# FlexTDMA for Delay-Stable Communications in Asynchronous Industrial Control Networks

David A. Miller and Ahmed E. Kamal
Department of Electrical and Computer Engineering, Iowa State University, Ames IA 50011

*Abstract*—**To support closed industrial control networks we consider constant end-to-end delay with minimal delay-jitter support in an asynchronous network. The Rate Constrained Static Priority rate-jitter (RCSP-RJ) minimizes jitter while not providing full restoration of traffic at each switch. RCSP-delay-jitter (-DJ) provides jitter free network communication but requires a means of time coordination between switches. We introduce FlexTDMA that allows nearly constant delay bounds with minimal delay-jitter in an asynchronous network. FlexTDMA uses a token bucket to restore the arrival envelope of each flow. This eliminates delay-jitter, simplifies analysis of the network and avoids cyclic dependent issues in network analysis. A network of FlexTDMA switches operate asynchronously and offer nearly constant delay and minimal delay-jitter. FlexTDMA adapts the functional characteristics of RCSP-RJ while providing the performance characteristics of RCSP-DJ, through the use of periodic baselining (transmission at maximal delay bound) of each flow at each switch. When a flow has been baselined the receiving switch establishes a reserved time slot in which the next frame arrival is expected (thus the term flexible TDMA). The value of a baseline event degrades relative to switch clock drift rates. Infrequent baseline collisions cause delay-jitter, and baseline transmission opportunities (TO) can be scheduled thereby minimizing the probability of alignment.**

## I. Introduction

THERE is a need in industrial control networks, which are mission-oriented local networks, for frame forwarding techniques that offer a fixed delay to a subclass of traffic [7]. Industrial control networks have a portion of the exchanged bandwidth dedicated to support tight inter-process control and control loop communication. There are three types of latency related service descriptions the network can offer: ***1)*** Low delay bound on communicated traffic, ***2)*** Minimal delay-jitter, which is the extent of compression between any two arriving frames [2], and ***3)*** Stable delay bound value independent of the other network utilizations.

Other packet scheduling schemes that attempt to minimize delay-jitter, suffer from either requiring inter-switch clock coordination (i.e. RCSP-DJ) (rather than asynchronous operation), or maintain a fixed priority so that the highest priority flows must contend without regard to past frame arrival treatment (i.e. RCSP-RJ). The result is that asynchronous switching networks are restricted to use of RCSP-RJ which can not guarantee maximal delay bounds.

Techniques utilizing an 'earliness timestamp' [8] are avoided to preserve the fault independence of switches as each switch is functionally dependent on the fault-free operation of predecessor switches. Industrial control switching network systems will benefit from FlexTDMA when the complexity of system level synchronization is unacceptable, but the component switches must operate fault independently. FlexTDMA does not require asynchronous clock coordination, and limits the affect of a switch fault within the network.

We introduce FlexTDMA scheduling policy which is intended to offer nearly RCSP-DJ service in an asynchronous network, i.e., without switch coordination. This is accomplished by a periodic maximally delayed frame transmission on each flow (called baselining). This allows a RCSP-RJ policy to closely conform to the RCSP-DJ performance properties.

The FlexTDMA function value is isolated to limited closed industrial networks with well defined system engineering delay requirements established to insure a functional system.

This paper is organized as follows. Section 2 gives some background, and Section 3 details the proposed FlexTDMA policy and its properties. Section 4 discusses the performance of FlexTDMA, followed by conclusions in Section 5.

## II. Background

### A. Network Model and Assumptions

This paper considers a network of switches that operate independently with no clock coordination. Each switch has no knowledge of the internal clock timing of any adjacent switch, and there is no direct way to validate the timing of each received frame other than evaluation of the received traffic envelope of each flow to the switch. Additionally there is no 'earliness timestamp' exchanged between switches.

### B. Policing and Regulation

Arriving traffic must be constrained to a defined traffic envelope [2, 4]. This can be accomplished by restoring the arriving traffic envelope through traffic shaping [6]. This eliminates delay-jitter restoring each flow to the initial transmission definition at the network ingress [2, 3]. This allows per switch analysis of delay bound and avoids classic cyclic dependency analysis issues [5]. The reshaping algorithm restores each flow to have frames separated by the initial minimum frame transmission interval [3]. When a flow is bounded by a rate, *r*, maximum burst size, *b,* and traffic envelope

$$A^*(I) = r \cdot I + b \tag{1}$$

within time *I*, reshaping of the flow does not increase the maximal delay bound from the ingress of the previous switch packet scheduler to the egress of the reshaping process in the current switch [3, 5]. The holding time in the regulator is limited to the ahead-of-time transmission in the previous switch packet scheduler. A re-shaping regulator is used to determine the eligibility time (ET) of a frame so that the flow conforms to the initial traffic definition [1, 2, 3].

