Probabilistic Modeling of Network-induced Delays in Networked Control Systems

Manoj Kumar, A.K. Verma, and A. Srividya

***Abstract*—Time varying network induced delays in networked control systems (NCS) are known for degrading control system's quality of performance (QoP) and causing stability problems. In literature, a control method employing modeling of communication delays as probability distribution, proves to be a better method. This paper focuses on modeling of network induced delays as probability distribution.

CAN and MIL-STD-1553B are extensively used to carry periodic control and monitoring data in networked control systems. In literature, methods to estimate only the worst-case delays for these networks are available. In this paper probabilistic network delay model for CAN and MIL-STD-1553B networks are given. A systematic method to estimate values to model parameters from network parameters is given. A method to predict network delay in next cycle based on the present network delay is presented. Effect of active network redundancy and redundancy at node level on network delay and system response-time is also analyzed.

***Keywords*—NCS (networked control system), delay analysis, response-time distribution, worst-case delay, CAN, MIL-STD-1553B, redundancy

I. Introduction

NEtworked Control System (NCS) contains a large number of interconnected devices that exchange data through communication networks. Examples of such systems are found in industrial automation, building automation, office and home automation, intelligent vehicle systems and advanced aircraft and spacecraft. NCS provides several advantage such as modular and flexible system design, simple and fast implementation and powerful system diagnosis and maintenance utilities [1], [2]. But it has few disadvantages when NCS has real time demands i.e., it requires guaranteed bounded time delays [1], [2], [3], [4].

Schematically a typical NCS is shown as in Figure 1. Sensor node samples the process parameters with a given sampling period, convert physical parameters to digital and pack the message to send to controller. The controller node unpacks the message from sensor node and use control algorithm to calculate control signals to be sent to actuator node. Actuator node according to control signal takes the corrective action in process. All these messages are sent over control network. In NCS, time delay has one more factor in addition to node computation delay, it is network-induced delay, i.e. from sensor to controller and controller to actuator.

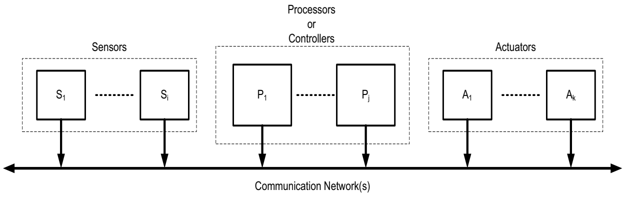

Fig. 1. Schmatic of a typical NCS

The specific application imposes different degrees of timing requirements to the NCS implementation. Control systems pose timing constraints on computer-controlled systems because control theory assumes a highly deterministic timing on its implementation [1]. Consequently, the insertion of the communication network in control loops makes the analysis and design of such application complex, due to the network-induced time delays. Network delays are time varying based on the network technology, MAC and traffic etc. [1], [5], [6], [7], [8]. These time varying network-induced are responsible for degradation of control system's quality of performance (QoP) and causing instability [9], [3], [10].

Refs. [1], [6], [7] point out control method employing modeling of communication delays as probability distribution and design controller to account for this to have better QoP. NCS field buses [11], [12], [13] such as FIP, PROFIBUS, SAE token bus, SAE token ring, CAN and MIL-STD-1553B have mostly periodic and small size data. These buses based on delay characteristics [14] can be classified in i) cyclic service networks and ii) random access networks. For both kind of delays, probabilistic models do not exist [15], [16].

In this paper, methods to probabilistically characterize network-induced delay for two networks CAN and MIL-STD-1553B is proposed. The method requires network parameters as random variables (r.v.). A detailed method to determine *pdf* (probability density function) of these r.v.s from system parameters is outlined.

In section 2, an overview of CAN and MIL-STD-1553B is given along with their worst-case delay analysis. Section 3 gives probabilistic network delay model for both networks. Effect of redundancy on network delay is given in section 4. Derivation of sample-to-actuation and response-time of NCS is given in section 5. Effect of node redundancy on system response-time is analyzed in section 6.

II. Overview of field bus networks

In this paper, for stochastic network delay modeling CAN and MIL-STD-1553B are chosen. In this section details of these two buses is provided.

Manoj Kumar is with the Control Instrumentation Division, BARC, Mumbai, 400085, India, email: kmanoj@barc.gov.in

A.K. Verma and A. Srividya are with Reliability Engineering, Indian Institute of Technology Bombay, Mumbai, 400076, India, email: akv@ee.iitb.ac.in, asvidya@ee.iitb.ac.in

A. CAN

CAN is a broadcast bus where a number of nodes are connected to the bus. It employs carrier sense multiple access with collision detection and arbitration based on message priority (CSMA/AMP) [17]. Data is transmitted as *message*, consisting of up to 8 bytes. CAN message format is shown in Figure 2.

The basic features [17], [18] of CAN are:

1) High-speed serial interface: CAN is configurable to operate from a few kilobits to 1 Mega bits per second.
2) Low cost physical medium: CAN operates over a simple inexpensive twisted wire pair.
3) Short data lengths: The short data length of CAN messages mean that CAN has very low latency when compared to other systems.
4) Fast reaction times: The ability to transmit information without requiring a token or permission from a bus arbitrator results in extremely fast reaction times.
5) Multi master and peer-to-peer communication: Using CAN it is simple to broadcast information to all or a subset nodes on the bus and just an easy to implement peer-to-peer communication.
6) Error detection and correction: The high level of error detection and number of error detection mechanisms provided by the CAN hardware means that CAN is extremely reliable as a networking solution.

A data may be transmitted periodically, sporadically, or on demand. Every message is assigned a unique *identifier*. The identifier serves two purposes, filtering messages upon reception and assigning priority to the message. The use of identifier as priority is the most important part of CAN regrading real-time performance. The identifier field of CAN message is used to control access to the bus after collision by taking advantage of certain electrical characteristics. In CAN terminology, a 0 bit is termed *dominant*, and a 1 bit is termed *recessive*. In case of multiple stations transmitting simultaneously, all stations will see 0 if any one of the node puts 0 bit (dominant), while all stations will see 1 if all transmitting node put 1 bit. So, during arbitration, by monitoring the bus a node detects if there is a competing higher priority message and stops transmission if it is the case. A node transmitting the last bit of the identifier without detecting a higher priority message must be transmitting the highest priority ready message, and hence can continue.

For more details on CAN, please refer to [17], [18].

1) Delay analysis: Tindell et al. [15], [16] present analysis to calculate the worst-case latencies of CAN messages. This analysis is based on the standard fixed priority response-time analysis.

