Real Time Voice and Multimedia over IP

Comms Group Symbian Seminar
Friday 16th of March 2001

presented by John Pagonis, PAN Team

symbian

Today’s Menu

•Intro
•Buzzword compliance - Protocols
•Classic comms stuff - FEC and codecs (briefly:-)
•Apps and usage scenarios for multi medium WID comms
•eBoard
•epoCall
•epoCast
•Fin

Focus and Scope

This talk is about the protocols, considerations, constraints and techniques of transmitting various media like audio, video and still images over IP networks in real-time, with a bias towards WID applications.

This talk will be very ‘RFCentric’

Note: this talk is _not_ about Internet telephony, the Mbone, RSVP, QoS or routing.

‘Interrupts help unblocking’

- also they keep us awake
- stop me from talking bananas
- help understanding
- give me some time to have some water

How real time is ‘real-time’ ?

Real-time usually means ‘fast enough’

Real-time usually implies ‘flow at a constant rate’

Real-time doesn’t need to happen ‘now’ but it needs to be experienced as such.

Real time is an experience and it can sometimes exist only as a state of mind.

A falling star is experienced in real-time, in reality this occurred few thousand of years prior to this presentation.

Real-Time services

- Interactive - 2 to n-way

- Non-interactive

different constraints for each, but the same Internet laws apply for both.

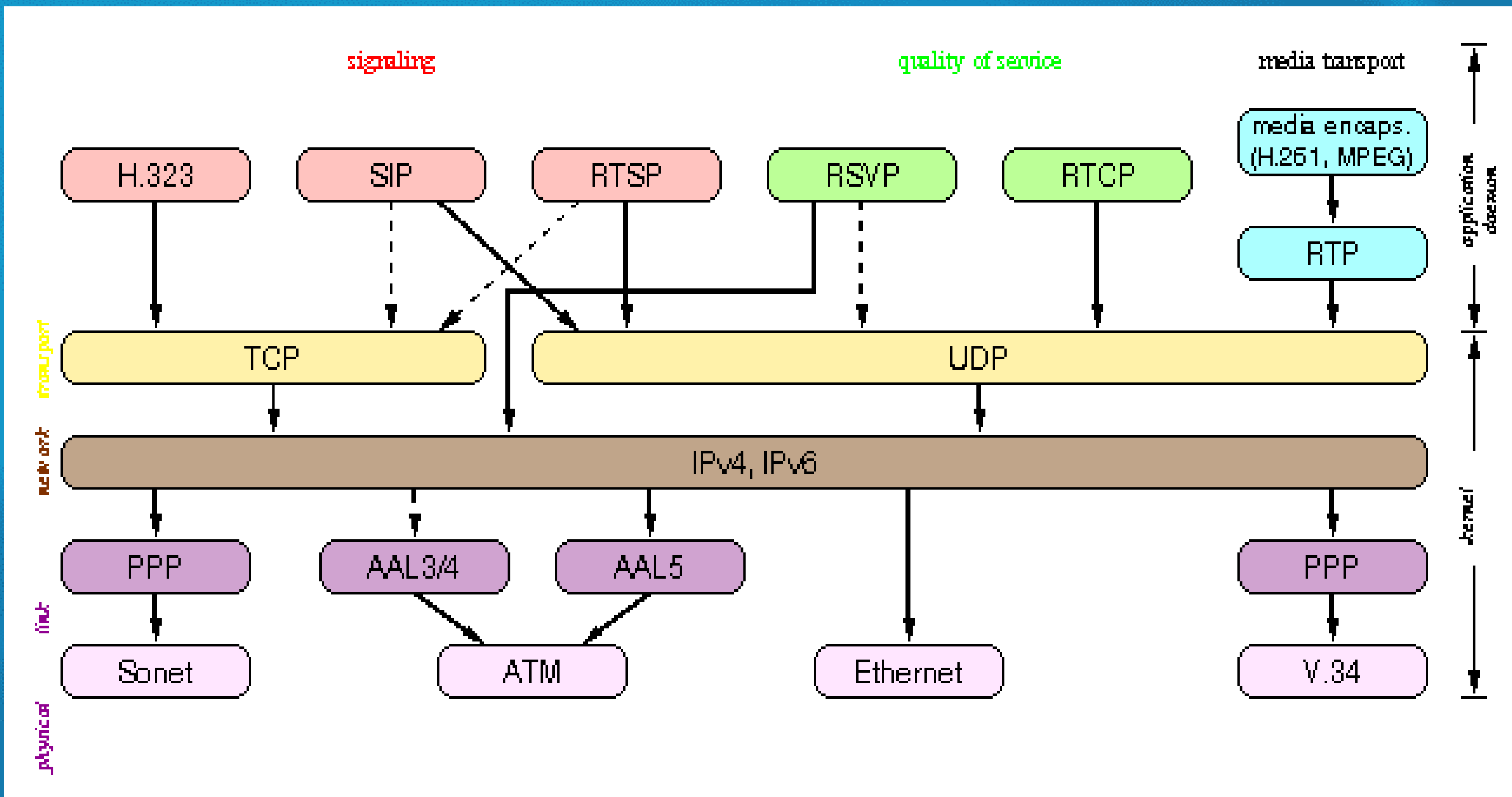
- best effort delivery
- unbounded delays
- lossy channels
- etc

