Title: A Network Tool for Multimedia File Distribution
Issue: Final Report
Reference: GEOC-UOA-2400-3
Document Type:
Category: R (Document for Review)
Date: 1 May 2002
Pages Number: 32
Contract: GEOCAST
Originating Entity: University of Aberdeen (UoA) & University of Surrey (UniS)
Authors: M. Koyabe, A. Matthews, G. Fairhurst (UoA)
S. Iyengar, M.P. Howarth, H. Cruickshank (UniS)
ENREGISTREMENT DES EVOLUTIONS / CHANGE RECORDS

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>DATE</th>
<th>DESCRIPTION DES EVOLUTIONS</th>
<th>REDACTEUR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>9/11/2001</td>
<td>First Draft</td>
<td>M. Koyabe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>G. Fairhurst</td>
</tr>
<tr>
<td>01</td>
<td>8/02/2002</td>
<td>Second comments on draft (UoA &amp; UoS)</td>
<td>M. Koyabe, G. Fairhurst, Matthews</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S. Iyengar</td>
</tr>
<tr>
<td>01</td>
<td>8/03/2002</td>
<td>Final comments on draft (UoA &amp; UoS)</td>
<td>M.P. Howarth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>H. Cruickshank</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>M. Koyabe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>A. Matthews</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>S. Iyengar</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>M.P. Howarth</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>B. H. Cruickshank</td>
</tr>
</tbody>
</table>
ABSTRACT

This document describes a framework for delivery of multimedia objects (data files, installation software and multimedia video clips). This framework will be used as a part of a study investigating the practical issues in delivering clips over the GEOCAST Testbed. GEOCAST is an Information Society Technology (IST) Project funded by the European Commission (EC). The transfer protocol used in this tool is based on a previous framework developed under the JISC/JTAP-288 programme of the Joint Information Systems Committee of the UK. It is therefore made available only for use in the GEOCAST project, by the University of Aberdeen (UoA). Further use, or derivative developments of the transfer protocol require the explicit permission of the University of Aberdeen (UoA). University of Surrey (UoS) developed the application tool’s security components.

This document provides a framework that defines the network aspects of the application. The framework comprises three protocols - a session protocol to co-ordinate client and server interactions, a multicast transfer protocol, and a security tool to authenticate clients and provide encryption and decryption of the multimedia objects. Particular emphasis is placed on the fundamental issues concerning the design of the protocol and the problems, which may be encountered.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>III</td>
</tr>
<tr>
<td>ABBREVIATIONS</td>
<td>VI</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 OVERVIEW OF SERVICE</td>
<td>2</td>
</tr>
<tr>
<td>3 APPLICATION OPERATION</td>
<td>4</td>
</tr>
<tr>
<td>4 SESSION PROTOCOL</td>
<td>5</td>
</tr>
<tr>
<td>4.1 ASSUMPTIONS</td>
<td>5</td>
</tr>
<tr>
<td>4.2 SESSION SERVER OPERATION</td>
<td>5</td>
</tr>
<tr>
<td>4.3 SESSION CLIENT OPERATION</td>
<td>7</td>
</tr>
<tr>
<td>4.4 CLIENT/SERVER APPLICATION</td>
<td>7</td>
</tr>
<tr>
<td>5 THE TRANSFER SERVICE</td>
<td>10</td>
</tr>
<tr>
<td>5.1 RELIABLE MULTICAST TRANSFER SERVICE ISSUES</td>
<td>10</td>
</tr>
<tr>
<td>5.1.1 Reliable Multicast</td>
<td>10</td>
</tr>
<tr>
<td>5.1.2 Congestion Control</td>
<td>11</td>
</tr>
<tr>
<td>5.2 SAT-RMTP PROTOCOL OPERATION</td>
<td>12</td>
</tr>
<tr>
<td>5.2.1 Data Transmission</td>
<td>12</td>
</tr>
<tr>
<td>5.2.2 Transmission Loss and Repair</td>
<td>12</td>
</tr>
<tr>
<td>6 SAT-RMTP FILE TRANSFER TESTS</td>
<td>14</td>
</tr>
<tr>
<td>6.1 SETUP AND CONFIGURATION</td>
<td>14</td>
</tr>
<tr>
<td>6.1.1 Test Configuration</td>
<td>14</td>
</tr>
<tr>
<td>6.1.2 Session Directory (SDR) Integration</td>
<td>15</td>
</tr>
<tr>
<td>6.2 TRACING FILE TRANSFER</td>
<td>21</td>
</tr>
<tr>
<td>6.2.1 Service Announcement</td>
<td>21</td>
</tr>
<tr>
<td>6.2.2 Data transfer</td>
<td>21</td>
</tr>
<tr>
<td>7 ASYMMETRY IMPLICATIONS ON SAT-RMTP PROTOCOL</td>
<td>23</td>
</tr>
<tr>
<td>7.1 RETURN TRAFFIC</td>
<td>23</td>
</tr>
<tr>
<td>7.1.1 Session feedback</td>
<td>23</td>
</tr>
<tr>
<td>7.1.2 Congestion and flow control feedback</td>
<td>24</td>
</tr>
<tr>
<td>7.1.3 Reliability feedback</td>
<td>24</td>
</tr>
<tr>
<td>7.1.4 Status feedback</td>
<td>25</td>
</tr>
<tr>
<td>7.1.5 Lower layer feedback</td>
<td>25</td>
</tr>
<tr>
<td>7.2 UNICAST VS MULTICAST RETURN TRAFFIC</td>
<td>26</td>
</tr>
<tr>
<td>8 SECURITY SERVICE</td>
<td>27</td>
</tr>
<tr>
<td>8.1 SCOPE</td>
<td>27</td>
</tr>
<tr>
<td>8.2 ARCHITECTURE</td>
<td>27</td>
</tr>
<tr>
<td>CONCLUSIONS</td>
<td>31</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>32</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 6.1: Parameter setting and configuration used for test experiments .............................................. 15
Table 6.2: Details of SDR plugin for SAT-RMTP protocol ................................................................. 16
Table 6.3: Comparison between unicast and multicast return traffic ....................................................... 26