The worst-case response-time of message is the longest time between the queueing of a message and the time message reaches at destination nodes. In case of CAN, it is defined to be composed of two delays, (i) queueing delay, q_i, (ii) transmission delay, C_i [16]. The queueing delay is the longest time that a message can be queued at a node and be delayed because of other higher- and lower- priority messages are being sent on the bus. The transmission delay is the actual time taken to send the message on the bus. Thus, worst-case response-time is defined as:

$$R_i = q_i + C_i \tag{1}$$

The queueing time, q_i is itself composed of two times, (i) longest time that any lower priority message can occupy the bus, B, (ii) the longest time that all higher priority messages can be queued and occupy the bus before the message i is finally transmitted.

$$q_i = B_i + \sum_{j \in hp(i)} \left\lceil \frac{q_i + J_j + \tau_{bit}}{T_j} \right\rceil C_j \tag{2}$$

Where J_i is the queueing jitter of the messages, i.e., the maximum variation in the queueing time relative to T_i, $hp(i)$ is the set of messages with priority higher than i, τ_{bit} (bit-time) caters for the difference in arbitration start times at the different nodes due to propagation delays and protocol tolerances. Equation(2) is recurrence relation for q_i. Considering effect of external interference and error, the worst-case response time [19] can be given as:

$$R_i = B_i + \sum_{j \in hp(i)} \left\lceil \frac{q_i + J_j + \tau_{bit}}{T_j} \right\rceil C_j + C_i + E\left(q_i + C_i\right) \tag{3}$$

B. MIL-STD-1553B

MIL-STD-1553B is a military standard that defines the electrical and protocol characteristics for a data bus. The data bus is used to provide a medium for exchange of data and information between various nodes of a system. This standard defines requirement for digital, command/response, time division multiplexing techniques for a 1 MHz serial data bus and specifies the data bus and its interface electronics [20]. Originally, this standard was intended for Air Force applications. But with its wide acceptance and usage, it is being used in a large number of critical applications, such as, space shuttles, space stations, surface ships, submarines, helicopters, tanks, subways and manufacturing production lines.

A summary of the characteristics of MIL-STD-1553B is given in Table 1 [21].

The standard defines four hardware elements. These are:

1) Transmission media
2) Remote terminals
3) Bus controllers
4) Bus monitors

A typical network consisting of a bus controller and a remote terminal with dual redundant bus is shown in Figure 3.

The control, data flow, status reporting, and management of the bus are provided by three word types:

1) Command words
2) Data words
3) Status words

Word formats are shown in Figure 4.

The primary purpose of the data bus is to provide a common media for the exchange of data between terminals of system.

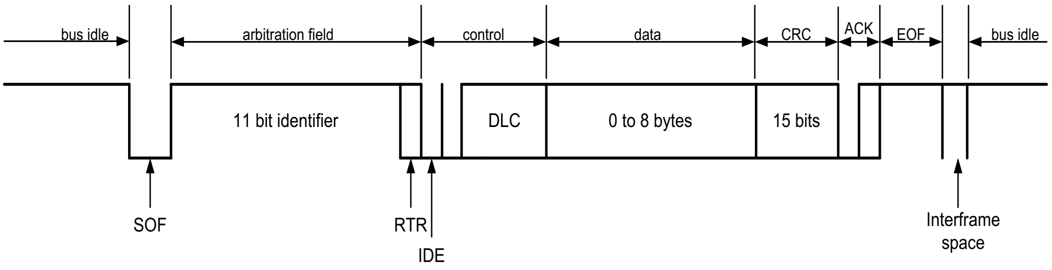

Fig. 2. CAN message format

TABLE I
CHARACTERISTICS OF MIL-STD-1553B

Data Rate	1 MHz
Word Length	20 bits
Data Bits/Word	16 bits
Message Length	Maximum of 32 data words
Transmission Technique	Half-duplex
Operation	Asynchronous
Encoding	Manchester II bi-phase
Protocol	Command/response
Bus Control	Single or Multiple
Fault Tolerance	Typically Dual Redundant, second bus in ”Hot Backup” status
Message Formats	Bus Controller to terminal Terminal to Bus Controller Terminal to Terminal Broadcast System control
Number of Remote Terminals	Maximum of 31
Terminal Types	Remote terminals Bus Controller Bus Monitor
Transmission Media	Twisted shielded pair
Coupling	Transformer and direct

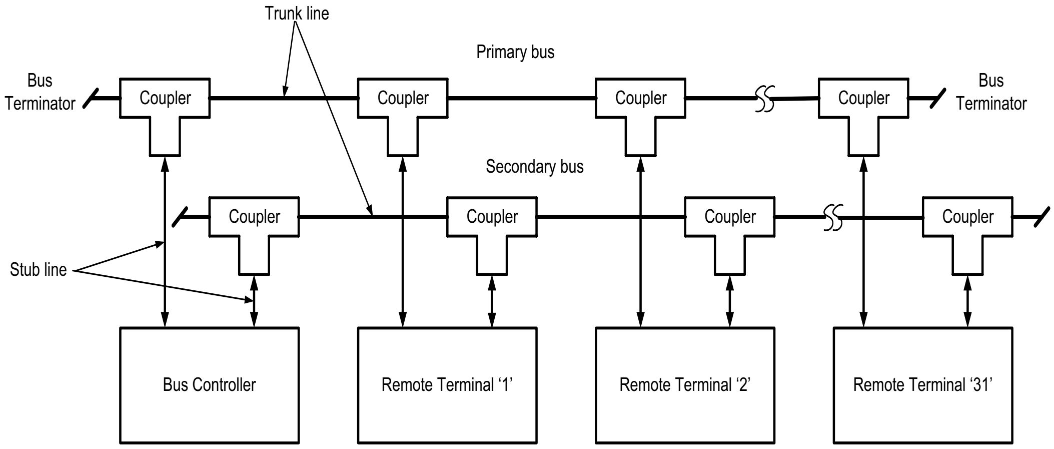

Fig. 3. MIL-STD-1553B network

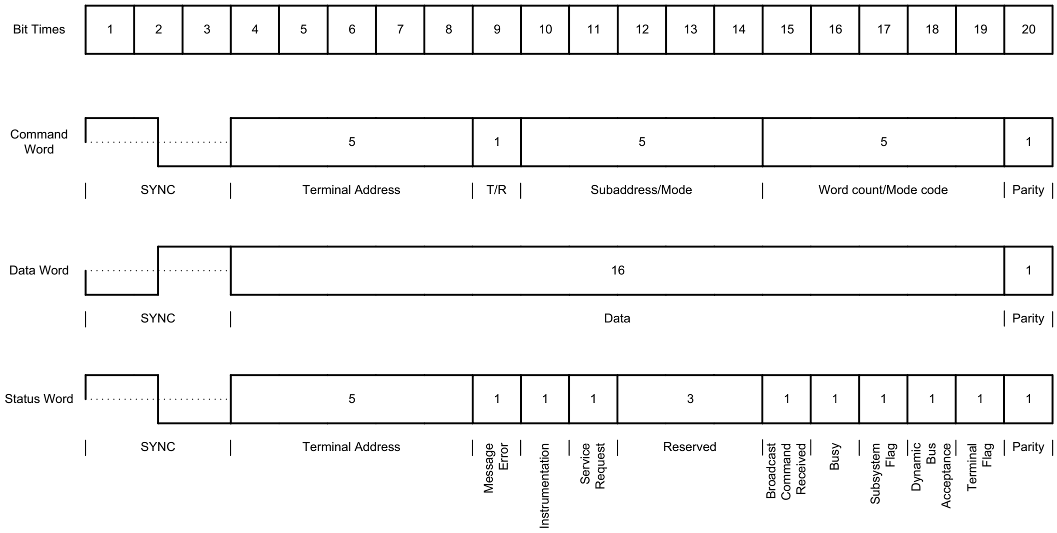

Fig. 4. Messages formats

The exchange of data is based on message transmission formats. The standard defines ten types of message transmission formats. All of these formats are based on the three word types defined in succeeding paragraph. A RT-RT information transfer format is shown in Figure 5.

