Efficient Workspaces through Semantic Reliability

Jeremiah Scholl, Stefan Elf[†], Peter Parnes
Luleå University of Technology
Centre for Distance-spanning Technology
Department of Computer Science
SE-971 87 Luleå, Sweden
{jeremiah,self,peppar}@cdt.luth.se

Abstract. Transport level protocols typically provide "best-effort" data delivery where no attempts are made to recover lost packets, or "reliable" data delivery where techniques are used to ensure that all data sent out eventually reaches the receiver(s). However, it has been suggested that some applications will perform better when using dynamically configurable reliability based on application level semantics. A general framework has been created with the intent of delivering this type of semantic reliability to a wide variety of applications, but to date little research has been done to demonstrate how any applications can benefit from the use of such a framework. This paper addresses this problem in that it discusses semantic reliability in the context of collaborative workspaces.

1 Introduction

IP-Multicast provides efficient one to many best-effort delivery. However, a wide variety of reliable multicast protocols have been introduced over the last several years because some applications perform poorly when they suffer from packet loss. While designed around separate application requirements and network constraints these protocols have in common that they are engineered to guarantee that each packet transported eventually reaches its designated receiver. A comprehensive study of multicast transport protocols including the reliable transport protocols mentioned above is given by Obraczka [1].

Alternatively, it has been suggested that some applications could benefit from more relaxed reliability rules where only a subset of packets lost during transmission would be recovered. Ideally this would save bandwidth by reducing the amount of repairs sent on the network that recover data, which is no longer useful to the receiver. This idea has not been explored as extensively as "absolute" reliability but several ideas have been presented for solving specific application scenarios by using the technique. In addition, a general framework has recently been described which aims to make relaxed reliability based on application semantics available to a wide variety of applications (see section 2).

Little work has been done to date in order to demonstrate how this type of framework can enhance specific applications. This is in practice a potentially time consuming process because the semantic rules that determine when to send repairs must be worked out separately for each application. Despite this drawback, the potential benefits in terms of efficient use of bandwidth and stability when facing packet losses are compelling. In particular this technique could be very useful when deploying distributed multimedia applications on wireless telecommunication networks, as these links tend to be lossy and have limited bandwidth.

This paper addresses this issue by examining semantic reliability in the context of IP-Multicast collaborative workspaces, which are complex applications for enabling group communication. Focus on a specific implementation is required before the necessary Application Level Framing (ALF) [2] can be performed due to the fact that collaborative workspaces are inherently different from one another. In order to obtain this focus while maintaining a real-world view, an existing partnership between Luleå University of Technology and Marratech AB has been utilized. Thus, the analysis in this paper focuses on features contained in Marratech's commercially available collaborative workspace implementation, Marratech Pro [3]. This will help this paper serve developers of similar applications who are interested in an analysis based on a highly deployable solution because Marratech Pro is based on the well known and the widely used protocols RTP [4] and SRM [5] as well the standardized video codec H.261 [6].

2 Related Work

Several approaches exist for using application or data semantics in order to classify or handle information. Mauve and Hilt [7] have presented an Application Programming Interface (API) for reliable multicast

[†] Stefan Elf is also with Ericsson AB, SE-931 87 Skellefteå, Sweden. Views expressed in this paper are his own and not necessarily shared by his employer.

that allows the sender to control the use of forward error correction (FEC) in a subscription-oriented interface, which allows a receiver to proclaim its interest, or lack thereof, in lost packets. Several other authors have also explored having the sender include application level semantic information in packets. For example, Pereira et al. [8] describes a method where a buffer occupancy monitor is combined with an obsoleteness marker in packet headers in order to purge redundant messages from buffers or to immediately drop incoming packets. This allows a buffer to make space for useful, non-obsolete packets but has also the drawback that the sender must mark all packets accordingly.