LIST OF FIGURES

Figure 1.1: System components for the client ..........................................................................................1
Figure 2.1: Simple illustration of transfer and session protocols communicating with three remote clients (left) connected to one or more UES terminals ......................................................... 3
Figure 3.1: Simple illustration of communication between the various clients and servers ...................... 4
Figure 4.1: Files to be transmitted are encrypted by the security server when requested by the session server. ........................................................................................................................................ 6
Figure 4.2: Encrypted files requested by the session client are decrypted by the security client .......... 7
Figure 4.3: Packet exchange between client and server applications for return path clients (note: * Security request and key may be sent any time after SAT-RMTP clients starts)................. 8
Figure 6.1: Testbed equipment setup and configuration ............................................................. 14
Figure 6.2: SDR session information and type window ..................................................................... 17
Figure 6.3: SDR session timing and distribution scope window .................................................... 17
Figure 6.4: SDR session media and contact details window ......................................................... 18
Figure 6.5: SDR session security parameters window ................................................................. 18
Figure 6.6: SDR session review of session details ........................................................................... 19
Figure 6.7: Main SDR session and Client Session window with announcement packet details .......... 20
Figure 6.8: Session client joining a well-known session address .................................................. 21
Figure 6.9: Session server transmits announcements periodically ................................................... 21
Figure 6.10: SAT-RMTP client joining data stream ......................................................................... 21
Figure 6.11: SAT-RMTP client starts receiving encoded data ........................................................... 22
Figure 6.12: SAT-RMTP client requesting for packet retransmission ............................................. 22
Figure 6.13: SAT-RMTP client leaves a data stream group ............................................................. 22
Figure 6.14: SAT-RMTP client leaves the session ........................................................................... 22
Figure 8.1: Security server architecture ..........................................................................................28
Figure 8.2: Security client architecture ........................................................................................... 29
Figure 8.3: Security message flow diagram .................................................................................... 30
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>Acknowledgement</td>
</tr>
<tr>
<td>CA</td>
<td>Certification Authority or Conditional Access</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>GEOCAST</td>
<td>Multicast over Geostationary Satellite</td>
</tr>
<tr>
<td>GES</td>
<td>Ground Earth Station</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hashed Message Authentication Code</td>
</tr>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>JISC</td>
<td>Joint Information System Committee</td>
</tr>
<tr>
<td>JTAP</td>
<td>JISC Technology Applications Programme</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LKH</td>
<td>Logical Key Hierarchy</td>
</tr>
<tr>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Picture Expert Group</td>
</tr>
<tr>
<td>NACK</td>
<td>Negative Acknowledgement</td>
</tr>
<tr>
<td>PGP</td>
<td>Pretty Good Privacy</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RMT</td>
<td>Reliable Multicast Transport</td>
</tr>
<tr>
<td>SA</td>
<td>Security Association</td>
</tr>
<tr>
<td>SAT-RMTP</td>
<td>Satellite Reliable Multicast Transport Protocol</td>
</tr>
<tr>
<td>SDP</td>
<td>Session Description Protocol</td>
</tr>
<tr>
<td>SDR</td>
<td>Session Directory</td>
</tr>
<tr>
<td>TBC</td>
<td>To Be Confirmed</td>
</tr>
<tr>
<td>TBD</td>
<td>To Be Determined</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UES</td>
<td>User Earth Station</td>
</tr>
</tbody>
</table>
1 Introduction

The system consists of a number of components (Figure 1.1). For the purpose of the GEOCAST demonstrator, it is assumed the content (multimedia objects being transferred) consists of MPEG-1, MPEG-2 or Quicktime format video clips. At the client, each user uses a client session tool to select the required multimedia content. The tool will initially be implemented using a command-line interface.

![Diagram of system components for the client]

**Figure 1.1:** System components for the client

The client hardware consists of a PC (running Mandrake Linux) or SUN platform (Solaris 2.6 or later version) with high performance Ethernet interface. A hardware or software decoder is required for display of the MPEG video on the computer screen. The client also allocates a portion of local disk to act as a temporary store for objects. This is used to hold a copy of the down-loaded multimedia content.

The multimedia content is distributed by a network server (e.g. a SUN Ultra 10 with high performance disk and network interfaces). Communication between the client and the server is via three purpose designed protocols, each operating over UDP using IPv4: A session protocol which identifies the client needs, a transfer protocol which transfers an object to the client, and a security protocol which handles the encryption/decryption of the transferred object.

A transfer proceeds in four phases:

i) The network transfer protocol known as SAT-RMTP (Satellite Reliable Multicast Transport Protocol), provides network delivery of the content over a multicast enabled network. This transfer protocol has been developed to mitigate the demands on the server/network, and to tolerate packet loss due both to congestion at network
bottlenecks, introduced by receiver performance bottlenecks, and from link impairments (fading).

ii) Encoded multimedia clips will be stored at the content server. Using the content tool, these will be encrypted and session information created. An MPEG-1, MPEG-2 or QuickTime format will be used.

iii) The security protocol, which identifies individual end-users, associates security state (keys) with the clients and decrypts content. Note that the encryption and decryption is provided at the application level, and when encryption is required the transfer protocol only distributes encrypted content.

iv) A session protocol (SDR) is provided, which uses a menu-driven interface. The session protocol describes the objects (files) available for download. The session protocol may in future be extended to encrypt this information.

2 Overview of Service

Multicasting is the networking technique of delivering the same packet simultaneously to a group of clients. This is useful if the clients require a common set of data at the same time, or the clients are able to store (cache) common data until needed. Where there is commonality between the data required by a group of clients, multicast transmission may provide significant bandwidth savings (up to 1/N of the bandwidth for N separate unicast clients). Multicast protocols use a one-to-many paradigm for transmission; typically using class D Internet Protocol (IP) addresses to specify multicast groups [1-3]. IP multicast is currently being used for a variety of applications, such as teleconferencing, remote conference participation, file distribution, TV delivery [3].

One important property of unicast communication is “fate sharing” between the client / server and the reachability provided by the network. Should a system fail, or the network partition, the session may be aborted. This is the basis of the TCP transport protocol. When a server has many multicast clients the concept of fate sharing does not apply to the server. The protocol mechanisms for multicast must therefore be much more robust and survive the failure of clients and partial network failures.

Figure 2.1 shows the interaction between the session server and its clients. The application has been constructed in a way that allows the session server process and a number of transfer / security protocol instances to be located on a common server platform. These may alternatively be distributed among a number of server platforms.
Figure 2.1: Simple illustration of transfer and session protocols communicating with three remote clients (left) connected to one or more UES terminals.
3 Application Operation

The application tool operation, as shown in figure 3 below, consists of three main protocols: the security protocol, session protocol and transfer protocol (SAT-RMTP protocol). These protocols interact and exchange information before, during or after an advertised session.

Figure 3.1: Simple illustration of communication between the various clients and servers
4 Session Protocol

The main function of the session protocol is to convey information (through regular announcements) of services being offered (i.e. files, video clip files); and co-ordinate the transmission and scheduling of various file transfers requested by receivers. Sessions are initiated by the session server, which periodically transmits session announcements to listening receivers. These session announcements use SDP to convey useful information (such as session start and end time, media type and other session related parameters) that is associated with a particular session that may have been advertised to receivers. Receivers, there after, use this information to decide whether to either join or reject participating in the advertised session.

4.1 Assumptions

The session protocol design adopted in this proposed application is based on a number of assumptions. These assumptions aim at enabling the session protocol to function and operate first within the scope of GEOCAST network model [4] and secondly provide session advertisement service, using the Session Description Protocol (SDP) [5], that is seamless in design to both the SAT-RMTP file transfer protocol [6] and the security protocol.

The session protocol design assumes that:

- Both the session server and session client have knowledge of a well-known communication channel (i.e. a multicast address with a corresponding port number) used by the session server (to multicast announcements) and also by receivers (for tuning in and listening to announcements).

- Receivers interested in a particular service must tune in (i.e. join the advertised session) on or before the time that session (i.e. where transmission of a data file (or video clip) is scheduled to start. Receivers are expected to join on or before the start of a scheduled session, in order to receive the service.

- A session server can advertise more than one service (i.e. titles of multiple files), each with a unique session identifier. Clients there after can request more than one advertised service. However, in the current session protocol design, receivers are restricted to only one multimedia object per advertised session.

