Applied Science and Innovative Research
ISSN 2474-4972 (Print) ISSN 2474-4980 (Online)
Vol. 7, No. 2, 2023
www.scholink.org/ojs/index.php/asir

Original Paper

TSN-Based Automotive E/E Architecture

Yang Hu[1]

[1] Civil Aviation Electronic Technology Co., Ltd., Chengdu, China

Received: March 9, 2023 Accepted: March 24, 2023 Online Published: March 28, 2023

doi:10.22158/asir.v7n2p60 URL: http://doi.org/10.22158/asir.v7n2p60

Abstract

Time-Sensitive Networking, also known as TSN, is a deterministic network based on traditional Ethernet. It offers a bunch of standards or profiles specified by IEEE 802.1 task group which has been evolved from the former IEEE802.1 Audio Video Bridging task group. In Automotive Industry, especially in ADAS domain, TSN backbone communication will gradually merge with or even replace the traditional in-vechile communication like CAN/CANFD/LIN/MOST/FlexRay due to below properties, it plays a key bridge role in heterogeneous SOC communication network.

Keywords

TSN, automotive, ADAS, SOC, network

1. Introduction

TSN is much more useful in domain based E/E architecture. In the future, however the E/E architecture tends to become centralized + partial domain, but TSN still can have its place in it. In this chapter, we would like to propose the architecture model based on those two types of E/E architectures. Anyway, every car is different and OEMs & Tier 1s will be the end responsible about how TSN is deployed inside the vehicles.

2. System Context

2.1 Domain Based Architecture

Below system diagram shows the typical domain based E/E architecture (No TSN redundancy). It includes central gateway domain, ADAS+IC domain, IVI domain and other domain like powertrain domain and chassis domain. The GW with TSN related protocols integrated becomes TSN backbone network among domains. The clock source will be provided by GNSS and the NTP could be the source backup. In order to use TSN, all NICs must support hardware timestamp, with gPTP help, the time will be synchronized through all the domains. Radio sensors can either be connected directly to SOC or handled by sensor ECU (preferred). Sensor ECU can also do the time sync through CAN.

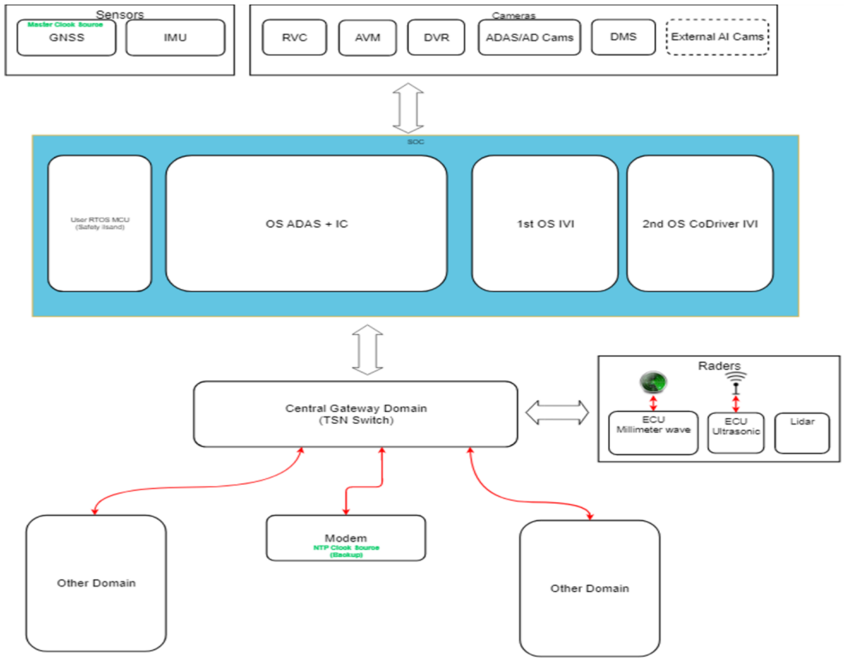


Figure 1. Domain Based E/E Architecture

2.2 Centralized + Partial Domain Architecture

Below system diagram shows the future centralized based E/E architecture (No TSN redundancy). ADAS+IVI+IC domains are in one SOC which would communicate with other domains through TSN GW (Switch). With regard to the SOC internal communication, there could be two proposals, one is to create virtual network (vNIC) for each domain, another one is to still depend on the TSN backbone network. Below diagram is the latter.

