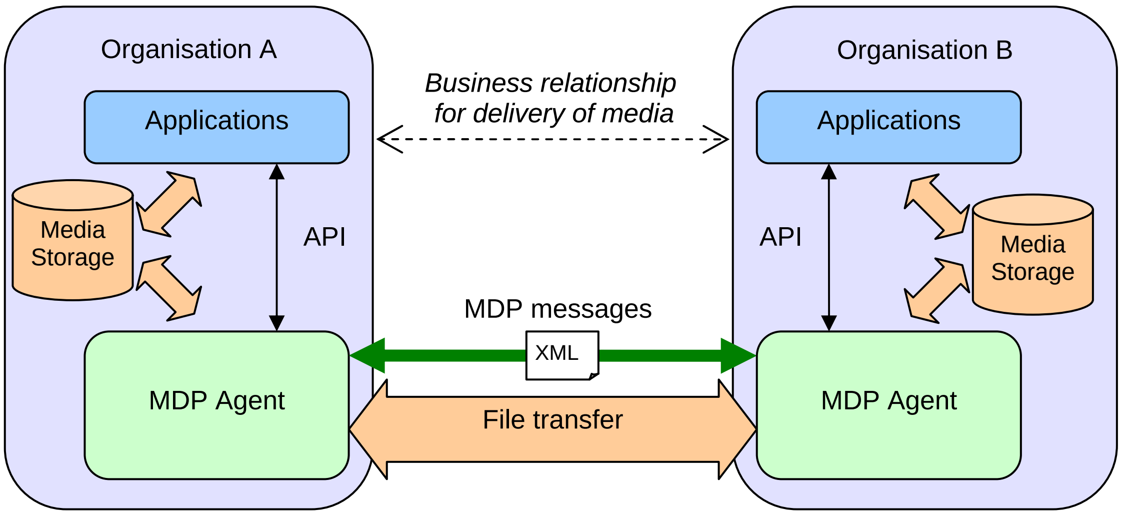


Figure 1 – MDP in context

MDP agents communicate with each other using a set of simple **messages**; these have meanings such as:

- I wish to initiate a new delivery, and here are the details.
- This is what I agree to for this delivery.
- Please tell me how the delivery is progressing.
- I wish to temporarily suspend the file transfers for this delivery.
- I wish to abort the delivery.
- The delivery has completed, and here are details of what happened.

The messages take the form of HTTP requests and responses, rather like those between a web browser and server. Many of the messages contain a manifest document as a payload, while others include other XML documents, for example to send error information.

MDP allows an organisation to package together the media files for the delivery along with any other files required (e.g. subtitles, rights information); the manifest document can be seen as the electronic equivalent of a consignment list or delivery note.

It is important to note that MDP is not a file transfer protocol. Instead it allows agents to talk to each other to negotiate the details of which protocol will be used, along with other details such as at what time the transfers should start, and which files such have higher priority. Then, once the transfers have started, the agents use MDP to talk about how they are progressing, and whether they are successful.

2.1 Extensions and Web Services

As is often the case with network protocols, coping with changing requirements may mean adding extensions to MDP, for example the ability to specify partial file transfers. Because it is important that any such changes do not cause compatibility problems, the MDP message set also allows Agents to declare which extensions they support, as well provide useful information such as how large a file can be transferred.

According to W3C, a web service is ‘a software system designed to support interoperable machine-to-machine interaction over a network’. So in effect an MDP agent provides a web service for the delivery of media files. Adoption of standards from the web services community could help with future incorporation of MDP into a wider service-oriented architecture. Of particular importance are:

- SOAP [4], a protocol for exchanging XML-based messages.
- WSDL [5], a language for describing web services.

Support for these standards is provided as an extension to MDP.

3 Uses of MDP

MDP can be used in any part of a production workflow where media delivery and business transactions coincide. It allows organisations to apply technical, security and business logic policies on transfers, and supports automated checking and auditing of deliveries.

In particular, MDP can be used support automation and policy enforcement for:

- Delivery of media into or out of post-production companies through a **digital dispatch department**. A partial implementation of MDP is already being used for delivery of commercials in the London post community.
- Delivery of rushes content to a broadcaster’s ingest facility.
- Delivery from a broadcaster's production asset management system to a media delivery department for playout, webcasting, etc.
- Delivery of material to newsrooms from roving reporters, where time is of the essence, yet checking before airing material is still essential.