Getting to know the basic tools for the job

Executive summary of :

- UDP
- RTP/RTCP
- SDP (Not the Bluetooth one:-)
- SIP
- RTSP
- H.323

An incomplete Internet protocol layer overview



User Datagram Protocol

- Uses IP
- Multiplexing
- CRC
- Minimal overhead - 8 bytes header
- Multicast capable
- Connectionless operation

Real-time Transport Protocol - RFC 1889

Principles of application level framing and integrated layer processing
Provides end-to-end delivery services to data with real-time characteristics:

- Payload type identification
- Sequence numbering
- Time-stamping
- Media source identification
- Delivery monitoring, other session info - via RTCP packets
- Supports multicasting
- 12 byte header

Real-time Transport Protocol - RFC 1889

No guarantees on:

- QoS
- Delivery
- or of timely delivery
- or of ordered delivery

It is in effect a protocol framework, which for different applications has a different spec according to app profile and payload format. (see RFC 2736 and RFC 1890 for guidelines and profiles)

Started in 1992, first proposed in 1995, became RFC in 1996 and version 07 of the RTP2 draft was submitted in Mar 2000

Session Description Protocol - RFC 2327

- Text-based well defined format for session description.
- Intended to be used by various session/transport protocols like SAP, SIP, HTTP, RTSP and mail (MIME) clients !
- Not intended for doing the session negotiation
- Text payload <1K
- MIME type is application/sdp

Session Description Protocol - RFC 2327

SDP payload includes:

- Session name and purpose
- Time that the session should be active
- Media and profiles comprising the session
- Receiver info (addr,ports,media formats etc)
- Bandwidth needed
- Contact info of the ‘admin’ of the session
- Transport protocol used

Session Initiation Protocol - RFC 2543

- Application level signalling protocol for creating, modifying, terminating and negotiating sessions
- Text-based
- Comms in the session can be multicast, unicast or hybrid
- Transport independent
- Supports ‘Personal Mobility’, proxying and redirections
- Call oriented (setup, transfer, terminate etc)
- Works with invitations that carry session descriptions (SDP)

ITU H.323

- Protocol suite (H245, H225.0, H332, H450.1, H450.2, H450.3 H235, H246)
- Basically does session signalling and conference control
- Binary representation of messages based on ASN.1
- Allows for huge parameterisation of messages, many things can be done in many ways
- Stateful operation that needs TCP (for the IP domain)
- Originally conceived for use on a single LAN segment without any QoS guarantees - NOT the Internet
- Base on older ISDN signalling protocols (Q.931)
- 768 pages long (not including ASN.1 messages)

ITU H.323

- Can be used only with ITU registered and standardised codecs
- Overlapping functionality with RTCP

it provides primitives for:

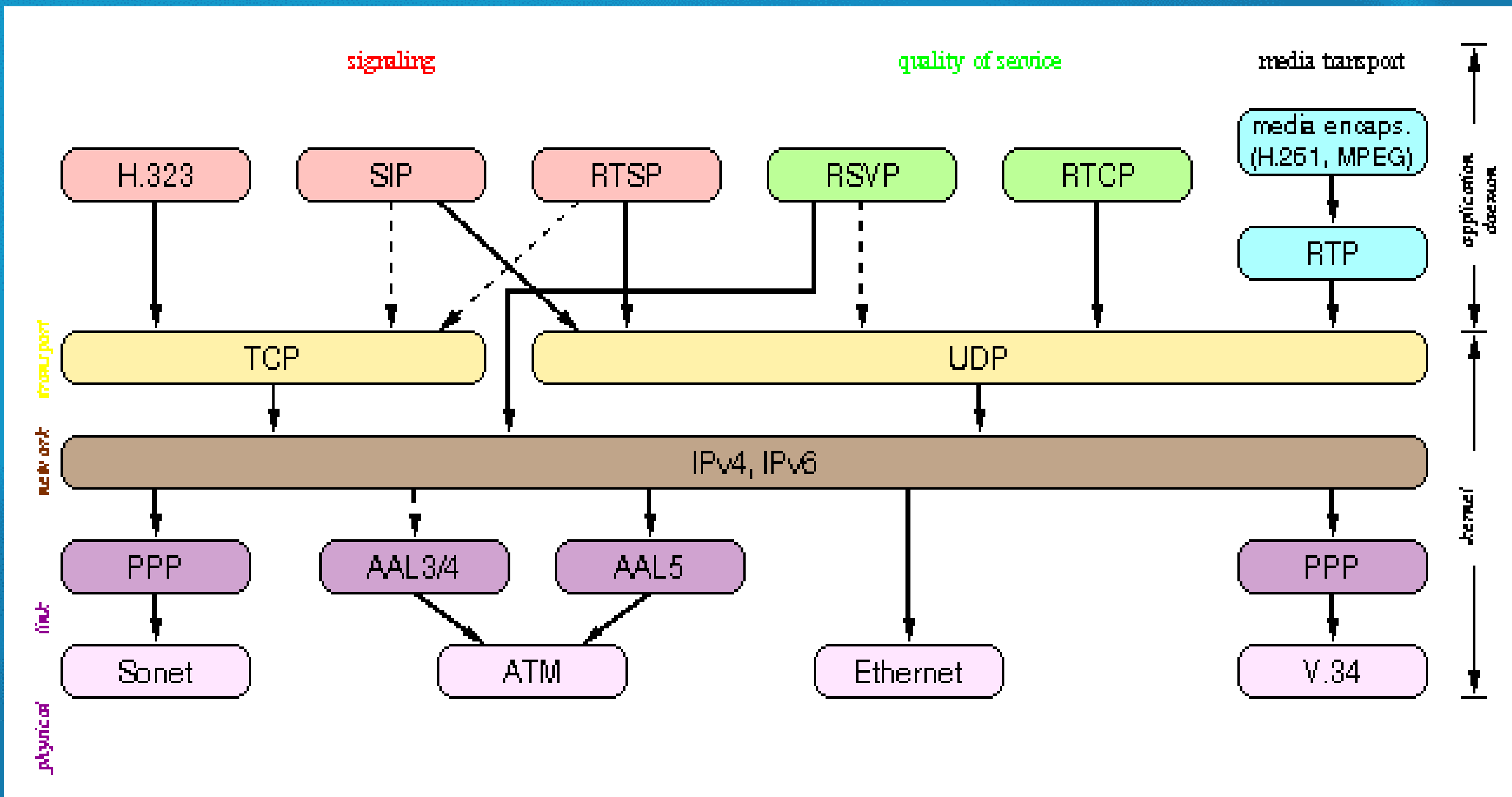
- Conference control
- Capability exchange
- Session maintenance
- Basic signalling
- QoS functions
- Registration
- Service discovery
- Session establishment and negotiation.

Real Time Streaming Protocol - RFC 2326

A ‘network remote control’ for multimedia streams

- It allows for video-like control of streams with real time characteristics
- Text-based
- Uses either RTP, UDP,multicast UDP, TCP
- Came from Netscape and RealNetworks

An incomplete Internet protocol layer overview, again



Comms building blocks - codecs

A codec is a device that encodes or decodes signal.

Also stands for compressor/decompressor, referred as a technology (SW,HW or both) that is used for compressing and decompressing data.

