

The Satellite Reliable Distribution Protocol (SRDP)

Franco Tommasi, Simone Molendini, and Andrea Tricco

Original scientific paper

Abstract – The Satellite Reliable Distribution Protocol (SRDP) is proposed and discussed. SRDP is designed for satellite-based data broadcasting and aims at transmitting reliably multicast via satellite Internet applications' data in an efficient and scalable way.

SRDP operates in three steps, called phases. In phase 1 the sender transmits a file through the satellite internet adding a reasonable degree of redundancy. In phase 2 the sender retransmits a set of new redundant packets to recover some packets that were lost during phase 1. Phase 3 allows each receiver to request all its missing packets by a TCP connection to the sender. In order to evaluate the performance of the SRDP, we have considered two different types of loss behaviours: random and burst. Our results show how to choose the amount of redundancy in order to optimize the performance of the SRDP.

Index terms: Reliable multicast, satellite, scalability

I. INTRODUCTION

The satellite is an almost ideal medium for wide-scale broadcasting: today satellites reach a number of households and businesses far beyond that of traditional broadcast methods [1]. Satellite broadcasting is not as expensive as one might think if the potential number of receivers and the covered geographic area are taken into account. The satellite then can today be used to transmit reliably bulk data like software products, off-line multimedia content, CD-ROM images and product catalogues. A protocol to allow Internet hosts to receive such data through satellites is then required.

Many reliable multicast protocols have been proposed until now and several comparisons exist [2-7], but the use of these protocols in the Internet has been lamented [8]. Since a "one-size-fits-all" solution to the reliable multicast problems does not seem to exist, in this document we propose and discuss the Satellite Reliable Distribution Protocol (SRDP), a protocol designed for satellite-based data broadcasting. From an architectural point of view, SRDP is an application and some of its messages are encapsulated in UDP, others in TCP.

A common weakness of other protocols supporting reliable multicast is that they do not carefully consider satellite networks' peculiarities. Satellite networks suffer from relatively high bit error rates compared with terrestrial fibre trunks. Moreover, the feedback channel could be unavailable for some receivers. SRDP is designed for the networks where the satellite is used as a means to distribute data to a group of receivers. From this point of view, the SRDP protocol aims at

transmitting reliable multicast information in an efficient and scalable way for the star-based topologies that are intrinsic to the satellite transmissions. In fact, SRDP can profit from feedback coming from receivers provided with an outbound connection but it can be used as well to transmit data to receivers provided only with an incoming satellite connection. For example, it would be meaningful to have SRDP transmissions without feedback if receivers are terminals that wait for SRDP services (an instance of such service is the updating of information panels).

The rest of the paper is organized as follows. Section II characterizes the reference topology and Section III provides a basic description of the SRDP. Section IV deals with the performance of SRDP in two contexts of random and burst losses presenting the results of simulations. Conclusions are drawn in Section V.

II. INTERNETWORKING WITH THE SRDP

In this section we analyse the reference topology and the design steps to give SRDP efficiency and fairness.

A. Reference topology and definitions

Satellites provide broadcast transmissions in the fullest sense, since their footprints can be made to cover large areas of the earth. In general terms, basic satellite services provide for simplex communication. Telecommunication and broadcasting services today are showing an increasing trend to add interactivity to traditional TV. There are several ways of implementing a feedback channel for satellite multicast services: commonly-proposed terrestrial return channels include PSTN (Public Switched Telephone Network), ISDN (Integrated Services Digital Network) and GSM (Global System for Mobile communication). Moreover, there is a huge worldwide interest in defining a satellite return channel (duplex transmission) and standards, as RCS (Return Channel via Satellite) [10], to use it already exist: among other benefits, satellite based return channels allow to cover rural or remote areas not covered by terrestrial services.

In our topology, see figure 1, we discuss the SRDP protocol with reference to its use in a "satellite internet", that is a connection of IP networks containing at least a satellite link. Such network might be (or not) connected to the Internet, in both instances using IP protocol.

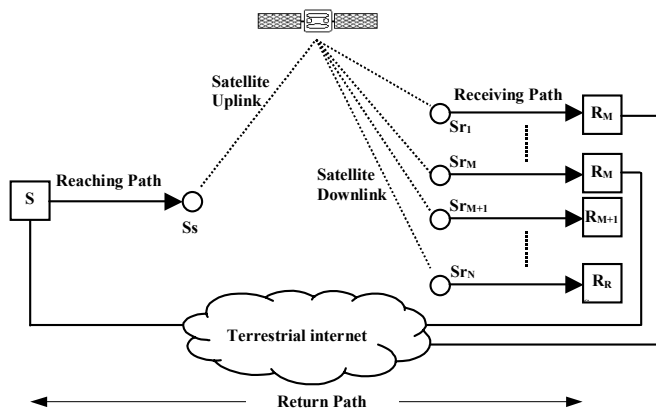


Figure 1. Topology and entities.

A Sender host (S) in this internet is able to transmit multicast data to R Receiver hosts (R_i , $1 \leq i \leq R$), among which M are endowed with a return channel. The last IP (Internet Protocol) node before the satellite uplink is called Satellite Sender (Ss), while the first node after the satellite downlink is called Satellite Receiver (Sr). The satellite link is then the hub of the transmission and the topology is a star-based one. The communication is essentially one-way, though some of the receivers can also have a Return path. Such "Return paths" allow them to send information to S. For convenience we identify three different kinds of subpaths: Reaching path (from the Sender to the Ss), Receiving paths (from the Sr the Receivers), the Return paths (from the Receivers to the Sender) - see figure 1.