- A session server schedules a service only when it receives a request that uniquely identifies a service that has been advertised (i.e. with a unique session ID). This enables the session server to have knowledge of interested receivers downstream, before scheduling an advertised service. Similarly, a session client can also reject a request prior to the start of a scheduled service that was requested earlier.

4.2 Session Server Operation

As shown in figure 4.1 below, when the session server (located at the transmitter) receives a session request from a receiver, it interacts with both the security and SAT-RMTP file transfer server, in order to provide securely encrypted data and reliably transmitted data respectively to receivers downstream. The session server requests from the security server a session key (with corresponding control key) for the requested service. After authenticating the request, the security server generates the session keys (used to decrypt the encrypted data) and may also generate the control key(s) (this
could be used to securely send new session keys during periodic re-keying). Corresponding session and control keys (sent to the receivers) are stored in the secure database.

The security server uses the generated session/control keys to encrypt source data, which is then passed to the SAT-RMTTP file transfer protocol (for transmission to registered receivers). All messages between the security server and the client are cryptographically secured using a security association (SA) established by the protocol (see section § 6.0).

Figure 4.1: Files to be transmitted are encrypted by the security server when requested by the session server.
4.3 Session Client Operation

![Diagram](image)

**Figure 4.2:** Encrypted files requested by the session client are decrypted by the security client.

As shown in figure 4.2, a receiver anticipating to join a session, receives session announcements via the session client. Once the session receiver decides to join a session, it passes the session information to the file transfer (SAT-RMTTP) client for client configuration. This process also invokes the file transfer (SAT-RMTTP) client to start receiving data. Once a transfer completes, the encrypted content received by the file transfer (SAT-RMTTP) client, is passed onto the security client, which decrypts the encrypted file (using either the session or control keys received from the secure database located at the servers). The successfully decrypted file is then stored at the receiver to be accessed by the multimedia application (e.g. MPEG-1, MPEG-2, or Quicktime player).

4.4 Client/Server Application

Figure 4.3 shows how a server tool uses the session server to initiate a file transfer service. The content provider decides what content is to be sent and using the security service prepares each multimedia object by inserting details in the session database. If encryption is required, the keys for the content are then generated and then the content is encrypted. The session server periodically transmits announcements (containing information regarding the service/set of services) to receivers at a pre-determined frequency. Clients use the session announcement to discover the group and port number associated with the transfer and awaits configuration information from the server. A receiver application tool wishing to join an advertised session uses the session client to identify the
corresponding transfer server and set appropriate IP multicast group and port information. The security server transmits a set of session/control keys to the security client to establish a Security Association (SA) to the server. Receivers may acknowledge receipt/request transmission of security keys at their own time before decrypting the encrypted file.

**Figure 4.3:** Packet exchange between client and server applications for return path clients (note: * Security request and key may be sent any time after SAT-RMTP clients starts)
Once started by the session server, the file transfer (SAT-RMTP) server is autonomous, and does not directly communicate with the session server. The content tool may choose to monitor the transfer progress by subscribing to appropriate multicast groups. The content tool schedules the start of file transfer. Receivers requesting a service, have to wait for the next scheduled transmission start time for the service requested. During this time, the session server periodically sends multicast announcements of services that have been scheduled to start. Encrypted data is transmitted to downstream receivers over different layers with varying data throughput. Receivers participating in a session may subscribe to more additional layers (based on their local bandwidth capacity and network congestion level) for the same session. The design aspect of SAT-RMTP protocol, based on layered transmission, ensures that both network congestion (along the data path) and data flow rate are controlled in a TCP-friendly manner [7, 8]. Once a complete encrypted file is received, the client security component decrypts the file using either the session/control keys received from the security database.
5 The Transfer Service

The application tool uses Satellite Reliable Multicast Transfer Protocol (SAT-RMTP) for reliable transmission of data between the source (transmitter) and receivers registered in a session. The SAT-RMTP protocol, adopts a number of simple design paradigms, which are efficient and robust, especially over channels experiencing high bit-error rates and persistent outages [9]. The protocol is designed to inter-operate with other protocol standards currently being considered by the IETF [10] and is also extensible for future modifications (as technology changes).

During data transfer, the SAT-RMTP server does not have to keep track of the number or location of clients (registered in a session). Clients start receiving data by transmitting session requests to the server for an advertised session (see section § 4.3). When invoked by the session server, the SAT-RMTP server starts transmitting data to registered receiver(s) in the session. At the receiver, a SAT-RMTP client reliably receives encrypted data. The encrypted data is temporarily stored and later decrypted when the whole file has been received. Receivers also leave a group when all the required data packets have been received (which may occur before decryption).

In the rest of this sub-section, we outline some of the critical issues in designing reliable multicast protocols and highlight how these issues are addressed in SAT-RMTP design. We also present results of test experiments carried out using SAT-RMTP.

5.1 Reliable Multicast Transfer Service Issues

One of the key issues in designing multicast transfer protocols is the optimisation of the application performance, while minimising use of the server and network capacity (such as satellite networks) [11]. Different multicast applications have varying requirements for the underlying transport protocol [2, 12, 13]. Although a number of researchers have sought to propose generic multicast protocols, recent work has shown that different multicast applications have widely varying requirements [14], and the notion of one-size-fits-all is near impossible to achieve [15]. In addition, characteristics of the network topology (such as delay, high bit-error rate and persistent link outages—inherit in satellite and a number of wireless networks) may further exacerbate these challenges—outlined in the following sub-sections.

5.1.1 Reliable Multicast

Reliability, in reliable multicast protocols, implies that the sender gains knowledge of the status of receivers [16]. The same is also true for unicast reliability—although in this case a relatively simple technique may be used to obtain the remote state. In a multicast network, however, there is no such one-on-one reliability (as in unicast case) [15, 17]. One must deal with partial delivery failure (i.e. sender is unaware of receivers that do not complete a file transfer successfully) and problems caused by maintaining data reliability to large receiver groups (e.g. feedback implosion). The SAT-RMTP protocol partly addresses these issues by adopting a receiver-based transmission [18], where multicast receivers are ultimately responsible for the delivery of the data and receivers are assumed to have received all the packets, unless the transmitter receives a retransmission request. More so, in this design, a server may always commence data transfer (for an advertised session) as long as there is at least one client registered (for that session)—a philosophy adopted by SAT-RMTP protocol.

Performance of reliable multicast transfer protocols deployed over topologies with long latencies (such as satellite networks) may be constrained by the round-trip delay between the server and clients. Longer round-trip time delays may cause the server to take a longer time to respond to receivers’ retransmission requests (caused by data loss over the link). This may subsequently result in
low application throughput, since receivers have to wait twice the path round-trip times before receiving data retransmissions. SAT-RMTP addresses this problem by using packet-based Forward Error Correction (FEC) [19, 20]. FEC encoded redundant information is transmitted with the original data. With sufficient redundancy, receivers may then reconstruct any missing data (i.e. packets) using the redundant information without requesting for retransmission.

For large receiver groups, as envisioned in the GEOCAST network model, the server has to be wary of feedback implosion. Feedback implosion is caused by simultaneous transmission of control messages (i.e. ACKs or NACKs), from the receivers to the source (server). This implosion of control messages may overwhelm either the network (i.e. in-bound satellite capacity) or the server resources [21]. In order to minimize this effect, SAT-RMTP adopts feedback suppression techniques similar to those based on time-multiplexing (client retransmission messages) [17, 21]. These techniques may further be used to implement retransmission slots at receivers similar to the sloting approach used in [22].