Examples of policy enforcement include:

- Checking that approval exists for a delivery; e.g. the client has paid for the job.
- Automated checking for technical conformance of files entering or leaving, e.g. file type, aspect ratio, compression details, audio levels, flashing images, etc.
- Automated checking of accessibility requirements, e.g. whether subtitles are present.
- Automated checking that all required metadata information is present, and that any unnecessary metadata has been stripped out.
- Checking that only files that are intended to be seen by the client are delivered, e.g. that internal company information is not leaked to the client

4 BBC Production Gateway

As part of its move towards becoming a tapeless organisation, the BBC initiated the **Production Gateway** project to help clarify the benefits, costs and risks of file-based exchange with its production partners, and to identify workable business models [6].

One of the first activities of the project has been a technical trial of exchange between BBC and five production facilities across three cities, over links with capacity 7 Mbit/s to 1 Gbit/s (see Figure 2).

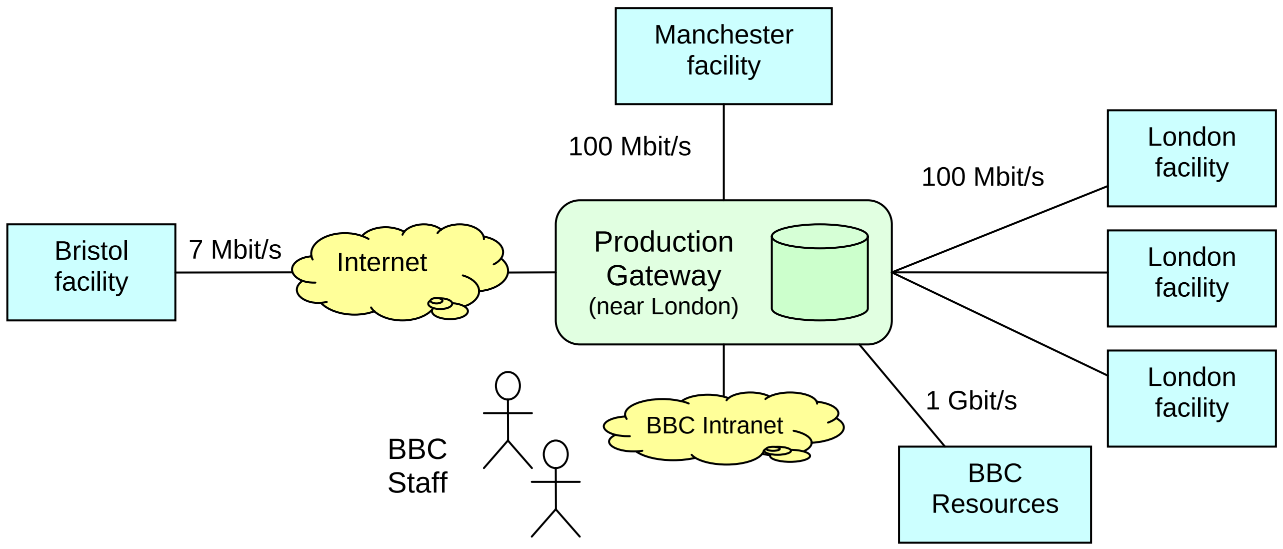


Figure 2 – Simplified view of Production Gateway technical trial

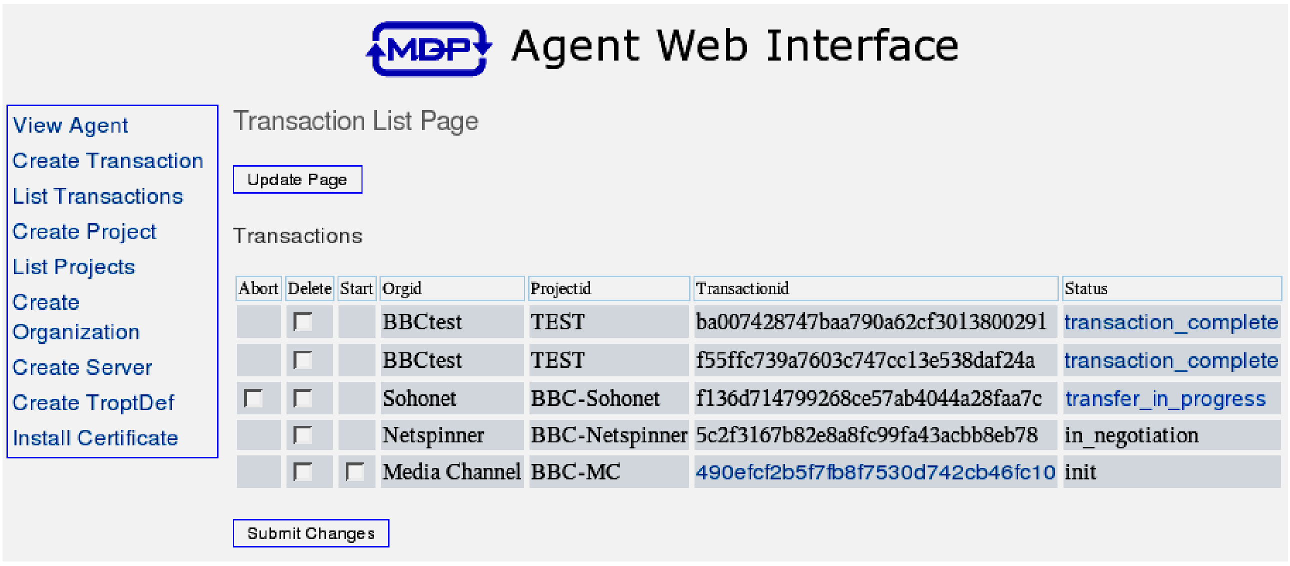
The technical trial included a Java application running on each organisation's server. This included a user interface to allow an operator to select files for delivery and monitor progress. It also included delivery agent code, which implemented a subset of the full set of MDP messages, and initiated an FTP transfer. On arrival, the transferred content was held on the gateway server, and a notification sent to a named contact.

Although not all aspects of MDP were implemented (e.g. no encryption was used for the messages or the file transfer), the trial demonstrated the viability of the protocol for exchange with post houses. The trial also provided other valuable information for the project, e.g. about the throughput of the different types of connectivity at various times of day.

The findings of the Production Gateway project have since informed the wider work of the BBC's Digital Media Initiative [7], which is enabling BBC business areas to migrate their production activities to file-based workflows.

5 Open source reference implementation

