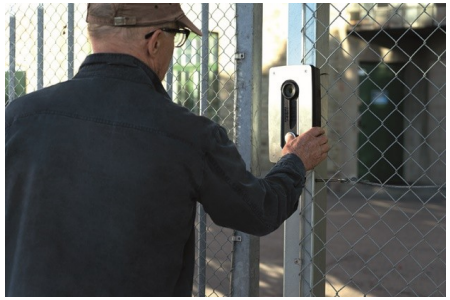
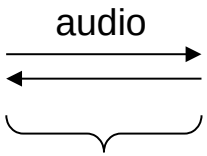




Cutting audio latency with bidirectional WebRTC

Albert Sjölund, AXIS Communications

Outline

> Background

> Measurement method

> Findings and possible optimizations

Background

Axis & Axis products

- Security cameras
  - Other security products
- Servers and software for video management
- Cloud services
- Headquartered in Lund, Sweden

Camera

Camera
(can rotate)

Camera
(can rotate)

Camera

Bodyworn
camera

Speaker

Door station
(intercom)

Recorder

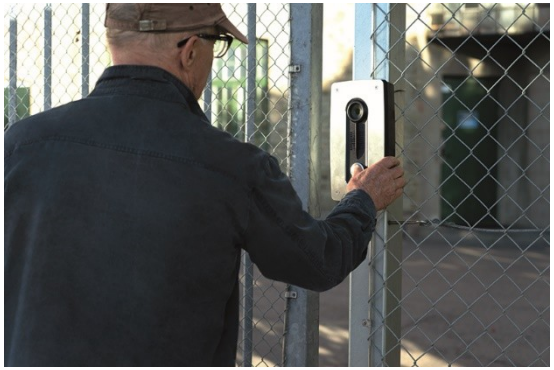
AXIS & Streaming

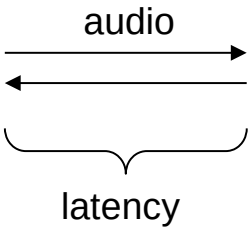
> Streaming protocols
  > SIP

  > RTSP

  > WebRTC – modern and low latency
    > Versatile
    > WebRTCBin & Gstreamer

AXIS two-way-audio/intercom use case





> Two-way audio not a usecase previously
> Intercoms, medical devices
> Implementation existed – needs improvement

INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.114

(05/2003)

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

International telephone connections and circuits – General Recommendations on the transmission quality for an entire international telephone connection

One-way transmission time

ITU-T Recommendation G.114

e of connections involving various impairments, including delay; and ITU-T
[5] maps transmission rating predictions of the model into categories of speech
n quality. Thus, while this Recommendation provides useful information regarding mean
elay as a parameter by itself, ITU-T Rec. G.107 [3] (and its ITU-T Rec. G.108 [4] and
. G.109 [5] companions) should be used to assess the effects of delay in conjunction with
irments (e.g., distortions due to speech processing).