### 5.1.2 Congestion Control

The successful operation and proliferation of the Internet Protocol (IP) [23] technology, has mainly been due to the wide-scale deployment of the standardized, Transport Control Protocol (TCP) [24]. TCP provides a reliable unicast data transport protocol with standardized end-to-end congestion mechanisms [23, 25]. Unlike TCP, reliable multicast transport protocols lack such congestion control standards, that can operate robustly across a wide range of network scenarios and support traffic fairness when deployed alongside existing network transport standards.

Several TCP-friendly congestion control mechanisms (applicable to RMT protocols) have been suggested [7, 26-28], and subsequent surveys conducted [29]. Despite these efforts, there is still no (Internet-based) multicast transport protocol standard, capable of providing effective and dynamic congestion control methods, safe enough to be deployed in large-scale over end-to-end rate adaptive file transfer applications [30]. At least for small-scale transfers (less than 100 clients) this, however, is likely to change in the near future, especially with the on-going research and standardization efforts in the IETF (where these issues are currently being addressed) [10].

One of the key requirements of widespread deployment of RMT protocols via satellite links (attached to terrestrial networks such as GEOCAST) is, an adequate congestion control mechanism [11]. The common philosophy adopted in implementing congestion control mechanisms for hybrid satellite-terrestrial networks, is to base it on a rate adaptive design space [26, 29]. This may be either sender-based (where the transmitter throttles the transmission rate based on appropriate congestion signals along the traffic path) or receiver based (where receivers implicitly choose to regulate the receiving rate based on the local congestion signals). In either case, the congestion control mechanism must first be able to accurately detect when the network is at optimum capacity; and adopt desirable behaviour (i.e. TCP-friendly with competing network flows). The other specific concern especially in hybrid satellite-terrestrial networks, is the ability of the congestion control mechanism to be able to react separately to congestion signals (such as packet loss) that may occur either over satellite links (due to frequent outages) or along the network path (due to queue congestion).

SAT-RMTP addresses these issues by adopting a receiver based rate congestion control mechanism. The transmitter sends data over different layers at varying transmission rate. Receivers then implicitly tune in and receive data from any number of layers (including the base layer). During transmission the receiver may choose either to unsubscribe or subscribe to additional layers depending on the local congestion level. It is important to note that rate control of the source is an orthogonal issue to that of recovering packets dropped by the network. Rate control, in this case,
reduces the likelihood of extended periods of heavy packet loss, but does not eliminate congestion losses [21]. The SAT-RMTP protocol employs fixed rate transmission by the transfer server. Although not implemented, a future SAT-RMTP protocol version may also make use of the bandwidth reservation services (e.g. lightweight QoS) to adapt the sender rate.

5.2 SAT-RMTP Protocol Operation
As outlined in section § 1.0, the file transfer provided by the application tool service is performed by the SAT-RMTP protocol. The SAT-RMTP protocol makes use of both unicast and multicast data transfer techniques, for transmission of data between the SAT-RMTP server and client. The transfer protocol uses a set of UDP sockets [31], each bound to either a multicast or unicast IP address, for data exchange between the server and client. Both the server and client protocol operations and functions use different packet types each containing varying sets of parameters.

5.2.1 Data Transmission
Transmission of data objects (i.e. encrypted video clip files) is initiated by the session protocol (see section § 4.2). Each file transfer instance is started by a single transfer request call from the session protocol. Since SAT-RMTP protocol is receiver-based, the server is not necessarily aware of client receivers or the progress of any individual client. Instead, each SAT-RMTP client takes responsibility of its own data transfer. Once initiated, the server transmits an entire multimedia object as sequence of FEC encoded data blocks (referred to as a DATA packet type) using UDP, over configured data layers (channels), each associated with a different IP multicast group address.

The SAT-RMTP protocol also uses different other sets of packet type. These packets include: STATUS—transmitted by both sender and receiver to control the operations and functions of the protocol and NACK STAT—transmitted by receivers for data retransmission request from the server. Each data packet transmitted has a common header. This common header contains a unique session identifier (i.e. session ID) that uniquely identifies an advertised session. Both the server and the receiver, use the session ID to differentiate between the different sessions (i.e. transfers of different multimedia objects) that may have been requested from the transmitter. The SAT-RMTP protocol also uses a flow identifier (i.e. flow ID) to identify the different data layers (channels) used for transmitting data in a given transfer.

5.2.2 Transmission Loss and Repair
A SAT-RMTP server assumes that receivers correctly receive the data transmitted, unless it receives a retransmission request. This implies that there is no need for the receivers to acknowledge blocks of data that may have been received successfully. Each block consists of data packets with additional FEC encoded redundant information. Clients use the FEC redundant information (sent along with the original data), to perform local repair of missing packets in a block without the need for retransmission from the server. Although, transmission of redundant FEC information may require additional link capacity [32], its application in protocol design makes SAT-RMTP protocol more robust, especially when uncorrelated data loss occurs over the link1.

SAT-RMTP receivers may request for retransmission of lost data packets (from the server), after the expiration of a random timer. SAT-RMTP receivers adopt a random timer that is based on an exponential distribution similar to [34]. Using an exponentially distributed timer reduces the potential of retransmission request packets (NACK STAT), sent by the receivers, causing implosion

---

1 This is appropriate for uncorrelated loss pattern experienced over the link when uncorrelated rain fades occur in a satellite footprint [33]
(at the server) when a large number of receivers fail to receive packets. The initial random seed used for randomising the feedback packets is determined by the low-order Ethernet MAC address (or if not available by the IP source address) of each client [22].

Upon receipt of a retransmission request packet (NACK_STAT) from a receiver, the SAT-RMTP server adds the requested data packets in a block to its retransmission list. The server then schedules all marked data packets (with respective missing packets) on the list, for transmission to the receivers. Receivers may receive one or more copies of data packets in an arbitrary order—but since the protocol is idempotent [35] duplicate packets have no effect and are discarded by the receiver.

The SAT-RMTP server may, in future version, also sends control packets (STATUS) periodically with synchronization information. Receivers use this information to synchronize receiver state (i.e. recovering from a failed data layer join, or attempting to perform receiver-driven congestion control). When the server reaches the end of the object transfer (i.e. a video file clip), the server waits for a pre-defined period for any retransmission request from receivers. If no repair request is received on/before the expiration of waiting period (i.e. idle time), the server assumes that the delivery was successful. After successfully completing an object transfer, a receiver may leave the session without informing the server. As discussed earlier in section § 4.1, SAT-RMTP also assumes that there are no late joiners once a transfer service has begun.

The current version of SAT-RMTP protocol does not have a mechanism to detect typical Crying Baby Problems—where a client either through software, hardware, or configuration error persistently fails to receive some packets, thus transmitting excessive repair requests (perhaps even for the same block) to the detriment of other receivers. This problem was, however, minimized in the current SAT-RMTP protocol design by allowing future versions of the SAT-RMTP sender to eject the would be Crying Baby receivers from an advertised session.
6 SAT-RMTP file transfer tests

This section of the report, discusses the test experiments conducted using the SAT-RMTP protocol. The experiments were conducted, in this project work, to show the correct operation of each of the various mechanisms implemented in SAT-RMTP protocol.

