

FT-LAN, A Fault Tolerant Local Area Network Architecture for Mobile Mission Critical Systems

Paul Richardson¹, Larry Sieh²

¹Department of Electrical and Computer Engineering, University of Michigan-Dearborn

²U.S. Army Tank-Automotive Research, Development, and Engineering Center, Warren, MI, 48397-5000

Abstract: This effort presents a fault-tolerant LAN architecture for Mobile Mission Critical Systems (MMCS's). MMCS's are real-time computing platforms that perform continuously in harsh environments. It is desirable that they continue to perform a subset of critical tasks under fault conditions. The network in a MMCS can be characterized as an isolated, real-time, device-control network. A key consideration for MMCS's is the ability of the network to support critical operations during fault conditions. The architecture we propose combines a priority-driven, real-time, LAN protocol with the adaptive earliest deadline first (AEDF) scheduling approach. Under nominal conditions the real-time protocol efficiently guarantees all message time constraints with respect to *bandwidth utilization efficiency*. If a fault(s) occurs, AEDF attempts to sustain critical tasks by ordering message transmissions based on *value and system-state*, ignoring *bandwidth utilization efficiency*. A significant capability of AEDF is its ability to detect and respond to the occurrence of a fault before critical message transmissions are late. A case study and simulation are presented to evaluate performance during overloads.

I. INTRODUCTION

This effort presents a fault-tolerant LAN (*FT-LAN*) architecture for Mobile Mission Critical Systems (MMCS's). MMCS's are mobile, embedded, real-time, computer systems that can be found in applications such as *robotic vehicles, military vehicles, off-road vehicles, and aircraft*. These systems are required to perform continuous operations in harsh, dynamic environments. It is desirable that they continue to perform a subset of critical tasks under fault conditions. Failure to perform critical tasks can result in serious injury or costly damage.

A significant trend in the evolution of MMCS's is the migration towards local area network (LAN) based computer architectures. This evolution is motivated by rapid increases in system complexity and the resultant growth in development, maintenance, and reliability issues. Network-based computer architectures introduce the potential to significantly alleviate these concerns. The network in a MMCS can be characterized by a *limited geographic span, tightly controlled inter-network access*, and availability of *apriori knowledge* regarding operational parameters such as *process arrival rates and message lengths*. It is expected that applications for MMCS's will continue to grow. Further, as user performance demands continue to increase, system complexity will also grow, exacerbating cost and reliability concerns.

A significant obstacle to fully exploiting the potential of computer networks in MMCS's is the inability to predict temporal behavior during network overloads. Overloads result from network faults that occur during continuous operations in

harsh, dynamic environments. Several approaches have focused on *efficiently* guaranteeing time constraints under nominal conditions [6,8,11]. However, these approaches tend to perform poorly during overloads because they don't consider *value* or *system state*. The *FT-LAN* architecture described in this paper provides conditional guarantees for the time constraints of critical messages during overloads.

In the remainder of this paper, we (ii) discuss previous relevant work, (iii) present a network model, (iv) describe the operation of *Adaptive Earliest Deadline First* (AEDF) for real-time networks, (v) present a case study using the weapons element of a combat vehicle, and (vi) summarize our results.

II. PREVIOUS RELATED WORK

The *FT-LAN* architecture combines the results of several past efforts regarding real-time LANs and fault-tolerance in real-time multi-tasking systems.

In [6], *Zhao et al.* describe the augmented transmission time of a message for IEEE 802.5 token ring protocol as:

$$C_i = \begin{cases} \theta(K_i + 1) & \text{if } F \leq \theta \\ L_i F + (K_i - L_i) \max(C_i - L_i \cdot F_{data} + F_{ovhd}, \theta) + \theta & \text{otherwise} \end{cases} \quad (2.1)$$

$$\text{where } K_i = \left\lceil \frac{C_i^b}{F_{data}^b} \right\rceil \text{ and } L_i = \left\lfloor \frac{C_i^b}{F_{data}^b} \right\rfloor,$$

θ is worst case token circulation time, C_i^b is the length of the data for the i^{th} message, F_{data}^b is the length of a frame's data, and F_{ovhd}^b is the length of a frame's overhead.

In [8] a method was presented for determining the feasibility of a message set in a real-time LAN for a combat vehicle. This method uses the *non-preemptive* earliest deadline first (EDF) scheduling policy [3] and the IEEE 802.5 token ring LAN protocol [15]. It was shown to be particularly efficient for networks characteristic of those found in MMCS's. For a set of n messages, $\{m_n\}$, with deadlines that coincide with inter-arrival rates and an n^{th} message that consists of at least one full frame, from [8] we have $\{m_n\}$ is feasible if and only if the following conditions hold:

$$(1) \text{ The network utilization, } U_n = \sum_{i=1}^n \frac{C_i'}{P_i} \leq 1 \quad (2.2a)$$

where C_i' is the *augmented message transmission time* and P_i is the *inter-arrival rate* for the i^{th} message.