commendations for one-way transmission time

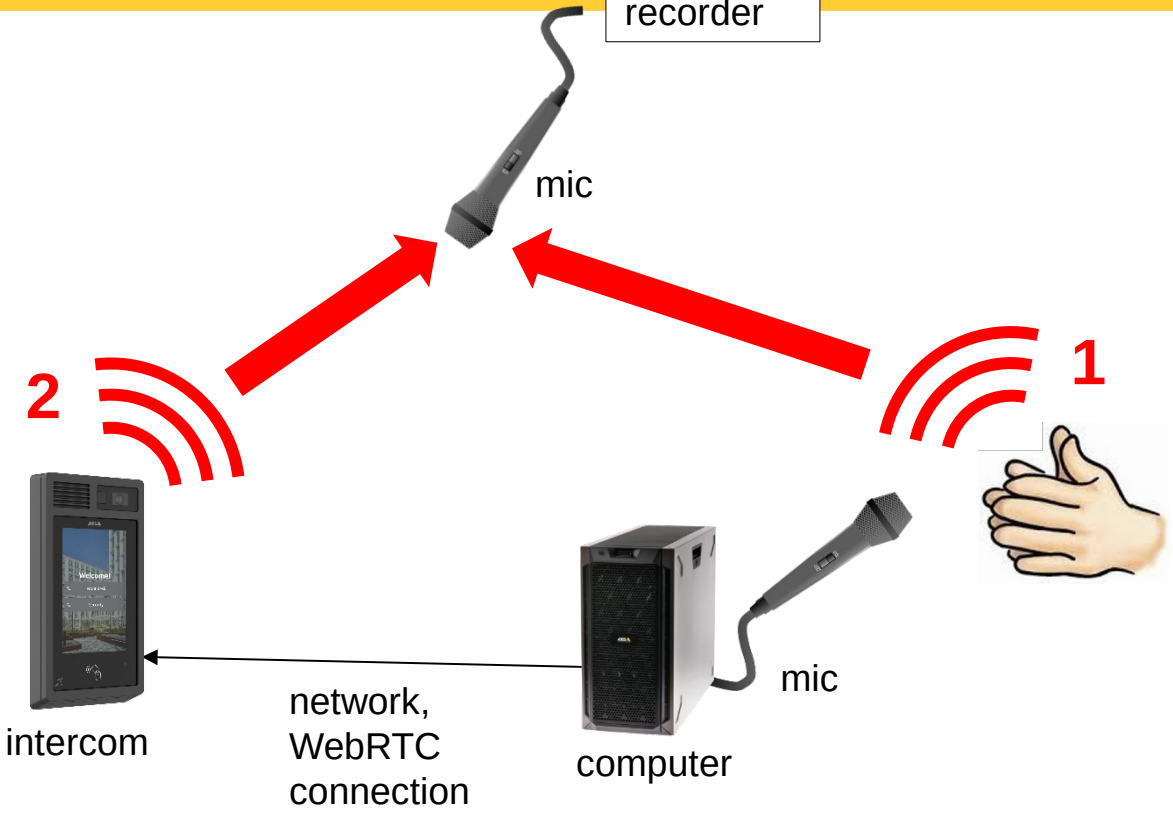
of the type of application, it is recommended to not exceed a one-way delay of 400 ms
l network planning (i.e., UNI to UNI, as illustrated, for example, in ITU-T
-1 [13]), a value that allows flexibility in deploying global networks, without making an
umber of user experiences unacceptable.

it is desirable to keep the delays seen by user applications as low as possible. The
ould be used to estimate the effect of one-way delay (including all delay sources, i.e.,
ear") on speech transmission quality for conversational speech as shown below. For
applications such as interactive data or video, there are no agreed-upon assessment tools
-model, so the effects of delay on such applications must be carefully monitored.
few applications may be slightly affected by end-to-end (i.e., "mouth-to-ear" in the case
delays of less than 150 ms, if delays can be kept below this figure, most applications,
and non-speech, will experience essentially transparent interactivity.

ays above 400 ms are unacceptable for general network planning purposes, it is
that in some exceptional cases this limit will be exceeded. An example of such an
s an unavoidable double satellite hop for a hard-to-reach location, the impact of which
nated by use of the advantage factor in the E-model.

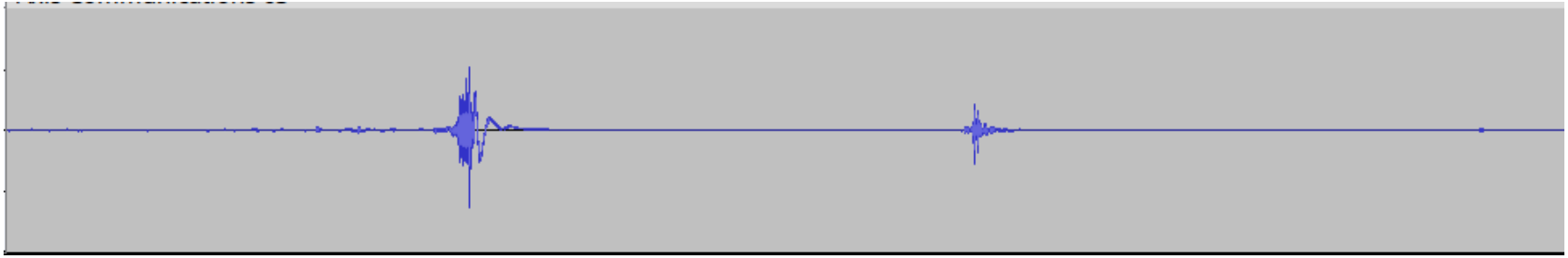
Measurement method

Measurement method – out to in



Example diagram (Audacity)

> Many programs, Audacity is a great choice



Findings and possible optimizations

Measurements

AXIS Intercom

Current WebRTC release:

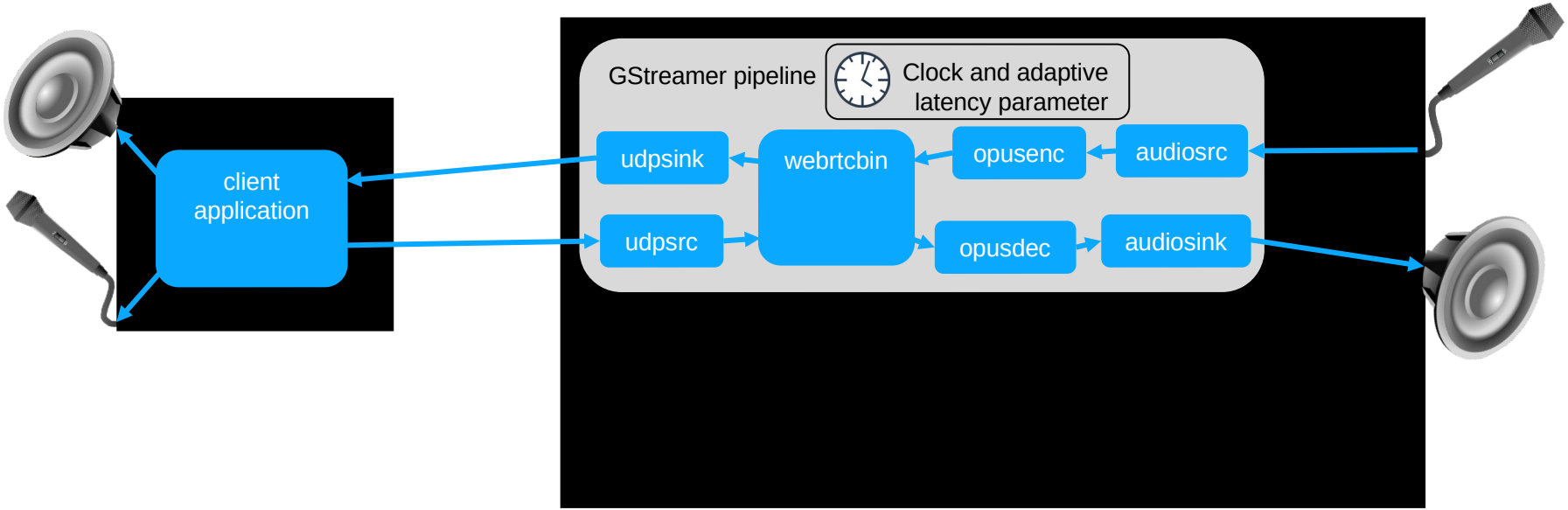
450 ms

650 ms

> Unusable for voice communication, overlap between speakers.
> Incorrect setup likely