Intermessage gap shown in Figure 5, is the minimum gap time that the bus controller shall provide between messages. Its typical value is $4\mu s$. Response time is time period available for terminals to respond to a valid command word. This period is of 4 to $12\mu s$. A time out occurs if a terminal do not respond within no-Response timeout, it is defined as the minimum time that a terminal shall wait before considering that a response has not occurred, it is $14\mu s$.

1) Delay analysis: As per [14], the delay for cyclic service network can be simply modeled as a periodic function such that $\tau_k^{SC} = \tau_{k+1}^{SC}$ and $\tau_k^{CA} = \tau_{k+1}^{CA}$ where τ_k^{SC} and τ_k^{CA} are the sensor-to-controller delay and the controller-to-actuator delay at k^{th} sampling time. This argument about delay works perfectly in ideal cases. In practice, NCS may experience small variations on periodic delays due to several reasons. For example, the mismatches in clock generators on a controller and a remote terminal may result in delay variation.

III. Network delay

A. CAN

Figure 6 presents the communication procedure of CAN where the MAC is non-destructive CSMA/CD. In CAN arbitration during collision is based on message identifier also called the bit-wise arbitration [17].

Flow chart of transmission procedure of CAN is given in Figure 6. For probabilistic delay modeling, decision boxes are modeled as probabilities and predefined process boxes (i.e. wait for bus) is modeled as probability density (*pdf*). Ref. [22] gives a detailed model of CAN response-time. Evaluation of these probabilities and *pdfs* (items 1-6 of Figure 6) [22] for a specific message follows below.

Let a set of messages of the system should be denoted by M. Parameter are estimated for a message m from set M. For parameter estimation, $i \in M$ stands for all messages of set except message m for which the parameter is being estimated.

Probability of finding medium free(1): Probability that a message finds the medium free when it gets queued, is estimated based on the utilization of network. This utilization is by other messages of network.

$$P_{free}^{m} = 1 - \sum_{i \in M} \frac{C_i}{T_i} \tag{4}$$

Probability of finding the bus free, by a message m is the complement of utilization. This is because in a closed system (with fixed number of messages/customers) with n messages, a message on arrival finds the system in equilibrium with $n-1$ messages [23].

Probability of no collision with high priority message(2): When network is free, a node with ready message can start transmission. Node will abort and back off transmission if it finds any higher priority message concurrently being transmitted. This can happen if a node start transmitting a higher priority message within the collision window τ_w.

$$\begin{gathered} P_{Suc}^{m} = \prod_{\substack{i \in M \\ i \in hp(m)}} P_C^i \\ \text{where} \\ P_C^i = \text{Prob}\left[\text{non occurrence message i in } \tau_w\right] \\ = 1 - \left(\frac{1}{T_i} \times \tau_w\right) \end{gathered} \tag{5}$$

Blocking time (4): A message in queue can be blocked by any message under transmission by any of the other nodes. This is because CAN messages in transmission cannot be preempted. *pdf* of this blocking time $p_b(t)$ is obtained by following steps:

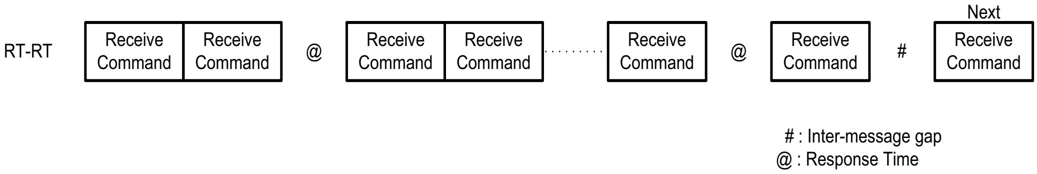

Fig. 5. RT-RT infromation transfer format

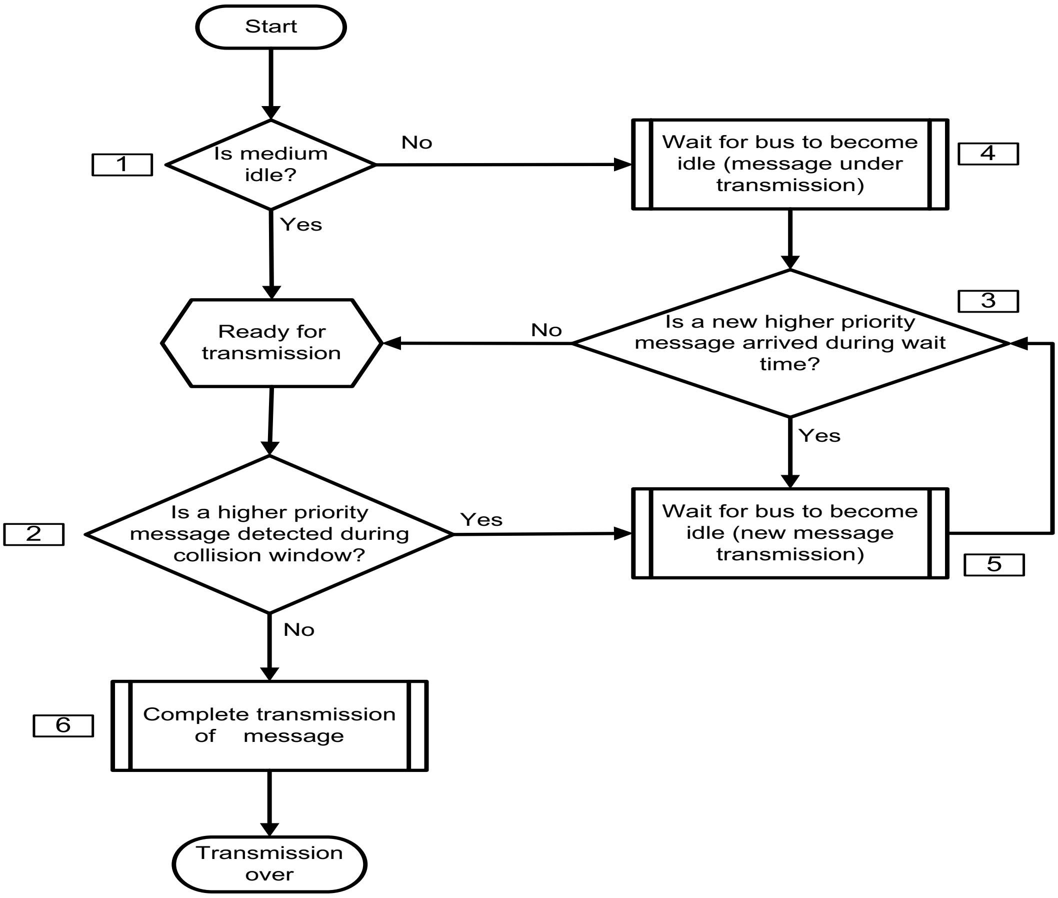
Start
Is medium idle?
1
No
Yes
Wait for bus to become idle (message under transmission)
4
Is a new higher priority message arrived during wait time?
3
No
Yes
Ready for transmission
Is a higher priority message detected during collision window?
2
Yes
No
Wait for bus to become idle (new message transmission)
5
Complete transmission of message
6
Transmission over

Fig. 6. Transmission procedure of CAN

1) Find the ratio r_i of all the messages. $r_i = \frac{1}{T_i \sum\limits_j \frac{1}{T_j}}$, for $i, j \in M$
2) Construct a pdf, $p(t)$ of total blocking time by other messages
$$p\left(t\right) = \sum_{i \in M} r_i \cdot \delta\left(t - C_i\right) \qquad (6)$$
3) Message can get ready at any time during the blocking time with equal probability. So, effective blocking time is given by following convolution

$$p_b^m\left(t\right) = \frac{1}{\max\left(C_i\right)} \int\limits_0^t p\left(\tau\right) \left[\begin{array}{c} U\left(t+\tau\right) \\ -U\left(t+\max\left(C_i\right)+\tau\right) \end{array} \right] d\tau \qquad (7)$$

Blocking time by high priority message (5): When the ready node finds bus free and start transmission of ready message, then if within the collision time window, another node starts transmitting a higher priority message, node backs off. And the message need to wait till the time of completion of this transmission. pdf of blocking time by high priority message, p_{bhp} is obtained by following steps 1-3 of *Blocking time* with one variation, instead of all messages only high priority message of network are considered.