The use of the ubiquitous IP as an interconnection protocol allows any host to send data via satellite through Ss, be it directly connected to Satellite Service Center (i.e. the physical location of devices and apparatus to broadcast data via satellite) or not. The SRDP protocol uses UDP-based datagrams (User Datagram Protocol) on the Reaching and Receiving paths and TCP connections on the Return path. It is an end-to-end protocol in the sense that all mechanisms introduced involve only the Sender and the Receivers: the other nodes along the path between sender and receivers have to be able to forward UDP multicast (from sender to receivers) and TCP flows (from receivers to sender).

B. Reliability

During multicast transmissions receivers may experience either correlated errors (multiple receivers observe a loss of the same set of packets) or uncorrelated ones. Possible causes of correlated losses in an Internet based satellite transmission can be sender's failure, generic losses (e.g. congestion) on the Reaching path and up-link fading. Uncorrelated losses are due to generic losses on the Receiving path, down-link fading and receivers' failures. File transfer protocols require total reliability. That implies that either all the application data units are received correctly or a failure message is sent to the receiving application. In the multicast case there is a second and stronger type of reliability, we call it complete reliability: it requires total reliability for all receivers. The SRDP protocol assures total reliability but not complete reliability. In other

words, we don't assume that all receivers must receive the file, but the ones that have received it can safely assume its correctness and integrity. The complete reliability goal conflicts with scalability: although the protocol is designed to reach the largest number of receivers, SRDP does not guarantee that all the recipients receive the file.

The mechanisms SRDP protocol uses to attain reliability are layered in three steps, called phases. During first phase the file is transmitted via multicast with a packet-level Forward Error Correction (FEC) scheme [11] to recover losses generated at any point of the paths. FEC schemes are common in communications and particularly useful when some of the receivers do not have a Return path. The second phase uses a distributed Automatic Repeat reQuest (ARQ) mechanism to collect loss information from the receivers that have a Return path. Using this information the Sender sends through the satellite a new sequence of packets to all receivers. This mechanism allows to reduce losses occurred in the first phase. During the third phase, receivers are individually allowed to request and receive packets from the Return path using the traditional Internet TCP (if they can).

C. Scalability

An Internet protocol is called scalable when the cost paid (number of transmitted packets, service time and so on) to get certain results increases less than linearly when the number of receivers increases. In the ideal case, the performances of the protocol do not change when the number of users increases. The classic TV system, for example, scales very well.

A way to implement reliability while preserving a good scalability is using FEC algorithms to add redundancy to the applications' data. Although FEC mechanisms are highly scalable, their massive use reduces the goodput. Moreover these mechanisms are not sufficient to deal with burst errors. For this reason, SRDP introduces distributed ARQ mechanisms to give feedback information to the sender in order to adjust the needed quantity of redundant packets. This operation recovers burst errors. To avoid the message implosion introduced to the ARQ mechanisms, the number of receivers providing feedback is reduced in an exponential back-off way and the identity of the receivers providing feedback is randomly selected using the Scalable Feedback Suppression (SFS) algorithm that we will detail in the next section. The purpose of this algorithm is to try to guarantee the sender the reception of the same amount of feedback, independently from the number of receivers.

After phase 2, the last chance for receivers to correct errors is phase 3. This phase does not involve multicast communications but it is based on a TCP client/server architecture. Since some receivers could be not powerful enough to work with the transmission rate (they are called "crying babies") the protocol must include a mechanism to limit the number of requested packets from each single host or, in other words, to forbid the crying babies the access to the phase 3. Strictly speaking phase 3 does not scale but the protocol can try to reduce the number of requests by excluding the worst receivers from this phase. Particularly, a cut-off decisions are taken to reduce the number of requests from the

receivers, based on the sender's estimate of the number of receivers, the available outgoing bandwidth etc.

D. Congestion control

Congestion collapse in today's Internet is mainly prevented by the congestion control schemes defined by TCP [12]. It is then important to be careful about the congestion-related issues raised by the use of UDP-based protocols. Particularly, UDP multicast applications have the potential to do more congestion-related damages to the Internet than UDP unicast ones. Some general-purpose reliable multicast protocols have been published before the impact of congestion caused by reliable multicast was fully appreciated. These cases were recently deprecated [8].

To analyse the problem of congestion in our topology let us consider it in four separate sub-paths: a) the Reaching path, b) the Satellite link, c) the Receiving path and d) the Return path.

a) The Reaching path. This is the most critical part of the network. Ss is available to a potentially large number of senders some of which are possibly hops away. There is therefore an obvious and classical risk of congestion connected to such transmissions. There are a number of possible solutions: TCP tunnelling, Explicit Congestion Notification, QoS mechanisms, to name a few. By adopting some of those mechanism it is possible to access Ss from a larger set of locations. On the other hand if no such mechanism is adopted, the use of Ss must be restricted to senders which are close to Ss and connected to it by well-provisioned links. Another problem to be considered is the amount of bandwidth to be assigned to each transmission. Since more than one sender can access Ss at the same time, an out-of-band method for coordinating and assigning available bandwidth is needed (for example the Service Scheduler designed in [13]).

b) The Satellite link. Since each Satellite link is used by only one Ss at the time the problem of congestion on such a link is automatically solved by the out-of-band method quoted in the last paragraph.

c) The Receiving path. First of all it must be noted that the present price of a satellite receiver is so affordable that the simplest way to avoid congestion problem is to attach directly a Receiver host to a dedicated Satellite receiver, thus avoiding all congestion problems. If a Satellite receiver is used to serve an entire LAN the risk of congestion is still low as normally the satellite bandwidth and the LAN bandwidth are magnitude orders away. A TCP tunnel from the Sr to each receiver Host is activated in the case that the LAN could be congested or in the case that the Satellite receiver is used to serve distant LANs.

d) The Return path. SRDP uses TCP on the Return channel so no special congestion issue is raised.