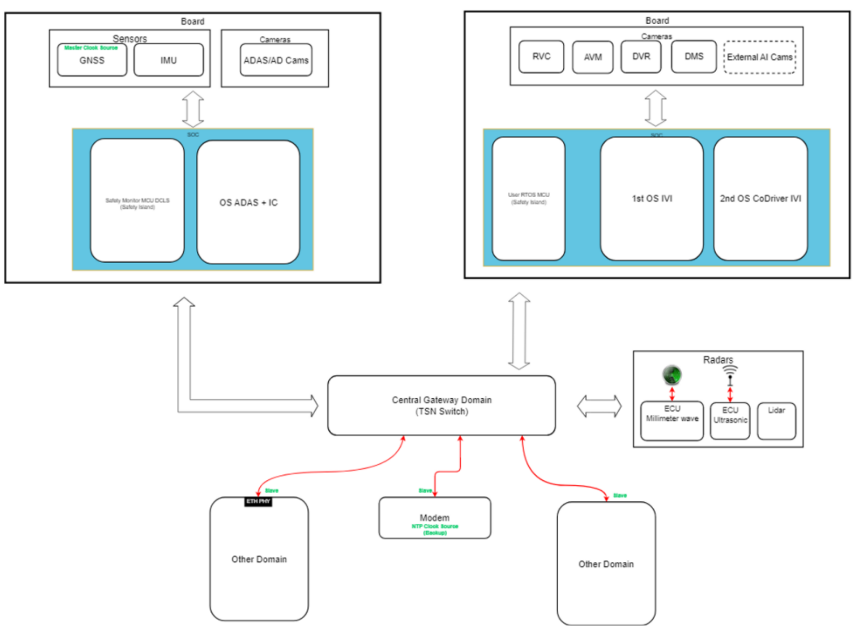


Figure 2. Centralized Based E/E Architecture

Published by SCHOLINK INC.

3. Software Context

Below diagram shows the TSN software stack diagram. Considering this is heterogeneous SOC, OS is not the same, the A core is not the same either. Each domain TSN related software is probably different. Apart from SWITCH domain software, The gPTP server should be in each domain as the bridge for time sync.