Approaches based on buffering and FEC can however be viewed as problematic when implementing them in collaborative workspaces. This is because buffering introduces delay, which can harm interactivity between users, and the abundant use of FEC introduces overhead into the data stream, which harms scalability by reducing the number of effective senders that can participate in a session. The solutions described in this paper thus focus on using application semantics along with recovery-based reliability, as this is low-weight compared to FEC without introducing the interactivity problems associated with buffering. Other approaches for semantic reliability include different packet header markers for scenarios other than obsoleteness. These include message deadline as in Rodrigues et al. [9] and the similar lifetime semantic in Baldoni et al. [10].

Elf and Parnes [11] have presented a robust framework based on several semantics, which suits our needs for classifying and deploying semantic information to recovery based feedback. The approach described focuses specifically on how to classify data based on different application scenarios, which are "tunneled" down to the underlying reliable multicast protocol in use by the application. This enables the framework to be used with a wide variety of protocols and applications.

The framework is also flexible in that the sender can distribute configuration information in data packets (the semantic would alter the behavior of all receivers) and each receiver can independently configure their own behavior based on specific local application properties or network conditions. Therefore, the semantics used are associated with application behavior rather than QoS levels. Because each host can be configured independently of all other hosts it offers the advantage that a complex application with several media (for example a collaborative workspace) can be dynamically configured to perform in an optimal way.

The framework describes several semantics in which the traditional view of reliable delivery is considered a special semantic case, referred to as "absolute". Similarly the opposing view in which no attempts are made to retransmit lost packets is labeled as "best-effort". In addition, several other semantics are available, which can be used to benefit an application in situations where neither the "absolute" nor "best-effort" semantic seems to provide optimal performance. Table 1 provides a list of possible semantics, which may be useful in different scenarios. The semantics listed in the table are those that were identified in the reference publication. However, the framework was intended to be expandable so other semantics may be utilized when necessary.

Table 1. Example semantics and properties (from[11]).

Semantic	Meaning	Property	Configured by
ABSOLUTE	All packets received	None	Sender/receiver
OBSOLESCENCE	Packets made obsolete	Seq.no.	Sender
REDUNDANCY	Redundant packet	Seq.no.	Sender
LIFETIME	Packet life time	Time	Sender
EXPENDABLE	No crucial information	Boolean	Sender
DEADLINE	Latest arrival time	Time	Receiver
CATCH-UP	Disregard packets	Seq.no.	Receiver
BEST-EFFORT	Fully relaxed	None	Receiver

Each semantic is intended to relate to one of several parameters that together describe the desired behavior of the transport protocols, with respect to each packet. Certain semantics may exclude others, such as "absolute" obviously is mutually exclusive to "best-effort". Some semantics relate to the sender, such as "lifetime", while others relate to the receiver, the example being "deadline".

The next section discusses Marratech Pro in the context of this framework. Several ideas are presented, which attempt to save bandwidth, reduce latency or provide stability through the use of semantics other than the traditional "absolute" and "best-effort". This is intended both to provide a view of how more efficient collaborative workspaces can be implemented in the future and help the reader obtain a clearer picture of how relaxed reliability can be utilized to create more efficient applications in general.

3 Semantic Reliability in Collaborative Workspace Tools

Marratech Pro is a commercially available collaborative workspace, which offers common and intuitive tools for enabling group communication. These tools are interactive audio, video, chat, whiteboard, and a shared web-browser. Currently, with only one exception, Marratech Pro classifies data in the traditional manner treating audio and video as "best-effort" while the browser, chat, and whiteboard are viewed as needing "absolute" reliability, which they obtain from the use of an SRM-like [5] mechanism.

The one example of relaxed reliability based on application semantics that has already been implemented into the application occurs when one or more receivers require information about whiteboard pages after reestablishing contact or joining a session for the first time. In some situations these updates end up being sent to the entire group because collaborative workspaces need to run in environments with Network Address Translators, firewalls and reflectors, which will result in all traffic being sent via multicast [12]. Thus, this will cause some participants in the session to inevitably receive packets describing whiteboard pages for which they have already obtained an up-to-date copy.

When this happens, hosts that are not interested in the update ignore losses and do not generate NAKs for missing packets related to the update. It is important to note that this feature has existed in Marratech Pro for several years, which illustrates that the idea of using relaxed reliability is not new by any means. However, it is equally important to note that this behavior can be obtained by utilizing the "redundancy" semantic associated with the framework. This demonstrates that the framework is flexible enough to incorporate old as well as new ideas, which can be valuable when adding it to existing applications.