BBC Research has made available a **reference implementation** of an MDP Agent that is compliant with the specification. This allows a user to use either a web-browser or a command-line interface to request and monitor the delivery of files, and to perform a range of administration activities (see Figure 3). It also provides a SOAP-based API to allow applications to access these functions over a network.



Transaction List Page

Transactions

Abort	Delete	Start	Orgid	Projectid	Transactionid	Status
	☐		BBCtest	TEST	ba007428747baa790a62cf3013800291	transaction_complete
	☐		BBCtest	TEST	f55ffc739a7603c747cc13e538daf24a	transaction_complete
☐	☐		Sohonet	BBC-Sohonet	f136d714799268ce57ab4044a28faa7c	transfer_in_progress
	☐		Netspinner	BBC-Netspinner	5c2f3167b82e8a8fc99fa43acbb8eb78	in_negotiation
	☐	☐	Media Channel	BBC-MC	490efcf2b5f7fb8f7530d742cb46fc10	init

Figure 3 – Reference implementation web interface

All messages are secured using HTTPS, with server and optionally client certificates used to authenticate the organisations' Agents. Security matters are discussed further below.

At the time of writing, the receiver always pulls the files across the network, using HTTP, HTTPS or SFTP. Pulling the content (rather than pushing it with e.g. an FTP PUT command) means that the receiver has better control over the rate and priority of transfers; this is important where an organisation may be receiving large numbers of files and has to meet deadlines, e.g. at a playout centre. It also makes it easier for the receiver to check the start of a long file, and abort the transfer if there is any problem.

The implementation supports scheduling and prioritisation, and allows transfers to be suspended and resumed. It also automatically detects a stalled transfer and resumes it later. The integrity of a transfer can be verified using a hash checksum.

The implementation has been made available for free download as **open source software** [8]. A 'live-CD' version is also available: this allows a user to try out the software without having to install it on a computer. The Agent is coded in Perl and implemented using open source software components throughout. For example, widely-used Perl modules are used to manipulate XML documents, create HTTP requests and access databases, and the Agent is typically used with the well-known Apache HTTP server and MySQL database.

By choosing the open source software approach, BBC has signified its wish for the end-user community to take ownership of the implementation. This will encourage organisations to adopt an open approach to delivery, rather than adopt proprietary and non-interoperable solutions.