In conclusion, because of the peculiarities of the satellite internet we felt that no special congestion related mechanism should be included in SRDP. Congestion controlling methods such the ones quoted above must be used both in the cases of

satellite internets where the sender or the receivers are far from the satellite link and in the use with non-satellite internets.

III. DESCRIPTION OF THE SRDP PROTOCOL

SRDP operations are performed in three different phases. During phase 1, also called "Satellite phase", a file is transmitted through the satellite internet adding a reasonable degree of redundancy. During phase 2, also "Hybrid phase", a set of redundant packets are retransmitted to attempt the recovery of packets lost during phase 1. To this purpose, the sender periodically polls the receivers for feedback about the number of lost packets (Stat Req and Stat Rep messages). During phase 3, also "Terrestrial phase", each receiver is allowed to request all its missing packets by opening a TCP connection to the Sender. Phase 3 is then aimed at eliminating the remaining losses. These three phases are separated in time and identified by SRDP synchronization messages.

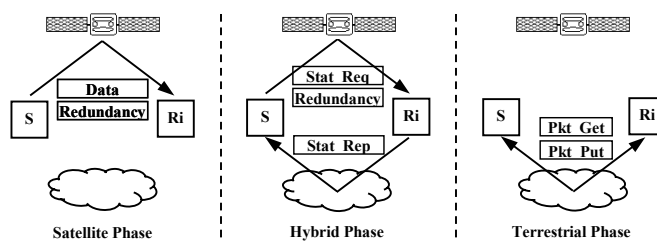


Figure 2. The Three phases.

SRDP has been designed to be used with the satellite networks where an important goal is to reduce the waste of resources. In these environments the time that is dedicated to the transmission is often limited and it constrains the total number of redundant packets that can be added. The typical process of bandwidth purchase can be outlined as follows. A certain amount of data packets K must be transmitted. If the channel bandwidth available is B (packets/s), one has to decide a priori (based on experience and/or on the results of our simulations) the amount of redundancy packets, H , to add to the data. The time slot to be purchased will then at least be $(K+H)/B$ seconds. For this reason, SRDP assumes the parameter H as fixed.

In order to understand how the SRDP protocol uses redundancy, let H be the number of redundant packets that are available to the sender, and K be the number of original packets obtained from the file. The choice of the H parameter influences the performances of the protocol. If H is large, the reliability increases but the goodput of the protocol diminishes. Instead, if H is small, the protocol has a better performance in terms of goodput but the reliability is reduced.

We also define the following relation: $H = H1 + H2$, where $H1$ and $H2$ are the quantities of redundancy that are added in phases 1 and 2 respectively. For instance, if $H2 = 0$ the behaviour of the SRDP protocol is equal to that of the protocols that use unidirectional FEC techniques. Instead, if $H1 = 0$ the protocol relies on an ARQ mechanism only. The results of the simulations, presented in the following sections, aim at helping us in the right choice of these parameters. It

will be shown that it is possible to find a fair trade-off among parameters.

A. Phase 1: Satellite Phase

The purpose of the Satellite phase is to transmit a file from the sender to the receivers through the satellite: the Return path is not used. During the transmission the sender uses Forward Error Correction (FEC) as a powerful technique to prevent loss of information. Packet level FEC adds redundancy packets and transmits them together with the file's ones: when an original packet is lost it can be recovered if enough redundant information is provided. The main drawbacks of such technique are a decrease in transmission efficiency and an increase in the encoding and decoding complexity.

The packets are divided in Transmission Groups (TG) consisting of k original packets and h redundant packets derived from the original ones [14]. The receiver is able to recover the original sequence of data packets when at least k out of the total number of $k+h$ have been received. For each TG, only the number of received packets matters. Losses of the original packets are recovered using redundant packets.

Let $h1$ be the number of derived redundant packets for each TG, and N be the number of TGs. The parameter $h1$ is simply derived as follows: $h1 = H1 / N$. That is, the available redundancy in the phase 1 is equally divided between all the N TGs. Before being transmitted the TGs are normally interleaved to improve the performance in case of burst losses. Moreover the interleaving is transparent to the packet level FEC mechanism.

B. Phase 2: Hybrid Phase

The objective of this phase is to inject the redundancy $H2$ where most needed, that is on the TGs that have suffered more losses during phase 1. Normally this happens when the interleaving and redundancy $h1$ have not been sufficient to correct a burst error. To this purpose, the sender transmits in multicast an SRDP Stat_Req message requesting statistics. The receivers build an SRDP Stat_Rep message containing the number of lost packets for each TG and send the reply opening a TCP connection through the Return path. Each Stat_Req/Reps exchange, between sender and i -th receiver, is independent from the other ones. It is the sender's responsibility to determine when transmitting the Stat_Reqs. Moreover, although these messages functionally belong to phase 2 they may be temporally anticipated in phase 1: soon after transmitting a TG during the Satellite phase, the sender may ask receivers for an acknowledgement with such mechanism.