The remainder of this section examines additional ideas for how to provide optimized reliability based on application semantics. More specifically, several features of the whiteboard, shared web-browser and interactive video streams provided by Marratech Pro are discussed and concrete examples of how the framework can be applied to enhance these tools are given.

3.1 Minimizing transmission of Obsolete Data from Shared Web-Browsers and Whiteboards

One common tool provided by collaborative workspaces is a shared web-browser, which is used in order to provide an effective method of distributing presentation slides. This is accomplished by allowing a user to control the shared browser so that each page it loads is automatically multicast out to the group. The presenter can then load a slide into each user's browser by simply clicking on a hyperlink or typing a URL.

At times congestion or other factors may result in the presenter moving on to a new slide before all the participants have finished receiving the current page. For example, this will occur if the presenter follows several hyperlinks in order to reach the page that she intends to discuss with the group. In this situation continuing to repair packets for the old pages may be unnecessary as this data can be viewed as obsolete from a presentation standpoint. By applying the "obsolescence" semantic from the framework these useless repairs can be avoided, which will free up bandwidth and allow the current slide to more quickly reach the group. Table 2 describes two semantic rules that apply this idea.

Table 2. Semantics for reducing the retransmission of obsolete data in a shared web-browser.

Semantic	Use
OBSOLESCENCE	The **sender** will stop repairing packets for old pages as soon as it begins to send a new page.
CATCH-UP	When a new page begins to arrive a **receiver** will stop sending nack's for packets representing old pages.

Both of these rules can be used at different times in order to benefit the application in different scenarios. For example, use of the first rule will limit all data recovery to packets for the current slide only. This will keep bad receivers from clogging the network with increasing amounts of control data as each new slide is loaded.

On the other hand, during some deployments the users may prefer to allow each receiver to act independently and would then employ the second rule only. This would allow a particular host to free up bandwidth for the current slide without affecting other hosts. This situation would also allow a host to have the option of replaying the presentation later on by attempting to recover the previous slide if for example a reference to this slide was made by the user clicking the back button in her browser.

A similar optimization can be made to whiteboards, which are included in collaborative workspaces in order to give the participants the ability to create and dynamically manipulate shared slides. Whiteboards are generally object-based, allowing the users to create and delete pages that can contain many objects. The objects themselves can be created, modified, or deleted at any time, and can consist of several different forms which may include hand drawn curves, simple geometric shapes, text, and images.

Because pages and objects can be deleted at any time it is possible for recovery packets to be sent out, which describe modifications for an object or page that no longer exists. For example, this can occur if a page is deleted while packets describing modifications to objects on the page are still en route to members of the session. A host could then avoid resending data referring to objects or pages that it identifies as deleted by using the "obsolescence" semantic in a similar way to that described for shared web-browsers.

However, unlike with a shared browser it may not be obvious to every host that the information being requested is in fact obsolete. Therefore, a host should send out an "obsolescence" message to the group when it would normally send a repair. This will keep hosts from continuing to send NAKs when their SRM back off timer has expired. In addition, this will be more efficient than sending a repair because these messages can be quite small compared to data packets and one message will be able to declare a large number of packets as obsolete.

3.2 Whiteboard text-node semantic reliability using local dictionary

Another idea for applying semantic reliability to a whiteboard application occurs when text is written into a page. Often whiteboards (including Marratech Pro) implement this feature by multicasting each letter individually to the users while the user is typing the text. While this does introduce a fair amount of overhead in terms of packet headers it allows the participants in the session to see each word as it is being added to the whiteboard. Because letters are sent individually an interesting possibility for packet recovery based on application semantics becomes available, which would not be available if each sentence were sent individually as is usually the case with chat tools.

This becomes apparent because at times it may be possible to accurately guess a word that has been only partially received by using a local dictionary as well as the semantics of the adjacent words. In this situation it will not be necessary for missing letters to be repaired because enough information has already been received to reconstruct the complete word at the receiver. By avoiding repairs, in addition to saving bandwidth, another benefit is that the amount of time it takes for a receiver to gain a view of the word transmitted will be decreased. This could be especially useful when using a whiteboard on a wireless link where a fair amount of uncorrelated losses are often experienced.

