INTERNATIONAL TELECOMMUNICATION UNION

**TELECOMMUNICATION STANDARDIZATION SECTOR**

STUDY PERIOD 2022-2024

**SG13-TD033/PLEN**

**STUDY GROUP 13**

**Original: English**

**Question(s):** 6/13 Geneva, 4 - 15 July 2022

**TD**

**Source:** Editors

**Title:** Draft new Recommendation ITU-T Y.3118 (formerly Y.IMT2020-jg-lsn): “Requirements and framework for jitter guarantee in large scale networks including IMT-2020 and beyond” – for consent



| | | |
|---|---|---|
| **Contact:** | Jinoo Joung<br>Sangmyung Univ.<br>Korea (Republic of) | Tel: +82-10-2086-0248<br>E-mail: jjoung@smu.ac.kr |
| **Contact:** | Chongho Yoon<br>Korea Aerospace Univ.<br>Korea (Republic of) | Tel: +82-10-2735-0124<br>E-mail: yoonch@kau.ac.kr |
| **Contact:** | Guosheng Zhu<br>Hubei University<br>China | Tel: +86-27 88666186<br>E-mail: zhuguosheng@hubu.edu.cn |
| **Contact:** | Taesang Choi<br>ETRI<br>Korea (Republic of) | Tel:+82-10-2740-5628<br>E-mail: choits@etri.re.kr |



**Abstract:** This draft Recommendation specifies the requirements and framework for jitter guarantee in large scale networks including IMT-2020 and beyond.

## Summary

With the base text TD036/WP1 (03-2022), this TD was created during Q.6/13 4-15 July 2022 SG13 Plenary meeting. The meeting agreed to accept proposals made by C74 with some revisions and editorial changes based on the discussions during the meeting, as the following:

- Summary reflects only the essential information on the Recommendation.
- The requirements are numbered.
- The contents in Clause 8 are reordered such that the three functional entities are explained one-by-one clearly.
- The whole document is checked with the author’s guide.

The meeting agreed to request for consent at this meeting.

# Draft Recommendation ITU-T Y.3118 (formerly Y.IMT2020-jg-lsn)

## Requirements and framework for jitter guarantee in large scale networks including IMT-2020 and beyond

**Summary**

This Recommendation specifies the requirements and framework for an effective and efficient solution of jitter guarantee for dynamic traffic with arbitrary input patterns in large-scale networks including IMT-2020 and beyond. The framework in this Recommendation is composed of the time-stamping and the buffering functions at the network boundary. It is scalable and does not rely on time synchronization or slot scheduling.

**Keywords**

Deterministic network, jitter, large scale network, latency, time synchronization.

**Table of Contents**

# Draft Recommendation ITU-T Y.3118 (formerly Y.IMT2020-jg-lsn)

# Requirements and framework for jitter guarantee in large scale networks including IMT-2020 and beyond

## 1 Scope

This Recommendation specifies the requirements and framework for jitter bound guarantee in large scale networks including the Internet, IMT-2020 networks and beyond, as follows:

- the requirements for achieving jitter bound guarantee in large scale networks;
- the framework to achieve the jitter bound guarantee effectively and efficiently.

Routing and upper layer functions lie outside the scope of this Recommendation.

## 2 References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-R M.1645] Recommendation ITU-R M.1645 (06/2003), *Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000*.

[ITU-R M.2083] Recommendation ITU-R M.2083-0 (09/2015), *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond*.

[ITU-T Y.3113] Recommendation ITU-T Y.3113 (2021), *Requirements and framework for latency guarantee in large scale networks including IMT-2020 network.*

## 3 Definitions

### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

**3.1.1 IMT-2020** [ITU-R M.2083]: Systems, system components, and related technologies that provide far more enhanced capabilities than those described in [ITU-R M.1645].

NOTE – [ITU-R M.1645] defines the framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000 for the radio access network.

### 3.2 Terms defined in this Recommendation

**3.2.1 large scale network**: A network or a set of interconnected networks, with diameter of 16 or larger, in which the numbers of flows and nodes are proportional to the diameter of the network.

