Audio/Video Transport Working Group:
Meeting Agenda
Proposed Changes to Real-time Transport Protocol (RTP)

Stephen Casner
University of Southern California
Information Sciences Institute

March 29, 1994

Agenda for Tuesday, March 29, 1330–1530

- Review agenda
- Status of Real-time Transport Protocol and working group
- Presentation on changes proposed in response to Area Directorate review
- Report on test implementations of proposed changes
- Discussion of proposed changes
  - What issues, problems may have been overlooked?
  - Gauge consensus for or against changes

Agenda for Wednesday, March 30, 1330–1530

- Continuation of discussion as needed
- Presentation on RTP and Synchronization Protocol
- Presentation of new encoding specifications over RTP
  - JPEG
  - MPEG1 / MPEG2
  - Cell–B
  - H.261 (?)
- Assess schedule for RTP completion, any other work

Status of RTP and AVT Working Group

- RTP spec draft-ietf-avt-rtp-04 submitted October 1993 to Trans. Area Dir. for “IESG Last Call” as “proposed std.”
- No comments received at November 1993 IETF meeting
- Informal feedback from Area Directorate review:
  - Change status to “experimental”
  - Requested protocol changes to incorporate ALF
- Two discussions by Casner, Frederick, Jacobson on vat and nv as test cases for RTP with proposed changes
- Email summary of proposed changes sent March 21

Application Level Framing

As proposed by David Clark and David Tennenhouse in their SIGCOMM ’90 paper, two design principles:

Application Level Framing (ALF) — lower layers (transport and presentation) process data in frames specified by the application such that they can be processed out of order

Integrated Layer Processing (ILP) — structure protocols to permit implementations to perform all data manipulation steps in one or two integrated loops, including:

- Minimize the number of multiplexing points
- Minimize the number of inband control operations

Overview of Proposed Changes

As applied to RTP, the following changes are proposed (though not all of these are exactly motivated by ALF):

- Control and data on separate ports
- Remove Channel ID, put multiplexing into encapsulations
- Media-specific timestamps
- Application-specific sync marker bit
- Global source identifiers, for loop detection
- Replace data options with a bit field
- Control packet format / Reception reports
  (and specify more precisely how they should be used)

Control and data on separate ports

No RTCP functions in RTP data packets; use separate packets on another port (e.g., +1).

Advantages:

+ Allows monitors to collect feedback from receivers without having to sort through the data packets
+ Control/data multiplexing on port to streamline data processing

Disadvantages:

– Consumes more port number space, more “sockets”
– May be harder if firewall ports manually administered (moot if port numbers are randomly allocated)

Eliminate Channel ID

UDP port number provides multiplexing in usual case; add multiplexing field in encapsulation for multiple RTP packets in UDP packet, and if needed for RTP-over-IP or -AAL5

Advantages:

+ Eliminates a multiplexing point
+ Frees bits in fixed header for other uses

Disadvantages:

– RTP-over-IP and RTP-over-AAL5 packets become longer

Media-specific timestamps

RTP timestamp remains 32 bits but may tick at any rate specified by encoding rather than always 65536 Hz, and is not adjusted by sender for drift with respect to real time.

Control packet carries both RTP timestamp and 64-bit NTP timestamp to establish relationship to real time (if needed).

Advantages:

+ Avoids time conversion calculations at the sender and receiver, and potential arithmetic errors at the receiver
+ Doesn't require sender & receiver to know what time it is
+ Reduced number of drift corrections for continuous transmission

Media-specific timestamps (cont.)

Disadvantages:

- Relationship to real time must be communicated separately
- Monitoring and recording tools would need to know about the predefined formats and their clock rates, whereas before they could be format-independent
- A receiver must wait for the periodic control packet before synchronization can be done

Application-specific sync marker bit

Main RTP spec provides sync marker bit in fixed header, but leaves meaning of the bit to application or encoding profiles

- Most applications will want to mark something
- Loss monitor may see relationship to marker

Advantages:

+ Application profile spec may choose
+ Don't require both end and start packets to find start

Disadvantages:

– Application profile spec must choose

Global source identifiers

Current: Sources identified by triplet (16-bit ID's usually 0): *[transport address/port, sync source ID, content source ID]* ID's are locally unique, assigned by bridges, translators.

Proposal: Sources identified by 32-bit number unique within a particular session (shared across media). A single source on a host with an IPv4 address may use that value; otherwise chosen at random from the "Class F" IP address space (26 bits). Collisions of random numbers resolved by:

- Observing others' IDs before transmitting at startup
- Challenge issued by any site when new site conflicts; random delays used to avoid implosion

Global source identifiers (2)

For some applications, identifiers should be stable across invocations to avoid loss of ownership of previously generated information. Random ID's or local ID's would have to be remembered in persistent storage (e.g., a file).