B.1 Pre-encoding technique

SRDP uses a pre-encoding technique to optimise the transmission. As previously stated, during phase 1, the amount of redundancy packets is given and fixed for all the TGs. Now, the purpose phase 2 is to transmit further information to the receivers. Instead of retransmitting some of the packets already transmitted in the first phase, SRDP transmits a new

sequence of redundant packets into the TG and pre-computes it to save time. For each TG an amount of newly computed $h2,i$ redundant packets is transmitted to all receivers, where:

$$\forall i \in [1,N], \quad h2,i \in [0,k].$$

During phase 2, the maximum number of required packets is k per each TG, because if all packets of the i -th TG have been lost during the phase 1, k new packets are needed. On the other hand, if the i -th TG has been decoded correctly (i.e. as gathered from all SRDP Stat_Rep messages received) other redundant packets are not needed. Before phase 1 the sender codes for each TG k data packets and h redundant packets, $h = h1+k$, transmitting $h1$ packets in phase 1, and $h2,i$ ($0 \leq h2,i \leq k$) packets during phase 2.

The purpose of this mechanism is to reduce the overall amount of bandwidth used for Phase 2: instead of retransmitting the packets missed by each receiver (which are moreover unknown: only the number of missing packets for each receiver and each TG is known) the sender transmits a common excess of redundancy packets which can be exploited by all receivers. Such solution allows also the receivers which do not have a return channel, to decode pending TGs exploiting the new redundancy. It has been shown [11] that it is possible to code and to decode TGs with good throughput also when the redundancy is large.

B.2 The Scalable Feedback Suppression (SFS) algorithm

As stated in the definition of phase 2 until now, the sender should receive, for each TG, acknowledgment information from all the receivers. This ideal condition implies ACK-implosion problems. In order to prevent them, the Scalable Feedback Suppression (SFS) algorithm calculates a rough estimation of RM (number of receivers with a return channel) and uses it to randomly select a group of receivers which transmit ACK-information back to the sender. This group of designated receivers has the task of representing all the receivers by providing information about the reception of each TG that is taken as valid for all receivers.

The SFS algorithm begins with the sender transmitting a multicast Stat_Req message to the receivers. Such message contains also a number ND , called the designation number. Using ND , each receiver decides whether to answer or not. When the response is positive the receiver builds a Stat_Rep message containing the number of lost packets for every TG. Only the receivers that are endowed with a return channel can participate to the algorithm; moreover, all receivers are independent from each other.

Let E be the number of Stat_Rep messages that the sender considers the minimum needed to build a reliable statistic and E be the actual number of Stat_Rep messages that it receives. The SFS algorithm can be summarized as follows.

- 1) The sender sets ND equal to 106;
- 2) The sender transmits a Stat_Req message containing the designation number ND ;
- 3) Each receiver that is endowed with a return channel replies sending (or not) a Stat_Rep message with a designation probability $1/ND$;
- 4) If ($E \geq E$ OR $ND = 1$) the algorithm ends. Otherwise the sender sets $ND = ND / 10$ and goes to the step 2.

Following the SFS algorithm the sender will send a sequence of Stat_Req messages until it receives a number of replies equal to or larger than E, or the parameter ND reaches the value 1. In this latter case the sender will receive a Stat_Rep message from all the receivers endowed with a return channel. In other words, the purpose of the SFS algorithm is to dynamically adjust the value of the parameter ND in order to limit the number of Stat_Rep messages received and to avoid the implosion problems. Such algorithm allows to quickly converge with a $O(\log(RM))$ complexity. It is important to note that during a SRDP transmission the number of receiving members is not supposed to increase because no one can join the SRDP session after it has just started.

B.3 Repair Strategy

After collecting the Stat_Rep messages as adequately filtered by the SFS algorithm, the sender has to identify the sequence of redundant packets that will be transmitted in phase 2. Several algorithms can be adopted, we will propose a simple algorithm in the following. In order to determine the number of redundant packets to transmit, the following definitions are given:

- $h1$ is the number of redundant packets transmitted in phase 1 for each TG;
- $h2[i]$ is the counter of redundant packets transmitted in phase 2 for the i -th TG;
- Packets_Count is the counter of redundant packets transmitted for all the TGs;
- Statq[i] is the number of lost packets for the i -th TG ($1 \leq q \leq E$) as derived from each of the E Stat_Rep messages;
- $\max_vector[i] = \text{Max}_q(\text{Statq}[i]); 1 \leq q \leq E; \forall i \in [1, N];$

The algorithm goes as follows:

- 1) $h2[i]$ is set to 0, $\forall i \in [1, N]$.
- 2) Packets_Count is set to 0.
- 3) TGs are scanned cyclically starting from the first ($i = 1$).
- 4) At each step a redundant packet is transmitted if ($h2[i] < \max_vector[i]$ AND $\max_vector[i] > h1$). If a redundant packet is transmitted, Packets_Count and $h2[i]$ are increased by 1.
- 5) Then the next TG is considered ($i = i + 1$) and the above operation is performed until $h2[i]$ reaches $\max_vector[i]$ or Packets_Count reaches H2.
- 6) If the algorithm stops before Packets_Count reaches H2 then the sender consumes the remaining redundancy ($H2 - \text{Packets_Count}$) cyclically sending a new redundant packet for each TG.

The sender could use other more sophisticated algorithms to exploit the similitude between Stat_Rep vectors and to optimise the number of receivers that can benefit from the new

sequence of redundant packets. But it must be kept in mind that such algorithms increase further the load on the sender.

C. Phase 3: Terrestrial Phase

If a receiver has not correctly received all the TGs, after the satellite phase and the hybrid phase, the only chance for it to receive the missing packets is to use its return channel. During this phase receivers request all missing packets to the sender and receive them in a client/server manner using the Pkt_Get and Pkt_Put messages.