1) Find the ratio r_i^H of all the messages. $r_i^H = \frac{1}{T_i \sum\limits_j \frac{1}{T_j}}$, for $i, j \in M, \quad i \in hp(m)$
2) pdf of blocking after back off is given by
$$p_{bhp}^m\left(t\right) = \sum_{i \in hp(m)} r_i^H \cdot \delta\left(t - C_i\right) \qquad (8)$$

where $\delta(\cdot)$ is Dirac delta function.

Probability of no new higher priority message arrival in BlockTime (3-4): When waiting time is 4 of flow chart.

$$\begin{aligned} P^m_{T_B} &= \prod_{\substack{i\in M\\ i\in hp(m)}} P^i_{T_B} \\ &\text{where} \\ P^i_{T_B} &= \text{Prob}\left[\text{non-occurrence of message i in } t_B\right] \\ &= 1-\left(\tfrac{1}{T_i}\times E[t_B]\right) \end{aligned} \tag{9}$$

Probability of no new higher priority message arrival in BlockTimebyNew (3-5): When waiting time is 5 of flow chart.

$$\begin{aligned} P^m_{T_{Bhp}} &= \prod_{\substack{i\in M\\ i\in hp(m)}} P^i_{T_{Bhp}} \\ &\text{where} \\ P^i_{T_{Bhp}} &= \text{Prob}\left[\text{non-occurrence of message i in time } t_{Bhp}\right] \\ &= 1-\left(\tfrac{1}{T_i}\times E[t_{Bhp}]\right) \end{aligned} \tag{10}$$

Time required to reach state *Ready for transmission* from state numbered 5 in Figure , may require a number of iterations from state 3 to 5. Let it takes i iterations, then it can be modeled as a r.v. with $pdf B^m_{hp}(i,t)$

$$B^m_{hp}(i,t) = \left[\left(1-P^m_{T_{Bhp}}\right)^{i-1} P^m_{T_{Bhp}}\right]\eta^i(t) \tag{11}$$

where

$$\begin{aligned} \eta^i(t) &= \eta^{i-1}(t)\otimes p^m_{bhp}(t) \\ \eta(t) &= p^m_{bhp}(t) \end{aligned}$$

$$B^m_{hp}(t) = \sum_i B_{hp^m}(i,t) \tag{12}$$

Symbol $\otimes$ is used to denote convolution.

In the same way, time to reach state 6 from state *Ready for transmission* may require a number of iteration of states 5 and 3. This time is modeled as a r.v. with $pdf t^m_{rdy}(i,t)$

$$t^m_{rdy}(i,t) = \left[(1-Pfree)^{i-1}Pfree\right](B^m_{hp})^{i-1}(t) \tag{13}$$

where

$$\begin{aligned} (B^m_{hp})^i(t) &= (B^m_{hp})^{i-1}(t)\otimes B^m_{hp}(t) \\ (B^m_{hp})^0(t) &= \delta(t) \end{aligned}$$

$$t^m_{rdy}(t) = \sum_i t^m_{rdy}(i,t) \tag{14}$$

From the instant message is queued, it can reach state 6 either directly, via 1-2, or via 1-4-3-2, or via 5. All these paths have different associated time delay. So, using total probability theorem [24], blocking time distribution is given as:

$$\begin{aligned} b^m(t) &= P^m_{free}t_{rdy}(t) \\ &+ \left(1-P^m_{free}\right)P^m_{T_B}\left[p^m_b(t)\otimes t_{rdy}(t)\right] \\ &+ \left(1-P^m_{free}\right)\left(1-P^m_{T_B}\right)\left[p^m_b(t)\otimes p^m_{bhp}(t)\otimes t_{rdy}(t)\right] \end{aligned} \tag{15}$$

TABLE II
SAE CAN MESSAGES

Message ID	No. of bytes	Ti (ms)	Di (ms)
17	1	1000	5
16	2	5	5
15	1	5	5
14	2	5	5
13	1	5	5
12	2	5	5
11	6	10	10
10	1	10	10
9	2	10	10
8	2	10	10
7	1	100	100
6	4	100	100
5	1	100	100
4	1	100	100
3	3	1000	1000
2	1	1000	1000
1	1	1000	1000

Network delay time density of a message is sum of its queueing time and transmission time.

$$d^m(t) = b^m(t)\otimes C_m\delta(t) \tag{16}$$

Network delay time distribution of message m, denoted as $D^m(t)$ is integration of delay time density $d^m(t)$

Network delay on CAN has variation mainly due to traffic and priority among messages. Message with lower priority has higher variation than high priority messages. Network delay of a message in one cycle is not related with the delay in the previous cycle, as the interacting traffic is independent. So, network delay in each cycle is independent and follow the *pdf* given by (16).

B. Example

To illustrate the method, we take a standard message set of SAE [25]. The list of messages along with other details are shown in Table II.

Using the proposed method, parameters are calculated for each message considering worst-case transmission time. CAN operating speed is 125Kbps (bit-time=7.745ms) [25]. For *pdf* ($B_{hp}(t)$ and $t_{rdy}(t)$) estimation number of attempts i is truncated such that accumulated probability is ≥ 0.9999. The probability density function of blocking time and blocking time by high priority message, random variables for message ID 9 is shown in Figure 7.

Once all the parameter values are available, delay time distribution is estimated.

The delay time distribution of 3 messages (message ID=1,9 and 17) is shown in Figure 8. From the response-time distribution, probability of meeting two worst-case times R_i and R^{sim}_i [25] is evaluated. R_i is the worst-case response-time from analysis while R^{sim}_i is from simulation. Column 2 and 4 in Table III give these values, corresponding probabilities from response-time distribution analysis is given in column 3 and 5. Let for all messages, probability of meeting a response-time is fixed to 0.999, then corresponding time value from response-time distribution is given in column 6.