(2) With P_i 's ordered in increasing order, for every interval $L: P_1 < L \leq P_n$, the normalized network demand, $\hat{D}(0, L) \leq 1$.

$$\text{Where } \hat{D}(0, L) = \frac{1}{L} \left(F_{max} + \sum_{j=1}^{n-1} \left\lfloor \frac{L-1}{P_j} \right\rfloor C_j' + C_n \right) \quad (2.2b),$$

F_{max} is augmented transmission time of the longest frame

and \overline{C}_n represents the augmented transmission time of the n^{th} message with one full frame removed:

$$\overline{C}_n = \left\lfloor \frac{L}{P_n} \left(\left\lfloor \frac{C_n^b}{F_{\max}} \right\rfloor - 1 \right) F_{\max} + (K_n - L_n) \max(C_n - L_n F_{\max} + F_{\text{ovhd}}, \theta) \right\rfloor + \theta \quad (2.3).$$

AEDF was first described in the context of an adaptive, preemptive, scheduling algorithm for scheduling multiple processes in a real-time system [9]. Under nominal conditions AEDF uses EDF scheduling to *efficiently* guarantee all time constraints [7]. If the system becomes overloaded, AEDF provides conditional guarantees for the time constraints of critical processes by (1) detecting the occurrence of faults before critical processes fail and (2) using *value based scheduling* (VBS) to order process execution based on their *value* and *system state*. The detection of faults before critical processes fail allows a proactive response in mitigating overloads. Value based scheduling sacrifices *throughput* in order to sustain critical processes during fault conditions. In section 4 we describe a non-preemptive variation of AEDF for scheduling non-preemptive frame transmissions on a LAN.

III. NETWORK MODEL

A network model for MMCS's is described that allows us to reason about message set feasibility under *nominal* and *fault* conditions. In order to facilitate the requirement to satisfy *message transmission time constraints*, it is critical to leverage the *a priori* knowledge available during the design-phase of a MMCS: Extensive information exists concerning message set parameters such as *inter-arrival rates*, *deadlines*, *lengths*, and *importance*. If these parameters aren't explicitly known, there is often enough information available to make reasonable estimates. Typically, inter-network access is strictly controlled through one or more wireless gateways (radios), allowing the impact of inter-network traffic to be bound. The topology of the LAN is fixed and has a limited geographic span, which bounds the maximum propagation delay and deadspace [8].

In this effort, we neglect inter-network traffic and restrict our concern to local message transmissions. It is assumed that message lengths are known, synchronous messages have an inter-arrival rate equal to their deadline, and asynchronous messages have an estimated inter-arrival rate equal to their deadline. Section 4 describes how AEDF can compensate for the case when actual inter-arrival rates exceed their estimates. Given the stated assumptions, the approach described in [8] based on *EDF* and *IEEE 802.5 token ring LAN* is selected as the underlying LAN for *FT-LAN*. Feasibility for nominal conditions can thus be determined by eqn 2.2.

A. Network Faults

Since we are explicitly concerned with feasibility under fault conditions, it is necessary to consider the impact of faults on network loading. The general effect of a fault is to cause an increase in *effective network loading*. If this increase causes a violation of the conditions in eqn 2.2, then the network is overloaded and some transmissions will unavoidably be late.

Network faults can be classified as either *transient surges* in loading or the *permanent loss* of some network resource.

Transient surges occur when frame arrivals exceed specified inter-arrival rates. Sources of transient surges include: *frame retransmissions* caused by lost/corrupt frames, *asynchronous frames arrivals* that exceed estimated values, or *unforeseen state changes* that causes a burst of unexpected traffic. Transient surges will subside returning the network to a nominal state. If a transient surge occurs as the arrival of k unspecified messages, feasibility is determined by modifying eqn 2.2 to incorporate the transmission times, C'_k , and inter-arrival rates, P_k , of each additional message. A permanent network fault occurs when some component fails (*e.g. a segment of transmission media or network interface card*). This results in a delay, τ_i , to each message's transmission time, C'_i . To determine feasibility in this case, the transmission times in eqn 2.2 must include the delay: $C'_i + \tau_i$. A delay of infinity represents a failed message transmission. Recovery from permanent faults requires some form of reconfiguration or repair. Redundancy and dynamic reconfiguration are common ways of handling permanent faults.