The protocol must take into account two important effects that could reduce the performance of the sender in an unacceptable way. First of all some of the receivers might ask for a large number of packets, for example if the receiver's hardware is inadequate. SRDP is inherently a satellite protocol and this phase must be confined to the retransmission of just a little number of packets to a reduced number of members of the original receivers' community. The SRDP sender, which has the session manager role in the protocol reduces then the group's membership eliminating the worst of such receivers. That is, the sender limits the maximum number of packets that can be demanded (Max_Pkt). Beyond such threshold the receiver cannot benefit from this phase. A second effect that may have an impact on the sender's performance is the race between receivers, that immediately after the phase 2, compete to obtain the sender's resources. To reduce this effect the sender uses the spread factor parameter (SF). The sender manages the access defining a temporal value SF, proportional to the expected number of connections that are estimated by the use of the SFS algorithm. Such value is broadcasted together with Max_Pkt and Pkt_Conn (see below) inside the SRDP End Message (see table 1). A receiver must wait, before trying to connect the sender, for a time randomly chosen into the range $[0, SF]$. Moreover, a single connection must not request more than a fixed number of packets per connection (Pkt_Conn) in order to limit the connection time for each receiver. Each receiver must perform the following steps when accessing the sender:

- 1) if the number of missing packets is greater than Max_Pkt, then the algorithm stops and the reception fails;
- 2) waits randomly for a time chosen in the interval $[0, SF]$, then opens the connection;
- 3) sends a request for a maximum of Pkt_Conn packets;
- 4) if more packets are needed, iterates step 2, otherwise ends.

The sender is the session manager of all the phases. Nonetheless, communication in this phase is a receiver-initiated one because each receiver is the initiator of the connection. A receiver may even decide to connect the sender hours or days after the end of the satellite transmission; a sender may keep the file for several days: in any case, SRDP does not guarantee the receiver that the sender is able to correctly answer the request.

D. Message Types

Table 1 lists the message types currently defined in SRDP. The Begin message is the first message transmitted by the

sender and starts phase 1. This phase is normally concluded by the Continue message that separates phase 1 from phase 2 (though, as before stated, nothing prevents Stat_Req messages to be sent before the separation occurs). The End message is used to closed phase 2. An End message also starts phase 3, but all the following SRDP messages, from the End on, are transmitted on Return paths only. Begin, Continue, End messages together with the Abort are called synchronization messages and are used to delimit the SRDP phases. They play therefore the critical task to synchronize the transmission: if the Begin message is lost, for instance, the entire transmission is lost. To minimize such probability, a synchronization sequence that replicates these messages is used. Data messages carry the file's packets during all the phases, while Redundancy messages are used to transport redundant packets.

TABLE I
SRDP MESSAGE TYPES.

Message Types	Description	Main fields
Begin	Packet to begin the transmission	F (file length), S (packet length), k, h1
Continue	Packet to pass from phase 1 to phase 2	Nothing
End	Packet to end the transmission	SF, Pkt_Conn, Max_Pkt
Abort	Packet to abort the transmission	Error_code
Data	Data packets	Pkt_Identifier
Redundancy	Redundant packets	Pkt_Identifier
Stat_Req	Packet to request statistic information	N_D (designation number)
Stat_Rep	Packet to reply with the statistic information	Stat _i (number of lost packets for each TG)
Pkt_Get	To request Data in phase 3	list of the missing packets' identifiers
Pkt_Put	To send Data in phase 3	list of the missing packets

After the phase 2 End message the receiver is enabled to transmit a Pkt_Get message opening a TCP connection to the sender. The Pkt_Get contains a list that specifies the packets that the receiver has not received. The Sender replies to the request transmitting a Pkt_Put message containing the required Data.

TABLE II
SRDP PARAMETERS.

Phase	SRDP Parameters	
1	H_1	Redundancy used in phase 1
	k	Number of data packet for each TG
	Interleaving	Number of interleaved TGs
2	H_2	Redundancy used in phase 2
	E	Number of Stat_Rep needed to build a reliable statistic
	N_D	Initial value for N_D
3	SF	Spread Factor parameter
	Max_Pkt	Maximum number of allowed packet in phase 3
	Pkt_Conn	Number of allowed packets for single connection
1,2,3	S	Data packets size in octets
	Sync_Length	Length of the synchronization sequences

IV. SIMULATION RESULTS

The main concern in our simulations is to identify the percentage of receivers that have correctly received the file after phase 2 (hereinafter referred to as R_{OK2}) for a given value of H. We have neglected phase 3 because its reliability is guaranteed by the use of TCP connections. R_{OK2} reflects the reliability of the protocol: the more R_{OK2} is approaching the 100% value, the more SRDP reliability becomes complete.

A. Simulation Model

Our simulations are based on the ns2 -Network Simulator [15]. The simulated network topology is the one in figure 1, where an SRDP sender is connected to R receivers by means of a satellite network. We have assumed the use of a GEO satellite and we have assigned to the satellite up-link and down-links a bandwidth of 2 Mbps and 125 ms of delay. We have simulated the transfer of a 100 MB length file to R receivers, of which M have a return channel. We have assumed during all the simulations a fixed length for the IP-packets (1500 bytes). The number of the data packets k in each TG is equal to 100 and a Vandermonde matrix has been used to generate the redundant packets from the original ones.

In the following, two different sets of simulations are presented to analyse the behaviour of the protocol in the condition of random and burst losses respectively. To simulate packet losses we have used for each of the two sets R+1 error distributions:

- R distributions have been used to simulate uncorrelated errors, which occur in the Receiving path and/or on the downlink;
- one distribution has been used to simulate correlated errors, which occur in the Reaching path and/or on the uplink.