Advantages:

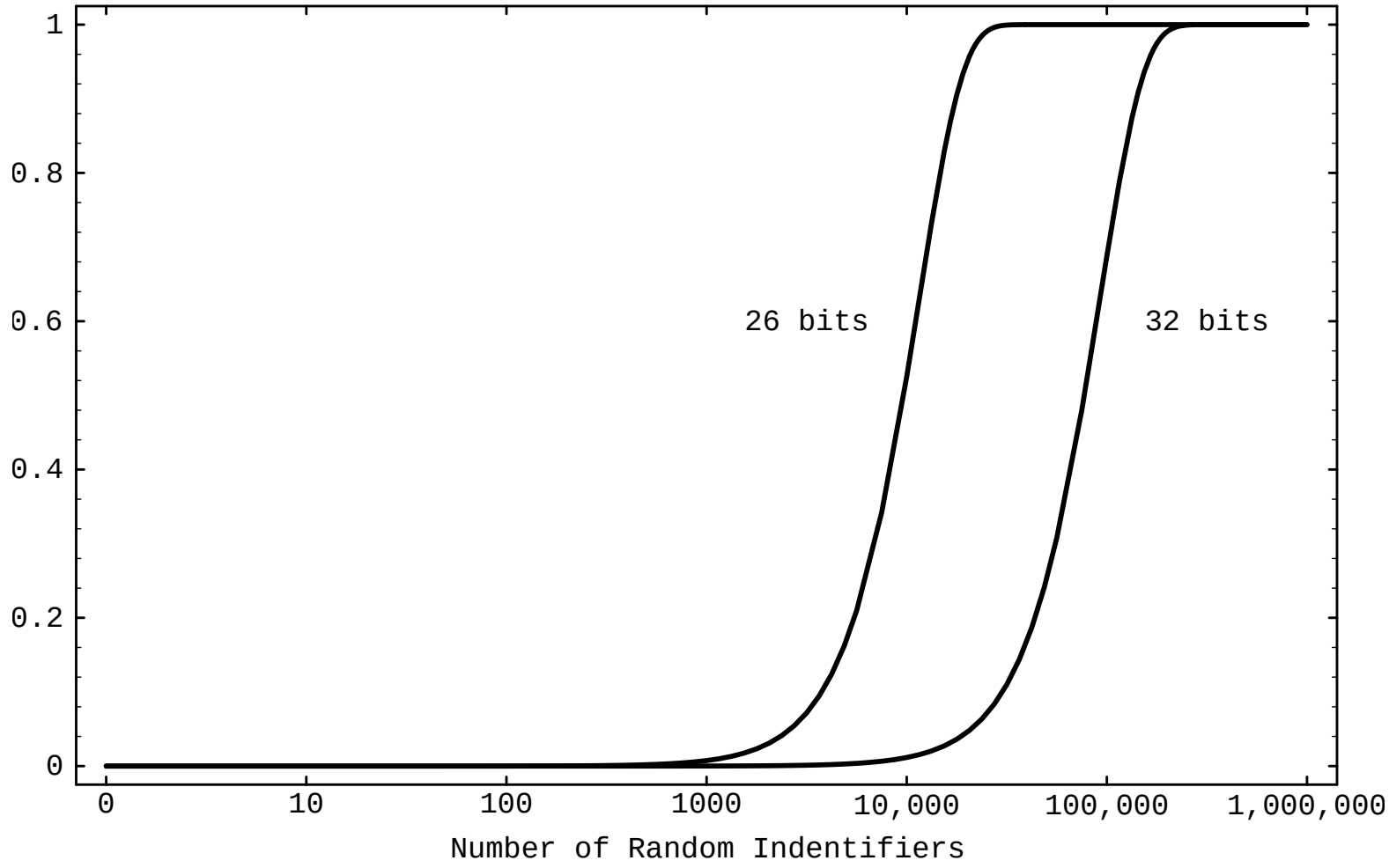
+ Loop detection is possible, but collisions look the same
+ No ID mapping by bridges and translators (this may be a requirement for some encryption or authentication schemes under the new option mechanism)
+ Easier for sources than B&T's to keep persistent storage
+ Shorter than truly globally unique [type,length,value] ID

Global source identifiers (3)

Disadvantages:

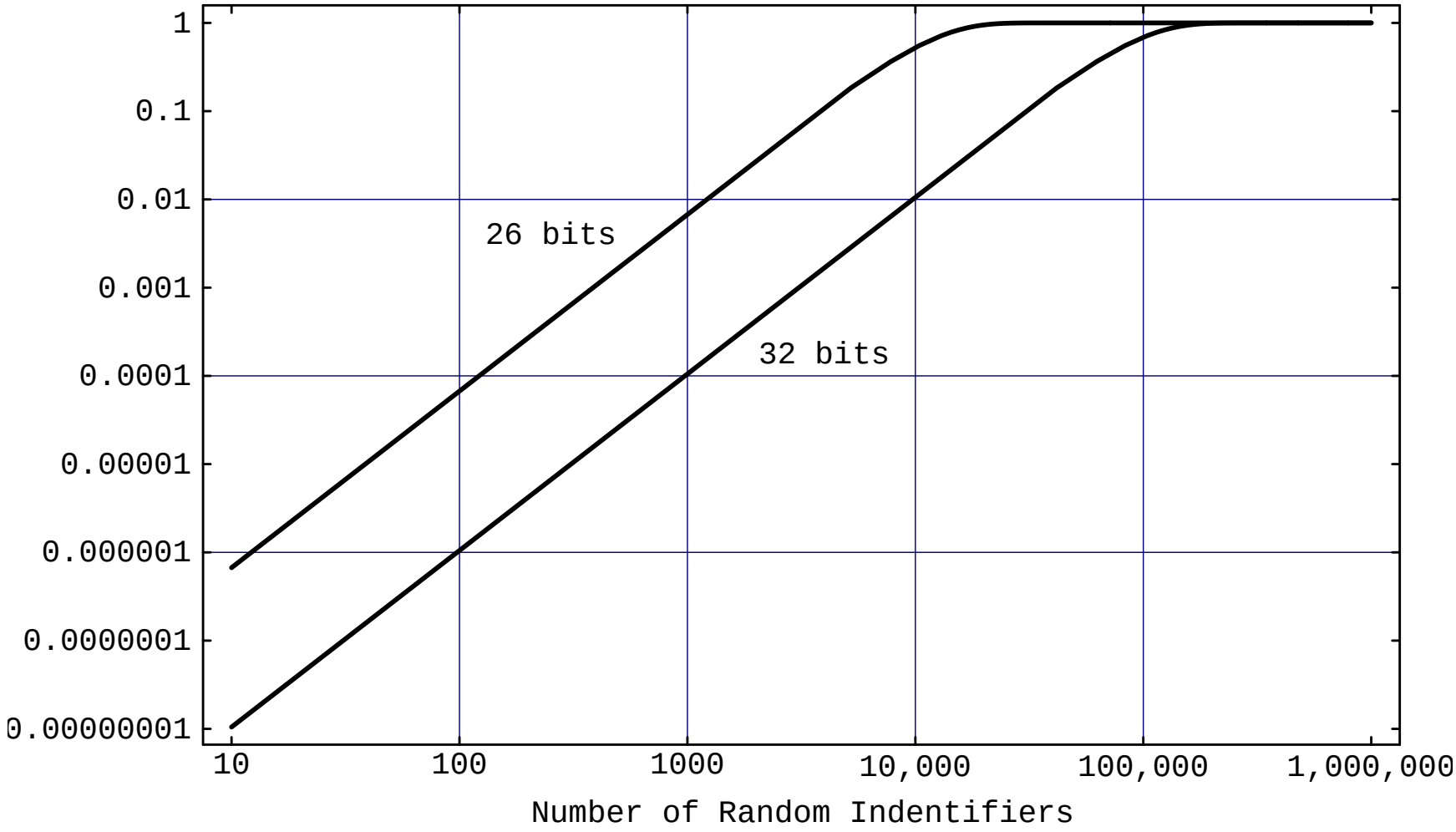
- Global identifiers are 32 bits instead of 16 for local
- You must be convinced by probability arguments
- Random allocation only scales to about 1K – 10K
  (but control packet rate backoff is similarly limited)
- Hosts behind a firewall that reuse IPv4 addresses would have to be configured to always use random identifiers
- Multiple sources on one host need means to coordinate
- Ties to IPv4 are cause for concern

Probability of Collision with Random Indentifiers



Thanks to Sally Floyd for probability analysis and assistance with Mathematica graphs

Log Probability of Collision with Random Indentifiers



Replace data options with a bit field

The [code,length] structure of RTP (data) options would be replaced by a bit field in the fixed header as follows:

- *Depends upon separate control/data ports and elimination of Channel ID*

CSRC (content source) — 5 or 6 bits to count the number of content source identifiers to follow, for packets produced by a bridge (mixer). Zero indicates packet not from bridge.

SSRC (synchronization source) — One bit to indicate that a sync source identifier follows.