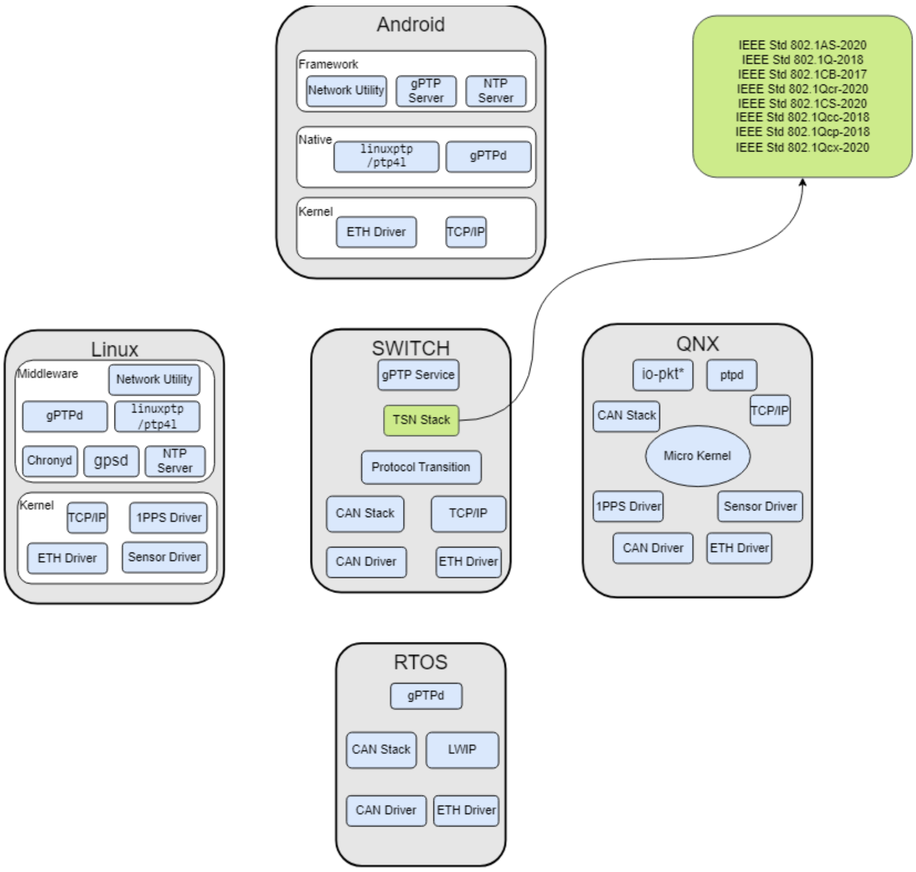


Figure 3. TSN Software Stack

QNX - As time clock source master, this domain should get the accurate time from GNSS and sync it to system clock. QNX has its own ptpd to do the sync.

Linux - This domain can also be the time source master in IVI, so GNSS driver is also needed. The open source linuxptp supports the TSN time sync protocol IEEE 802.1AS.

Android - This OS may not need the TSN in application level, NTP server is just enough, however we can also provide a gPTP server.

RTOS - This domain may need gPTP server to sync time for CAN data.

SWITCH - This device which should be provided by TSN SWITCH vendor will provide the TSN protocol mplementation.

4. Solution

TSN protocols overview:

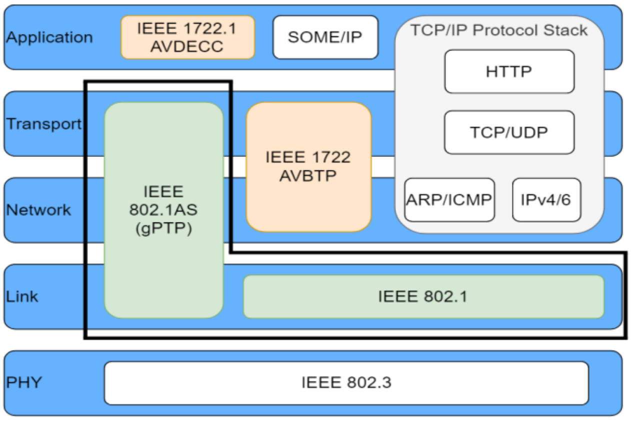


Figure 4. TSN Protocol

4.1 Time Synchronization

IEEE Std 802.1AS-2020 is the standard of gPTP, which is the base for the entire TSN network. Normally, the implementation is as a service daemon at user space or middleware process in each domain. All devices hardware MUST support TSN like hardware timestamp in PHY. The gPTP works like this:

- Choose a clock as master clock (GNSS) in on domain. In Automotive field, the master clock should be chosen statically in advance, no BMCA needed (Maybe in redundancy usage for backup master clock).
- Master clock will broadcast local time to the other domain periodically.
- The other domain will synchronize their own local clock as slave clock.

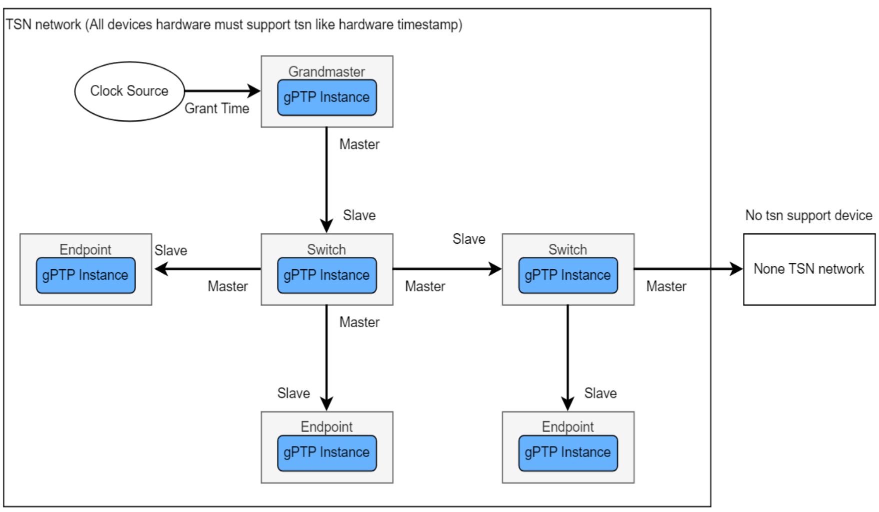


Figure 5. Time Synchronization

Synchronization shall include:

Clock time sync - simple, just send the local timestamp to slave.

