On Multi-Level Preemption in Ethernet

Mubarak Adetunji Ojewale, Patrick Meumeu Yomsi and Geoffrey Nelissen
CISTER Research Centre, ISEP, Polytechnic Institute of Porto, Portugal
Email: {mkaoe, pamyo, grrpn}@isep.ipp.pt

***Abstract*—Ethernet is increasingly being considered as the solution to high bandwidth requirements in the next generation of timing critical applications that make their way in cars, planes or smart factories to mention a few examples. Until recently, ethernet frames used to be transmitted exclusively in a non-preemptive manner. That is, once a frame starts transmitting on a switch output port, its transmission cannot be interrupted by any other frame until completion. This constraint may cause time critical frames to be blocked for long periods of time because of the transmission of non-critical frames. The IEEE 802.3br standard addressed this issue by introducing a one-level ethernet frame preemption paradigm. In this approach, frames transmitted through a switch output port are classified as express frames or preemptable frames, depending on their priority levels. Express frames can preempt preemptable frames and two frames belonging to the same class cannot preempt each other. While this partially solves the problem for express frames, all preemptable frames can still suffer blocking irrespective of their priority level. In this work, we investigate the feasibility and advantages of multi-level preemptions in time-sensitive ethernet networks.**

I. Introduction

Most systems today are made of several embedded devices interconnected through networks. Cars, planes, train and factories for instance contain tens to hundreds of sensors, actuators and computers that must communicate with timing guarantees. Real-time applications require responsiveness i.e., timely and correct reaction to events, which largely depends on the ability of data to move in a predictable manner on the network. Ethernet is the emerging communication technology in industrial and automotive domains. Its relatively cheaper price and its high bandwidth capacity make it the ideal replacement for previous communication infrastructures generally adopted in these domains. However, the legacy ethernet standard was mainly targeting non real-time applications and desirable capabilities like preemption, global time synchronisation across the network, frame duplication and re-transmission were initially missing. In order to provide system designers with these desirable features, several modifications have been made to the standards over the years. The IEEE 802.1p task group, for example, introduced a mechanism to specify a Class of Service (CoS) for ethernet frames in order to expedite the transmission of high priority frames [1]. The CoS of a frame signifies its priority and the frames are transmitted according to their CoS, highest priority first. Other features like time-triggered transmission, global clock synchronisation, credit based shaping, among others, have also been added to the ethernet to make it more suitable for real-time applications [2].

One modification made to the ethernet to support real-time communication is reported in the IEEE 802.3br and 802.1Qbu standards, which specify a frame preemption protocol for ethernet networks. Preemption implies that a frame that has already started its transmission on a switch output can be suspended in order for a more "urgent frame" to be transmitted through the same port. The transmission of the preempted frame is resumed only after the urgent frame has been fully transmitted. Preemption allows a high priority frame with stringent timing requirements to be transmitted more promptly, but this is achieved at the cost of some overheads. Fig. 1 illustrates a scenario where a high priority frame is transmitted by preempting a low-priority one. Upon the occurrence of each preemption, the standards specify some additional information to be added to the preempted frames so as to notify the network devices about the preemption, thereby impacting the transmission link utilization.

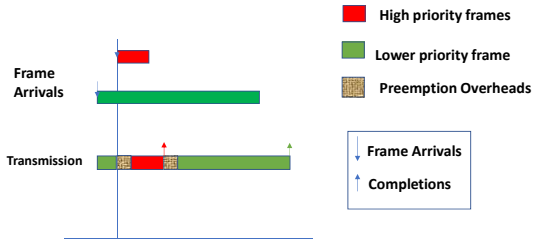

Fig. 1: Illustration of frame preemption.