## 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

| | |
|---|---|
| E2E | End-to-End |
| FIFO | First In First Out |
| QoS | Quality of Service |
| RTP | Real-time Transport Protocol |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |

## 5 Conventions

The keyword "is required to" indicates a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keyword "is recommended" indicates a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keyword "can optionally" indicates an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

## 6 Introduction

There are emerging applications that require latency and jitter bounds in large scale networks. For example, machine-to-machine communications for 'cloudified' industrial and robotic automation involves moderate-to-large scale networks. This type of communications requires very fine-grained timing accuracy for the dispersion of control commands and for the collection of data over wide area [b-ITU-T Y-Suppl.66]. ITU-T SG-13 has defined such services need to support critical grade reliability and extremely low as well as highly precise latency for the delivery of packets [b-ITU-T Y-Suppl.66].

In large scale networks, the input traffic is dynamic and the nodes operate asynchronously with each other. The large number of nodes over a geographically wide area are hard to be synchronized. The large number of flows makes a slotted operation undesirable as well. In order for guaranteeing a jitter upper bound in large scale networks, possibly over multiple administrative domains, a solution must be flexible and simple enough to be scalable.

This Recommendation specifies the requirements and framework for an effective and efficient solution of jitter bound guarantee for dynamic traffic with arbitrary input patterns in large scale networks including IMT-2020 and beyond.

## 7 Requirements

Req_1. It is required to specify the source, the destination, and the characteristics (e.g. average data rate, maximum burst size, etc.) of a flow.

Req_2. It is required to determine the latency upper bound of a flow within the network.

Req_3. It is recommended to determine the latency lower bound of a flow within the network.

Req_4. It is recommended to specify the requested end-to-end jitter upper bound of a flow.

Req_5. It is required to specify the network arrival times of packets belong to a flow. The clock of the specifying entity is not required to be time-synchronized with the other entities of the network.

Req_6. It is required to specify the network departure times of packets belong to a flow. The clock of the specifying entity is not required to be time-synchronized with the other entities of the network.

Req_7. It is required to hold a packet, at a functional entity located between the network boundary and the destination of a flow inclusive, for an interval based on information such as the network arrival times of the packets, the network departure times of the packets, the latency upper bound, and the requested jitter upper bound for a flow.

## 8 Framework

We consider the framework of guaranteeing the jitter bound in arbitrary sized networks with any type of topology, with random dynamic input traffic. The jitter is defined to be the latency difference of two packets within a flow, not a difference from a clock signal or from an average latency, as it is summarized in [b-IETF RFC 3393].

The framework is composed of

- a network that guarantees latency upper bounds;
- time-stampers for packets with clocks that are not necessarily synchronized with the other nodes, which reside between source and network ingress interface inclusive;
- and buffers that can hold the packets for predetermined intervals, which reside between destination and network egress interface inclusive.

Figure 1 depicts the overall framework architecture. Only a single flow is depicted between the source and the destination in Figure 1. The arrival ($a_n$), departure ($b_n$), and buffer-out ($c_n$) times of $n^{th}$ packet of a flow are denoted. The end-to-end (E2E) latency and the ‘E2E buffered latency’ are defined to be ($b_n$-$a_n$) and ($c_n$-$a_n$), respectively.

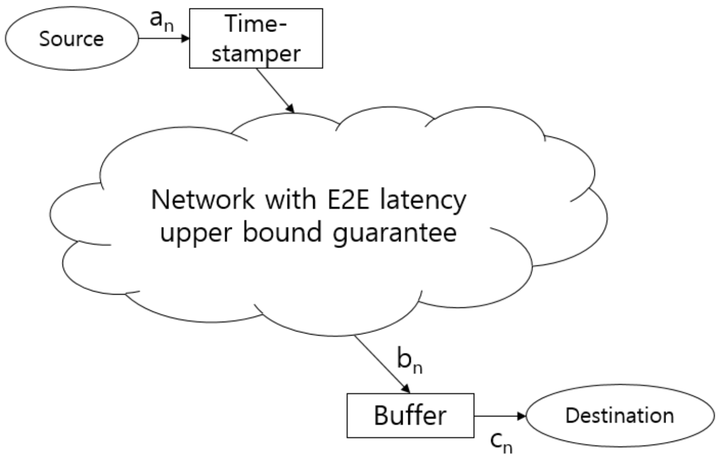


**Figure 1 - The framework architecture for jitter bound guarantee**

The buffer supports as many as the number of the flows destined for the destination. The destination in Figure 1 can be an end station or a deterministic network. The buffer holds packets in a flow according to predefined intervals. The decision of the buffering intervals involves the time-stamp value within each packet.