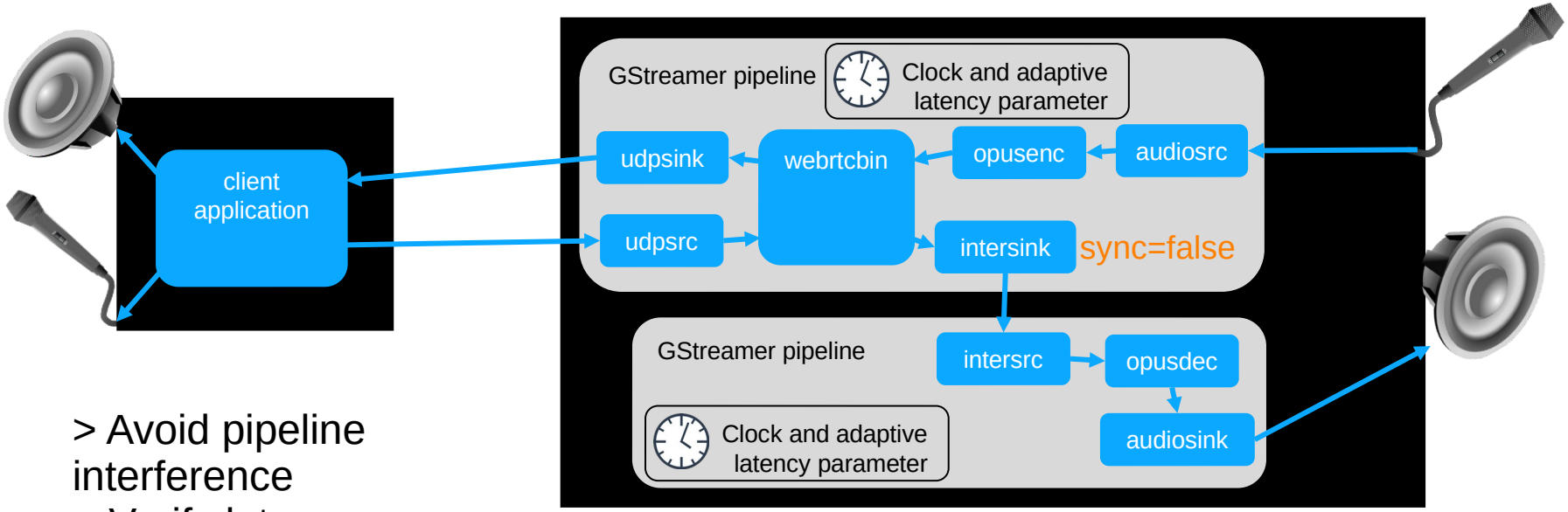
Demo client, WebRTC Firefox

Pipeline setup - latency



> Global latency ~400ms
> Two directions causing interference

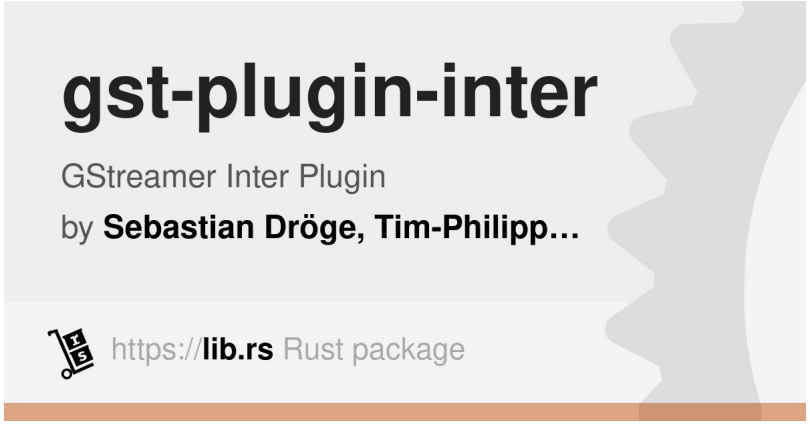
Split with pipeline decoupling



> Avoid pipeline interference
> Verify latency calculations!

Upstream work – pipeline decoupling

- gst-plugin-inter (intersrc/intersink) – turnkey solution for pipeline interoperability
- Rtpbin2
  - Rtpsend, rtprecv
  - Automatic pipeline decoupling, splitting send and receive



Measurements

AXIS Intercom

Current WebRTC release: 400 ms

With split gstreamer pipeline: 230 ms

> Not affected by incoming latency

650 ms

430 ms

> What happened here?

Test client for WebRTC - Firefox

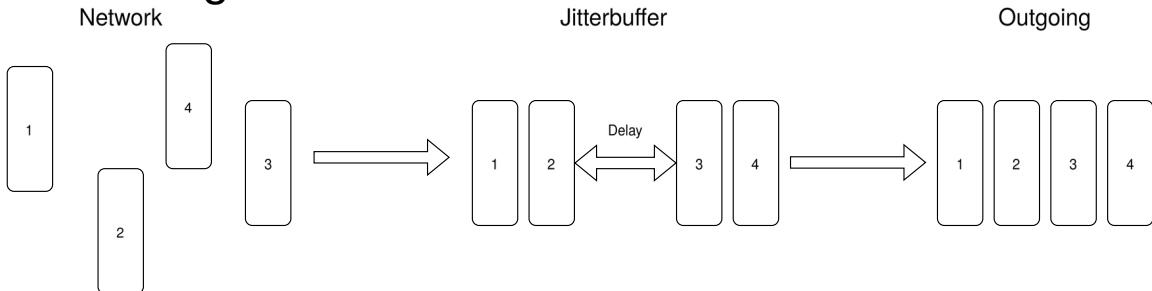
The Jitterbuffer

> Handles jitter by buffering packets
  > In UDP, reordering is common, ensure outgoing is consistent