Codecs are a science by themselves, an electronics engineering discipline and a 'black art' at the same time :-)

Their history is almost as old as this of comms.

They are also very difficult to explain … (for me at least)

Comms building blocks - codecs

For our purposes, codecs are used to encode in digital compressed format, streams of data from various sources, usually for transmission over a lossy channel.

The selection of appropriate codecs is critical and central to the design of a communication system.

Considerations when choosing one:

- Bit rates
- Output Quality
- Latency
- CPU power needed
- Dynamic range (audio)
- Motion in the source (video)

Comms building blocks - FEC

Forward Error Correction we call the system/technique where a Rx of a data transmission can detect and correct a symbol or code block (up to a predefined length) without the need for a return path to (the Tx) or retransmission.

Basically adding redundant data to a frame in order to protect it (or another one) from loss or random noise. This is achieved by using a predetermined algorithm between Tx and Rx.

FEC is good for correcting random (single bit many times) errors and possibly drooped frames. FEC cannot cope with bursts of errors that span across many frames.

Forward Error Correction - a simple example

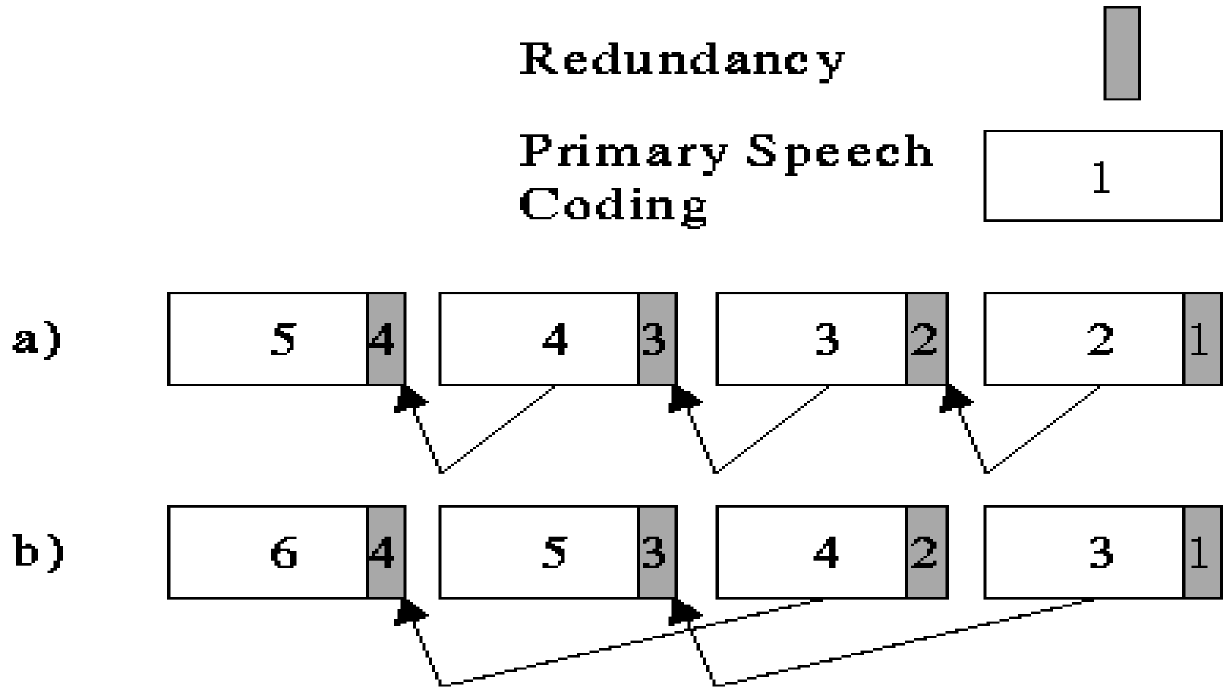


Figure 1: The Positioning of Redundancy Relevant to the Primary Coding for a) low and b) higher loss rates

Apps for Multimedia over IP

Things to consider:

- Usage scenarios and usability
- Domain specific knowledge
- Constraints - bandwidth, CPU, latency etc
- Target audience
- Mechanisms and responsibilities
- Existing framework, cost (unimportant:-), schedule (??)
- Interop and standards

Also, which medium is most important ?

e.g who cares about the lecturer's shirt ? slides and his voice are more important.

eBoard - the epoc whiteboard app

- Two users shall be able to share a virtual whiteboard and doodle in real time over the Internet. Connections will happen on pre-determined ports on the users' known IP addresses.