A demonstration of an MDP-based delivery using the open source implementation can be found on the Pro-MPEG website [1].

6 Security

As all automated systems have potential vulnerabilities, MDP Agents should be deployed within an infrastructure that follows IT security best practices. The security integrity of the protocol itself has been ensured by encrypting the HTTP messages with SSL/TLS [9] and supporting both server and client certificates.

Agents should be deployed with the understanding that they will come under attack, and it should be expected that patches and updates will be required to the Agent code as well as the systems they are running on. It is also possible that the protocol itself may require updating. In such a case, the MDP message set allows Agents to communicate version information before any delivery is initiated: this could allow an Agent to reject requests from a compromised Agent.

One of the design criteria for the protocol from the beginning has been not to invent or re-implement security processes. Solutions such as the public key infrastructure (PKI) are widely adopted and used to exchange data of a high value per bit (e.g. financial information); such solutions are expected to come under continuous heavy attack, and so are well-engineered and frequently updated to cope with new threats. The MDP specification and reference implementation will continue to adopt such solutions as they mature.

7 Transfer Performance

A van carrying 100 Digibeta tapes from London to Manchester in five hours has an effective bandwidth of about 3.5 Gbit/sec. Although there are obvious drawbacks to this approach, clearly network transfer performance is highly relevant to the cost-effectiveness of file-based delivery. A number of factors influence the transfer rates that will be obtained in practice:

- The raw bandwidth of the link (including any switches, routers, etc.)
- The network performance of the servers. Modern CPUs can easily saturate a 100 Mbit/s Ethernet link and often a Gigabit Ethernet link
- The amount and shape of traffic on the link: this is especially important for delivery over the Internet.
- The amount of latency in the link. In particular the throughput of TCP (and hence FTP and HTTP) decreases significantly once round-trip times become too high [10]. This is likely to become an issue when transferring large files overseas.
- The read and write speeds of the storage. Modern SATA drives deliver several hundred megabits per second but with RAID, several gigabits per second can typically be achieved.
- The choice of protocol, especially what type of encryption is used. With HTTPS, each connection involves an exchange of security context and then the payload is encrypted. If a processor-intensive cipher is used for this (e.g. AES-256), the number of CPU cycles consumed may limit the throughput obtained.

Table 1 compares various protocols in terms of their functionality and the average transfer rates measured between two dual 3.0 GHz Xeon servers with non-RAID storage on a Gigabit Ethernet network.

Protocol	Transfers files?	Secure protocol?	Provides higher-level functionality?	Average transfer rate (Mbit/s)
Raw TCP	No	No	No	937.3
FTP	Yes	No	No	322.5
HTTP	Yes	No	No	326.9
SFTP	Yes	Yes	No	228.9
SCP	Yes	Yes	No	285.8
HTTPS (AES-256)	Yes	Yes	No	270.6
MDP with HTTPS	Yes	Yes	Yes	223.9 (not optimised)

Table 1 – Comparison of network protocols

The measured disk read/write speed was about 500 Mbit/s, so disk performance may have some influence on the results. Nevertheless these results suggest that the performance drop suffered through use of secure protocols is relatively small, and so the additional bandwidth cost should not necessarily be seen as a significant drawback where secure transfers are required. Although SCP (secure version of remote copy) performs slightly better than HTTPS, the latter was chosen for the reference implementation because it was already being used to encrypt the MDP messages.

The results also show that there is a small additional overhead when HTTPS transfer is used with the MDP reference implementation (the basic HTTPS figure was obtained using ‘wget’, an open source file transfer client). Some of this overhead may be due to the Agent monitoring the transfer. However, as little optimisation of the Perl code has yet been performed, it should be possible to reduce the overhead significantly in practice. Furthermore, as MDP can be used to orchestrate multiple simultaneous transfers, a modified version of the Agent could make use of multi-core or multiple CPUs to achieve better overall performance than a simple FTP tool. It could also support the delivery of large files in ‘chunks’ using multiple simultaneous transfers.