The network in between the time-stamper and the buffer can be of arbitrarily sized network, including the Internet, IMT-2020 network, and beyond. The input traffic can be dynamic. It is required, as in Req_2, the network be able to guarantee and identify the latency upper bounds of the flows. The network is also required to let the buffer be aware of the upper bounds of the flows it has to process. ITU-T Y.3113 specifies the framework for latency upper bound guarantee in large scale networks [ITU-T Y.3113]. It is recommended that the latency lower bound information is provided by the network as well. The lower bound may be contributed from the transmission and propagation delays within the network.

The time-stamper marks on the packets their arrival times. One may use the time-stamping function in the real-time transport protocol (RTP) over user datagram protocol (UDP), or the transmission control protocol (TCP). Either the source or the network ingress interface may stamp the packet. In the case that the source stamps, the time-stamp value is the packet departure time from the source, which is only a propagation time away from the packet arrival time to the network. Note that the time-stamper clock and the buffer clock are not required to be time-synchronized. All we need to know at the buffer is the differences between the time-stamps, i.e. the information about the inter-arrival times.

Let the arrival time of $n^{th}$ packet of a flow be $a_n$. Similarly let $b_n$ be the departure time from the network, of $n^{th}$ packet. Then $a_1$ and $b_1$ are the arrival and departure times of the first packet of the flow, respectively. The first packet of a flow is defined to be the first packet generated by the source, among all the packets that belong to the flow. Further let $c_n$ be the buffer-out time of the $n^{th}$ packet of the flow. Let $g_n$ be the processing delay within the buffer of the $n^{th}$ packet of the flow. The $g_n$ includes but not limited to the time interval to look up the timestamp and to store/forward the packet. However, it does not include the intentional buffer-holding interval. By definition, $c_n - b_n \geq g_n$. Let $\max_n g_n = g$, the maximum packet processing delay in the buffer. It is assumed that a buffer can determine the value of $g$.

Let m be the jitter control parameter, which has a value between W and U, W ≤ m ≤ U. The rules for the buffer holding interval decision are given as follows:

- The buffer holds the first packet with the interval $(\mathrm{m} - \mathrm{W})$, for some m, W ≤ m ≤ U. The buffer-out time of the first packet $c_1$ is then $(b_1 + \mathrm{m} - \mathrm{W})$.

- The buffer holds n[th] packet until the time max $\{g + b_n, c_1 + (a_n - a_1)\}$, for any n > 1.

The second rule states that a packet should be held in the buffer to make its inter-buffer-out time $(c_n - c_1)$ equals to the inter-arrival time $(a_n - a_1)$. However, when its departure from the network is too late, thus the inter-buffer-out time should be larger than the inter-arrival time, then just pass the buffer as soon as possible, i.e. $c_n = g + b_n$.

The jitter between packets i and j is defined as $|(c_i - a_i) - (c_j - a_j)|$. By following the above rules, the upper bounds of both the E2E buffered latency and the jitter are guaranteed. The theorems regarding the guaranteed bounds can be found in Appendix I. The proofs for the theorems can be found in [b-Joung-2021] and [b-Joung-2022].

With the framework defined in this Recommendation, any requested jitter bounds, including zero jitter, can be achieved by adjusting the parameter *m*, while still guaranteeing a latency bound.

## 9 Security Considerations

The QoS management of IMT-2020 network includes user equipments, access networks, and core network that are subject to security and privacy measures. Sensitive information should be protected as a high priority in order to avoid leaking and unauthorized access. Security and privacy concerns should be aligned with the requirements specified in [b-ITU-T Y.2701] and [b-ITU-T Y.3101].

# Appendix I

## The theorems for the latency & jitter bounds and their usage

(This appendix does not form an integral part of this Recommendation.)

This appendix provides three theorems for the latency and jitter bounds guaranteed by the rules, and an example usage applying the rules.