- Low bandwidth (suits current GSM technology)
- Touch screens :-)
- Small latency - delay in updates is important but not critical
- Minimal feedback between Rx/Tx pairs
- Best effort delivery
- Interactivity - better have something there than all late

The app usage defines the protocol

eBoard - things to consider

- Transport
- Packetisation
- FEC
- Feedback
- Connection establishment

eBoard - design

- Treat the XxY screen as a grid of NxK cells
- Package each cell in a single frame - size ?
- Frame format includes cell coordinates, timestamp, cell subset that was last updated, so that Rx can check and possibly request updates in case of great loss.
- Use UDP only, on the predefined ports - dead easy
- Variable but low bitrate - how low ?
- No packet reordering is necessary on the Rx !!!

Control plane:

- Connect, disconnect, reject
- Retransmit, info packets

Yes it sounds remarkably like…… ER5 Battleships !!!

epoCall- epoc Voice over IP app

A user shall be able to use its WID to place and receive full duplex audio traffic over IP. Extremely useful when this is combined with 802.11 or Bluetooth, where the user will use the company's/hotel's/etc access point to make free Internet calls.

- Low bitrate ~20k on average
- Telephone quality audio
- Very small latency <=250ms
- Highly interactive
- Continuous and constant flow of data
- Best effort delivery
- The app and human perception defines the protocol

epoCall- things to consider

- Codec(s) ?
- FEC
- Transport
- CPU needed
- Packet re-ordering and loss
- Medium aware stream fragmentation - small frames = lower throughput, more reordering but less latency - larger frames = more latency
- Comfort noise
- session management and mobility

epoCall - design

- Use GSM 06.10 audio codec ~13K runs on a 25MHz 486sx
- Use LPC codec for FEC ~ 4.8K
- FEC, use redundant audio piggybacked on SeqN-1 frames
- Use RTP
- Stop re-ordering and do FEC scheme if many packets out of seq
- Insert comfort noise, at the Rx, when there is not traffic
- could use RTCP more if variable quality and many codecs were used for adaptive Tx channels