Of course a word that has been guessed by a local dictionary may or may not be correct. This fact must be appropriately displayed for the user for example by changing some display attribute. In addition, the local host must not use a guessed word when sending repairs to other hosts until the exact word has been confirmed.

Two separate scenarios for avoiding the retransmission of lost letters are illustrated in Figure 1.

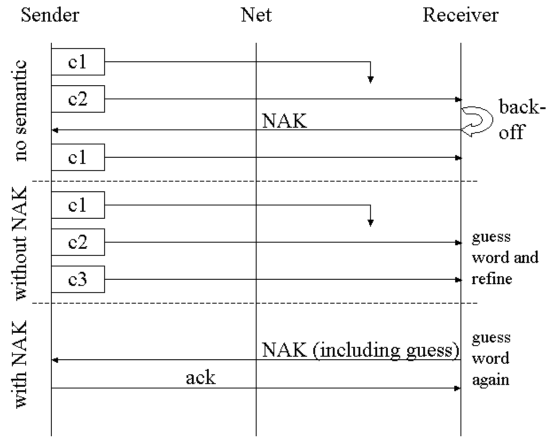


Figure 1. The principle for text-node semantics using a local dictionary.

In the *without NAK* scenario the receiving host does not send NAKs for missing letters after a guess. This will be quite effective when the word guess is correct. However, even if the dictionary cannot give a correct guess it will still be effective if enough information was made available from the received letters for the human user to deduce the correct the word, in which case a NAK would still be unnecessary.

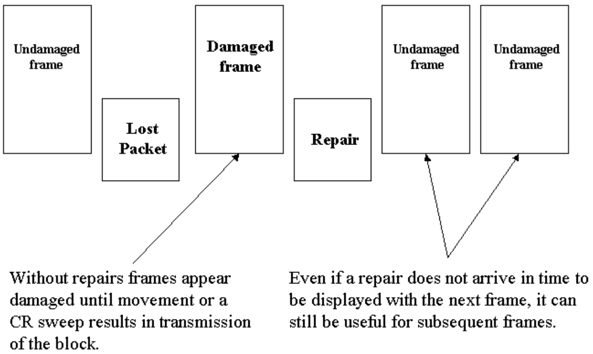
In the *with NAK* scenario, a NAK is sent in order to verify that the dictionary guess is correct. The current guess for the word is included in the NAK in order to accomplish this. Then, after receiving a NAK with a guess a host can then simply respond whether the guess is correct or not. When the guess is correct then the receiver can choose not to NAK any additional losses for the word as it already has enough information to display the correct word to the user. When the guess is not correct, the entire word can be re-sent in one packet to the receiver.

3.3 Efficient Reduction of Error-Propagation in Multicast Interactive Video

While the ideas presented above deal with the more natural idea of using application semantics in order to make traditionally "reliable" media perform more efficiently it is also interesting to discuss semantics in the context of media that are normally treated as "best-effort" by developers. Rather than making these media run more efficiently, semantics instead open up the possibility to introduce error control techniques that are normally thought of as grossly inefficient. Such is the case with real-time video streams, which are commonly provided by collaborative workspaces in order to give the participants a visual presence of each other and their surroundings.

One of the most challenging aspects in obtaining high quality interactive video is that many commonly used error control techniques for non-interactive video are unavailable because they introduce latency in frame playout times. For example, without the benefit of buffers, repairs typically arrive too late to be displayed in the intended frame. An alternative is to use FEC, but as stated in section 2 this can consume a considerable amount of bandwidth, reducing the number of effective senders that can participate in a multi-sender session.

However, repairs can be used with interactive video in order to reduce error propagation even if they do not arrive in time to be displayed with the current frame. This technique has been implemented for unicast MPEG traffic and has shown to be low-weight and effective [13]. With a bit of additional design work this concept can be used in order to reduce error propagation for multicast H.261 traffic as modeled in Figure 2.



