An Adaptive Mechanism for Real-time Secure Speech Transmission over the Internet

A. Aldini [a] **R. Gorrieri** [a] **M. Roccetti** [a]

The 2nd IP-Telephony Workshop, April 2nd-3rd 2001, Columbia University, New York City, USA

[a]University of Bologna, Dipartimento di Scienze dell'Informazione, Mura Anteo Zamboni 7, 40127 Bologna, Italy. E-mail: {aldini, gorrieri, roccetti}@cs.unibo.it

Outline

- Introduction
  - Real-time Secure Audio over the Internet
- An Adaptive Playout Control Algorithm
  - Securing the Mechanism
- Performance Analysis
- Conclusion

Introduction

Motivation

The Internet provides a *best effort* service over public network without security guarantees.

Sophisticated applications (such as voice-based communications over the Internet) have both strict temporal constraints and security requirements.

Goal

The aim is providing a mechanism which guarantees:

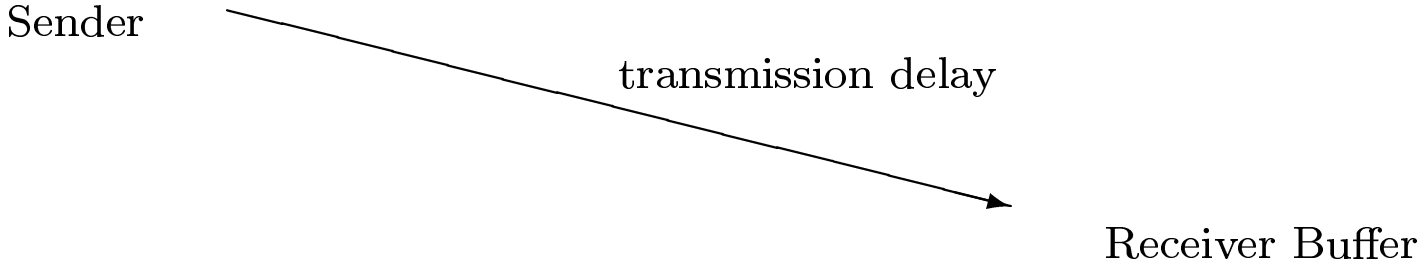
- an adequate QoS in spite of packet losses and jitter (variable delay in transmission),
- authentication, confidentiality, and integrity in spite of the adoption of insecure networks.

Packet Audio over the Internet

Available network bandwidth is not the only requirement to meet for QoS.

Problem: transmission delays depend on network conditions.

Approach: adapting the applications to the jitter present on the network.



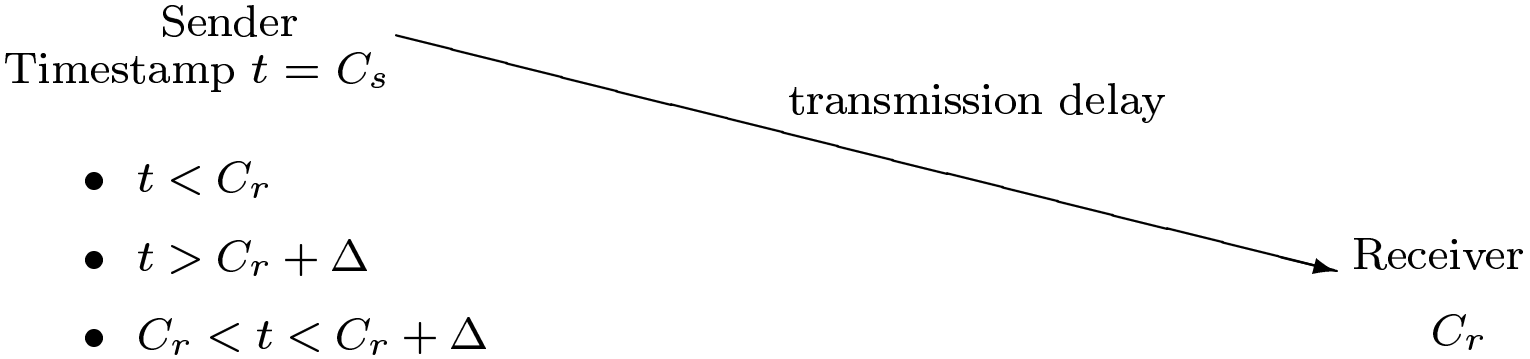
Received packets are queued into the buffer and the playout of each packet is delayed.