During an overload, there is an indeterminate delay from the time that a fault occurs until the first frame transmission is late. The duration of this delay depends on the network loading and nature of the fault. The next section will describe how this delay can be leveraged to provide support for critical message transmissions.

IV. AEDF APPLIED TO NETWORKS

This section describes the application of AEDF to the real-time, LAN protocol described earlier [8]. During nominal conditions EDF scheduling is used to order frame transmissions and the network is non-intrusively monitored for the occurrence of faults. If a fault occurs, AEDF provides conditional guarantees for critical frames by (1) detecting faults before critical frames are late, (2) suspending non-critical frame transmissions, and (3) using *VBS* to order critical frame transmissions based on their value to the system and current system state. Although VBS is less efficient than EDF from the perspective of *bandwidth utilization efficiency*, its advantage lies in its ability to select *high-value frames* for timely transmission. This is a desirable trait during overloads, when some frames will unavoidably be late. These operations are described below.

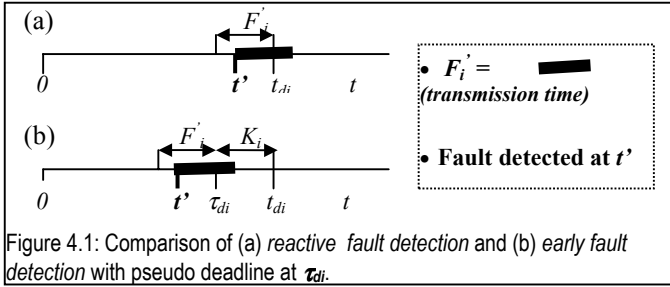
Critical frames carry data whose late arrival can result directly in serious injury or costly damage (*e.g. closed-loop stabilization or emergency stop*). The late arrival of non-critical frames may cause degradation in performance but will not cause any immediate danger to the system or crew. Examples include *oil-pressure* and *coolant temperature*. The suspension of non-critical frame transmissions frees additional network capacity to support critical frame transmissions.

A. Early Fault Detection

AEDF proactively responds to the occurrence of faults by using spare network capacity to detect faults before critical frames are late. This is a noteworthy improvement over *reactive* methods which wait until a frame is late before realizing a fault condition exists. Since *early fault detection*

requires additional network capacity, it is only provided for critical frames. *Reactive fault detection* is provided for non-critical frames. *Early fault detection* is achieved by allocating unused capacity to create *pseudo* deadlines well before actual deadlines. If the resulting time constraints are feasible, then a violation of a *pseudo* deadline will signal the occurrence of a fault before any critical frame transmission is late. If enough spare capacity exists, a scheduling event can be guaranteed for each critical frame transmission before it is late. In section V, we will show that the amount of spare network capacity to achieve this is not unreasonable.

Figure 4.1 depicts this idea in terms of the time-line for a frame transmission using (4.1a) *reactive fault detection* and (4.1b) *early fault detection*. Assume an undetected overload condition exists and that a frame f_i arrives at time $t=0$, with deadline t_{di} , and augmented transmission time F'_i . In 4.1a, the overload is detected when a scheduling event takes place at time $t' \geq t_{di} - F'_i$. If f_i is scheduled immediately, it will complete transmission late, at time $t' + F'_i$. If the message set is feasible, then this could only happen if a fault has occurred, causing the network to overload. In 4.1b, spare capacity is used to create a *pseudo* deadline at $\tau_{di} = t_{di} - K_i$. If the *pseudo* deadline is feasible, then a scheduling event occurring at time $t' \geq \tau_{di} - F'_i$ indicates a fault condition exists by violating the *pseudo* deadline. If f_i is scheduled before $t_{di} - F'_i$ it will arrive on time.



The transmission time for the maximum length frame, F'_{max} , has been shown to be the worst case blocking time and hence the longest interval between scheduling events using the IEEE 802.5 protocol [8]. If $K_i = F'_{max}$ for each critical frame, then during a *transient* overload each critical frame is guaranteed a scheduling event before it is late. During a *permanent* overload F'_{max} will increase by some undetermined amount, and such a scheduling event cannot be guaranteed. In our example, we will use $K_i = F'_{max}$.