Let the arrival time of $n^{th}$ packet of a flow be $a_n$. Similarly let $b_n$ be the departure time from the network, of $n^{th}$ packet. Then $a_1$ and $b_1$ are the arrival and departure times of the first packet of the flow, respectively. The first packet of a flow is defined to be the first packet generated by the source, among all the packets that belong to the flow. Furthermore, let $g_n$ be the processing delay within the buffer of the $n^{th}$ packet of the flow. The $g_n$ includes but not limited to the time interval to look up the timestamp and to store/forward the packet. However, it does not include the intentional buffer-holding interval. By definition, $c_n - b_n \geq g_n$. Let $\max_n g_n = g$, the maximum packet processing delay in the buffer. It is assumed that a buffer can determine the value of $g$.

Let m be the jitter control parameter, which has a value between W and U, $W \leq m \leq U$. The rules for the buffer holding interval decision are given as follows:

- The buffer holds the first packet with the interval $(m - W)$, for some m, $W \leq m \leq U$. The buffer-out time of the first packet $c_1$ is then $(b_1 + m - W)$.
- The buffer holds $n^{th}$ packet until the time $\max\{g + b_n, c_1 + (a_n - a_1)\}$, for any $n > 1$.

The second rule states that a packet should be held in the buffer to make its inter-buffer-out time ($c_n$-$c_1$) equals to the inter-arrival time ($a_n$-$a_1$). However, when its departure from the network is too late, thus the inter-buffer-out time should be larger than the inter-arrival time, then just pass the buffer as soon as possible ($c_n = g + b_n$).

The buffer requires information on W, U, $b_1$, $c_1$, $b_n$, and $(a_n - a_1)$. W is required to be informed by the network. The knowledge about $b_1$ and $b_n$ are obtained by the buffer with its own clock. The buffer is required to keep the record of the time $c_1$. Finally the time difference $(a_n - a_1)$ can be calculated from the difference of the time-stamps of the packets, which have been added at the source or the network ingress interface. Note that the time-stamper clock does not have to be synchronized with the buffer clock, i.e. the buffer does not need to know the exact values of $a_n$ or $a_1$. The implementation of the second rule is also feasible since $\max\{g + b_n, c_1 + (a_n - a_1)\}$ is greater than or equal to $g + b_n$, the packet departure time of the $n^{th}$ packet from the network plus the maximum processing delay in the buffer, by definition.

Since the time-stamping and the buffering functions defined in this Recommendation are implemented at the boundary of a network, the complexity does not increase as the network size increases.

The buffer has to be able to identify the first packet of a flow, in order to identify the time $b_1$ and the relative time-stamp values representing $a_1$. If a first-in first-out (FIFO) property is guaranteed in the network, then this is trivial. Otherwise, this is achievable by a flag at the header, indicating the packet is the first packet. A sequence number written in the packet header, such as the one in the RTP, would work as well. In networks without the FIFO property guarantee, if some of the earlier packets (e.g. $2^{nd}$ or $3^{rd}$ packets of a flow) arrive to the buffer sooner than the first packet, than they will be buffered until the first packet's buffer-out time plus the additional interval, as specified by the rule.

Figure I-1 depicts the relationships between the arrival, departure, and buffer-out times of packets of the flow under observation. The following theorems hold.