In Figure 8 the offset at time axis is due to blocking when the message is queued. It is same for all messages irrespective

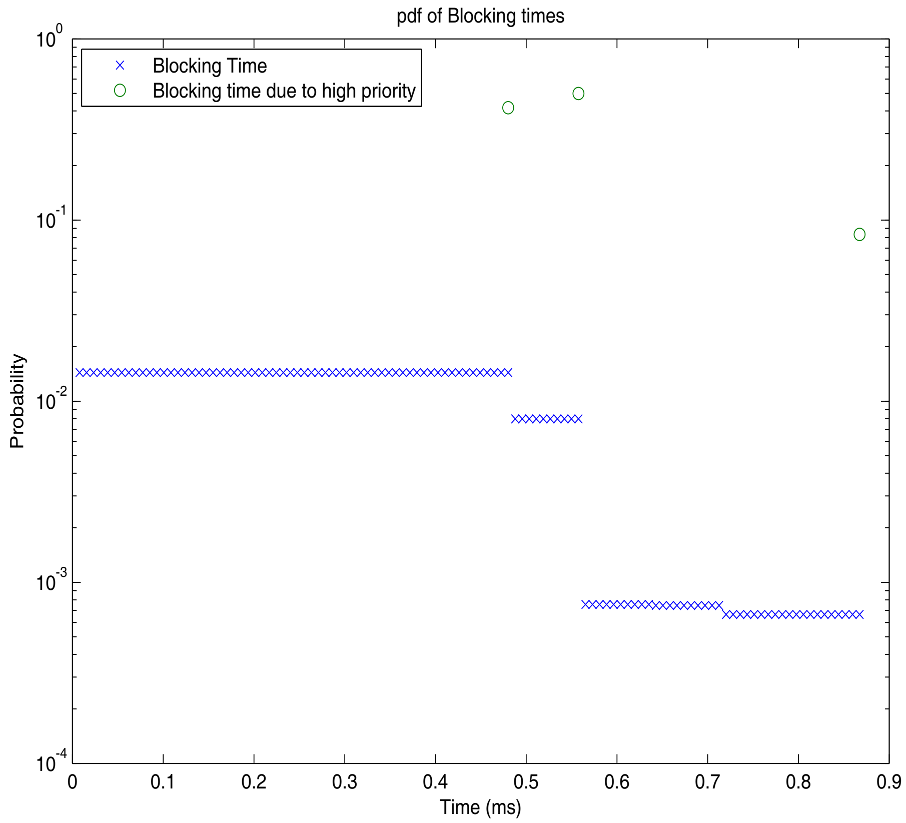

Fig. 7. pdf of blocking times

of message priority, because CAN message transmissions are non-preemptive. Slope of response-time curves are different. Slopes are dependent upon the message priority, higher the message priority higher is the slope.

Response-times from worst-case analysis are giving upper bound on network delay time, so probability at these times from delay time distribution is expected to be very high or even 1. Values in column 3 of Table III confirms this. Worst-case response-time from simulation is obtained from a limited simulation (2000000 ms [25]). Hence there is no consistence probability at these response-times.

Network delay time of message with probability 0.999, is comparable for higher priority messages, while it is almost 25% of worst-case for lower priority. This is because worst-case analysis assumes all higher priority message will get queued deteministically, while delay time distribution gives probabilistic treatment to this.

C. MIL-STD-1553B

Consider a network with 2 nodes as shown in Figure 9. BC of the network controls transfer of data on network. The network delay for data transfer from node A to B is defined as

$$\tau_{AB} = t^A_{suc} - t^A_Q \tag{17}$$

In this equation τ_{AB} is the network delay experienced by data at node A for transfer to node B. t^A_{suc} is the time of successful transfer of data from node A, t^A_Q is the time of queuing of data by node A for transfer to B.

Node A is allowed to transmit its data to B periodically under the command of BC. As node A and BC are not synchronized, waiting time (queuing time to the time of actual start of transmission) will have uniform distribution. This uniform distribution has range $\left(0, \tau^{AB}_{mil}\right)$. τ^{AB}_{mil} is time period or cycle time of data transfer from A to B. Once node A gets turn for data transfer to B, it starts putting the frame, then transmission delay has two components, frame size and prorogation delay. Frame size is proportional to number of bytes to be transferred, while propagation delay is because media length connecting nodes A and B. Now in terms of waiting time, frame time and propagation time, network delay can be written as:

$$\tau_{AB} = \tau^A_{wait} + \tau^A_{fr} + \tau^A_{prop} \tag{18}$$

For a given data and pair of nodes framing time and propagation time are constant. The sum of these two is referred

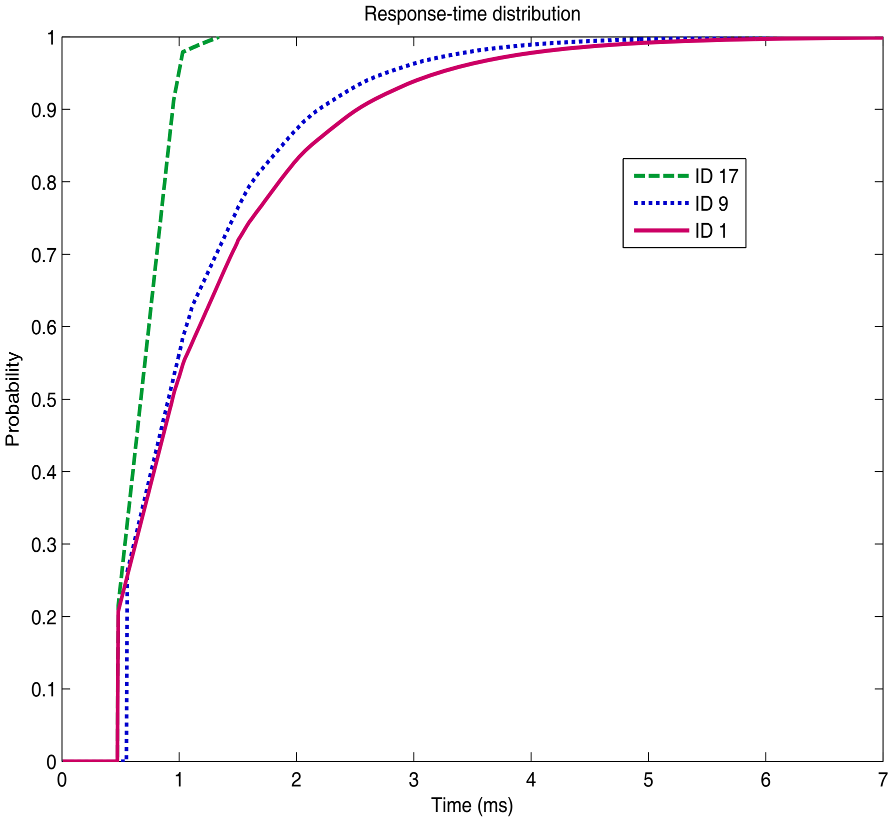

Fig. 8. Response-time distribution

as transmission time. With the assumption that data is not corrupted during framing or propagation (i.e no retransmission), transmission time is constant. So, network delay is sum of a random (waiting time) and fixed (transmission time) quantity.

Let transmission time of the data transferred from node A to B is denoted as τ_{tx}^{AB}. Then *pdf* of network delay is given as:

$$d^{AB}\left(t\right) = \frac{1}{\tau_{mil}^{AB}} \int\limits_{0}^{t} \left[U\left(\tau\right) - U\left(\tau - \tau_{mil}^{AB}\right)\right] \delta\left(t - \tau + \tau_{tx}^{AB}\right) d\tau \tag{19}$$

With the above equation it is clear under the assumption of no retransmission, network delay has time shifted uniform distribution. This time shift is by fixed transmission time.

