Draft Recommendation ITU-T Y.IMT2020-jg-lsn

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

Summary

There are emerging applications that require latency and jitter bounds in large scale networks. Machine-to-machine communication 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 telemetry data over wide area [ITU-T Y.Sup66].

IEEE TSN task group (TG) defines a set of solutions for latency and jitter minimization. The solutions rely on time-synchronization and slot scheduling at every node in the network.

The slot scheduling is a resource consuming task. When the network is large, it is quite a burden to the network operator. For dynamic environments where critical flows appear and disappear over time, runtime reconfigurations are necessary, which makes the scheduling problem even harder.

Moreover, the time-synchronization requirement across every node in the network has two difficulties. First, time synchronization function implementation may impose too much overhead to lightweight embedded nodes. Second, the synchronization accuracy may not up to the level of traffic requirements, especially in networks with a large number of hops.

With all the effort mentioned above, however, it is yet to come up a solution that can guarantee a jitter bound in general networks with dynamic traffic with arbitrary input pattern.

Therefore, this Recommendation specifies the requirements and framework for an effective and efficient solution for jitter guarantee in large-scale networks.

Keywords

Jitter, latency, deterministic network, large scale network, time synchronization

Table of Contents

Draft Recommendation ITU-T 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 guarantee in large scale network as follows:

- The requirements for achieving jitter bound guarantee in large scale networks, such as the Internet, IMT-2020 networks, and beyond.
- Overall framework and functional entities, and their interworking to achieve the jitter bound guarantee, effectively and efficiently.

Routing and upper layer functions are out of scope of this Recommendation. If necessary, the document will, instead, reference the existing works appropriately.

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 E.800] Recommendation ITU-T E.800 (09/2008), *Definitions of terms related to quality of service*.

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

[ITU-T Y.Sup66] Recommendation ITU-T Y.Sup66 : ITU-T Y.3000-series - Network 2030 services (2020), *Capabilities, performance and design of new communication services for the Network 2030 applications*.

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.

3.1.2 customer premises equipment [ITU-T E.800]**:** Telecommunications equipment located at the customer installation on the customer side of the network interface.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 large scale network: A network or a set of networks, whose longest end-to-end path includes 16 or more relay nodes.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AN Access Network

CN Core Network

CPE Customer Premises Equipment

DN	Data Network
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
UE	User Equipment

5 Conventions

None.

6 Introduction

There are emerging applications that require latency and jitter bounds in large scale networks. 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 [ITU-T Y.Sup66]. 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 [ITU-T Y.Sup66].

In moderate-to-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 for jitter bound guarantee in moderate-to-large scale networks, with dynamic traffic with arbitrary input pattern.

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 desired end-to-end jitter upper bound of a flow.

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

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

Req_7. It is required to hold a packet, at a point between including the network boundary and the destination of a flow, for an interval based on information such as the network arrival instances of the packets, the network departure instances of the packets, the latency upper bound, and the desired 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 RFC 3393 [b-IETF RFC 3393].

The framework is composed of

- a network that guarantees latency upper bounds;
- a time-stamper for packets with a clock that is not necessarily synchronized with the other nodes, which resides between including the source and the network ingress interface;
- and a buffer that can hold the packets for a predetermined interval, which resides between including the destination and the network egress interface.

Figure 1 depicts the overall architecture of the framework. 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) instances 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.

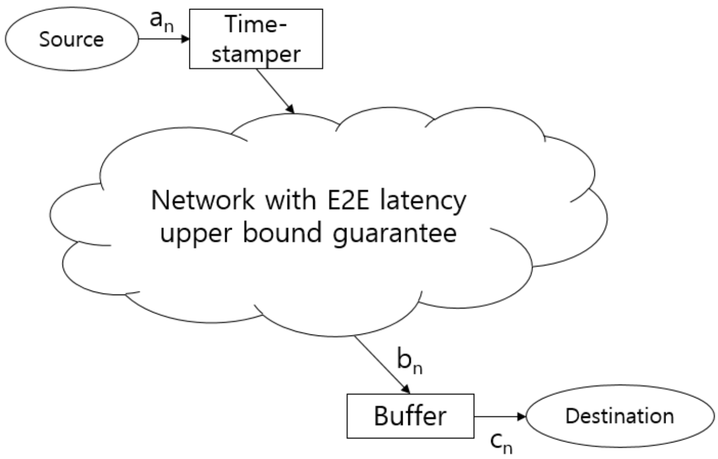

Figure 1. The framework for jitter bound guarantee

The buffer supports as many as the number of the flows destined for the destination. In cases where the buffer is not suitable to be placed within an end station, the network can attach the buffering function at the boundary. The destination in Figure 1 can also be a small deterministic network.

There is an entity for time-stamping the arrival instances to the packets. 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 instance from the source, which is only a propagation time away from the packet arrival instance to the network. Note that the source and the destination need not to share a synchronized clock. All we need to know is the differences between the time-stamps, i.e. the information about the relative arrival instances.