Crucial tradeoff between the length of the imposed additional delay and the amount of lost packets due to late arrivals.

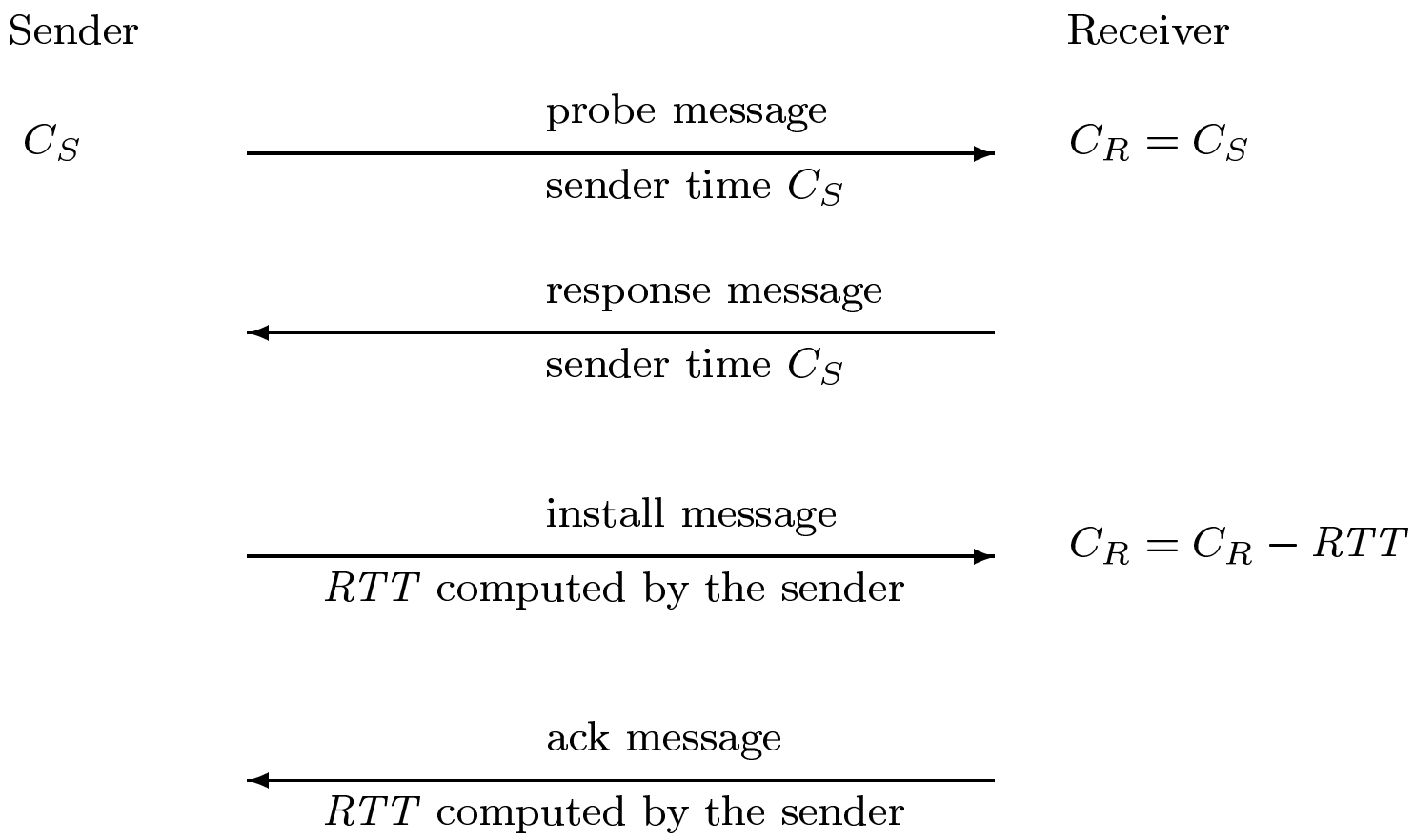
An Adaptive Control Mechanism

Roccetti et al. (1998)

- No assumptions (external synchronization, jitter delay distribution).
- The algorithm periodically estimates the upper bound for the packet transmission delays, by means of a packet exchange of a three-way handshake protocol.
- The callee is provided with the caller estimate of the RTT value experienced, so it may offset its own clock ($\Delta = C_S - C_R$).