As discussed earlier, network delay in case of MIL-STD-1553B has variation mainly because of mismatch in clocks of remote terminals and bus controllers. Most of the cases, this mismatch is very small. So, network delay in consecutive cycles will have small variations, means network delay is dependent on previous cycle. Let network delay of data from node A to B is measured in cycle k, then network delay in cycle $k+1$ is going to be near to the measured value with a high probability. Although network delay in a randomly chosen cycle will follow the *pdf* given by (19), but network delay in consecutive cycle will be dependent on current cycle network delay.

Let p_{pre} is the probability that network delay in any cycle is same as in the previous cycle, then network delay in successive cycles is given as:

$$d^{AB}\left(t, i\right) = \delta\left(t - x\right) \tag{20}$$

where x is the network delay in cycle i

$$d^{AB}\left(t, i+1\right) = p_{pre}\delta\left(t - x\right) + \left(1 - p_{pre}\right) d^{AB}\left(t\right) \tag{21}$$

and

$$\begin{aligned} d^{AB}\left(t, i+2\right) &= p_{pre} d^{AB}\left(t, i+1\right) + \left(1 - p_{pre}\right) d^{AB}\left(t\right) \\ &= p_{pre}^{2}\delta\left(t - x\right) + \left(1 - p_{pre}\right)\left(1 + p_{pre}\right) d^{AB}\left(t\right) \end{aligned}$$

similarly

$$\begin{aligned} d^{AB}\left(t, i+n\right) &= p_{pre} d^{AB}\left(t, n-1\right) + \left(1 - p_{pre}\right) d^{AB}\left(t\right) \\ &= p_{pre}^{n}\delta\left(t - x\right) + \left(1 - p_{pre}^{n}\right) d^{AB}\left(t\right) \end{aligned} \tag{22}$$

If $p_{pre} = 1$, i.e. there is no mismatch in clocks, then n^{th} cycle will also have the same network delay as i^{th} cycle. When $p_{pre} < 1$, then network delay in n^{th} cycle will be given be $d^{AB}(t)$.

TABLE III
CALCULATED PARAMETER VALUES

Priority (ID)	R_i [25]	$D(R_i)$	R_i^{sim} [25]	$D(R_iSim)$	$t_{min}[D(t_{min}) = 0.999]$
17	1.416	1.0000	0.680	0.5123	1.324
16	2.016	1.0000	1.240	0.9883	1.402
15	2.536	0.9997	1.720	0.9920	2.238
14	3.136	0.9997	2.280	0.9956	2.742
13	3.656	0.9995	2.760	0.9959	3.369
12	4.256	0.9995	3.320	0.9967	3.919
11	5.016	0.9991	4.184	0.9965	4.957
10	8.376	1.0000	4.664	0.9968	5.553
9	8.976	1.0000	5.224	0.9976	5.925
8	9.576	1.0000	8.424	0.9999	6.374
7	10.096	1.0000	8.904	0.9999	6.831
6	19.096	1.0000	9.616	0.9999	7.094
5	19.616	1.0000	10.096	1.0000	6.940
4	20.136	1.0000	18.952	1.0000	6.978
3	28.976	1.0000	18.952	1.0000	7.172
2	29.496	1.0000	19.432	1.0000	7.025
1	29.520	1.0000	19.912	1.0000	7.032

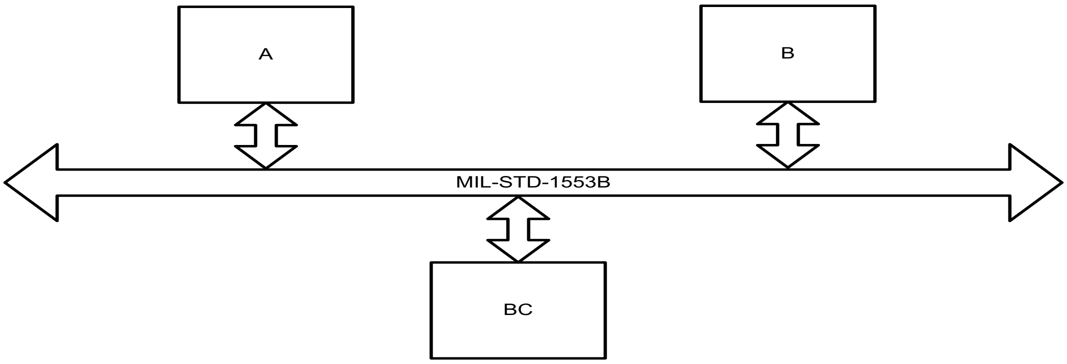

Fig. 9. Two nodes of a MIL-STD-1553B network

IV. EFFECT OF NETWORK REDUNDANCY

In this section effect of hot network redundancy on network delay is analyzed. Here n number of independent networks connects all the nodes. All nodes have independent network interface units (NIU) for each networks. A node after computation put the data for transmission on all the NIUs simultaneously. NIUs independently transfer the data on their respective networks.

A. CAN

As discussed earlier, CAN frame does not have source and destination identification. It only has message identifier, which is used to assign priority to message and receiving nodes used this for filtering relevant message for them. A typical diagram of a node with multiple CAN networks is shown in Figure 10. Node NIUs does not perform any computation except framing whose delay time can be neglected. With the assumption that every node put their data to all NIUs simultaneously, and under condition that CAN bus utilization is < 1, all networks will work in perfect synchronization. So, redundancy at CAN network level does not affect network delay. It only provides fault tolerance against failure on bus level or NIU level.

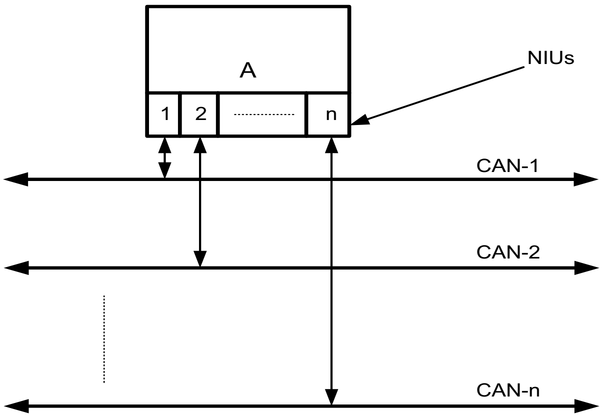

Fig. 10. Redundancy at network (CAN)

B. MIL-STD-1553B

A generic diagram of two nodes of a system connected via n independent MIL-STD-1553B networks is shown in Figure 11. A node puts the data for transfer on all the NIUs simultaneously and cycle time of data transfer on all networks is same. But due to phase difference among BCs and node network delay of each network will be independent and *s*-identical. Let network delay for data from node A to B on

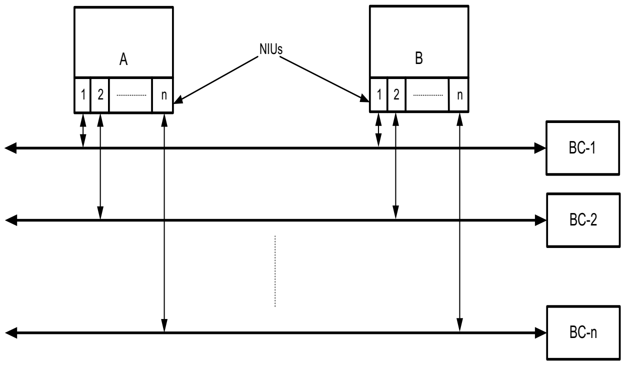

Fig. 11. Redundancy at network (MIL)

network i is denoted as ${}^{i}\tau_{AB}$, then

$$ {}^{i}\tau_{AB} = {}^{i}\tau_{wait}^{A} + {}^{i}\tau_{tx}^{AB} \tag{23} $$

Let network delay for data from node A to B is defined as minimum of all n network delays.