8 Recent activity

8.1 Standardisation

Since the publication of the original version of this paper at IBC 2006 [1], the MDP specification has been re-drafted in a more layered form, and has been submitted to SMPTE [10] for standardisation of:

- The **logical messages** that are sent between agents and the logical data structures that appear in these messages (the Manifest document is one such structure).
- A **mapping** of the logical messages and data structures onto HTTP(S) and XML.
- The **sequence** of messages, and **rules** on their usage, for an MDP transaction that is initiated at the sending end and where the receiver pulls the files across the network.
- A dynamic web-based **register of transfer protocols**. This will allow MDP transactions to be used with new transfer technologies as they are developed. The register provides a solution to the problem of how two implementations refer to a new protocol in the same way (e.g. is it “NEWFTP”, “newFTP” or “NFTP”?). MDP allows agents to exchange information about which protocols they support before initiating a transaction.

At the time of writing, this standardisation activity is very nearly complete, and soon the standard (SMPTE 2032) will be available for purchase from SMPTE.

8.2 Reference implementation

During the SMPTE working group phase, a number of minor changes to the protocol were made to provide better functionality or interoperability. The reference implementation has been updated to ensure that it remains compliant; an outline of these changes can be found at [8].

8.3 Transfer protocols

Some transfer protocols are better suited to certain types of delivery, or different network conditions. For example, a number of technologies provide good “out-of-the-box” performance over **high-latency** networks and networks that suffer from **packet loss**. The dynamic register of transfer protocols would allow these to be used within an MDP transaction. BBC Research is investigating the performance of a number of such protocols for delivery of large media files. Both open-source and proprietary technologies are available; in general, the proprietary solutions offer more sophisticated user interfaces than the open-source software available at the current time, and often include additional functionality such as the ability to restrict the bandwidth used between different times of day.

Open-source examples of such protocols are UDT [12] and UFTP [13]. These both use UDP packets, rather than TCP packets to carry the data; because UDP does not guarantee delivery, the algorithms implement their own acknowledgement and congestion control schemes.

An alternative to using these specialist protocols is to use computers, or standalone hardware, with TCP/IP stacks that are optimised for high-latency links (known a "TCP tuning [10]).

Figure 4 compares the performance of several protocols across a 100 Mbit/s network over a range of simulated latency conditions, with round-trip times (RTTs) of up to 1 second. The results suggest that by using protocols such as UDT and UFTP for high-bandwidth intercontinental transfers, we can expect to achieve an order of magnitude improvement in throughput compared to standard FTP. Some improvement also was obtained by tuning the Windows XP TCP window and FTP buffer sizes when using standard FTP. It is worth noting that SMB/CIFS, the protocol used for Windows Sharing, performs extremely badly at medium to high latencies. This is due to the "chatty" nature of the protocol, i.e. a large number of small messages are sent between the machines, causing additional congestion. So just "dragging-and-dropping" media files over Windows folders shared over a wide area is highly unadvisable.