Without repairs frames appear damaged until movement or a CR sweep results in transmission of the block.

Even if a repair does not arrive in time to be displayed with the next frame, it can still be useful for subsequent frames.

Figure 2. Reducing error propagation in video.

However, unlike with unicast traffic, NAK suppression will need to be utilized in order to keep the necessary loss-feedback for repairs from impacting scalability. This introduces a problem because today's most deployable NAK-suppression technique is based on random timers [5], which introduce delay in reporting losses back to the sender. The delay could result in repairs and/or NAKs sent out, corresponding to blocks that have since been updated due to new movement or a periodic conditional replenishment sweep. However, the number of unnecessary repairs can be kept to a minimum if instead of resending every lost packet, additional semantic rules are used as discussed in Table 3.

Table 3. Semantics for enabling efficient error recovery in interactive H.261 video.

Semantic	Use
OBSOLESCENCE	Before sending a repair the **sender** will check to see if the block described in the repair has been updated after the original packet was sent. If so, the repair is "obsolete" and should not be sent.
DEADLINE	In some situations a **receiver** can detect which block has been lost. A receiver will then avoid sending this NAK if it receives a new packet describing the block before its random timer expires.

These semantic rules will reduce the number of repairs sent which contain obsolete data to a rather small number, opening up an error-control technique that would otherwise be too inefficient to be generally useful.

4 Evaluation

Table 4 represents the bandwidth savings in bytes, not including control data, and improved latency in seconds when several of the tasks from section 4 are performed. Column 1: The task being performed; 2: The number of bytes saved not including control data; 3: The improved latency in reception by the user when considering the bandwidth savings from column 3.

Table 4. Savings in bandwidth (bytes) and delay (seconds) experienced by a receiver with a 256 Kb/s link experiencing 10% loss. See text.

Task	Bandwidth	Delay
Slide distribution	200000	6.5
Dictionary (*without NAK*)	2800	70
Dictionary (*with NAK*)	1700	40
H.261 repair	- 1040	10-15

Slide Distribution. A presenter follows several links in order to load the desired presentation page. Bandwidth savings ***S***, is defined by $S = n \cdot b \cdot r$, where ***n*** is the number of intermediate pages, ***b*** is the average bandwidth per page, and ***r*** is the loss rate incurred by the receiver. The decrease in latency in receiving the presentation page will be equal to S / D where ***D*** is the speed of the client link.

The numbers in the table reflect the savings incurred when 4 intermediate slides with an average size of 500 KB are loaded.

Dictionary. Lexical and semantic properties of words are used to guess the missing letters from a word. Two scenarios are examined, these being the *without NAK* and *with NAK* (see section 3.2). Assuming no feedback redundancies the first scenario will result in one packet per loss being saved on the incoming link. The second scenario adds the cost of the NAK carrying the guessed word, and one reply, possibly containing the correct word. This process will save bandwidth over traditional repairing methods because one repair packet can repair several lost letters.

The delay required for the user to view the completed word is calculated based on the assumption that it takes one half round trip time for a packet to reach the receiver. For the *without NAK* scenario the gain in delay is given by:

$$t = t_1 + (1+p)(t_1 + t_{bk}) + (1+p)^2 (t_1 + t_1) - t_1$$

where t_1 is half the round trip time, p is the probability for a loss occurrence, and t_{bk} represents the SRM back-off time.

The figures in Table 4 are based on the savings when 100 words of typically 5 characters are written into the whiteboard as text, with the typical values p=0.10, t_1=0.3, t_{bk}=0.35.

The *with NAK* scenario adds very little traffic since complete words are transported rather than characters. The table shows figures based on the assumption that 50% of the guesses are correct. There will be an additional delay for transporting the nack to the sender and for receiving the sender's NAK/ACK regarding the guessed word.

H.261 Repair. A packet containing H.261 data is retransmitted in order to correct future frames. Packet size for H.261 data is dependant on the number of blocks replenished and will vary, as several blocks may be included in one packet. The bandwidth "savings" in the table are negative and represent the additional bandwidth consumed for the worst-case scenario of a repair transmit equal to the maximum transmission unit for Marratech Pro.