To determine the feasibility of *early fault detection*, the conditions from eqns 2.2 and 2.3 are modified to reflect the allocation of additional network capacity. Constraints (1) and (2) are modified by ρ , where $\rho = F'_{max}$ for critical frames and $\rho = 0$ for non-critical frames. Constraint (1), network utilization for *early fault detection* is:

$$(1) U_{n_e} = \sum_{i=1}^n \frac{C_i}{P_i - \rho} \leq 1 \quad (4.1a)$$

Constraint (2), normalized maximum demand for *early fault detection* is: $\forall L: P_l < L \leq P_n, \hat{D}_e(0, L) \leq 1$. Where

$$\hat{D}_e(0, L) = \frac{1}{L} \left(F'_{max} + \sum_{j=1}^{n-1} \left\lfloor \frac{L-1}{P_j - \rho} \right\rfloor C_j + C_{n_e} \right) \quad (4.1b)$$

$$\text{and } \overline{C}_{n_e} = \left\lfloor \frac{L}{P_n - \rho} \left(\left\lfloor \frac{C_n^b}{F'_{max}} \right\rfloor - 1 \right) F'_{max} + (K_n - L_n) \max C_n - L_n \cdot F'_{max} + F_{ovhd} \theta \right) \quad (4.2)$$

It should be noted that these conditions are pessimistic in that any fault condition is detected, even if it don't result in an overload. If the conditions cannot be satisfied then there is not enough spare network capacity for early fault detection. Mitigation of these issues is discussed in more detail in [9].

B. Value Based Scheduling (VBS)

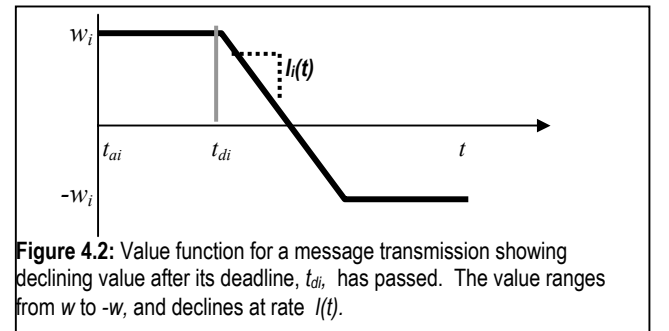
VBS [2,4,5,9] uses *value functions* to determine the cost or benefit of meeting or missing time constraints. *Value functions* are useful during overloads when relative value is more important than overall throughput. These functions are used to *assess scheduler performance* and *order critical frame transmissions*. For a message of type i , a value function with two parameters is used to assign values for each transmission:

$$v_i(w_i, l_i(t)) \quad (4.3).$$

The weight parameter, w_i , establishes bounds for cost and benefit. A weight of ∞ represents system failure. The lateness parameter, $l_i(t)$, is a function of time that captures the impact of tardiness. A lateness function of ∞ represents a step function decline in value.

Figure 4.2 depicts the value function for a message i that arrives at time t_{ai} and has deadline t_{di} . Initially the message is assigned a value of w_i . If the frames that comprise the message are scheduled in a timely manner, then the message will be on-time and its value function is $v_i = w_i$. If a frame is scheduled at time $t > t_{di} - F'_i$ then it is considered late and its value is decreased according to $l_i(t)$. In this example, the minimum value of m_i is bound at $-w_i$. Thus we have:

$$\begin{aligned} \text{if message } m_i \text{ is timely then } v_i(t) &= w_i, \\ \text{otherwise, } v_i(t) &= \min(w_i, w_i - l_i(t)), \text{ if } w_i - l_i(t) \geq 0 \\ &= \max(-w_i, w_i - l_i(t)) \text{ if } w_i - l_i(t) < 0. \end{aligned} \quad (4.4)$$



Changes in system state also affect the value of a transmission [9]. To capture changes in system state the parameters w_i and $l_i(t)$ can be modified to reflect new states. This results in a family of curves that captures the impact of state changes on the value of a message transmission. Network performance can be analyzed by examining mean values for each message type \bar{v}_i and their cumulative sum:

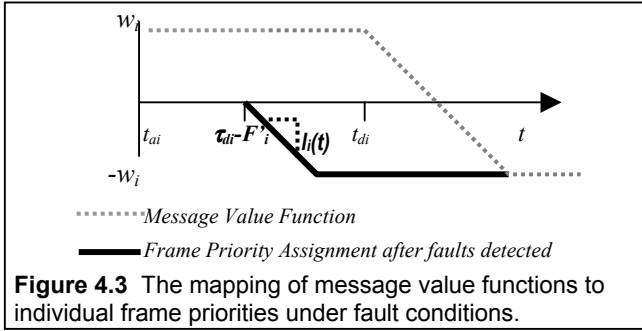
$$V = \sum_{i=1}^n v_i \quad (4.4).$$

If a fault occurs, AEDF also uses value functions to assign priorities to frames that comprise critical messages, as depicted in figure 4.3. Lower numeric values indicate higher priorities. For time t and critical frame f_i we have:

$$\begin{aligned} \text{if } t \leq \tau_d - F'_i \text{ then } \text{priority} &= t_{di} - t \quad (\text{EDF scheduling}) \\ \text{otherwise } \text{priority} &= \max(-w_i, w_i - l_i(t)) \quad (\text{VBS}) \end{aligned} \quad (4.5)$$

Once a fault is detected AEDF suspends the transmission of non-critical frames. This has the affect of releasing additional network capacity for critical message transmissions. Non-critical deadlines are still evaluated to determine the existence of a fault condition, however the frames are not considered for transmission. Non-critical frame transmissions resume when a scheduling event completes without any faults being detected.