Clock frequence sync - because not all the crystal oscillator frequencies are same, so all the slave clock frequency should align with master clock.

When master sends the local time to slave, the delay and rateratio should be considered. Below diagram shows the time synchronization algorithm (assume the residence time is ignored):

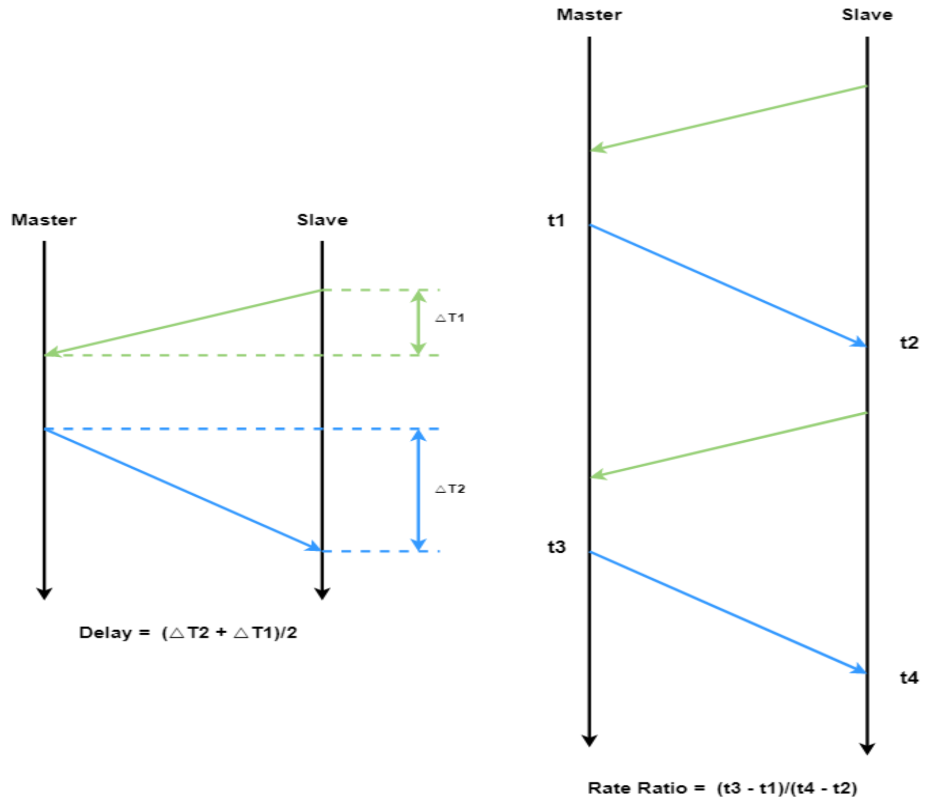
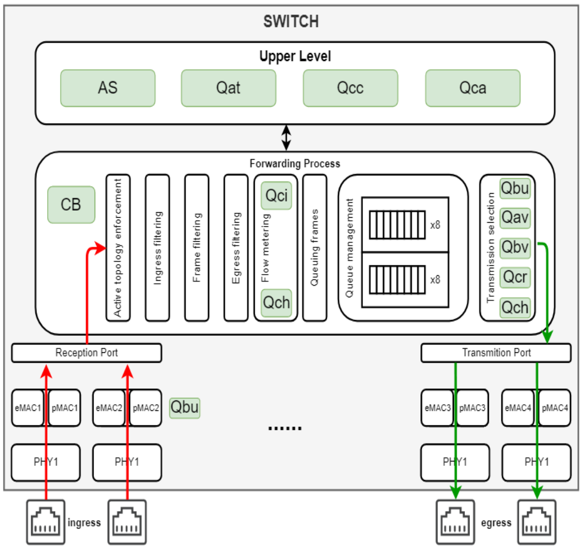


Figure 6. Time Synchronization Algorithm

4.2 Bounded Low Latency

In order to make the data frame transportation with low latency and jitter over the network, TSN provides some protocols, working along with gPTP to assure the data frame transportation deterministic. As you can see in below diagram, this part should be implementeted in SWITCH:



AS	gPTP
Qat	Stream Reservation Protocol
Qcc	Stream Reservation Protocol (SRP) Enhancements
Qca	Path Control and Reservation
CB	Frame Replication and Elimination for Reliability
Qci	Per-Stream Filtering and Policing
Qch	CQF
Qav	CBS
Qbv	TAS
Qcr	ATS
Qbu	Frame Preemption

Figure 7. TSN on Switch

You might need have some network basic knowledge like shaper, token, queue, etc.