The Handshaking Protocol



The handshaking protocol is periodically carried out (e.g. every second) throughout the entire conversation lifetime.

Securing the Mechanism

- A preliminary authentication phase, during which the first symmetric key is exchanged, precedes the conversation.
- Each packet of the handshaking phase is encrypted with the symmetric key by using a block cipher.
- During such a phase the parties exchange a session key K which is used by a stream cipher as a seed for the pseudorandom generation of the keystream.
- Each audio packet belonging to the chunk of conversation i between the 2 consecutive synchronizations i and $i+1$ is encrypted with the stream cipher which uses the session key K_i.

Securing the Handshaking Protocol

- The *ack* packet is needed in order to come to an agreement on the instant the session key changes.
- Tampering of packets is prevented by
  - the secrecy of the symmetric key,
  - the presence of the timestamps and the RTT values.
- Dropping of handshaking packets can be prevented by masquerading such packets as normal audio packets, by filling the audio sample with rubbish.
- Anyway, we can shut down the conversation if more than n consecutive handshaking phases are broken off.

Securing the Conversation

Sender

1. $P_j = \{t_s, M_j\}$
2. Send $P_j^* = \{\{P_j\}_{K_i}, MAC(K_i, P_j)\}$

Receiver

1. Receive P_j^*
2. Compute t_s and M_j with K_i
3. Verify the MAC

For each audio packet created with the above algorithm and received in time for its playout, the receiver can decide its playout instant and verify its integrity and the authenticity of the sender.

Security Conditions

Secrecy

The assurance of the privacy of the conversation is enforced by the very brief lifetime of the session keys used by the stream cipher.

Authenticity

The preliminary authentication phase and the handshaking protocol guarantee authentication of the parties. A *man in the middle* can neither spoof nor forge packets.

Integrity

Altered packets are revealed by checking the MAC.

Experimental Assessment

Scenario

Stream Cipher: RC4.

Block Cipher: Blowfish.

MAC: MD5 encrypted with the session key.

Interval between 2 consecutive synchronizations: 1 second.

	Computing Time (ms)
Block Cipher	0.008
Stream Cipher	0.0591
MAC	0.0474
Total Latency	0.1145

Comparison

- Some well-known application-level tools are considered for a comparison:
  - Speak Freely (`www.fourmilab.ch`)
  - PGPfone (`http://www.pgpi.com`)
  - Nautilus (`http://www.lila.com/nautilus/`)
- They employ codecs in order to reduce the quantity of data to be transmitted and block ciphers for the encryption/decryption of data.

Comparison: examples

Speak Freely

Computing Time (ms)	CODEC			
	GSM		ADPCM	
	Mean	Variancy	Mean	Variancy
Blowfish	2.47	0.01	5.22	0.15
IDEA	3.94	0.01	9.08	0.05
DES	9.77	0.20	20.8	0.16

PGPfone

Computing Time (ms)	CODEC			
	GSM lite 4.4		ADPCM	
	Mean	Variancy	Mean	Variancy
Blowfish	2.09	0.06	4.72	0.02
CAST	2.08	0.002	4.43	0.07
3DES	6.35	0.14	16.8	0.56

Conclusion

The mechanism we presented offers:

- a packet audio control mechanism,
- a complete security infrastructure.

The integration of these two aspects is realized in a simple way and with a negligible overhead (BoAT is about 2 orders of magnitude better than the other tools).

The packet audio playout control algorithm has been passed through intense formal functional and performance analysis. We aim at applying formal methods also for the analysis of the security properties of our mechanism.