Each run has been repeated 1000 times to allow for more accurate results. The Stat_Rep messages carry the information needed to manage phase 2. The SFS algorithm and the number of receivers endowed with a Return path M are the elements that determine the number of generated Stat_Rep messages. In order to control such number, we have fixed N_D equal to 1, and we have chosen to act only on the parameter M.

B. Random Losses

In this section all the error distributions are characterized by a Bit Error Rate b that is assumed to be constant over the time of a transmission for all links but the Return channel. Admittedly this is a simplification (in general b on the up-link is different from b on the down-link) needed to control the complexity of the simulation work. Moreover, we assume that the R+1 distributions are mutually independent. This implies an upper bound on the packet loss probability. A whole packet is assumed to be lost when at least one bit is corrupted. The packet loss probability P is hence dependent on the packet size S in octets, and can be calculated as $P = 1 - (1-b)^{8S}$ (e.g. a b of 10⁻⁶ gives a P, packet loss probability of about 0,012).

Because in this section our attention is devoted to study the protocol behaviours in the condition of random losses in the following simulations we have chosen an interleaving factor

equal to 1. The interleaving spreads the burst losses over more transmission groups partially transforming the burst errors in random losses. Therefore it is not useful to interleave for the purposes of this section.

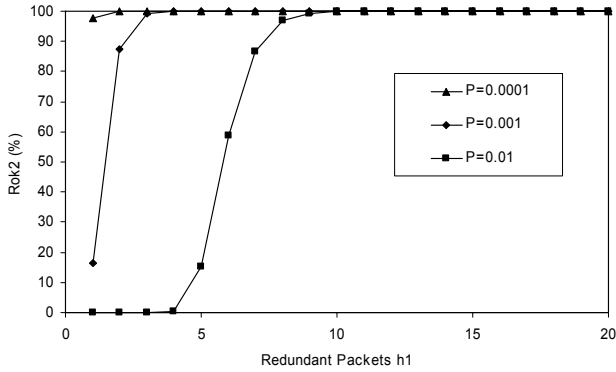


Figure 3. R_{OK2} as a function of $h1$ and P . $H2=0$, $R = 100$ and $M = 50\%$ of R .

The simulation whose results are shown in figure 3 investigates R_{OK2} as a function of the quantity of redundancy packets per TG, $h1$, and of the packet error rate P . R_{OK2} increases as $h1$ increases: for instance, with P is equal to 0.01, R_{OK2} is equal to zero when the number of redundant packets is lower than 4, while it reaches 100% (“complete reliability”, see section II B) for $h1 \geq 10$.

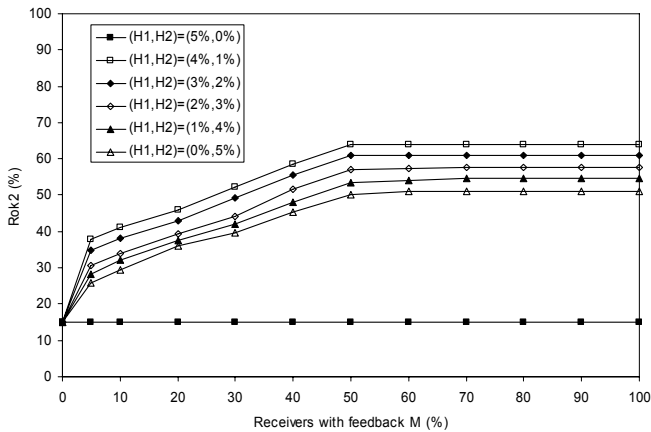


Figure 4. R_{OK2} as a function of M . $H=5\%$, $R = 100$, $P = 0.01$.

The purpose of the simulation whose results are shown in figure 4 has been to examine the role that the percentage of receivers endowed with a Return path M has on the reliability of the protocol. When the protocol uses only phase 1, corresponding to the curve (5%,0%) in the figure, R_{OK2} has a constant value (the packet level FEC used in phase 1 is independent from the feedback availability). Instead, the other curves show that R_{OK2} increases with the percentage of receivers M . It can be observed that, if M exceeds the 50%, the curves are levelling off near the maximum value. This behaviour suggests that to benefit from phase 2 it is not necessary that return channels are available for more than 50% of the receivers (and this value has been then used in the following simulations).

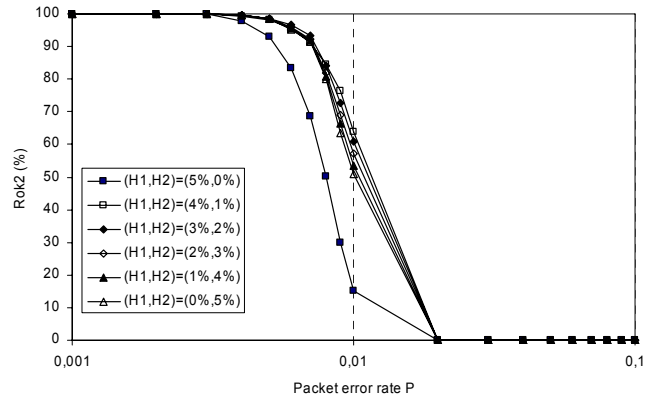


Figure 5 R_{OK2} as a function of P . $H=5\%$, $R = 100$, $M = 50\%$ of R .