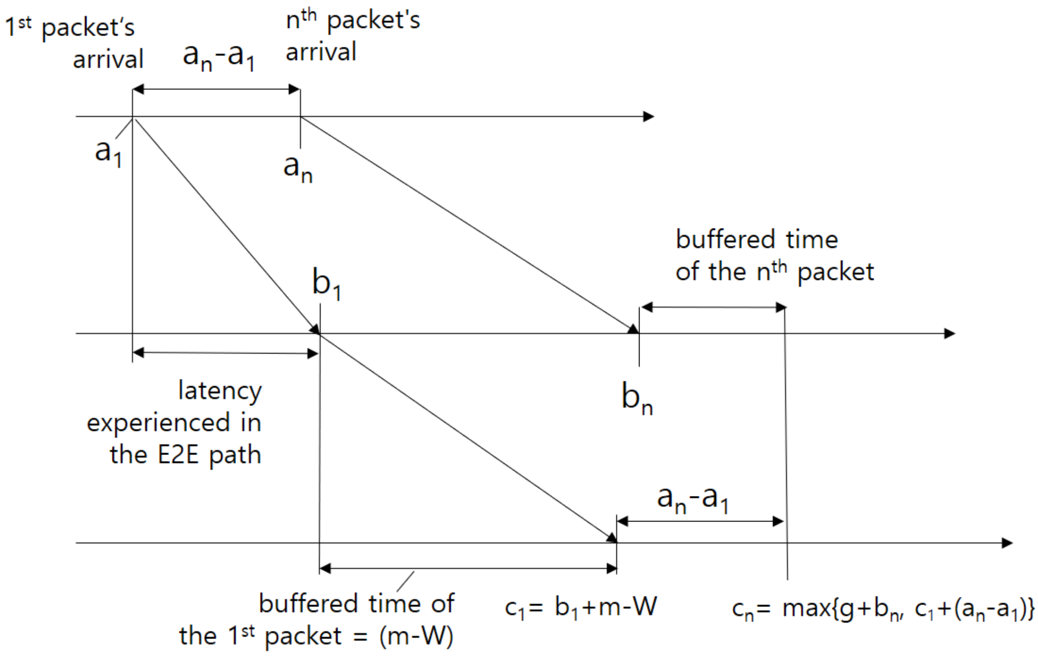


**Figure I-1 - Relationship of packets' arrival ($a_n$), departure ($b_n$), and buffer-out ($c_n$) times**

**Theorem 1** (Upper bound of the E2E buffered latency). The latency from the packet arrival to the buffer-out times, $(c_n - a_n)$, is upper bound by (m+U-W).

**Theorem 2** (Lower bound of the E2E buffered latency). The latency from the packet arrival to the buffer-out times, $(c_n - a_n)$, is lower bounded by m.

**Theorem 3** (Upper bound of the jitter). The jitter is upper bounded by $(\mathrm{U} + \mathrm{g} - \mathrm{m})^{+}$.

The jitter between packets i and j is defined as $|(c_i - a_i) - (c_j - a_j)|$. $x^{+}$denotes $\max(0, x)$. The proofs for the theorems can be found in [b-Joung-2021] and [b-Joung-2022].

For example, assume that we have a flow requesting the E2E buffered latency bound of 10ms, and the jitter bound 1ms. Then from Theorem 1, it is required that (m+U-W) = 10ms, and from Theorem 3, it is also required that (U+g-m) = 1ms. Assume that g is twice the packet transmission time, 1µs for 1000bit length packet at 1Gbps link. Therefore g = 2µs. Based on these equations we obtain U ≈ 5.5ms+W/2 and m ≈ 4.5ms+W/2. As such, during the call setup process, upon the flow's requested specifications, the network and the buffer may assign U = (5.5ms+W/2) for the E2E network latency upper bound, and m = (4.5ms+W/2) parameter for the buffering. If W is not known, then one can assume W = 0.

As an extreme case, if one wants to achieve an absolute synchronization, i.e. the inter-departure times of the output packets $(c_n - c_{n-1})$ are exactly the same with the inter-arrival times $(a_n - a_{n-1})$, then one may set the jitter to be equal to zero. In this case we can achieve this synchronization by setting m = U, i.e. by holding the first packet long enough that the E2E buffered latency certainly exceeds the U. Then the E2E buffered latency upper bound becomes 2U-W, which is close to 2U when W is negligible. Note that letting m=U does not always mean the E2E buffered latency becomes 2U. It is just the upper bound in the worst case in which the first packet had already experienced the latency U in the network.

# Bibliography


[b-ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[b-ITU-T Y.3101] Recommendation ITU-T Y.3101 (2018), *Requirements of the IMT-2020 network*.

[b-ITU-T Y-Suppl.66] ITU-T Y-series Recommendations – Supplement 66, (2020), *ITU-T Y.3000-series Network 2030 services: Capabilities, performance and design of new communication services for the Network 2030 applications*.

[b-IETF RFC 3393] IETF RFC 3393 (2002), *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM).*

[b-Joung-2021] Joung, J. and Kwon, J. (2021) *Zero Jitter for Deterministic Networks without Time-synchronization*, IEEE Access, Vol. 9, March, 2021.

[b-Joung-2022] Joung, J., Kwon, J., Ryoo, J., and Cheung, T. (2022) *Asynchronous Deterministic Network based on the Diffserv Architecture*, IEEE Access, Vol. 10, Jan., 2022.