$$ \tau_{AB} = \min\{{}^{1}\tau_{AB}, \ldots {}^{i}\tau_{AB}, \ldots, {}^{n}\tau_{AB}\} \tag{24} $$

As discussed earlier network delays on redundant networks are independent and identical. Evaluation of network delay is order statistics [24]. This problem can be expressed as Ist order statistics.

$$ D_{red}^{AB}(t) = 1 - \left[1 - D^{AB}(t)\right]^{n} \tag{25} $$

where

$$ D^{AB}(t) = \int_{0}^{t} d^{AB}(\tau)\, d\tau $$

and $d^{AB}(t)$ is the *pdf* of network delay of all individual networks.

C. Example

Consider a NCS with 3 independent and active MIL-STD-1553B networks. Node A is to transmit 8 bytes of data to node B, and network cycle time for transfer from A to B is 50ms. Neglecting propagation delay, transmission time is 1ms (4 data words, status word and response time). The delay time distribution of data transfer from A to B is evaluated using (25).

The plot of delay distribution is given in Figure 12. It shows plot of network delays of single as well as network with 3 redundancy. As network delay density of a single network is uniformly distributed, its distribution is linear while network with redundancy have different distribution. Network redundancy in addition to fault-tolerance improves delay characteristics of the system.

It is seen that hot redundancy in case of MIL-STD-1553B reduces the network delays, while in case CAN it does not.

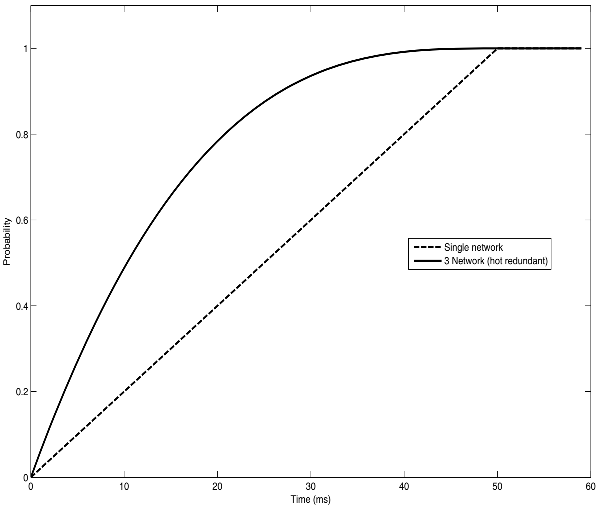

Fig. 12. Delay time distribution of data transfer

V. Sample to actuation delay and response-time

In NCS, two parameter of importance are sample to actuation delay and response-time. First is the time difference of actuating action to the corresponding sampling time. While the second is time taken by NCS to react (or respond) to an action of physical process.

Assuming that all nodes are time-triggered, i.e. they sample inputs from physical process or network periodically and generate output to physical process or network after computation time. So for a node i, sampling period is denoted by τ_{samp}^{i} and delay after sampling is computation delay is denoted as τ_{comp}^{i}.

In case of NCS, sampling of process input is carried out by sensor node and actuation for corrective action is done by actuator node. So, sample to actuation delay is given by:

$$ \begin{aligned} \tau_{a-s} &= \tau_{comp}^{S} + \tau_{SC} + \tau_{samp}^{C} \\ &+ \tau_{comp}^{C} + \tau_{CA} + \tau_{samp}^{A} + \tau_{comp}^{A} \end{aligned} \tag{26} $$

Computation delay is negligible for some nodes (sensor, actuator). For controller nodes it is finite for analysis purpose it can be assumed constant. Sampling time due to phase difference is modeled as uniform distribution.

$$ \tau_{samp}^{i} \models Unif\left(0, t_{samp}^{i}\right) \tag{27} $$

So, *pdf* of sample to actuation delay is convolution of all the variable of above equation:

$$ \begin{aligned} & d^{a-s}(t) = d^{SC}(t) \otimes unif\left(0, t_{samp}^{C}\right) \\ & \otimes \delta\left(t - t_{comp}^{C}\right) \otimes d^{CA}(t) \\ & \otimes unif\left(0, t_{samp}^{A}\right) \end{aligned} \tag{28} $$

Since sensor node samples the physical inputs periodically, response-time density is given as:

$$ r(t) = unif\left(0, t_{samp}^{S}\right) \otimes d^{a-s}(t) \tag{29} $$

VI. Effect of node redundancy

In NCS, to make the system fault-tolerant, its node group (sensor, controller and actuator) can have redundancy. This redundancy could of any form and type, active/passive, hot, cold or warm etc. In this paper *MooN* (M-out-of-N) redundancy is considered.

All nodes in a group independently sample the physical parameter. Parameters from network are common to all the nodes of a group. For parameters from network, nodes might wait for copies of same/similar data from multiple nodes of the sending group. For example, in case sensor group is having *2oo3* redundancy, controller nodes might wait for a data from two different nodes of sensor group for processing. As all nodes are time-triggered, a controller node performs *2oo3* of recent same kind of data from 3 nodes of sensor group.

Let redundancy of each group is denoted as M^iooN^i and $i \in \{S, C, A\}$. Receiving nodes of a data wait for data from sending group, i from at least M_i nodes. To model response-time in this scenario, we define group delay time. Group delay time of a group j, is defined as time difference of successful transmission of output data by M_j nodes from occurrence of an event at physical process or receipt of data at nodes of group j. Where i is the sending group.

$$\tau^i = \left(m^i\right)^{th} median \left\{ \begin{array}{l} \left(\tau^i_{samp} + \tau^i_{comp} + \tau_{ij}\right)_1, \ldots, \\ \left(\tau^i_{samp} + \tau^i_{comp} + \tau_{ij}\right)_{N^i} \end{array} \right\} \tag{30}$$

This is again a order statistics problem. As sample, compute and network delay of each node independent and identically distributed $(i \cdot i \cdot d \cdot)$, $d^i_1(t)$ is used to denote *pdf* of this sum. Corresponding distribution function is denoted as $D^i_1(t)$. So, group delay is given as:

$$D^i(t) = \sum_{n=M^i}^{N^i} \left[D^i_1(t)\right]^n \left[1 - D^i_1(t)\right]^{N^i - n} \tag{31}$$

With this response-time model of the system becomes as shown in Figure 13.

$$r(t) = d^S(t) \otimes d^C(t) \otimes d^A(t) \tag{32}$$

VII. Conclusion

Network-induced delays are important for NCS as they are the cause of system degradation and sometime system's stability. NCS control algorithms considering probabilistic network delay have better control QoP. In this paper, methods to probabilistically model network-induced delay of two field bus networks, CAN and MIL-STD-1553B is proposed. CAN is random access network, probabilistic model requires system parameters as probability and random variable. Effect of hot network redundancy on system delay time of these two networks is analyzed. The method is extended to evaluate sample-to-actuation delay and response-time of NCS. A fault-tolerant NCS has a number of nodes within sensor, controller and actuator groups, effect to these redundancy on system response-time is also analyzed.