In the following sub-sections of this section, we first describe the setup and configurations used for the test experiments. The basic OS platform used was consistent to the GEOCAST testbed configured in both ALCATEL Space Industries (ASPI) and University of Aberdeen (UoA). This is then followed by an analysis of data traffic monitored and captured at each respective client (or receiver).

6.1 Setup and Configuration

6.1.1 Test Configuration

As shown in figure 6.1 below, the test configuration consisted of a SAT-RMTP server running on a Sparc Ultra 10, Sun OS/Solaris 2.6 and linked via a 100 Mbps CISCO switch to two SAT-RMTP clients. The SAT-RMTP clients were running on Sun OS/Solaris 2.6 and Linux Mandrake (Intel) PC located on the same sub-network 139.133.204.0.

![Testbed Equipment Setup](image)

**Figure 6.1:** Testbed equipment setup and configuration

Parameter and configuration settings used for the LAN testbed, as listed in table 6.1 below, were kept constant for consistency in analysing data collected. The current version of the SAT-RMTP protocol enables receivers to tune in and receiver data over two different data streams\(^2\), with an adjustable minimum base rate i.e. 256 Kbps used in this case.

---

\(^2\) Later version of SAT-RMTP may have more data stream layers
Table 6.1: Parameter setting and configuration used for test experiments

<table>
<thead>
<tr>
<th>Component</th>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Announcement</td>
<td>Session IP Multicast address</td>
<td>224.2.2.9</td>
</tr>
<tr>
<td></td>
<td>Session IP Multicast port</td>
<td>6400</td>
</tr>
<tr>
<td>Server</td>
<td>Server IP Address</td>
<td>139.133.204.95</td>
</tr>
<tr>
<td></td>
<td>Server UDP unicast port</td>
<td>3450</td>
</tr>
<tr>
<td>Client</td>
<td>Client IP Address (Solaris SunOS)</td>
<td>139.133.204.81</td>
</tr>
<tr>
<td></td>
<td>Client IP Address (Linux OS)</td>
<td>139.133.207.13</td>
</tr>
<tr>
<td>Data Streams</td>
<td>256 Kbps - Group IP Address/port</td>
<td>239.2.2.4/5440</td>
</tr>
<tr>
<td></td>
<td>512 Kbps - Group IP Address/port</td>
<td>239.2.2.4/5441</td>
</tr>
</tbody>
</table>

6.1.2 Session Directory (SDR) Integration

The application uses the Session Directory (SDR) tool [36], to enable service advertisement (by the sender) and joining of advertised sessions (by the receivers) respectively. SDR uses the Session Announcement Protocol (SAP) [37], to periodically multicast session announcement packets describing a particular session. Receivers simply listen on a well-known multicast address and port, in order to receive SAP announcements. Session Description Protocol (SDP) [5] is used for describing the advertised sessions. When a receiver receives a SAP announcement packet, it simply decodes the SDP message and can then display the session information using the SDR tool.

Multicast sessions generated by the SDR tool enable anyone, receiving the traffic, to join in the session (unless the session traffic is encrypted). As mentioned earlier, SDP is used for two main purposes: first, to communicate the existence and timing of a session; and secondly, to convey sufficient information to enable receivers decide whether to/not to join and participate in an advertised session.

In the following sub-sections, we first illustrate how SDR tool was configured to support the SAT-RMTP protocol and present step-by-step command sequences (used in test experiment) to create or join a session. For further details of the SDR tool and its functions, we refer the reader to the user manual [36].

6.1.2.1 SDR plugin for SAT-RMTP protocol

The SDR tool was configured to support the SAT-RMTP protocol by means of plugin modules [36]. Plugins define the media, protocol and formats supported by an application tool. They also define, long-form names for these formats and the flags that are needed to start up the application tool. On the Solaris/Sun OS or Linux OS platform, used in the test experiments, SDR tool read the plugin files located in `/usr/local/etc/sdr/plugins` and `~/.sdr/plugins` in search for files whose name begins with `sdr2.plugin`. The naming convention for plugins is `sdr2.plugin.sequence.media.protocol.format.tool` where:

- `Sdr2.plugin` is a prefix for the plugin (always present)
• *Sequence* is form $S_\gamma y$ where $\gamma y$ is a sequence number of which $x$ seems be encoded as follows:

$x = 0$: audio  
$x = 1$: video  
$x = 2$: whiteboard  
$x = 3$:  
$x = 4$: text  
$x = 5$: games

• *Media* is the name of the media, i.e. audio, video, whiteboard, picture, text, web, game

• *Protocol* is the name of the protocol used to transfer the packages, i.e. udp, rtp, wb, ltmp, sat-rmtp

• *Format* represents the encoding format, i.e. for video: h261, mpeg, jpeg, mov. This field can be replaced by the wildcard character '*-' to indicate that several formats are possible, the user will choose to the last moment the most fitting format.

• *Tool* is the name of the tool such that exists on the local machine i.e. for SAT-RMTP the executable file sat_rmtp_client, must exit in the local machine in order to invoke the transfer client from SDR.

In addition to the default SDR plugins, table 6.2 below, lists the plugin file name and source code integrated in SDR tool to support the SAT-RMTP protocol.

**Table 6.2: Details of SDR plugin for SAT-RMTP protocol**

<table>
<thead>
<tr>
<th>File name</th>
<th>Sdr2.plugin.S61.file.sat-rmtp.-.sat_rmt_client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plugin source code</td>
<td></td>
</tr>
<tr>
<td><code>media</code> file</td>
<td></td>
</tr>
<tr>
<td><code>protosat-rmtp</code></td>
<td></td>
</tr>
<tr>
<td><code>tool: sat_rmt_client</code></td>
<td></td>
</tr>
<tr>
<td><code>fmt: sat-rmtp-file</code></td>
<td></td>
</tr>
<tr>
<td>`{</td>
<td></td>
</tr>
</tbody>
</table>
|   `flags: -$\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\langle\ associate::

6.1.2.2 Creating a Session

Creating a session using the SDR tool first requires the user to specify:-

• Session information and type

As shown in figure 6.2, SDR allows a user to specify session information which include: the session name, brief description of the session and optional URL specification (for further details). Session information enables receivers, of announcements, to have information regarding the service being offered. Session type, unlike information, specifies the type of session i.e. test (for anything note
intended for real users); meeting (for private and interactive sessions); broadcast (for sessions that are largely non-interactive); or directory (to be checked).

![SDR session information and type window](image1)

**Figure 6.2:** SDR session information and type window

- Timing and distribution scope

As shown in figure 6.3, SDR enables the user to specify time details and distribution scope for the session.

![SDR session timing and distribution scope window](image2)

**Figure 6.3:** SDR session timing and distribution scope window

Time details include: when the session will begin (day, date, month and time); the duration (approximate) of the session; and how often it will take place once started. Timing information is critical, since it enables users to know when to join the session if interested. Distribution scope value, based on TTL or administrative scooping [38], determines how far away the content provider...
wishes the traffic from the session to be received. This can either be local—within a campus or intranet; region—within a continent; or world—available worldwide.