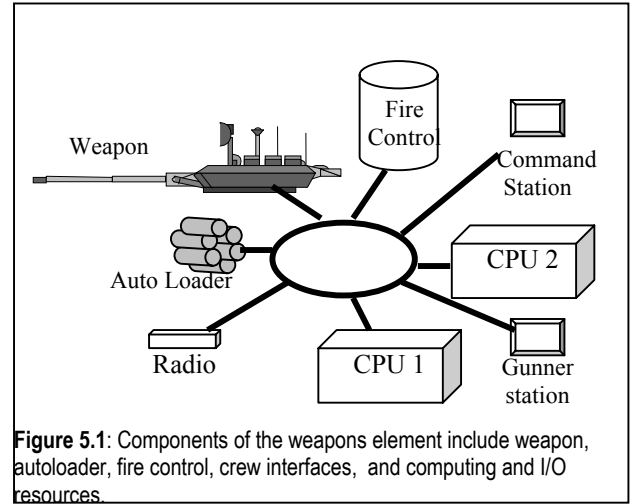
AEDF uses non-intrusive monitoring and a simple value function to avoid exacerbating operations during overloads. As a result AEDF doesn't add significant computational complexity to the EDF scheduling approach.



V. EXAMPLE SYSTEM: NETWORK OPERATIONS UNDER OVERLOAD CONDITIONS

The weapons element of a main battle tank is presented as an example of a MMCS LAN. After describing the essential characteristics for evaluating the system, we simulate the system under overload conditions using both AEDF and EDF. Our objective is to evaluate the ability of AEDF to facilitate critical operations under overload conditions.

The weapons element of a main battle tank consists of a number of components and the means for controlling and



monitoring those components. The primary components are the weapon itself, the fire control system, and the autoloader. Also included are processing resources, crew interfaces, a radio for communications external to the vehicle, and a digital intercom for internal voice communications. Figure 5.1 depicts a functional block diagram of the weapons element [12]. Table 5.1 describes the lengths and arrival rates of the messages in the element's message set [12][13]. Several assumptions are made to determine the feasibility of the message set for this network. It is assumed that the LAN consists of 32 nodes (2 crew input/output stations, 2 CPU's, a radio, 2 digital audio channels with 2 interfaces each, and 24 sensors and actuators of various types.) and estimate the total media length to be 200 meters. For a synchronous message, the inter-arrival rate is assumed to be its period, and the deadline is assumed to coincide with the period. The minimum inter-arrival rate of asynchronous messages is assumed to be their deadline. The operational state is a combat operation and loading is at its worst case.

A. Feasibility and AEDF Parameters

The selection of maximum frame size is significant and eqns 2.2 and 4.2 can be evaluated over varying frame sizes to determine which is optimal for a given message set [8]. For a data rate of 6M bits/second and a maximum frame payload of

TYPE	Size (bits)	Interarrival time (ms)	TYPE	Size (bits)	Interarrival time (ms)	TYPE	Size (bits)	Interarrival time (ms)
Gun Elevation Sensor	32	1	Target Gun Lock	32	16	Weapons Status	4096	333
Gun Azimuth Sensor	32	1	Proj Seize	32	32	Ammunition Status	4096	333
Gun Elevation Actuator	32	1	Proj Load	32	32	Sites Status	4096	333
Gun Azimuth Actuator	32	1	Proj Release	32	32	Situation Report,	16K	1000
Sight Azimuth Sensor	32	1	Proj Select	32	32	Order	16K	1000
Sight Elevation Sensor	32	1	Proj Position	32	7.5	Audio Channel 1	2048	33
Sight Azimuth Actuator	32	1	Gun Elev. Actuator Fluid Level	32	50	Audio Channel 2	2048	33
Sight Elevation Actuator	32	1	Gun Elev. Actuator Fluid Pressure	32	50	System Management 1	4096	64
Gun Azimuth Setpoint	32	20	Gun Azimuth Actuator Fluid Level	32	50	System Management 2	4096	64
Sight Elevation Setpoint	32	20	Gun Azimuth Actuator Fluid Pressure	32	50	System Management 3	4096	64
Sight Azimuth Setpoint	32	20	Sight Elevation Actuator Fluid Level	32	50	System Management 4	4096	64
Sight Azimuth Setpoint	32	20	Sight Azimuth Actuator Fluid Level	32	50	System Management 5	4096	64
Trigger Actuate	32	10	Sight Elev. Actuator Fluid Pressure	32	50	System Management 6	4096	64
Target Id Query	32	16	Sight Azimuth Actuator Fluid Pressure	32	50	System Management 7	4096	64
Target Id Status	1024	32	Loader Actuator Fluid Level	32	50			