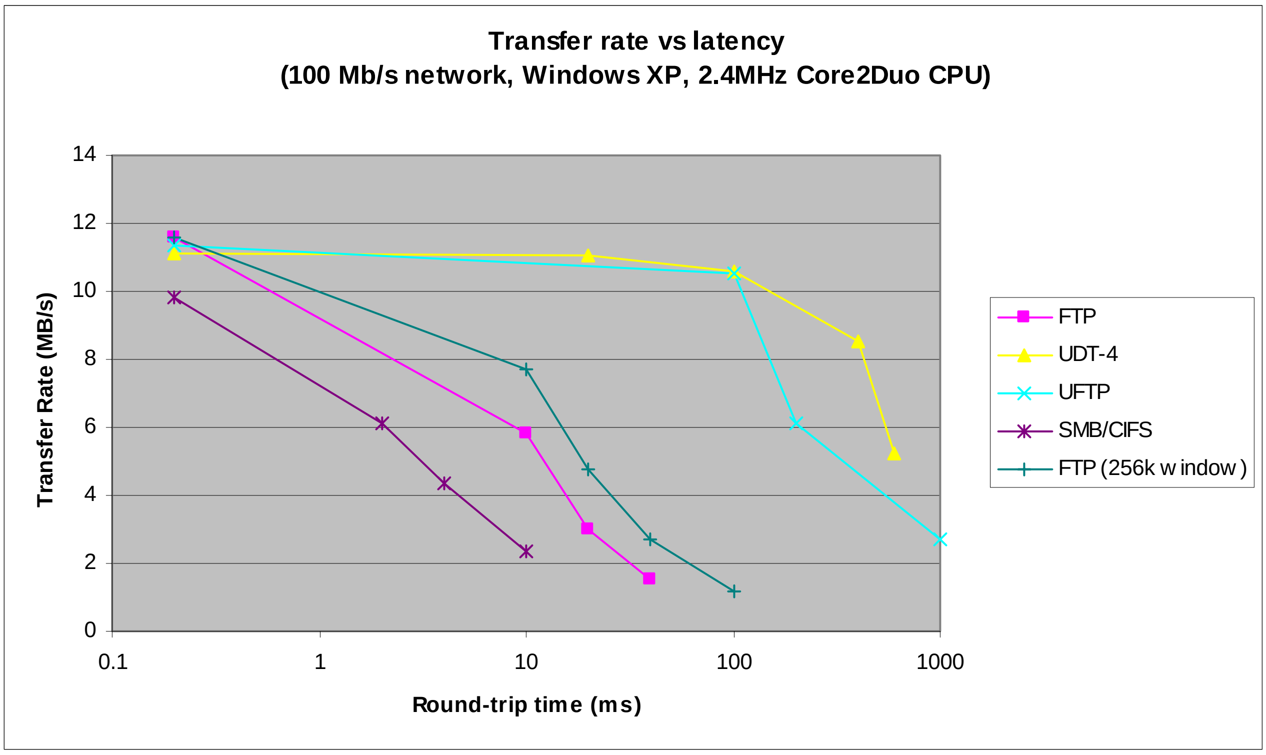


Figure 4 – High-latency transfer performance comparison

Further results of these investigations will be presented at a later date. The protocols are being assessed on the basis of the following criteria:

- Transfer rate over a range of raw network bandwidths (from satellite / ADSL rates to Gigabit Ethernet)
- Processor usage, including ability to use multiple processors or cores
- Ability to transfer very large files
- Performance when transferring a large number of files
- Effect of network latency
- Effect of packet loss

- Effect of link bit errors
- Effect of network disruptions (e.g. does transfer automatically restart?)
- Support for quality-of-service aspects, such as ability to complete the transfer by a specified deadline.
- Ability to pause and resume transfers in progress
- How the transfer protocol affects and is affected by other traffic sharing the network link; for example, voice-over-IP or web-browsing activity.
- Compliance with standards
- Facilities for integration into automated systems, including MDP agents

8.4 Future architectures

As the need to manage the BBC's internal and external file deliveries takes on a higher profile (for instance through the Digital Media Initiative's "Share" enabler [7]), the likely architectures for file delivery will become more clear. For example, Figure 5 shows a **service oriented architecture** (SOA) scenario in which the BBC itself and its trusted media partners make use of a range of common media services such as ingest, transcoding, and file movement. These services would be joined together through a middleware solution (such as that provided by the BBC's Digital Fabric [12]). However, files may also need to be exchanged with parties that are not trusted partners; this could be accomplished by providing a delivery service that invokes an MDP transaction; the details of the MDP messages are hidden from the clients of the delivery service.

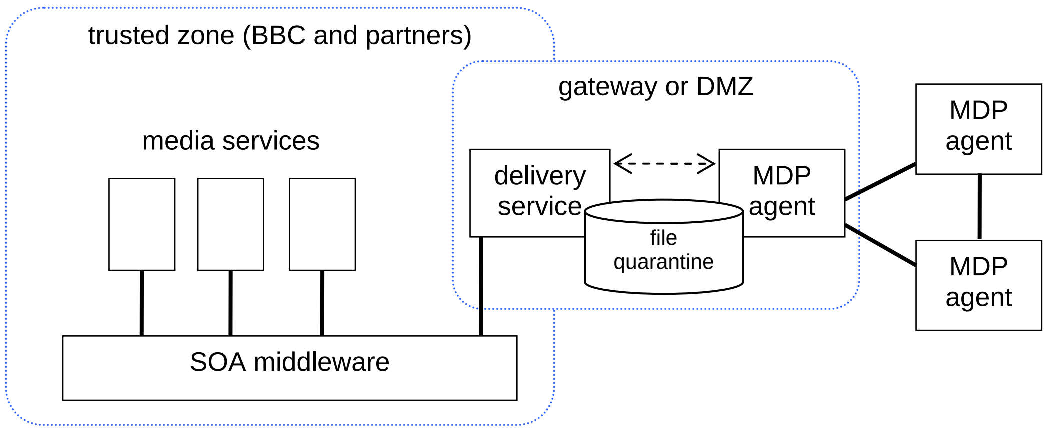


Figure 5 – SOA and gateway

Longer term, as organisations such as the BBC increasingly engage in **outsourcing** and **partnering**, the above "walled citadel" approach of network security is likely to change. Production, distribution and other partners are likely to require more direct and more flexible access to the content and services within an organisation, and a "hard-shelled" approach with firewalls is likely to be an impediment. Instead, organisations are likely to adopt a more "hard-centred" approach such as shown in Figure 6.