Before the specification of the IEEE802.3br standard, ethernet frames used to be transmitted in a non-preemptive manner [3]. Any low-priority frame could block any high priority frame for long periods of time depending on the low-priority frame's size. It is to circumvent this limitation that frame preemption was defined in the IEEE 802.1Qbu [4]. This standard specifies one-level preemption for ethernet frames since only two MAC service interfaces are supported: a preemptable MAC (pMAC) interface and an express MAC (eMAC) interface. Frames assigned to the eMAC service interface are referred to as *express* frames and those assigned to the pMAC interface as *preemptable* frames.

A critical look at the one-level preemption, however, raises some concerns. With the current specification, only express frames are allowed to preempt preemptable frames and frames of the same class cannot preempt each other [2]. In typical real-time applications, there are traffic classes that are not classified as express but nevertheless have timing constraints and should not be blocked for long periods by lower priority frames. To illustrate this, consider a medium priority frame (black frame in Fig. 2). With one-level preemption, the

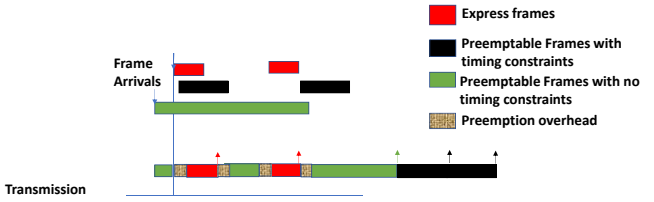

Fig. 2: Frame transmission under 1-level preemption.

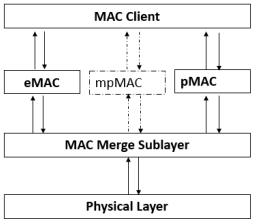

Fig. 3: The MAC merge sublayer managing service interfaces.

medium priority frame, which may be important to the smooth operation of a real-time application, can be blocked by a large lower priority frame (green frame) if both share the pMAC interface. This is because a preempted frame must complete its transmission before any other non-express frame can be transmitted. Consequently, this limitation has a negative effect on medium priority frames that are not uncommon in real-time applications. Note that if the medium priority frame was instead classified as express, then it could block more urgent frames, thereby defeating the whole purpose of introducing express frames in the first place.

Most work in the literature on this topic have been studying the effect of frame preemption on worst-case end-to-end transmission delays. The authors in [5] and [3], for example, showed that frame preemption reduces the transmission delays of express traffic significantly, but it has adverse effect on preemptable traffic. Thiele and Ernst [3] presented a Compositional Based Analysis (CPA) to provide guarantee on the end-to-end transmission delay of ethernet traffic with one-level preemption under Standard Ethernet and Time Sensitive Networking (TSN). To the best of our knowledge, no work has investigated the feasibility of multiple preemption levels on ethernet networks, especially for the scheduled traffic that are non-express, but with stringent timing constraints. In this work, we consider three levels of priorities and 2 levels of preemption to investigate the feasibility of multi-level preemption in ethernet.

II. Problem Statement

We consider a network traffic consisting of n streams $s_1, s_2, \ldots, s_n$ partitioned into traffic classes: *express traffic*, *medium priority preemptable traffic* (mpFrames) and *Best Effort preemptable traffic* (bpFrames). Stream s_i, with $i \in [1, n]$, consists of a potentially infinite number of frames s_i^j $(j \geq 1)$ with an inter-arrival time of at least T_i units between two consecutive frames. Frame s_i^j is characterised by its arrival time a_i^j and its size c_i^j. Express traffic frames have very strict timing requirements. Preemptable traffic frames are divided