### *C. Delay-Jitter Regulator*

[2, 3] establish the definition of a delay-jitter regulator. The values $ET_j^k$ and $AT_j^k$ are the ET and arrival times of frame k at switch j, respectively. The initial condition

$$ET_0^k = AT_0^k \tag{2}$$

assumes the initial arrival sequence to the first switch is conformant to the delay-jitter free traffic envelope. Under RCSP-DJ $ET_j^k$ is recursively calculated as

$$ET_j^k = ET_{j-1}^k + d_{j-1} + \pi_{j-1,j} \tag{3}$$

$d_{j-1}$ is the delay bound of eligible frames on this flow at the previous switch j-1. $\pi_{j-1,j}$ is the transmission time on the media from switch j-1 to switch j.

### *D. Full Restoration Following Regulation*

[2, 3] establish the ET, under RCSP-DJ, of two successive frames arriving to switch j which can be represented as $ET_j^{k+1} - ET_j^k = ET_{j-1}^{k+1} - ET_{j-1}^k$ by rearranging equation (3). By inductive application of (2) $ET_j^{k+1} - ET_j^k = AT_0^{k+1} - AT_0^k$. Therefore the initial arrival pattern to the first switch is fully restored at each switch. This is only possible assuming that the ET in the previous switch $ET_{j-1}^k$ is known, which is not possible in asynchronous systems where clock timing is not exchanged.

## III. FlexTDMA

### *A. Restoration Based on ET in Previous Switch*

The regulation equation (3) is based on knowing the three right side terms of the equation. The value $\pi_{j-1,j}$ is fixed at $P_k/l_j$, the frame k size divided by the line rate of switch j. The value $d_{j-1}$ is fixed in a static configuration based on the delay bound of the packet scheduler supporting the flow containing frame k in switch j-1. $ET_{j-1}^k$ is not directly known to switch j. The value $ET_{j-1}^k$ is known only within the clock domain of switch j-1.

### *B. Reimplementation of Regulation Using Token Bucket*

Recall that $\pi_{j-1,j}$, the transmission time from switch j-1 to switch j, is fixed. All the bits of the frame will arrive to switch j exactly $\pi_{j-1,j}$ after transmission was initiated at switch j-1 so $ET_j^k - ET_{j-1}^k = d_{j-1} + \pi_{j-1,j}$. The time from $ET_{j-1}^k$ to $ET_j^k$ is composed of $W_{j-1}^k$, the actual wait time in the packet scheduler of switch j-1, the transmission time $\pi_{j-1,j}$, and $H_j^k$, the actual holding time within the regulator of switch j so that $ET_j^k - ET_{j-1}^k = W_{j-1}^k + \pi_{j-1,j} + H_j^k$.

Thus $H_j^k = d_{j-1} - W_{j-1}^k$.

When using a token bucket as a regulator (*limits arrivals to* (1) *during any interval I*), the value $W_{j-1}^k$ can only be detected as the difference in arrival times of two successive frames to switch j of $AT_j^k$ and $AT_j^{k-1}$. The exact jitter applied to frame k can be detected when frame k-1 was delayed the maximal amount. Once a maximally delayed frame is received from switch j-1, the $ET_j^k$ will be computed as $\max\left(AT_j^k, ET_j^{k-1} + X_{\min} \cdot \left(1 - clockDrift_{ppm}\right)\right)$, where $X_{\min}$ is the minimum frame inter-arrival time on the flow.

### *C. Flow Baselining*

As each switch operates with an independent clock, when a token bucket is used at each switch (for independence) there is no way to know the delay time through the previous scheduler. Here we introduce the concept of a 'baselined' flow.

Definition *Baselined flow: A flow on which a frame has been recently transmitted at the delay bound, and each frame has been received before its eligibility time.*

A flow for which the current ET has passed with no frame receipt is considered not baselined. A flow is considered 'baselined' since subsequent switches experience the maximal relative receipt time. A newly baselined frame will cause a maximal series of ETs since the frame is transmitted at the maximal delay bound. When a flow is baselined, most frames on the flow can be transmitted with any delay less than the flow delay bound. Each switch tracks the baselined status of each flow. When switch i transmits a frame on flow j at the deadline time for the packet scheduling algorithm the flow is considered baselined at switch i. Subsequent frames transmitted by switch i are held at switch i+1 a duration so that the maximum delay bound is enforced.