A latency upper bound of the flow is guaranteed by the network. ITU-T Y.3113 specifies the framework for latency upper bound guarantee [ITU-T Y.3113]. It is recommended that a latency lower bound information is provided by the network. The lower bound may be contributed from the transmission and propagation delays within the network. The buffer holds packets in a flow according to predefined intervals. The decision of the buffering intervals involves the time-stamp within each packet.

Let the arrival instance 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 instances 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. Let's define m to 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 (m - W), for some m, W ≤ m ≤ U. The buffer-out instance of the first packet c_1 is then $(b_1 + \mathrm{m} - \mathrm{W})$.
- The buffer holds n^{th} packet until the instance max $\{b_n, c_1 + (a_n - a_1)\}$, for any n > 1.

The second rule implies that a packet should be held in the buffer to make its inter-buffer-out time (c_n- c_1) equal 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 ($c_n = b_n$).

The following theorems hold.

Theorem 1 (Upper bound of the E2E buffered latency). The latency from the packet arrival to the buffer-out instances, $(c_n - a_n)$, 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 instances, $(c_n - a_n)$, is lower bounded by m.

Theorem 3 (Upper bound of the jitter). The jitter is upper bounded by (U-m).

The proofs for the theorems can be found in [b-Joung]. The jitter between packets i and j is defined as $|(c_i - a_i) - \left(c_j - a_j\right)|$.

The three theorems state that with the framework, any desired 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 UE, ANs, and CN 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 buffering interval decision rules, their implementation feasibility, and the bounds from the framework.

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

This appendix provides the detailed description of the buffering interval decision rules, their feasibility for implementation, three bounds guaranteed by the rules, and finally an example scenario applying the rules.

Let the arrival instance 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 instances 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. Let's define m to be the jitter control parameter, which has a value between W and U, $W \le m \le U$. 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, $W \le m \le U$. The buffer-out instance of the first packet c_1 is then $(b_1 + \mathrm{m} - \mathrm{W})$.
- The buffer holds n^{th} packet until the instance max $\{b_n, c_1 + (a_n - a_1)\}$, for any n > 1.

The second rule implies that a packet should be held in the buffer to make its inter-buffer-out time (c_n- c_1) equal 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 ($c_n = b_n$).

The buffer requires information on W, U, b_1 c_1, b_n, and $(a_n - a_1)$. W, U are required to be informed by the network. The knowledge about b_1 and b_n are easily obtained by the buffer with its own clock. The buffer is required to keep the record of the time instance 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 written at the source. Note that the source 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 $\{b_n, c_1 + (a_n - a_1)\}$ is greater than or equal to b_n, the packet departure instance of the n^{th} packet from the network, by definition.

The buffer has to be able to identify the first packet of a flow, in order to identify the instance b_1 and the relative time-stamp values representing a_1. If a first-in first-out (FIFO) property is guaranteed in

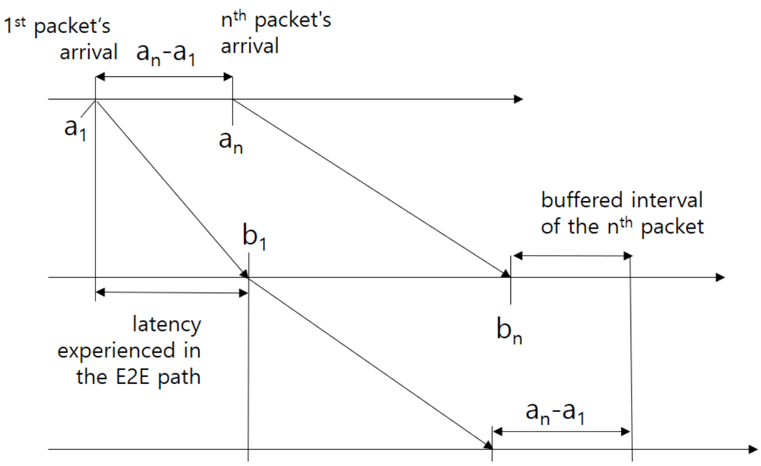

Figure I-1. Relationship of packets' arrival (a_n), departure (b_n), and buffer-out (c_n) instances.

the network, then this is trivial. Otherwise, this is achievable by a flag at the header, indicating the packet is indeed 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. 2nd or 3rd 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 plus the additional interval, as specified by the rule.

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

Theorem 1 (Upper bound of the E2E buffered latency). The latency from the packet arrival to the buffer-out instances, $(c_n - a_n)$, 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 instances, $(c_n - a_n)$, is lower bounded by m.

Theorem 3 (Upper bound of the jitter). The jitter is upper bounded by (U-m).

The proofs for the theorems can be found in [b-Joung]. The jitter between packets i and j is defined as $|(c_i - a_i) - \left(c_j - a_j\right)|$.

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, (m+U-W) = 10ms, and from Theorem 3, (U-m) = 1ms. 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 actual 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 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-IETF RFC 3393] IETF RFC 3393 (2002*)*, *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM).*

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

____