The most important part is that TSN offers some optional shapers which we can choose in different situation, choose them properly will improve the QoS significantly:

- 802.1Qav CBS

This is Real-time scheduling modeling for AVB. AVB traffic (SR_A and SR_B) is shaped through CBS (Credit-based Shaper) to ensure transmission delay in AVB network; in addition, there are BE messages in AVB network. Strict Priority Queuing (SPQ) scheduling algorithm is adopted to transmit the message.

- 802.1Qbv TAS

This is Real-time scheduling modeling for TSN. Time-Aware Shaper (TAS), which assigns the Gate Control List (GCL) for each queue of each out-port to achieve data scheduling. The data starts to be transmitted only when the queue door is opened. This shaper is appropriate for automotive control system and it needs to be applied in each domain, furthermore both master and slave time clocks MUST be synced or it won't work, so gPTP is very important for this protocol.

- 802.1Qch CQF

Cyclic Queuing and Forwarding protocol (), working together with 802.1Qci or 802.1Qbv, is a method of traffic shaping that can deliver deterministic, and easily calculated, latency for time-sensitive traffic streams. Only two queues, Q1 and Q2. Q1 sends frame and Q2 receives frame in cycle1; Q1 receives frame and Q2 sends frame in cycle2.

- 802.1Qcr ATS (Asynchronous Traffic Shaping)

No time synchronization needed. It includes:

1. Smoothen traffic patterns by reshaping per hop.

2. Prioritize urgent traffic over relaxed traffic.

Published by SCHOLINK INC.

- 802.1Qbu Frame Preemption Protocol

Technically this is not shaper, but it would coordinate the other shapers. By modifying preamble, the out-port that supports frame preemption mechanism can provide two MAC service interfaces: the preemptible MAC (pMAC) service interface and the express MAC (eMAC) service interface.

4.3 Reliability

We consider the reliability from redundancy and security point of view.

In order to improve the reliability of systems, TSN proposes fault-tolerance protocols (802.1CB and 802.1Qca) to guarantee the path redundancy and data seamless redundancy. The above protocols are combined to improve reliability by transmitting redundant replicas in network in parallel on disjoint paths; In other words, TSN mainly improves reliability through information redundancy (frame replicas) and link redundancy (multiple paths).

- 802.1CB Frame Replication and Elimination for Reliability

802.1CB provides the redundant mechanism of data transmission to solve the problems (e.g., frame loss and frame error) caused by network failures; the mechanism is mainly divided into two steps to achieve redundancy:

1. Frame replication.
2. Frame elimination.

- 802.1Qca Path Control and Reservation

802.1Qca calculates the data transmission path by combining the Intermediate Station to Intermediate Station protocol and the Shortest Path Routing (SPR) method. The redundant mechanism requires ring node topology, so star topology can not support this.

- 802.1AS Master Clock Redundancy part (Need BMCA to support)

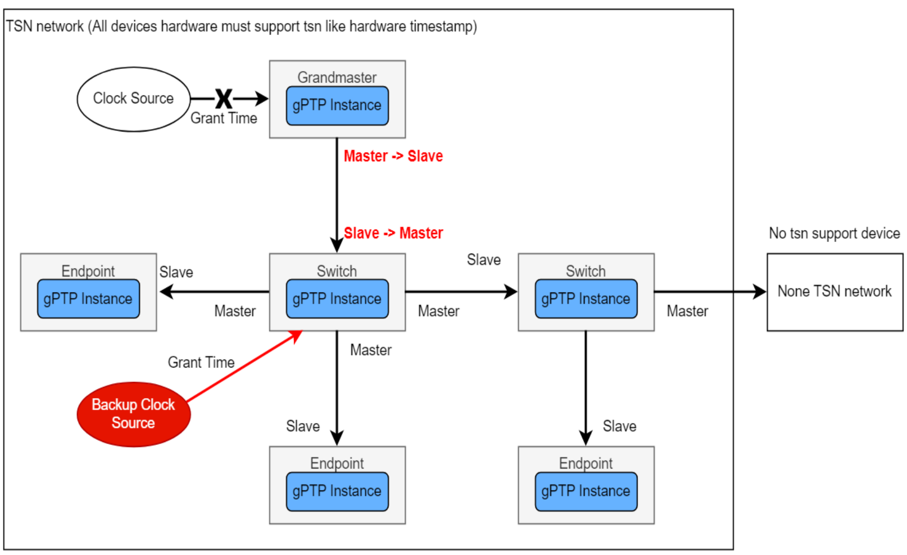


Figure 8. TSN Redundancy

Published by SCHOLINK INC.