In figure 5 we analyse the effects that different packets error rates P have on the performance of our protocol. The rain causes an attenuation between 8dB and 10dB for frequencies between 12GHz and 14GHz, increasing the bit error probability b by a factor between 10 and 200 [1]. Figure 5 shows R_{OK2} as a function of the packet error rate P for different values of the couple $(H1, H2)$: 6 curves are presented for each possible couple of entire percentage values obtained by a fixed value of H ($H=5\%$). For all figures, the values of the parameters H , $H1$, $H2$ are expressed in terms of the percentage of the total number of data packets. For example, the couple (5%,0%) means that the protocol is only using phase 1, on the contrary, when $(H1, H2)$ is equal to (0%,5%) SRDP is using all the redundancy in phase 2.

First of all, figure 5 easily shows that for each curve R_{OK2} decreases when the packets error rate P increases. It can be seen that all curves have the same threshold behaviour and that the chosen couples of values $(H1, H2)$ are the critical ones (as could be derived from figure 3). Comparing the curves (5%,0%) and (4%,1%) in the figure, one can observe that the latter one has a better behaviour. That is, even if only 50 percent of the receivers has a return channel, all the receivers benefit from the phase 2 redundancy.

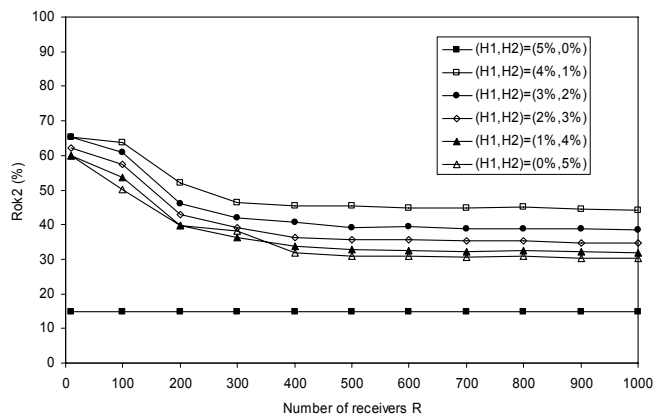


Figure 6. R_{OK2} as a function of R . $H=5\%$, $M = 50\%$, $P = 0.01$.

Figure 6 shows R_{OK2} as the function of the number of the receivers R for different values of the couple $(H1, H2)$. Like

the previous figure, the curve corresponding to (5%,0%) assumes a constant (and worst) value. Instead, the other curves reach the maximum value about $R = 100$. From 400 receivers on all curves look like horizontal lines: R_{OK2} reaches a stable value. After 100 receivers there is a lightweight degradation of the performance but however significantly better than unidirectional FEC (only phase 1). Again, even in such hard conditions, the advantages coming from a careful choice of the phase 2 redundancy amount can be appreciated.

C. Burst Losses

In this section we examine the protocol behaviour when losses are bursty. In particular, we assume that packet loss distributions are described by a two state Markov chain [16] which characterizes the transmission as a ON/OFF behaviour: the ON state was associated to error free periods, while the OFF state was associated with error periods. In this section, again, as defined in 4.2, we use $R+1$ error distributions: one for correlated errors and the others for uncorrelated ones.

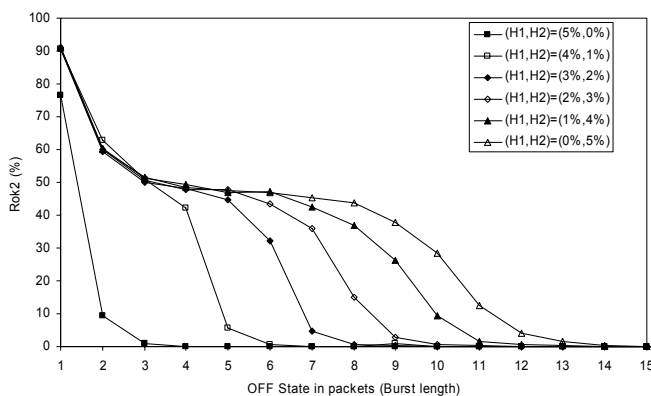


Figure 7. R_{OK2} as a function of burst length. $H=5\%$, $M = 50\%$, $ON_State = 10000$ packets.

Figure 7 shows R_{OK2} as a function of the OFF state duration for different values of the couple $(H1, H2)$. The length of the ON and OFF states is expressed in terms of number of packets. It can be observed from figure 7 that phase 2 brings improvements of the behaviour of SRDP also in scenarios where the losses are bursty. The curve (5%,0%) quickly reaches the value 0: one correlated burst error longer than $h1$, on a TG, is enough to compromise the reception of all receivers. By increasing the redundancy in phase 2 the protocol's behaviour improves.

Uniformly spreading the redundancy on the TGs, curve (5%,0%) in the figure, is not sufficient to recover from burst errors. Instead, phase 2 aims at injecting new redundant packets where they are more needed. Another important difference is that phase 1 uses the redundancy $H1$ in advance, knowing nothing about possible errors, while phase 2 uses all the available redundancy $H2$ after the burst errors have damaged the TGs and knowledge about the damages is obtained (by the *Stat_Rep* messages).

Also in this simulation we have chosen an interleaving factor of 1. The goal of the simulation is to examine the effect of a bursty error distribution has on R_{OK2} comparing the different values of the couples $(H1, H2)$. Additional

interleaving would clearly improve the protocol behaviour for both phase 1 and 2, but it would not significantly change the relative gap between the phase 1-only curve (5%,0%) and the others. The overall effect of an interleaving factor > 1 would mainly be a shift to right of all curves.