epoCall - FEC example

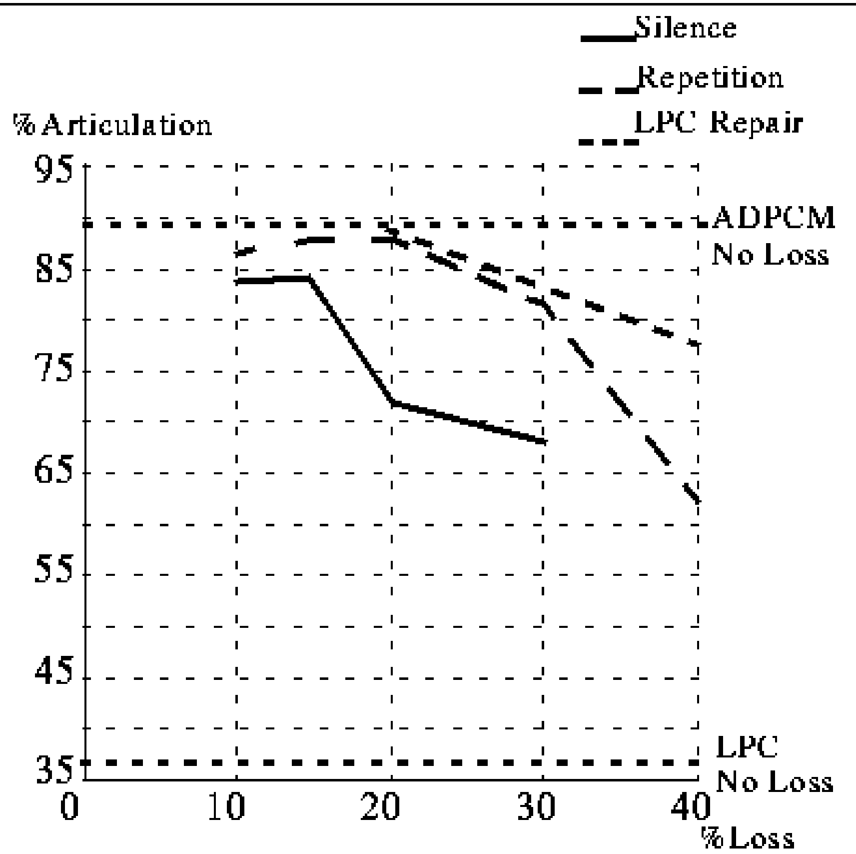


Figure 2: Voice Reconstruction for 20/40ms Packets

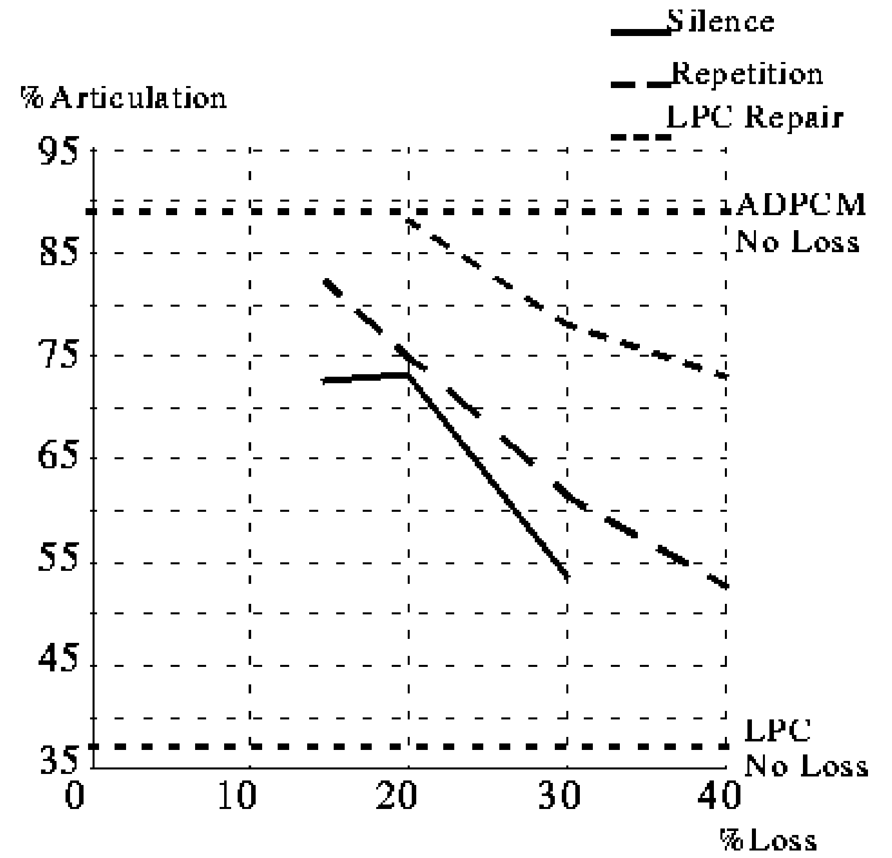
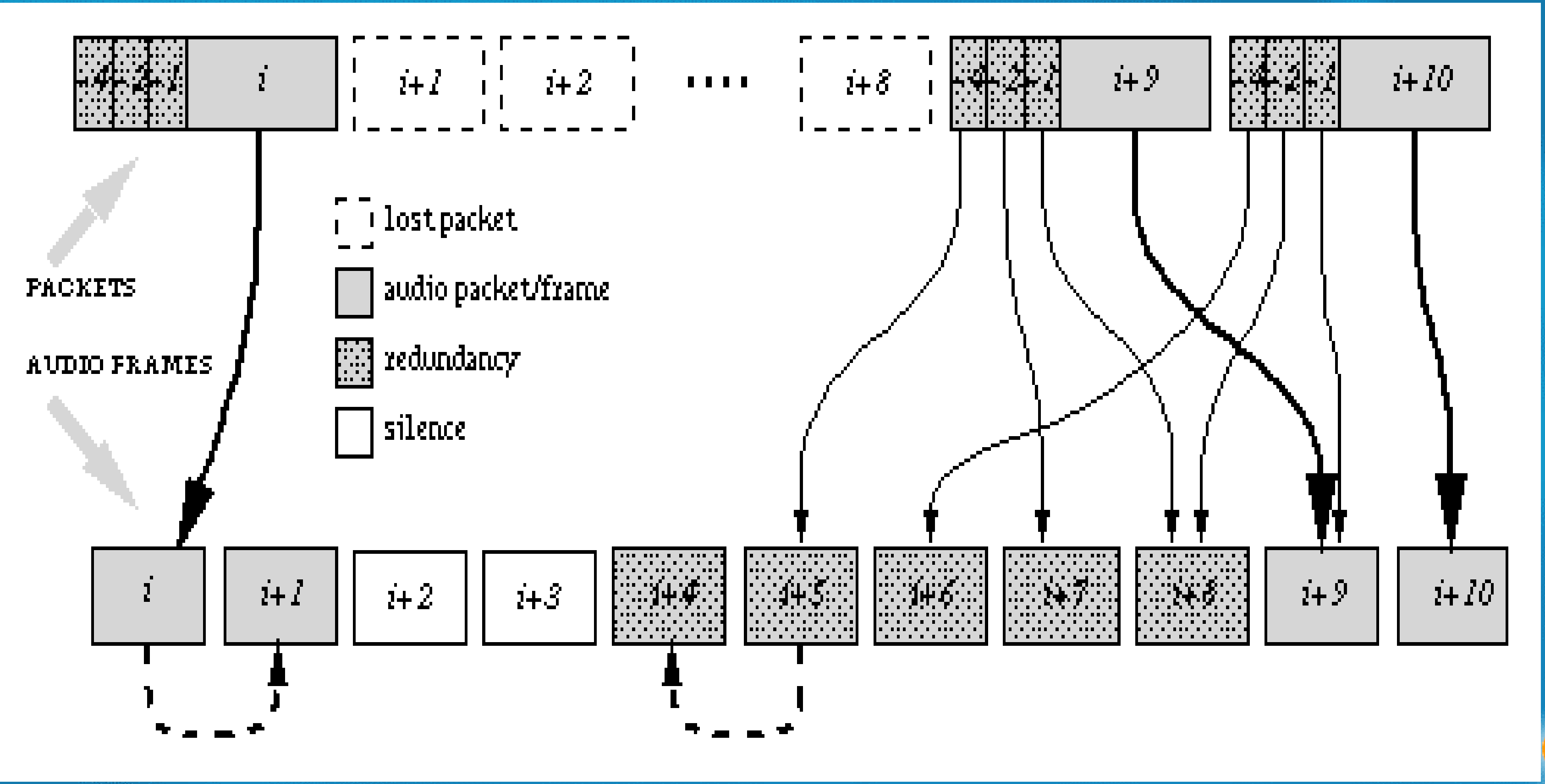


Figure 3: Voice Reconstruction for 80ms Packets

epoCall - FEC overdose



epoCast

• usage: remote surveillance (kids, dogs, garden etc), newscast traffic monitor, Live football on the bus !!!!

• GPRS/3G bitrates
• 2 streams, video and audio
• Stream synchronisation
• Latency not a big problem
• Video like control for non live feeds
• Opportunity for buffering and frame-loss recovery

epoCast - things to consider

- Many video codecs out there, motion in football is a problem :-(
- Lip-sync is very important
- Transport protocol
- Session control protocol(s)
- CPU considerations (video playback)
- Framing
- Video-like controls

epoCast - design

- H263 video stream using RTP - RFC 2190
- FEC is embedded in H263 frames
- H263 decoding can be done on 486sx 25MHz
- Timestamping and RTP source ids are very important for lip-synch at the receiver end.
- For the audio stream use the same technique as in epoCall, see RFC 2198 and utilise our favourite 2->n audio codecs
- RTCP is used from the server not only for reports but also in order to show which streams are associated, thus need synchin
- RTCP is used from the client to report quality of Rx to the server
- SIP/SDP is used from the to initiate the session, can happen either way e.g. important newscasts may be initiated from the server.
- RTSP is used for the ‘net remote’