When a flow is not baselined by transmitting with a maximum delay at the scheduler of a switch, the delay bound can range from $\Pi$ to $\sum_{i=1}^{n-1} d_i + \Pi$, where $\Pi$ is the sum of the transmission times at all switches. The delay-jitter is limited to the wait time in the packet scheduler of the final switch [2, 3]. A frame is received to a baselined flow can be transmitted at any time between the ET of the flow and the deadline.

### *D. Virtual Flow for Baselining*

A virtual flow $flow_{01}$ is added to the packet scheduler at priority level-0 (highest) for allocation of high priority TO for baselining of flows utilizing the packet scheduler.

When $flow_{01}$ is included in the flow set of the packet scheduler the RCSP schedulability equation becomes:

$$\left\lceil d_m / X_{\min_{0,1}} \right\rceil \cdot P_{0,1} + \sum_{k=1}^{m} \sum_{j=1}^{i_k} \left( \left\lceil d_m / X_{\min_{k,j}} \right\rceil \cdot P_{k,j} \right) + P_{\max} \le d_m \cdot l \tag{4}$$

The first term of (4) represents the amount of workload that is added by the inclusion of the baselining flow $flow_{01}$.

When B is the highest (maximal) priority level allowed to utilize $flow_{01}$ baselining opportunities, we can draw the following 3 conclusions.

***1)*** the delay bound for priority levels-m in [1,B-1] must consider the existence of flow$_{01}$ in the delay bound schedulability equation. All priority levels in [m+1,B] (those priority levels allowed to baseline and at higher priority than the priority level m) may utilize flow$_{01}$ for baselining. The resulting delay bound schedulability equation for priority level-m will be (4).

***2)*** the delay bound for priority level-m = B can be scheduled as if flow$_{01}$ had no bandwidth allocation at all. The rate at which frames arrive on flows within the priority levels [1,B] is restricted to the initial traffic envelope as each flow is restored. The workload of flows having frames that become eligible is constrained by the second term of (4). Considering the non-preemption characteristic of the scheduling policy the delay bound schedulability equation for priority level-B is the original RCSP defined as

$$\sum_{k=1}^{m}\sum_{j=1}^{i_k}\left(\left\lceil d_m / X_{\min_{k,j}} \right\rceil \cdot P_{k,j}\right) + P_{\max} \le d_m \cdot l \tag{5}$$

***3)*** the delay bound for priority levels-m in [B+1,M], where M is the maximum priority level can be computed as if flow$_{01}$ had no bandwidth allocation at all. Priority levels in [1,B] all have higher priority and are limited in arrival rate by the fully restored traffic envelope. Equation (5) establishes the delay bound for these flow levels, which is the original RCSP.

### *E. Flow Transition to a Non-baselined State*

When a frame k arrives on flow n at level 1 (flow$_{1,n}$) with an AT of $AT_n^k > ET_n^k$ where $ET_n^k$ is the ET of frame k given the arrival pattern of previous frames on flow$_{1,n}$, the flow$_{1,n}$ will be considered non-baselined. This occurrence will be very rare, other than an intentional baselining event in the previous switch.

### *F. Frame Receipt Function*

Determination of which arriving frame to baseline is carefully considered given the limited TO of flow$_{01}$. When a flow has not been baselined for a time period that allows the relative clock drift between network devices to reach a bound, the flow must be re-baselined.

Baselining is considered at each frame receipt allowing real-time consideration on a frame-by-frame basis. The specific event is a frame held by the switch becoming eligible for transmission scheduling. The eligibility time is $ET_j^k$ on flow k at switch j. The delay bound on priority level-m is d$_m$.

A minimum Baseline Interval (BI) table records the minimum interval in which each flow must be baselined. The acceptable interval is a function of the delay-jitter performance need of the flow given the clock drift rate capability of the current switch and the device transmitting to the current switch. The baseline deadline BD(k) records when the flow k must be baselined. The baselined status of each flow will be maintained. A sorted queue Flow$_{01}$Queue of scheduled frame transmissions on flow$_{01}$ is maintained. Each element holds a frame pending transmission, and a scheduled transmission time. The queue is sorted in ascending order of scheduled transmission times. A FIFO queue FIFOqueue will be used to store pending frames at priority level-1.