- Media configuration and contact details

Figure 6.4 shows the media configuration that enables a user to specify the correct media type and format for an advertised session. This allows a SAT-RMTP receiver announcement to start the SAT-RMTP application (i.e. sat_rtmp_client executable file) directly from SDR tool with all the correct flags and settings for joining the session. The applicable media formats include: audio, video, whiteboard, text, directory and file3.

![SDR session media and contact details window](image)

**Figure 6.4:** SDR session media and contact details window

Contact details provided, as shown in figure 6.4 above, enable receivers wishing to join (or who have already joined) a session, to make personal contact with the sender in case of a problem or further interest in the content being transmitted.

- Security parameters

![SDR session security parameters window](image)

**Figure 6.5:** SDR session security parameters window

---

3 *file* media format was configured using SDR plugin for SAT-RMTP protocol (listed in table 6.2)
• Confirmation of session configuration

SDR tool allows the user to review the session details, as shown in figure 6.6 below, before commencing transmission of the session. This gives a user the opportunity to change, omit or cancel altogether the session configuration and subsequent transmission.

![SDR Create New Session](Image)

**Figure 6.6:** SDR session review of session details

### 6.1.2.3 Joining a Session

When a client starts SDR the main session window will appear with a list of currently available public sessions (see figure 6.7). To join a session, a user simply clicks on the preferred session and the client session window will appear. The user is then able to join, invite, record, or dismiss the session.
When the user selects join, the SDR tool invokes the plugin supporting SAT-RMTP, which, in turn, launches the `sat_rmtp_client` executable file (to start the SAT-RMTP client module).

![Main SDR session and Client Session window with announcement packet details](image)

**Figure 6.7:** Main SDR session and Client Session window with announcement packet details
6.2 Tracing File Transfer

Traffic traces conducted between the sender and receivers was aimed at establishing correct operation at different modules implemented in the SAT-RMTP protocol. This was achieved, in this project work, by monitoring and capturing snapshots of packet transmissions either to- or from- the sender or receiver network interface. Using readily available traffic snooping tools, such as tcpdump [39], we were able to analyse sequences of operations and interaction between the SAT-RMTP components and the session protocol.

The following step-by-step sequences of traffic dumps show how data is reliably delivered from a SAT-RMTP server (source) to a SAT-RMTP client (receiver); and also the interaction with the session protocol. These traffic dump snapshots represent the interaction at one of the clients (i.e. client: 139.133.204.81).

6.2.1 Service Announcement

Figure 5.2 shows the receiver (client) joining a well-known session address (224.2.2.9), to listen for advertisements regarding services available at the receiver.

```
14:43:09.958208 139.133.204.81 > 224.2.2.9: igmp nreport 224.2.2.9 [ttl 1]
```

**Figure 6.8:** Session client joining a well-known session address

Figure 5.3 below, shows the periodic service announcement transmitted by the session server regarding a particular service/s (in this case file transfer), which may be of interest to receivers downstream.

```
14:43:11.731073 139.133.204.95.6400 > 224.2.2.9.6400: udp 1024
14:43:11.731653 139.133.204.95.6400 > 224.2.2.9.6400: udp 1024
14:43:11.732272 139.133.204.95.6400 > 224.2.2.9.6400: udp 1024
14:43:11.732964 139.133.204.95.6400 > 224.2.2.9.6400: udp 1024
14:43:11.733581 139.133.204.95.6400 > 224.2.2.9.6400: udp 1024
```

**Figure 6.9:** Session server transmits announcements periodically

6.2.2 Data transfer

Receivers interested in a particular service, join the group where the service has been advertised. In our case, figure 6.10 shows, the receiver joining a data stream group (239.2.2.4/5441) advertised by the SAT-RMTP server to registered SAT-RMTP clients. The current version of SAT-RMTP protocol paces data transmission over two data rates layers i.e. 256 Kbps or 512 Kbps (see table 5.1).

```
14:43:11.735879 139.133.204.81 > 239.2.2.4: igmp nreport 239.2.2.4 [ttl 1]
```

**Figure 6.10:** SAT-RMTP client joining data stream

Figure 6.11 below, shows encoded data packets transmitted by the SAT-RMTP server to the group multicast address corresponding to the respective data stream. SAT-RMTP receivers tune in to this group multicast address to start receiving encoded data.
When packet loss occurs, figure 6.12 below shows, a SAT-RMTP receiver requesting for retransmission of missing packets from the server. Retransmission request in the current version of SAT-RMTP is sent as UDP unicast to a well-known data port at the server. Return traffic from the receiver may also include control information (from the clients) requested by the sender.

Figure 6.12: SAT-RMTP client requesting for packet retransmission

A SAT-RMTP receiver network layer leaves a data stream group, as shown in figure 6.13 below, after either receiving adequate number of encoded data packets needed to decoded a packet. This may also occur when receivers react to congestion levels along the network.

Figure 6.13: SAT-RMTP client leaves a data stream group

Lastly, when a SAT-RMTP receiver has received the requested service and no more service is required, it leaves that particular session as shown in figure 6.14 below.

Figure 6.14: SAT-RMTP client leaves the session
7 Asymmetry Implications on SAT-RMTP protocol

In this sub-section of the report, we discuss the implications of network path asymmetry, on the SAT-RMTP protocol performance, and suggest appropriate mitigation techniques. Large-scale deployment of the SAT-RMTP protocol over satellite networks (that are inherently asymmetric), such as GEOCAST network, may from time-to-time experience performance degradation (as illustrated in section § 6.3). Performance degradation may be due to either persistent link outages (resulting in high packet loss); or the imperfection and variability of the return traffic (or feedback information) from receiver(s) to sender.

Asymmetry may manifest itself as: the differences in transmit and receive capacity; imbalance in the packet loss rate; or the differences between transmit and receive paths [40]. Data traffic generated by most reliable multicast protocols, such as SAT-RMTP, is mainly asymmetric (i.e. more traffic on the forward link than the return link) [41]. This traffic characteristic, however, may be exacerbated further (especially in satellite networks), when multicast flows carrying feedback or control information share the available upstream capacity with other traffic flows. Some flows may have reserved upstream capacity or guaranteed class of service, compared to other normal flows. This treatment further alienates flows over the available upstream capacity.

In the case of a one-to-many topology (e.g. Satellite case), where a single source transmits data to a large number of receivers downstream, the source often incurs less transmission overhead or latency in scheduling transmission. In contrast, however, the upstream transmission (from receivers to source) is mainly subjected to additional overhead and latency (e.g. due to either retransmission timeouts or contention intervals), causing network path asymmetry. Satellite systems deploying dynamic-reservation based capacity assignment (Bandwidth-on-Demand) schemes; incur significant overheads caused by MAC protocol signalling when reserving capacity for each packet sent on the return link [42]. These schemes also introduce overheads per upstream transmission, which can be significantly high irrespective of the packet size (i.e. transmitting short packet becomes as costly as transmitting MTU-sized data packets).

Since multicast return traffic (feedback traffic) is mainly of low volume and random in timing, there is a potential benefit, especially in satellite networks deploying dynamic-reservation BOD schemes, to optimise transmission of return traffic. In this sub-section of the report, however, we discuss and rationalise the composition of SAT-RMTP return traffic and suggest appropriate mitigation techniques that may be adopted, to enhance its performance over asymmetric network path.