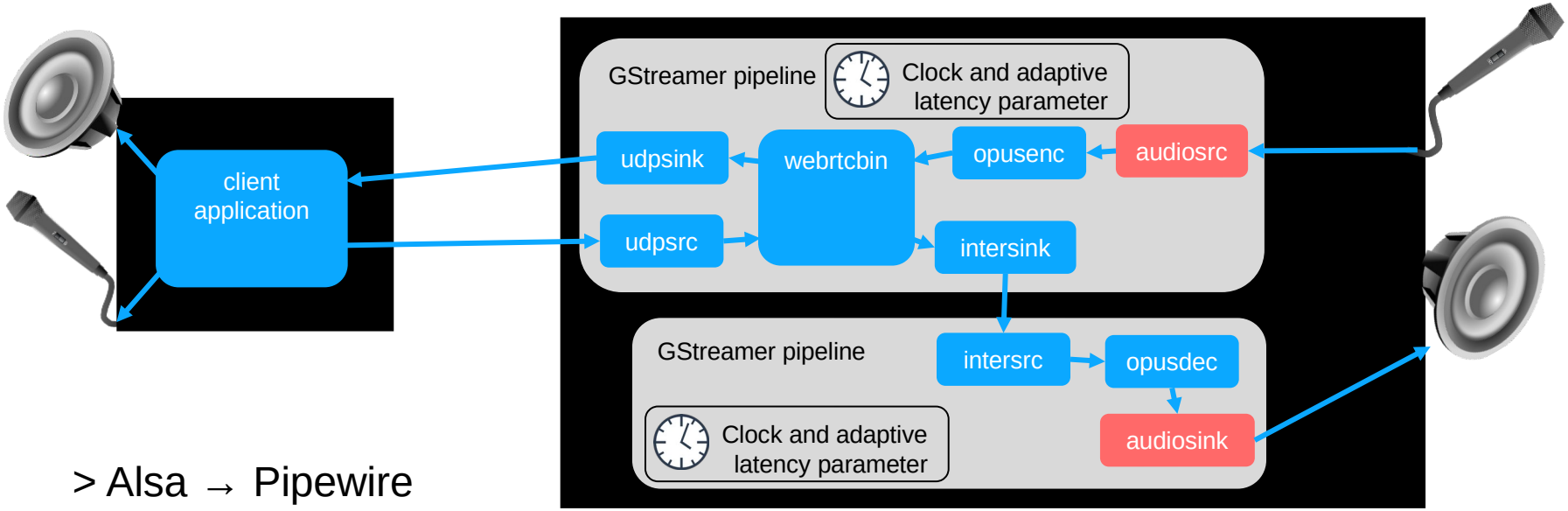
> Increases latency by static amount
  > Previously in both directions

> Missing update caused latency to be ignored
  > Caused timing issues
  > Thanks upstream!

> Dynamic algorithms – only increasing for now



The audio provider



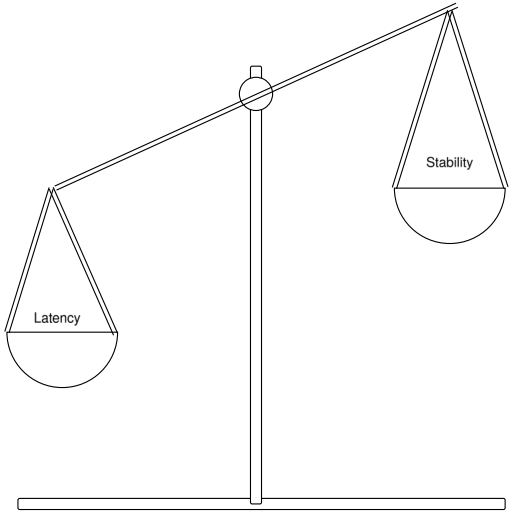
> Alsa → Pipewire
> Minimize overhead
> Buffer size

Pipewiresink – what can be changed internally?

> Quantum (buffer size) affects latency

> Tradeoff stability ↔ latency

> System dependent
  > pipewire-alsa
  > Different quantums



Was the goal reached?

GOAL - 150ms

Final measurements – almost!

AXIS Intercom

Current WebRTC release: 450 ms

With improvements: 250 ms

650 ms

220 ms

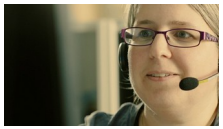
> Low perceived latency, good experience

Demo client, WebRTC Firefox

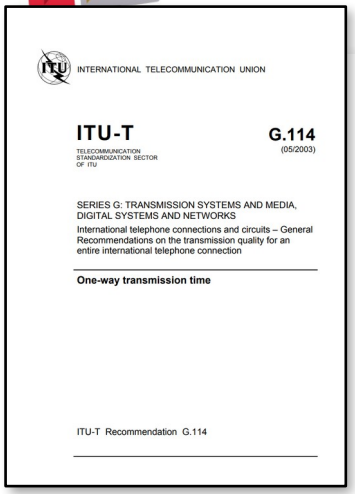
More work still needed

- Reaching 150ms – buffers, buffers, buffers
- Jitterbuffer work
- Hardware/Software limitations/possibilities
  - Audio features like echo/noise cancellation
- Gstreamer, client, hardware

Current WebRTC release:	450 ms	650 ms
With improvements:	250 ms	220 ms



Contact: alberts@axis.com



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T **G.114**
(05/2003)

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

International telephone connections and circuits – General Recommendations on the transmission quality for an entire international telephone connection

One-way transmission time

ITU-T Recommendation G.114