- 802.1Qci Per-Stream Filtering and Policing

TSN proposes 802.1Qci to provide stream filtering and policing function for messages arriving at the in-port of bridge to prevent traffic overload, thereby improving network security. The process occurs after data stream arrives at the in-port of bridge and before it queues at the out-port. 802.1Qci can solve network attacks and traffic overload problems. For example, stream filtering and policing functions will control the data stream by limiting the transmission bandwidth to improve the security of network when a DDoS network attack occurs in the network.

4.4 Resource Management

Since Automotive In-Vehicle tends to do the static configuration, resource related protocol may not be our concern.

Stream Reservation Protocol: Qat, Qcc

- 802.1Qat Stream Reservation Protocol (SRP)
- 802.1Qcc Stream Reservation Protocol (SRP) Enhancements and Performance Improvements

5. Conclusion

It has been a trend to apply TSN technology to the automotive field. With the increasing number of network applications on the vehicle platform, TSN protocol is becoming more and more important. This paper discusses the current popular on-board E/E architecture, and analyzes the feasibility of relevant protocols and applications in its architecture, such as time synchronization, low latency, reliability and resource management protocols.

References

Alderisi, G., Iannizzotto, G., & Bello, L. L. (2012). Towards IEEE 802.1 ethernet AVB for advanced driver assistance systems: A preliminary assessment. In *Proceedings of the 2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (pp. 1-4).* https://doi.org/10.1109/ETFA.2012.6489775

Atallah, A. A., Hamad, G. B., & Mohamed, O. A. (2020). Routing and scheduling of time-triggered traffic in timesensitive networks. *IEEE Transactions on Industrial Informatics*, *16*(7), 4525-4534. https://doi.org/10.1109/TII.2019.2950887

Gavriluţ, V., Zhao, L., Raagaard, M. L., & Pop, P. (2018). AVB-aware routing and scheduling of time-triggered traffic for TSN. *IEEE Access*, *6*, 75229-75243. https://doi.org/10.1109/ACCESS.2018.2883644

IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks. (2018). I*EEE Std 802.1Q-2018*.

Kleineberg, O., Fröhlich, P., & Heffernan, D. (2012). Fault-tolerant Audio and Video Bridging (AVB) Ethernet: A novel method for redundant stream registration configuration. In *Proceedings of the*

2012 IEEE 17th International Conference on Emerging Technologies Factory Automation (pp. 1-8). https://doi.org/10.1109/ETFA.2012.6489645

Raagaard, M. L., & Pop, P. (2017). Optimization Algorithms for the Scheduling of IEEE 802.1 Time-Sensitive Networking (TSN). Technical Report. Technical University of Denmark. Retrieved December 10, 2020, from http://www2.compute.dtu.dk/~paupo/publications/Raagaard2017aa-Optimization%20algorithms%20for%20th-.pdf

Thiele, D., Ernst, R., & Diemer, J. (2015). Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers. In *Proceedings of the 2015 IEEE Vehicular Networking Conference (pp. 251-258)*. https://doi.org/10.1109/VNC.2015.7385584

Umadevi, K. S., & Sridharan, R. K. (2017). Multilevel ingress scheduling policy for time sensitive networks. In *Proceedings of the 2017 International Conference on Microelectronic Devices, Circuits and Systems (pp. 1-4)*. https://doi.org/10.1109/ICMDCS.2017.8211725

Zhao, M. X., Yu, J.-J., Li, W.-T., Liu, D., Yao, S. W., Feng, W., She, C. Y., & Tony QS Quek. (2020). Energy-aware offloading in time-sensitive networks with mobile edge computing. *arXiv:2003.12719*. Retrieved from https://arxiv.org/abs/2003.12719