V. CONCLUSIONS

We have presented a reliable multicast data distribution protocol particularly designed for the use with satellites. The protocol has several interesting properties. Namely it is IP-based, it does not require a return channel for the receivers and it has a good scalability. The protocol operates in three phases. The first one is aimed at all receivers, assuming no Return channel at all. The second one tries to exploit the information obtained by the receivers that have a Return channel to benefit all receivers and the third one is dedicated only to the receivers that have a Return channel, thus guaranteeing total reliability for the involved receivers. Phase 2 and 3 require special attention to avoid excesses of traffic directed to the Sender, that is to preserve protocol's scalability (e.g. the SFS algorithm in phase 2 and cut-off thresholds in phase 3).

The simulations show that the use of the ARQ mechanisms of phase 2 largely improves SRDP performances, even if not all receivers are endowed with a return channel. They also show that a fair balancing between $H1$ and $H2$ is needed. Adding $h1$ redundant packets for each TG is a suggested conservative choice, as SRDP cannot rely on the ARQ mechanisms alone for the transmission's success (e.g. there could be a pathological case with little receivers sending misleading feedback information).

All the figures shown can be of help in correctly configuring the parameters of SRDP. The choice of the parameters k and h affects the dynamics of the management of the resources. The administrator of the network should set these parameters in order to avoid the errors and to minimize the number of redundant packets sent via satellite.

ACKNOWLEDGEMENTS

We would like to acknowledge Clio Srl [17] for the help provided during the real-world test of SRDP. On the November 2003 the IANA [18] assigned to SRDP the Registered Port n° 3942.

REFERENCES

- [1] G. Maral and M. Bousquet, "Satellite Communication Systems", 4nd Edition ed. Toulouse: John Wiley & Sons, 2001.
- [2] K. Obraczka, "Multicast Transport Protocols: A Survey and Taxonomy", IEEE Communications Magazine, 36(1), 94 - 102, (1998).
- [3] B.N. Levine and J. J. Garcia-Luna-Aceves, "A Comparison of Reliable Multicast Protocols", ACM Multimedia Systems Journal, 6(5), 334-348, 1998.
- [4] Akkor G., Hadjithedosiou M., Baras J.S., "Transport protocols in multicast via satellite", Journal of Satellite Communications and Networking 22:611-627, 2004.
- [5] B. N. Levine and J. J. Garcia-Luna-Aceves, "A Comparison of known Classes of Reliable Multicast Protocols", in IEEE/ICNP'96, 112-121, 1996.

- [6] D. Towsley, J. Kurose, and S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *IEEE/JSAC*, 15(3), 398-406, 1997.
- [7] M. W. Koyabe, G. Fairhurst, "Reliable Multicast via Satellite: A Comparison Survey and Taxonomy", *International Journal of Satellite Communications (IJSC)*, Vol:24(1), 21-26, 2001.
- [8] RFC2357 A. Mankin, A. Romanow, S. Bradner, V. Paxson. "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols". June 1998.
- [9] K. Sollins, "The TFTP Protocol (Revision 2)", IETF RFC1350, July 1992.
- [10] ETSI EN 301 790, "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems", V1.2.2, European Standard (Telecommunications series), December 2000.
- [11] L. Rizzo, "Effective erasure codes for reliable computer communication protocols". *ACM SIGCOMM Computer Communication Review*, vol.27, no.2, p. 24-36, April 1997.
- [12] V. Jacobson, 'Congestion Avoidance and Control', in *SIGCOMM '88: Communications Architectures and Protocols*, Stanford, USA, 18, 314 - 329, 1988.
- [13] European Space Agency (ESA) MODUS project, <http://telecom.estec.esa.nl/artes/artes3/fileincludes/projects/162ssi/ssi.cfm>.
- [14] J. Nonnenmacher, E. W. Biersack, D Towsley "Parity-Based Loss Recovery For Reliable Multicast Transmission". *IEEE/ACM Transaction on Networking*, 6(4): 349-361, August 1998.
- [15] UCB/LBNL/VINT. Network Simulator – ns2, <http://wwwmash.cs.berkeley.edu/ns/>.
- [16] L. Kleinrock, "Theory, Volume 1, Queueing Systems", John Wiley & Sons, 1975.
- [17] Clio srl. Available: <http://www.clio.it>
- [18] IANA, Internet Assigned Numbers Authority, <http://www.iana.org>.



Francesco Tommasi graduated from University of Pisa, Pisa, Italy, in electronic engineering in 1984. In the same year he joined CSELT, Turin, Italy, where he became involved in image processing. In 1988 he joined the Istituto per la Ricerca Scientifica e Tecnologica, Trento, Italy, where he made researches in image motion analysis. He has been head of software development at Cigraph, Venice, Italy. He is now associate professor at the Engineering Faculty of the University of Lecce, Italy. His present research interests include networking and the applications of satellite to distance learning.



Simone Molendini received the M.S. and Ph.D. degrees in computer engineering from the University of Lecce, Lecce, Italy, in 1999 and 2004, respectively. Currently, he is a Research Assistant in the Department of Innovation Engineering, University of Lecce. His currently research interests are in Internet signaling, mobility and satellite applications. Active in several IETF working groups, he is co-author of the RFC2961 - "RSVP Refresh Overhead Reduction Extensions", year 2001.



Andrea Tricco received the M.S. and Ph.D. degrees in computer engineering from the University of Lecce, Lecce, Italy, in 2001 and 2005, respectively. Currently, he is a Research Assistant in the Department of Innovation Engineering, University of Lecce. His currently research interests are in Internet signaling, mobility and satellite applications.