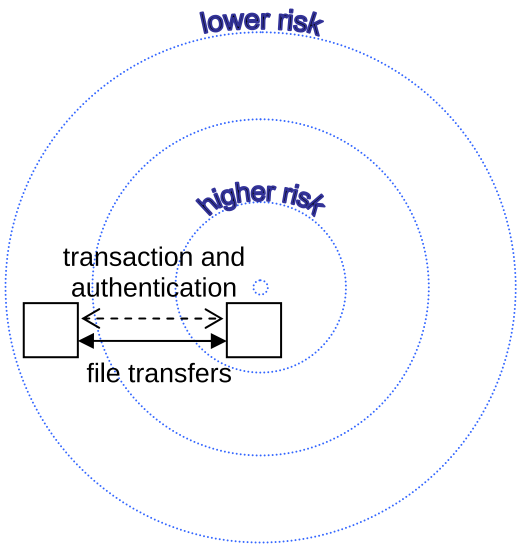


Figure 6 – Possible future security model

This architecture allows the level of security to be set appropriately to the risk involved in a security breach, so for an example a TV broadcast server would be have higher protection than a public internet forum. Systems and services would present authentication information to access each other, and this checked against the appropriate security policies. This is one aspect of the work of the DTI-sponsored PRISM project that is investigating the use of grid-based computing in media domains [15]. In such a context elements of MDP (or a successor to the current protocol) might be used within an authenticated file delivery service.

9 Conclusions

As the media industry moves towards completely tapeless workflows, the need for an open and interoperable approach to file-based delivery becomes more important. The Media Dispatch Protocol (MDP) can play an important part in this approach by providing an open mechanism for orchestrating, controlling and securing file transfers. MDP makes use of standard web technologies such as HTTP(S) and XML.

Partial implementations of MDP have already found use in the production community, and in the BBC's Production Gateway initiative, while a fuller reference implementation is now available as open source software. Initial tests with this implementation suggest that using HTTPS to transfer files securely can provide sufficient performance to be used for deliveries over public networks, when insecure protocols such as FTP would be unacceptable. Furthermore, MDP allows the use of new transfer protocols that are designed for high performance over long-distance or unreliable links.

SMPTE is currently standardising MDP's messages and data structures MDP together with their usage and how they are represented on the network, and is defining a dynamic register to allow different transfer protocols to be used in an MDP transaction.

Current research work includes performance testing of different transfer technologies, and identification of options for secure file delivery in future architectures.

10 References

1. P. Brightwell, N. Harris, B. Roeder R. Atherton, File-based delivery using the Media Dispatch Protocol, Proceedings of the International Broadcasting Convention, September 2006.
2. Professional MPEG Forum, www.pro-mpeg.org.

3. P. Brightwell, Broadcast Media Exchange for B2B Applications, Proceedings of the International Broadcasting Convention, September 2004
4. W3C, May 2001, Simple Object Access Protocol, www.w3.org/TR/soap/
5. W3C, March 2001, Web Services Description Language, www.w3.org/TR/wsdl
6. BBC Production Gateway project, www.bbc.co.uk/rd/projects/index.shtml
7. BBC Digital Media Initiative, backstage.bbc.co.uk/news/archives/2007/09/bbc_dmi_project.html
8. MDP Reference Agent project, www.sourceforge.net/projects/mediadispatch
9. IETF, January 1999, The TLS Protocol, www.ietf.org/rfc/rfc2246.txt
10. Pittsburgh Supercomputing Center, Enabling High Performance Data Transfers www.psc.edu/networking/projects/tcptune/
11. Society of Motion Picture and Television Engineers, www.smpte.org
12. UDP-based Data Transfer Protocol, udt.sourceforge.net
13. UDP-based FTP www.tcnj.edu/~bush/uftp.html
14. Computing, 04 May 2006, BBC lays foundations for move to SOA, www.computing.co.uk/computing/news/2155301/bbc-lays-foundations-move-soa
15. DTI PRISM project, www.bbc.co.uk/rd/projects/prism/