Additionally, improvements in latency will depend on the content of the video stream because movement reduces the amount of time that an effective repair can be delivered. Therefore, the latency value represents a scenario with no moment occurring in affected areas before the next background sweep. The time between sweeps varies greatly depending on many factors so the values shown are representative of typical values observed while running sessions at Luleå University of Technology.

5 Conclusions and Future Work

In this paper we have described how semantic reliability can be applied to a real-world collaborative workspace application. We show, using appropriate examples, that dynamic behavior based on semantics can be useful in various aspects of the workspace. In interactive video, which is normally not handled with reliability, we introduce reliability based on application semantics in order to strengthen the transport to a degree at which the application can benefit from it. For the whiteboard and shared web-browser tools ideas have been presented for relaxing reliability based on application semantics, which will allow the applications to handle errors in a manner tolerable by the user.

It is important to note that the default behavior of the framework is always the standard behavior for the application or the protocol in question.

Future work includes implementation of these and possibility other modifications into Marratech Pro and since there is a user base, we intend to perform real-life testing and measuring. This is intended in order to give more concrete conclusions about how the ideas presented will enhance the user experience. While this paper has used more of a "big picture" view, in the future each idea will be explored into more minute detail. For example, improvements in video quality, which is the main goal of adding recovery based reliability, cannot be measured without testing the implementation on a user base.

References

[1] Obraczka K, Multicast transport protocols: a survey and taxonomy *IEEE Communications Magazine* , vol. 36, pp. 94-102, Jan, 1998.

[2] Clark DD and Tennenhouse DL, Architectural considerations for a new generation of protocols *Computer-Communication-Review. vol.20, no.4; Sept. 1990; p.200-8,* vol. 1990.

[3] Marratech. - The e-meeting company, http://www.marratech.com, URL visited August 2nd, 2002.

[4] Schulzrinne H, Casner S, Frederick R, and Jacobson V, "RTP: A Transport Protocol for Real-Time Applications," *IETF RFC 1889,* Jan. 1996.

[5] Floyd S, Jacobson V, Liu C-G, McCanne S, and Zhang L, A reliable multicast framework for light-weight sessions and application level framing *IEEE/ACM Trans. Networking,* vol. 5, no. 6, pp. 784-803, Dec, 1997.

[6] Anonymous, "Video Codec for Audiovisual Services at p x 64 kbit/s," *ITU Recommendation H.261 (03/93),* Mar. 1993.

[7] Mauve M and Hilt V, An application developer's perspective on reliable multicast for distributed interactive media *Computer-Communication-Review. vol.30, no.3; July 2000; p.28-38,* vol. 2000. 0146-4833.

[8] Pereira J, Rodrigues L, and Oliveira R, "Semantically reliable multicast protocols," *Proceedings of 19th IEEE Symposium on Reliable Distributed Systems,* Nurnberg, Germany, pp. 60-9, 2000.

[9] Rodrigues L, Baldoni R, Anceaume E, and Raynal M, "Deadline-constrained causal order," *Proceedings Third IEEE International Symposium on Object- Oriented Real-Time Distributed Computing (ISORC 2000),* Newport, CA, USA, pp. 234-41, 2000.

[10] Baldoni R., Prakash R., and Raynal M./Singhal M., "Efficient ? -causal Broadcasting," *Journal of Computer System Science and Engineering,* 1998.

[11] Elf S and Parnes P, "Applying Semantic Reliability Concepts to Multicast Information Messaging in Wireless Networks," *IRMA Conference Proceedings: Issues & Trends of Information Technology Management in Contemporary Organizations,* Seattle, 2002.

[12] Scholl J and Parnes P, "Low-Weight Congestion Control for Multi-sender Applications," *IFIP/IEEE International Conference on Management of Multimedia Networks and Services 2002,* Santa Barbara, Oct. 2002.

[13] Rhee I, Error control techniques for interactive low-bit rate video transmission over the Internet *Computer-Communication-Review. vol.28, no.4; Oct. 1998; p.290-301,* vol. 1998. 0146-4833.