7.1 Return traffic

A number of reliable multicast protocols, such as SAT-RMTP, rely on the availability of a return link for reliability. These protocols mainly use the return path to transmit either data or control information to other receivers; the source; or parent node. A part from data transmission, SAT-RMTP receivers also transmit feedback control information, which may relate to: the session service; congestion and flow control indicators; acknowledgement or retransmission request of missing data; transfer status (reject, abort, complete); or other signalling parameters (triggered over the protocol stack).

7.1.1 Session feedback

Before an IP multicast source starts transmitting data for a particular session, it is the philosophy of the SAT-RMTP protocol that, at least there should exist a receiver downstream interested in that session. Receivers listen on a well know multicast address and port number for announcements and
thereafter, send a *join request* (for the announced service) to a router upstream. When the source receives at least a single request, it then commences transmission.

In a single-hop satellite case, where large numbers of receivers are a single-hop away from the source and probably listening on more than one multicast channels, simultaneous receiver *join request* may incur considerable MAC signalling overheads, caused by requesting of capacity from the NCC for each receiver *join request* transmitted. Although BoD schemes may mitigate these MAC overheads at the network level, in practice, this may have far-reaching consequences for delay sensitive applications [12].

In such cases, especially where Source Specific multicast (SSM) [43] is deployed, a solution based on layer 3 may be more desirable. One particular alternative may be the use Multicast Source Notification of Interest Protocol (MSNIP) [44]. MSNIP operates between a multicast source and shared first hop routers or network node, to provide information on the presence of receivers to the source system. Using MSNIP services an application wishing to source multicast data can register to be notified when receivers join and leave a session.

MSNIP allows the multicast source to avoid the work of transmitting packets onto their first hop when there are no joined receivers. This may have significant benefit, in wide-scale deployment of SAT-RMTP over satellite, since the source does not have to waste bandwidth by transmitting packets to unavailable receivers downstream. MSNIP use may also reduce session return traffic significantly, since receivers (downstream) can be notified of sessions, and decide to join, without sending *join request* to upstream router.

### 7.1.2 Congestion and flow control feedback

As mentioned in section § 5.1.2, the common philosophy adopted in implementing TCP-friendly congestion control techniques, is based on rate adaptive design space [29]. These techniques are based on either the explicit approach—where sender throttles the transmission rate, based on an appropriate congestion signal along the traffic path, or the implicit approach—where receivers choose to regulate their own receiving rate, based on local congestion levels. It is desirable, therefore, that either technique must be able to accurately detect when the network is at optimum capacity; so as to adopt an appropriate behaviour (i.e. be TCP-friendly with competing network flows).

Adopting TCP-friendly characteristics, in these techniques, may require some form of signalling between the transmitter and the receiver. In the SAT-RMTP protocol, both congestion and flow control is implicitly based (with minimal or no return traffic required to negotiate rates). However, when deployed over satellite networks, there is a need for the transmitter to have adequate information regarding the available uplink capacity (which may vary with link conditions), and also existing congestion levels over the extended terrestrial network. An IP layer based technique, such as Explicit Congestion Notification (ECN) [45], based on setting the ECN bit, may be used to transmit relevant signals back to the transmitter regarding either downstream congestion levels or bandwidth capacity availability over the satellite channel [11]. This information could be transmitted either periodically (based on a preset or random timer), or when requested by an upstream router/node. Priority scheduling suggested for BoD schemes, may be used to mitigate overheads caused by (congestion and flow control) upstream transmission [11].

### 7.1.3 Reliability feedback

In wide-scale deployment of reliable bulk data multicast, where a source transmits data to a very large group of heterogeneous receivers, not all receivers complete file reception (or receive all data packets
sent) after a file transfer session. In reliable multicast techniques (such as SAT-RMTP), which rely on a return path for reliability, such receivers experiencing loss, may request for retransmission of lost data by sending either an acknowledgement (ACK) of data received or a negative acknowledgement (NACK) of data not received [10], to the source upstream.

As mentioned earlier in section § 6.4.1.1, simultaneous upstream transmission of feedback, especially over satellite networks employing dynamic-reservation based capacity assignment schemes, may incur significant delay overheads [11]. Although, BoD schemes may be used to mitigate these overheads, end-host techniques may also be used to limit these overheads.

Feedback suppression techniques, i.e. NACK/ACK suppression, may be used to reduce the amount of NACK feedback traffic (as in the case of the SAT-RMTP protocol) transmitted upstream to the source. In the SAT-RMTP NACK suppression scheme, receivers defer NACK transmission until the expiration of an exponential random timer. Each NACK transmitted is a bit-map list of missing packets, which aggregates further the upstream traffic. As discussed in section § 6.4.2, use of multicast over unicast, to transmit return traffic, may be desirable option for suppressing further duplicate retransmission request from receivers in the same group.

7.1.4 Status feedback

In a number of reliable multicast protocols, the transmitter may from time-to-time, probe or require downstream receivers to send status feedback. This status information (such as a rejected file, abort transfer, completed session, security keys needed) may be useful to the transmitter, in terms of making accurate decisions or service adjustments (i.e. user logs, billing, capacity usage etc) for a particular session (especially in a closed service). In wide-scale deployment, however, keeping track of all receiver status may impede scalability of a protocol [41]. In such cases, it may be desirable that similar status information is aggregated either on a per-group or per-session or per-source basis, at the upstream router/node. Aggregating information in this way, although not effective in the closed service model, may significantly reduce upstream traffic in wide-scale multicast deployment. In extreme cases, where a particular receiver is experiencing cry-baby problems within a group or session, self-ejection based on the expiration of a preset timer may be desirable. It is therefore hoped that, application triggered techniques such as MSNIP [46], discussed in section § 6.4.1.1, may be used to provide useful information to mitigate overheads caused by status feedback.

7.1.5 Lower layer feedback

In satellite networks, although not noticeable in the upper layers (i.e. transport, session and presentation layer), the lower layers (i.e. network, link and physical layer) also generate a significant amount of feedback traffic [42, 47]. Both the transmitter and receiver generate traffic, at the link layer, when transmitting/requesting frames or negotiating capability. Receivers that receive frames with bit errors (caused channel noise) or delayed over the link (and perceived lost), may request for retransmission. High frame errors over the link may exacerbate upstream link traffic. Similarly, the network layer signalling (as discussed briefly in sections § 6.4.1.1 and § 6.4.1.2) also generates a fair share of traffic. This volume of traffic is mainly due to either router or network signalling. Control information regarding Quality of Service (QoS), Class of Service (CoS) or router operation, may constitute the upstream network layer traffic. Multiplexing different upstream network layer traffic (e.g. on a per-flow ID basis) may be desirable, to mitigate upstream transmission overheads in the lower layers.
7.2 Unicast Vs Multicast Return Traffic

IP multicast transport technique [48], provides the middle ground between one-to-all delivery provided by broadcasting (where packets are delivered to all end hosts within the same IP network) and one-to-one delivery provided by unicast (where packets are delivered to a single known destination) [23]. Compared to broadcast, multicast technique is selective, i.e. allows simultaneous delivery of data to a set of end host with a common interest.