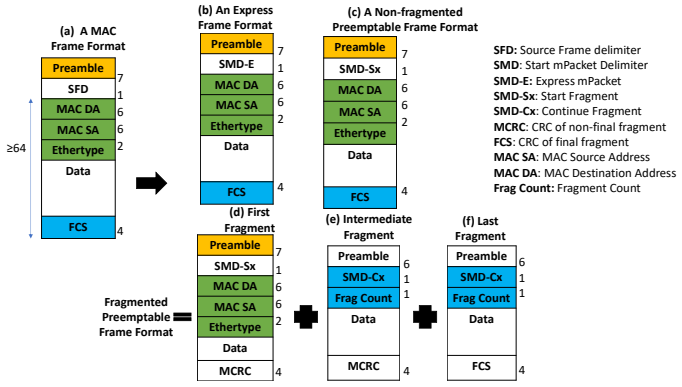

Fig. 4: Ethernet frame formats as specified in IEEE 802.3 Standards. Numbers represent sizes of each field (in bytes)

into two classes: mpFrames with strict timing constraints yet not as stringent as those of express traffic frames and the bpframes with no timing constaints at all. We assume that each stream is assigned a unique fixed priority and all frames generated from this stream inherit its priority. We also assume that any express frame has a higher priority than all preemptable frames and the following constraints are enforced:

- ▷ Any express frame can preempt all preemptable frames,
- ▷ Any mpFrame has higher priority than all bpFrames and therefore, can preempt these,
- ▷ Frames from the same class cannot preempt each other.

With the above assumptions and contraints, we investigate the feasibility of supporting multi-level preemption to limit the blocking of mpFrames by bpFrames.

III. Preemption in Ethernet Networks

Preemption occurs at the MAC merge sublayer, which is between the physical and the MAC layers (See Fig. 3). Frames at this sublayer are called *mFrames*. The sublayer may preempt a preemptable mFrame currently being transmitted and may also prevent it from starting its transmission citeStandard802.3br-2016. Before each mFrame transmission, the sublayer verifies if the next switch/node supports preemption by performing a verification operation (see citeStandard802.3br-2016, page 42 for details). Preemption capability is enabled only after the verification operation confirms that it is supported. When this is the case, additional information are added to the mFrame headers, describing its preemption characteristics. In addition, it is important to preserve the ethernet frame format when mFrames are preempted. IEEE 802.3br ensures this by defining mFrame formats in a preemption enabled environment. Fig. 4 shows that express frames (see Fig. 4b) differ from normal MAC frames (see Fig. 4a) by only 1 octet, referred to as "Start Frame Delimiter" (SFD) by replacing the MAC frame SFD with "Start Mframe Delimiter-Express" (SMD-E) in the frame format. In practice, the SFD and SMD-E have the same value. Similarly, a preemptable frame that is not preempted (see Fig. 4c) differs from a normal MAC frame only in that the SFD is replaced with "Start MFrame Delimiter Start Fragment" (SMD-Sx). When a frame is preempted, the

first fragment of the frame differs from a non preempted preemptable frame only in that the error checking code (FCS) of the fragment is replaced with a newly generated mFrame error checking code (mCRC) by the MAC merge sublayer (see Fig. 4d). All other fragment headers only contain a preamble, "Start Mframe Delimiter for Continuation fragment" (SMD-Cx) and frag_count (see Fig. 4e) to track subsequent fragments. The last fragment ends with the FCS of the original preempted frame (see Fig. 4f).

At the receiving node, a Medium Independed Interface (xMII) inspects the SMD for each frame upon arrival. The value of the SMD indicates whether the received frame is express or preemptable [1]. Express frames (containing SMD-E) are processed by an Express Filter and preemptable frames by a "Receive processing" construct. Receive processing ensures that fragments of a preempted frame are received completely and in correct order using the mCRC and the frag_count. The mCRC is computed such that all fragments of a preempted frame end with the same mCRC, except the last one which ends with the original frame FCS. Frag_count is used to monitor the correct order of frame arrivals and to detect missing frames. A mismatch in the mCRC after a sequence of arrival of fragments indicates the end of the reception of the preempted frame, i.e., the last fragment has been received and the frame transmission is complete.