512 bits, eqns 2.2 and 2.3 give: $\hat{D}(0, L) = 0.88$, $U_n = 0.89$ [8]. The feasibility of early fault detection is evaluated using eqns 4.1 and 4.2. In this case $\hat{D}_e(0, L) = 0.99$, $U_{ne} = 1.0$ and we have feasibility of early fault detection. Value functions are assigned to each message type as shown in table 5.2. The message set has been re-grouped into classes with identical time constraints, lengths, and value functions. *Qty* indicates the quantity of each message type. The *weight*, *w*, and *importance*, *l(t)*, are value function parameters. *Weight increment*, *dw*, is used to alter the value of *w*, to reflect a change in system state caused by late or timely arrivals. For this system, *w* is decremented by *dw* with each late arrival and incremented with each timely arrival, but to no more than the original value. *Min val*, is the least amount that *w* can decrease to. During the simulation $w_i(t)$ is calculated for each message transmitted, and $\bar{w}_i(t)$ for all transmissions of a message type. Message identifiers are listed in the *ID* column. Critical message identifiers range from M1..M18, non-critical identifiers range from N19..N46. Identifiers are used to identify specific messages when discussing simulation results.

MESSAGE TYPE	Qty	w	dw	min val	l(t)	ID
Trigger Actuate	1	60	3	-100	∞	M1
Real-Time Information	2	50	10	-80	$20vt/T$	M2, M3
Stabilization Actuators/Sensors	8	20	1	-60	∞	M4-M11
Stabilization Set Points	4	20	1	-60	$20vt/T$	M12-M15
Projectile Position	1	20	1	-30	∞	M16
Weapons Status	2	15	1	-30	$4vt/T$	M17, M18
Ammunition Status	1	15	1	-30	$4vt/0.11$	N-19
System Management	7	10	1	-20	$2vt/0.11$	N20-N26
Target Identification Query	1	8	1	-10	$2vt/T$	N27
Target Identification Status	1	8	1	-10	$2vt/T$	N28
Automatic Set Point Lock	2	8	1	-10	$2vt/T$	N29, N30
Audio Channels	2	8	1	-10	$2vt/T$	N31, N32
Projectile Handling	4	6	1	-8	$2vt/T$	N33-N36
Maintenance Sensors	10	1	1	-2	$2vt/T$	N37-N46

Table 5.2 : Message data grouped by *time constraint*, *length*, and *value function*. *Qty* indicates number of each message type. *Weight*, *w* and *importance*, *l(t)*, are value function parameters. *Min val* is the lower bound that value can decrease to and *dw* is used to alter *w* to capture changes in system state. *ID* lists the message identifiers.

B. Simulation Description and Results

The duration of the simulation was for 10 seconds during which EDF and AEDF were evaluated under identical conditions. Pending late synchronous messages are dropped,

since it would be more effective to send the most recent data, than to send data that is known to be late. A dropped message m_i is assigned a value of $-w_i$, and w_i is decremented by dw_i . An asynchronous message that exceeds its estimated inter-arrival time is considered a fault of the transient surge variety. Asynchronous messages continue to pend until they are scheduled and transmitted. In order to create an overload situation, a transient surge of messages was inserted for four seconds during the simulation run. The surge was achieved by decreasing the inter-arrival time of system management messages from 64 ms to 4 ms . During the surge the maximum normalized network demand and utilization are determined as:

$$\hat{D}(0, L) = 2.93, \quad U_n = 2.94$$

This indicates that the network loading during the transient surge is nearly three times the feasible network loading.

Figure 5.2 depicts the percentage of late message arrivals ordered by decreasing value for AEDF and EDF algorithms. In this figure the tendency's of EDF to minimize maximum lateness is clear. During an overload, this results in a relatively balanced percentage of late message arrivals. Conversely, AEDF's use of VBS during overloads is clearly evident in the low percentage of high value messages and high percentage of low value messages transmitted late. EDF's performance is further worsened because many of the high arrival-rate, high value stabilization messages (M4-M11) are blocked as EDF schedules relatively lower value messages (N33-N46). This situation does not occur using VBS.