### *G. Virtual Flow Allocation Considerations*

In order to support sufficient baselining epochs during the packet scheduling process, virtual flow flow$_{01}$ must be allocated a minimum inter-arrival time to support the baseline intervals of the FlexTDMA flows. This time (6) should be sufficient to allow a baseline event on each flow once each BI.

$$1 \Big/ \sum_{k=1}^{n}\left(1 / \min BI_{flow_k}\right) \tag{6}$$

The minimum BI for a flow (7) is a function of the clock drift rate and the maximum tolerable clock drift error.

$$\min BI_{flow_k} = \max ErrorPerBI_{flow_k} / clockDrift_{ppm} \tag{7}$$

### *H. Formal Description of FlexTDMA*

The FlexTDMA algorithm has three processing stages in the switch: 1) frame arrival to the switch, 2) frame eligibility, and 3) frame transmission. When a frame arrives at the input port of a switch, the arrival time (AT) is recorded, and the eligibility time (ET) of the flow is determined. The frame transmission deadline is determined from the ET and the switch flow deadline bound and the frame is stored pending eligibility. A frame, held until its ET, is scheduled for transmission on Flow$_{01}$ at its deadline time (flow baselining) when either the flow is not currently baselined or the baseline deadline has been exceeded, and Flow$_{01}$ is available. Otherwise the frame is placed into the appropriate FlexTDMA priority FIFO queue pending transmission. Flow$_{01}$ is considered available when no frame is currently scheduled for transmission within one Flow$_{01}$.period of the desired deadline time in order to respect the minimum transmission period of Flow$_{01}$. If a switch output port becomes idle, the switch selects a frame from Flow$_{01}$ queue when a frame is present and is scheduled in the window [now, now + frame$_{time}$], or a frame is selected from the FIFO queue.

### *I. FlexTDMA Properties*

FlexTDMA provides a level of service between asynchronous end systems and switches similar to a synchronized system supported with a TDMA scheduled interconnecting bus. The key criteria is: *deterministically bounded delay, constant delay bound, a low delay bound, and very low delay-jitter.*

## IV. PERFORMANCE

### *A. Performance Comparison Criteria*

The two key performance characteristics that differentiate FlexTDMA from TDMA and RCSP are stable delay bounds at the end-to-end value, and minimal delay-jitter. TDMA offers both minimal delay bounds and minimal delay-jitter service but only when the communicating elements are synchronized.

### *B. Simulation Results*

Figure 1 shows the network topology used to demonstrate the relative advantages of FlexTDMA. The topology has 6 switches connected in series. The traffic from node 0 is forwarded to node 10 where the end-to-end delay and delay-jitter are monitored. The nodes 1 to 9 introduce cross traffic forwarded to switch 10 where it is discarded. Each node transmits frames on 50 flows at a period of 2050 μs with 0 to 50 μs of jitter added to each period time. This topology and switching function were simulated using OPNET. Each switch can be configured to support FlexTDMA, RCSP-RJ or RCSP-DJ. The bounding delays were computed off-line and were configured into each switch. This allows the FlexTDMA and RCSP-DJ policies to target the bounding delays at each switch. Additionally a high delay bound was selected to simulate a requirement driven delay target for FlexTDMA.

Figure 2 shows the Cumulative Distribution Function (CDF) for end-to-end delay under each of the scheduling policies. The delay bounds for RCSP-RJ were nearly zero as there was little contention and therefore minimal actual delay at each switch. RCSP-DJ delay bounds were centered on the bounding delay of 4164 μs. RCSP-DJ has direct knowledge of the delay in the previous switch, and can simply delay each frame to the bounding delay of the previous switch. FlexTDMA suffers from baseline contention that causes some degree of delay less than the maximal delay bound. FlexTDMA had a few delay bounds ranging 3954 to 4164 μs due to baseline contention (this can be seen in the graph as the lower foot of the CDF for FlexTDMA), with the majority at the bounding delay of 4176 μs. Under FlexTDMA most frames are forwarded in a baselined condition so that no baseline contention is encountered. FlexTDMA was also tested with an artificially high required cumulative end-to-end delay bound of 6000 μs. There were a few delay bounds ranging 5018 to 6014 μs due to baseline contention (this can be seen in the graph as the lower foot of the CDF for FlexTDMA), with the majority at the bounding delay of 6014 μs. This demonstrates the ability of FlexTDMA to support delay bounds higher than the bounding delay values.