In wide-scale multicast deployment via satellite networks, where a single receiver delivers data to very large number of receivers, use of either unicast or multicast mode of transport over the upstream link, may be desirable. Multicast use for upstream traffic enables a set of end host receivers (registered to a particular group), to receive a common set of data (i.e. data packets for repair or caching) at the same time. End host receivers intending to request for retransmission, may suppress their respective NACKs if the same data packets are requested. This mode of transmission optimises the usage of upstream capacity, which is limited in satellite networks, compared to separate unicast upstream transmissions. Non-broadcast multiple access (NBMA) links (such as satellite upstream links), may need to be rebroadcast data packets to intended downstream receivers.

Unlike multicast, unicast transmission may be desirable for both frequent and non-frequent upstream traffic (consisting of either data or aggregated control information), to a source or a requesting end host receiver registered in a group. Upstream unicast transmission to a source or end host receiver (registered in a group), may generate less duplicate retransmissions to other receivers (registered in the same group). However, this may lead excessive usage of upstream capacity when the same data is transmitted to different receivers registered in the same group.

Use of both unicast and multicast transport techniques, for upstream traffic, may be desirable. End hosts receivers in a hybrid satellite-terrestrial network may request for data retransmission by multicasting NACK requests over an administratively scoped sub-network. Receivers listening in the same group with the requested repair packet may choose to unicast the repair packet to the requesting receiver. If there is no NACK respond, the requesting receiver may choose to expand further the scoped region or unicast a NACK request from the source. This is well suited for source specific multicast (SSM) [43], where receiver end hosts choose the source of multicast packets.

In table 6.3 below, a comparison between unicast and multicast upstream return traffic implications is presented. It should be emphasised this is based on the implications on receivers registered in the same group with a single source (transmitter).

Table 6.3: Comparison between unicast and multicast return traffic

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Capacity Usage Per Group</th>
<th>Redundancy Per Group</th>
<th>Resources Per Group</th>
<th>Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>More for set of receivers</td>
<td>Less duplicate retransmission</td>
<td>More transmitter resources needed</td>
<td>Desirable for one recipients</td>
</tr>
<tr>
<td>Multicast</td>
<td>Less to set of receivers</td>
<td>More duplicate retransmission</td>
<td>Less transmitter resources needed</td>
<td>Desirable for set of receivers</td>
</tr>
</tbody>
</table>
8 Security Service

All clients have a Secure Association (SA) with the security server. In general, this SA may either be secret key based or public key based. In a secret key SA each client shares a secret key with the server. In a public key SA each client has a certificate, which is stored at the server, and its own private key, which is stored locally. In the public key SA each client also stores the public certificate of the security server. For the demonstrator described here, public keys are used, and SA parameters, such as the choice of keying algorithms and length of keys, are manually configured.

8.1 Scope

The architecture described below assumes that each client has a return path to the server. Receive-only clients may be considered separately (for receive-only clients, authentication and initialisation has to be done offline, for example using email or fax).

Encryption of the session announcements may also be provided, and will be considered in a later version of the demonstrator. If the session announcements are encrypted and signed (to form a private group), then the session client needs to be configured with security information. Two options to achieve this are as follows:

- Including this security functionality in a custom session protocol and custom session server; or
- Session announcement authentication and encryption parameters are given to the authorised clients offline (e.g. using PGP) in order to configure an off-the-shelf session package such as SDR.

8.2 Architecture

The security services provided in the demonstrator are:

- Authentication of users (clients) - using public keys, which are manually configured prior to file transfer.
- Access control - ensuring that a user has permission to access the given file/object.
- Key exchange using public key algorithm.
- Confidentiality – encryption of object data, using secret key algorithm.
- Integrity – provided on the whole file / object using MACs, to confirm the sender identity and ensure that the file is unaltered.
- Detection of replay attacks using random numbers.

Figure 8.1 shows the architecture of the security server. There are two databases: a user information base (containing client access permissions) and a security database (where keys are stored). There are five functions of the security service: a certificate utility module, a key generation tool; an encryption tool (also known as the content tool); an authentication and access control module; and an initialisation module. A client/server security protocol provides authentication and access functionality, and securely distributes keys to clients (using the secure association which is assumed to be manually configured prior to file transfer).
The certificate utility module enables the security server to act as a Certificate Authority (CA) and issue certificates to clients on request. This is done offline before the security protocol takes place. Thus every client is in possession of its own certificate and password protected private key. For the purposes of this demonstration the certificates are issued as follows: the security server first generates its own CA public key pair and certificate. A client generates a public key pair and then using the public component creates a certificate request, which it sends to the CA. On receiving this request the CA then generates a certificate for that client, signed with the CA’s private key.

**Figure 8.1: Security server architecture**

An outline description of each security function for the server now follows:

- The certificate utility module generates its own public key pair and certificate. It is also used to generate certificates on receiving requests from clients.

- The key generation tool generates keys for each object once it has been notified by the content tool that an object is available to be transferred.

- The content tool is used to prepare each multimedia object for transmission, by encrypting the object using a chosen key or keys from the security database. At this stage, the keys are not sent to any clients. The content tool places the encrypted object in a transfer directory for access by the SAT-RMTP server.
• The authentication and access control module checks the client’s identity and access permissions when the security server receives a request from a security client (this request will be sent via the session server).

• The initialisation module issues keys for the object to authenticated security clients. Receipt of the corresponding keys allows each client to decrypt a received object.

Figure 8.2: Security client architecture

Figure 8.2 shows the architecture of the security client. This has four functions:

• The client certificate requester module generates a public key pair for the client and sends a certificate request to the CA.

• The authentication and access control module passes client ID information to the security server. This information is retrieved from the information database maintained at the client, where it has been manually configured into the client.

• The initialisation module receives keys from the server and stores them in the information database.

• Following transfer of the file, the content decryption tool decrypts each object. The tool first checks the file HMAC to verify the integrity of the file, and on successful verification proceeds to decrypt the file. The decrypted file is then passed to a multimedia player. Following successful decryption the security client also sends an acknowledgement to the security server to confirm successful decryption to the server.

Unlike the SAT-RMTP server, which does not have details of individual clients, the security server has a global perspective, and can receive packets from every participant in a transfer. The security server however executes at a lower priority, using a coarse-grain clock, and therefore contributes only a small additional processing overhead, even for a large number of clients.
Figure 8.3 summarises the message flows of the security system. The messages are as follows:

- Certificate request: this message is used to request a certificate from the CA;
- Certificate reply: the certificate payload is sent in response to the certificate request;
- File transfer: this is described elsewhere in this document;
- Authentication and key request: the client requests the decryption keys for the encrypted file, and provides authentication information to the server;
- Key download: on successful authentication and confirmation of access permissions, the server issues the session key(s) for the transferred file.
Conclusions

A summary has been provided indicating the important design issues for the delivery of multimedia objects using the proposed transfer protocol.

A number of trade-offs have been noted. There is a trade-off between network bandwidth, protocol complexity and recovery latency, these are described in detail in the documents relating to the SAT-RMTP protocol. The security requirements and implications are being studied, particularly in relation to the session protocol.

Further investigation is required to determine the impact of different network topologies and network components, and the impact of satellite multicast signalling and bandwidth-on-demand protocols.
References