Figure 5.3a and 5.3b depict the mean value for each message type sorted by decreasing value for AEDF and EDF respectively. These figures present an indication of how the system performed under severe overload. EDF suffers to some extent on all of the critical message transmissions, M1-M19. EDF also scores well for many non-critical messages reflecting again its inability to distinguish importance. It is probable that critical operations would have failed in this instance. For AEDF, all critical messages retained nearly nominal value, indicating a reasonable probability that the system would have sustained critical operations under a severe overload. This comes at the cost of negative scores for most of the non-critical message transmissions. Figure 5.4 depicts summarizes the earlier charts by presenting the normalized total value and total late arrivals for AEDF and EDF. This chart reinforces the earlier data regarding the ability of AEDF to facilitate network operations during severe overloads.

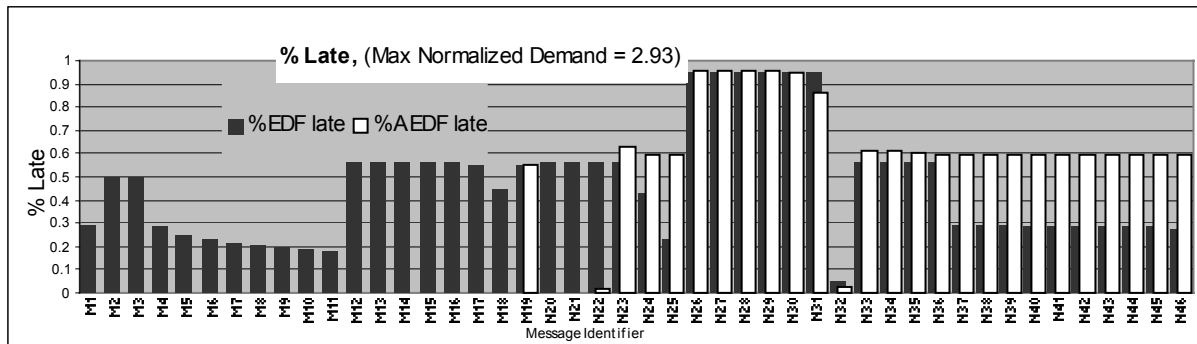


Figure 5.2: Percent late for each message stream from simulated overload. Message identifiers are on x-axis, %late on y-axis. AEDF results are shown in white, EDF results are shown in black.

VI. SUMMARY

This effort addresses the issue of fault tolerance in a real-time network for MMCS's. In this type of system, faults are expected to occur over lengthy operational cycles in harsh and dynamic environments. During an overload, critical messages require some level of continuous support until the system can be brought to a safe state. We propose the use of AEDF scheduling in conjunction with the IEEE 802.5 protocol for use in MMCS LAN's. A simulation study of the weapons element of a combat vehicle demonstrates the potential of this approach. A more formal description of adaptive scheduling can be found in [9]. The next phase of this work will be to implement the network and conduct validation experiments.