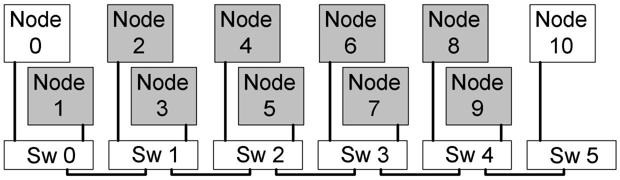

Fig. 1. The OPNET simulation network used a 6-switch serial topology.

Each policy tested offered very small (sub micro-second) delay-jitter performance, with delay-jitter measured as the reduction in the inter-arrival time of frames.

Figure 3 shows the average delay-jitter under FlexTDMA over a one second simulated run. The delay-jitter continually reduces as flows are baselined.

## V. Conclusions

This paper presents a switch frame processing methodology called FlexTDMA, which enhances the system level performances of a closed network by insuring a constant delay, a low delay bound, and low delay-jitter in a network of asynchronous end systems and networking switches. These results demonstrate the ability of FlexTDMA to offer nearly equal end-to-end delay and delay-jitter performance of RCSP-DJ in an asynchronous network. The difference in performance between FlexTDMA and RCSP-DJ results from the baseline contention existing in FlexTDMA.

## References

[1] R. L. Cruz, "A Calculus for Network Delay, Part I - Network Elements in Isolation," IEEE Trans. on Info. Theory, vol. 37, no. 1, Jan. 1991.

[2] H. Zhang and D. Ferrari, "Rate-Controlled Static-Priority Queueing," In Proc. IEEE Infocom, pp. 227-236, 1993.

[3] H. Zhang and D. Ferrari, "Rate-Controlled Service Disciplines," Journal of High Speed Networking, 1994.

[4] J. Liebeherr, D. E. Wrege and D. Ferrari, "Exact admission control for networks with a bounded delay service," IEEE/ACM Transactions on Networking, vol. 4, Issue 6, pp. 885-901, 1996.

[5] Hui Zhang, "Providing End-to-End Performance Guarantees Using Non-Work-Conserving Disciplines", Computer Communications, vol. 18, 1996.

[6] J. Loeser, T. Dresden and H. Haertig, "Using Switched Ethernet for Hard Real-Time Communication," Intl. Conference on Parallel Computing in Electrical Engineering (PARELEC), pp. 349-353, 2004.

[7] S. Gopalakrishnan, M. Caccamo and L. Sha, "Switch scheduling and network design for real-time systems," In Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium, 2006.

[8] H. Zhang and S. Keshav, "Comparison of Rate-Based Service Disciplines," In Proceedings of ACM SIGCOMM, pp. 113-121, 1991.

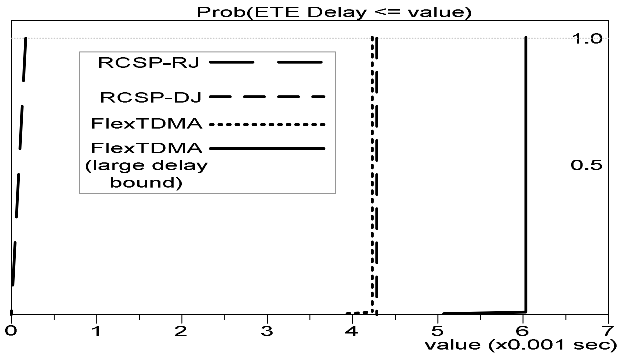

Fig. 2. A CDF of the end-to-end delay bound experienced by RCSP-RJ (large dashed), RCSP-DJ (medium dashed), FlexTDMA (small dashed), and FlexTDMA with a large required delay bound (solid).

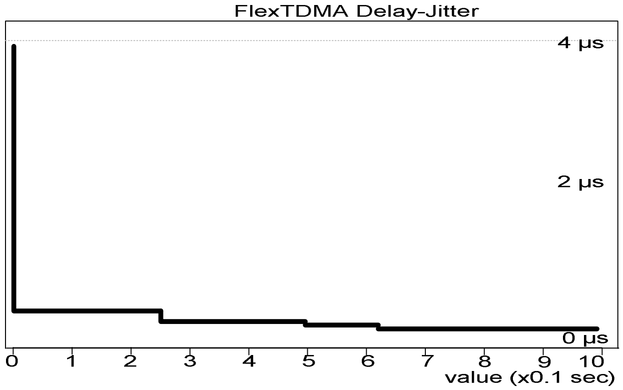

Fig. 3. A plot of the average delay-jitter under FlexTDMA over the course of a one second simulation.