IV. Feasibility of Multi-level Preemption and Recommendations

If an mFrame containing SMD-Sx (signalling the start of the transmission of a new preemptable frame) arrives at a node and Receive processing has not completed the reception of a previous preempted frame, Receive processing ensures that the MAC detects a "FrameCheckError" in the partially received frame (see [1], page 44). This mechanism implies that the node can detect the start of another preemptable frame, which is important to support multi-level preemption. Although the start of another preemptable frame would be flagged as an error, the IEEE 802.3br standard states that other techniques may be employed to respond to an incomplete frame transmission as long as the MAC behaves as though a FrameCheckError occurred. This submission opens the door to multi-level preemption specification, while still conforming to the standard. To this end, we recommend the specification of a mechanism to ensure the transmission of a frame in a higher preemption class without jeopardizing the integrity of the preempted frame. This operation should be performed such that the receiver node/switch correctly resumes the reception of the preempted frame later on.

The standards do not describe any mechanism to reassemble more than one frame in a buffer. We recommend the specification of such a mechanism to enable multi-level preemption as the buffer must be able to correctly reassemble and transmit a second frame, while already containing fragments of a first frame. In addition, the xMII that separates express frames from preemptable frames can be configured to distinguish between different priority levels for preemptable frames. As such, no additional frame filtering mechanism would be required for multi-level preemption.

We believe that the current preemptable frames format in the standards [6] is sufficient to handle multi-level preemption. To this end, we recommend that new values be defined for the one octet SMD contained in the header (See Fig. 4) to support more preemption levels. The standard currently defines eleven values for this octet. Additional values can be defined to check the level of preemption supported by the next node and to indicate the frame preemption levels.

We believe that a switch node supporting multi-level preemption can interoperate with those supporting only one-level or no preemption at all. With the new recommended SMD values, the MAC merge sublayer will be able to verify if the next node supports preemption and if this is the case, how many levels are supported. If just one level of preemption is supported, then all preemptable frames are transmitted on a single pMAC interface and multi-level preemption is disabled. In this case, all non-express frames are treated as preemptable frames and will not preempt each other. In the case preemption is not supported at all, frames are transmitted as already specified in the IEEE 802.1Q standards.

V. Expected Results and Conclusion

At this stage of this Work-in-Progress, we examined the feasibility of multi-level preemption in ethernet networks and provided a set of recommendations. Now, we seek to develop a formal worst case transmission delay analysis of frames assuming multi-level preemption and conduct experiments to demonstrate its effectiveness in time sensitive ethernet networks. An improvement is expected for medium priority frames with affordable preemption overhead in terms of buffer size and SMD definitions.

Acknowledgment

This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) and co-financed by ERDF (European Regional Development Fund) under the Portugal2020 Program, within the CISTER Research Unit (CEC/04234).

References

[1] "IEEE standard for local and metropolitan area networks–bridges and bridged networks," *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)*, pp. 1–1832, Dec 2014.

[2] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. Elbakoury, "Ultra-low latency (ULL) networks: A comprehensive survey covering the IEEE TSN standard and related ULL research," *CoRR*, vol. abs/1803.07673, 2018.

[3] D. Thiele and R. Ernst, "Formal worst-case performance analysis of time-sensitive ethernet with frame preemption," in *2016 IEEE 21st ETFA*, Sept 2016, pp. 1–9.

[4] "IEEE approved draft standard for local and metropolitan area networks-media access control (MAC) bridges and virtual bridged local area networks amendment: Frame preemption." *P802.1Qbu/D3.1, September 2015*, pp. 1–50, Jan 2015.

[5] W. K. Jia, G. H. Liu, and Y. C. Chen, "Performance evaluation of ieee 802.1qbu: Experimental and simulation results," in *38th Annual IEEE Conference on Local Computer Networks*, Oct 2013, pp. 659–662.

[6] "IEEE standard for ethernet amendment 5: Specification and management parameters for interspersing express traffic," *IEEE Std 802.3br-2016 (Amendment to IEEE Std 802.3-2015)*, pp. 1–58, Oct 2016.