CSRC count and SSRC bit could be combined into a "source ID" count if first ID in the list is sync source

Replace data options with a bit field (2)

BOS (beginning of sync unit) — Eliminated, but profile could define a similar field to be included in data section

APP (application-specific) — Application specific functions could be defined in profiles as extensions to the RTP header, but there is no provision for one implementation of an application to define its own optional information in a way that other implementations can simply ignore.

SDST (sync destination, or reverse-path option) — SSRC bit can be used for SDST on a reverse path packet because the directions are distinguished by the arrival port

Replace data options with a bit field (3)

ENC (encryption) — Two methods:

A. RTP header and data encrypted with constant IV; receiver depends upon header validity checks to detect decryption failure. SSRC would always have to be included since translators can't insert it; hard for translators to detect.

B. Fixed RTP header (64 bits) and SSRC field (if present) would not be encrypted in order to allow translators to insert or modify SSRC field (works with local identifiers). One bit in the header to indicate encryption. IV is either the first 64 bits of the RTP header, or an explicit value generated randomly for each packet, specified by out-of-band info.

Replace data options with a bit field (4)

MIC (authentication) — Two methods:

A. Authentication indicated by a bit in the header; message digest (with length?) appended to header. For locally unique identifiers, SSRC field is excluded from authentication. SSRC field must always be inserted, or SSRC flag bit must be excluded from authentication. Algorithm is out-of-band.

B. Authentication encapsulation header prepended to the RTP header and distinguished from RTP header by version number or some combination of bits. First word of the encapsulation should include a length field so receivers that don't care about authentication can skip it.

Option bit field allocations

Min	Max	
		Existing fields in fixed header:
1	2	version *(change to version 2-or-3)*
6	6	format
1	1	sync marker
8	16	sequence number
		New fields in fixed header:
5	6	CSRC count
0	1	SSRC present
0	1	encryption present
0	1	authentication present
21	34	(so 2 must be trimmed)

Advantages/disadvantages of bit field options

Advantages:

+ Can process options collectively as “packet format” *(seems unlikely for the options defined here)*
+ Bit field takes less space than [option code, length] if 32-bit global identifiers are used
+ More streamlined data processing
+ The order of the bits and the corresponding fields is fixed, but the receiver may process them in any order. This reduces the validation required.

  (For example, allows processing of SSRC first, but alternatively SSRC option could be required to be first)

Advantages/disadvantages of bit field options

Disadvantages:

- May not be any spare bits for new options

  (It may be argued that there just aren't that many variations that must be accommodated in the common part of supporting communication across a packet net, vs. application or media specific details which go later in the packet)
- Encryption and authentication key changes may not be indicated explicitly.

Control packets / Reception reports

Control packets carry RTCP information:

- Providing information about the sender, e.g., name
- Providing reception reports for all sources received
  - *Fundamental part of RTP transport functionality*
- Relating the sender's media timestamp to real time, and also marking the time of the reception reports
  - *Media timestamp must have sufficient resolution*

Control packet format (rough)

- Header (similar to data header, but need not be identical)
  - media timestamp
  - SSRC & CSRC info, encryption, authentication
- Sender info:
  - NTP timestamp (not required if unknown)
  - sender's packet count
  - sender's byte count
  - sender's name
- Reception reports
- Possibly additional options

Reception reports

- Reception info from all sources heard since last report, or round robin for as many as will fit (count and array of reception reports):
  - source identifier
  - count of packets received
  - count of bytes received
  - variance of interarrival time
  - last control packet timestamp from this source
  - delay time since that control packet was received

(last two items for round-trip propagation delay calculation)

Reception reports(2)

Differences from current QOS option specification:

- Absolute information rather than deltas so it does not matter if a report is missed
- Given two reception reports with timing information allows the counts to be translated into rates
- Minimum and maximum delay measures from RTP QOS option omitted because an outlyer can make the value useless as a description of the distribution

Control packet construction guidelines

Put more common information first so that application-independent monitors can process all the common information without having to know anything about the format of the application- or media-dependent information:

- sender info:
  - common
  - application dependent *(defined in profile)*
  - media dependent *(defined in profile)*
- reception report:
  - common, for all sources received
  - application dependent, for all sources received
  - media dependent, for all sources received

Additional RTCP options

These existing functions may need to be incorporated into the control packet structure in some way:

- **FMT (format description)** — Allows format codes to be defined dynamically if no higher−layer session protocol
- **BYE (goodbye)** — Source is terminating its participation in a session (could be a separate (trivial) control packet format)
- **APP (application−specific controls)** — Needed when multiple implementations of a single application interoperate but may have their own control information to communicate