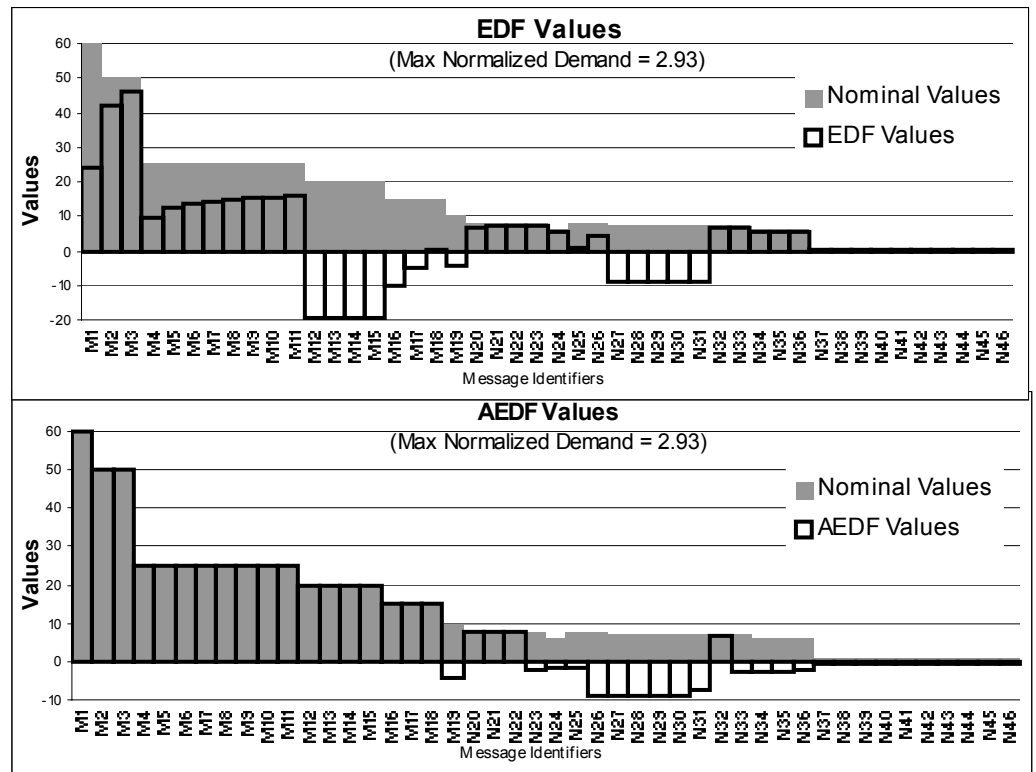


Figure 5.3: Value assignment for simulated overload, (a) EDF and (b) AEDF. Values for message streams are shown on x-axis in decreasing order, value assignments are on y-axis. Shaded areas represent nominal or ideal values, Actual results are indicated by dark lines.

REFERENCE LITERATURE:

- [1] K. Arvind, J Stankovic, "A LAN Architecture for Communications In Distributed Real Time Systems", *IEEE Real Time Systems, Vol 3, No. 2, May 1991*
- [2] F.R. Brown, "Scheduling Real-Time Processes Using Fuzzy Logic" *PhD Dissertation, Utah State University, 1998*
- [3] Jeffay, K., Stanat, D. Martel, C. , "On Non-Preemptive Scheduling of Periodic and Sporadic Tasks", *12th IEEE Real-Time Systems Symposium, San Antonio, TX, Dec 1991*
- [4] Jensen, E.D., C.D. Locke, H. Tokuda, "A Time-Value Driven Scheduling Model for Real-Time Operating Systems", *Proc. Symp. on Real-Time Systems, November 1985,*
- [5] D. Jensen, "Eliminating the 'Hard'/'Soft' Real-Time Dichotomy", *Embedded System Conference, Apr 94, pp84-96.*
- [6] S. Kamat, W. Zhao, "Real Time Performance of Two Token Ring Protocols", *Advances In Real Time Systems, Prentice Hall, 1996*
- [7] C. Liu, J. Layland, "Scheduling Algorithms for multi-programming in a Hard Real Time Environment", *ACM 20(1):46-61, 1973*
- [8] Richardson, Seih, "Real-Time LANs in Combat Vehicles: Feasibility Criteria For Non-Preemptive Messages and Multiple Message Streams Originating From Individual Nodes", *IEEE ICCCN '99, Boston, MA, Oct 99.*
- [9] Richardson, P., Sarkar, S "Adaptive Scheduling: Overload Scheduling For Mission Critical Systems", *5th IEEE Real-Time Applications Symposium, Vancouver, BC, June 1999*

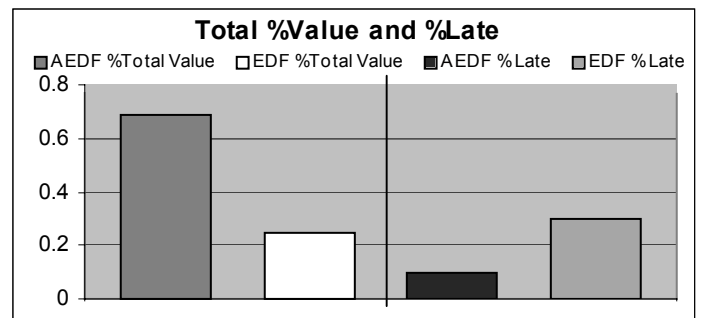


Figure 5.4: Cumulative Results for simulation. Two leftmost bars are %Nominal Value for AEDF and EDF. Two rightmost bars are %late for AEDF and EDF.

- [10] Richardson, P., "CVLAN: A Real-Time LAN Architecture for Combat Vehicles", *IEEE DASC Symposium, Ground Vehicles Session, Seattle, WA, Oct 98*
- [11] Strosnider, Lehoczky, and Sha, "Advanced Real Time Scheduling Using IEEE802.5 Token Ring", *Proc IEEE Real-Time Symposium, Dec 1988*
- [12] *M1 Series Tank Program, System Segment Design Document for Bus Operations*, U.S. Army TACOM (Limited Dissemination)
- [13] *Crusader Systems Segment Design Document (DRAFT)*, U.S. Army TACOM (Limited Dissemination)
- [14] *IEEE Standard 802.5-1989, A Token Ring Access Method & Physical Layer Specification*, IEEE, New York, 1989