Acknowledgment

The authors would like to thank Prof. Varsha Apte (IITB) for reviewing the initial work and critical guidance to make this work possible. We also, thanks Shri U. Mahapatra, P.P. Marathe and M.K. Singh (all from BARC) for their continual support and encouragement during the work; and reviewer(s) for going through the work and providing critical comments.

References

[1] J. Yepez, P. Marti, and J. M. Fuertes, "Control loop performance analysis over networked control systems," in *Proceedings of IECON2002*, 2002.

[2] T. Yang, "Networked control systems: a brief survey," *IEE Proc.-Control Theory Applications*, vol. 153, pp. 403–412, July 2006.

[3] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control System Magazine*, pp. 84–99, Feburary 2001.

[4] J.-P. Richard, "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, vol. 39, pp. 1667–1694, 2003.

[5] H. Yang and B. Sikdar, "Control loop performance analysis over networked control systems," in *Proceedings of ICC 2007*, pp. 241–246, 2007.

[6] B. Lu, "Probabilistic design of networked control systems with uncertain time delay," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings, 9 Part A*, pp. 355–362, 2008.

[7] D. Huang and S. Nguang, "State feedback control of uncertain networked control systems with random time-delays," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings, 9 Part A*, pp. 3–362, 2008.

[8] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, pp. 57–64, 1998.

[9] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Transaction on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.

[10] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transaction on Control System Technology*, vol. 10, pp. 438–446, 2002.

[11] J. Nilsson, *Real-Time Control Systems with Delays*. PhD thesis, Lund Institute of Technology, Sweden, 1998.

[12] F.-L. Lian, *Analysis, Design, Modeling, and Control of Networked Control Systems*. PhD thesis, University of Michigan, 2001.

[13] F.-L. Lian, J. Moyne, and D. Tilbury, "Performance evaluation of control networks: Ethernet, controlnet, and devicenet," *IEEE Control System Magazine*, vol. 21, pp. 66–83, 2001.

[14] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[15] K. W. Tindell, H. Hansson, and A. J. Wellings, "Analyzing real-time communications: Controller area network (CAN)," in *Proceeding of Real-time Symposium*, pp. 259–263, Dec. 1994.

[16] K. Tindell, A. Burns, and A. Wellings, "Calculating controller area network (can) message response times," *Control Engineering Practice*, vol. 3, no. 2, pp. 1163–1169, 1995.

[17] M. Farsi, K. Ratcliff, and M. Barbosa, "An overview of controller area network," *Computing & Control Engineering Journal*, pp. 113–120, June 1999.

[18] "CAN specification 2.0. part A and B, can in automation (cia)."

[19] T. Nolte, H. Hansson, and C. Norstrom, "Minimizing can response-time jitter by message manipulation," in *Proceedings of the 8th Real-time and Embedded Technology and Application Symposium (RTAS'02)*, 2002.

[20] "MIL-STD-1553B: Aircraft internal time division command/response multiplex data bus," 30 April 1975.

[21] "MIL-STD-1553 tutorial."

[22] M. Kumar, A. K. Verma, and A. Srividya, "Response-time modeling of controller area network (can)," in *Communicated to Int. Conf. on Distributed Computing and Networking (ICDCN09)*.

[23] J. K. Muppala, V. Mainkar, K. S. Trivedi, and V. G. Kulkarni, "Numerical computation of response-time distributions using stochastic reward nets," *Annals of Operations Research*, vol. 48, pp. 155–184, 1994.

[24] K. S. Trivedi, *Probability & Statistics with Reliability, Queueing, and Computer Science Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

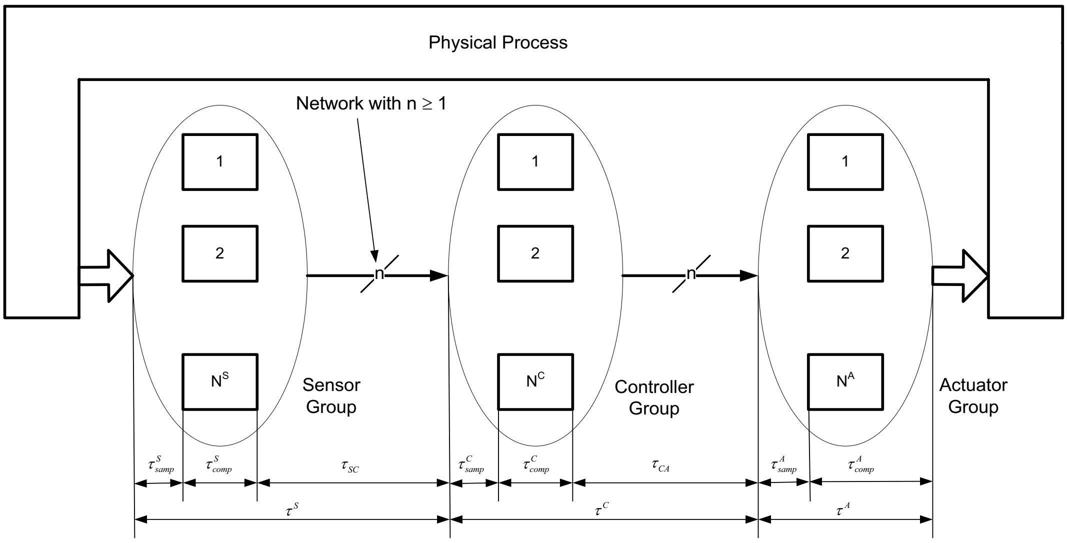

Fig. 13. NCS response-time with redundancies

[25] T. Nolte, H. Hansson, and C. Norstrom, "Probabilistic worst-case response-time analysis for the controller area network," in *Proceedings of the 9th Real-time and Embedded Technology and Application Symposium (RTAS'03)*, 2003.

A.K. Verma (B. Tech (Hons) (EE, IIT Kharagpur), Ph.D. (Engg.) (IIT Kharagpur)) has been with IIT Bombay as a faculty since 1988. He is currently a Professor in Reliability Engineering, Department of Electrical Engineering at IIT Bombay. He has supervised/co-supervised 21 PhDs and 75 Masters thesis in the area of Software Reliability, Reliable Computing, PSR, RCM and PSA. He has executed various research projects to promote industry interface and has been course co-ordinator for industry CEPs like Six Sigma, Software Inspections, Competency Tracking System and Software Reliability for IT industries. He has jointly edited books on Reliability Engineering and Quality Management, Quality, Reliability and Information Technology, and is also an author of book titled "Fuzzy Reliability Engineering-Concepts and Applications". He has over 165 publications in various journals and conferences. He has served as a Guest Editor of special issue of IETE Technical Review on Quality Management in Electronics, Telecommunication & Information Technology, 2001 and as a Guest Editor of Special Issue of International Journal of Reliability, Quality and Safety Engineering Dec 2004, June 2006 & April 2008, World Scientific. He has been a Guest Editor of special issue of DQM Communications (International Journal), 2006 & 2007 and special issue on Dependability in computing systems of International Journal of Performability Engineering, 2006 and is also a Guest Editor of special issue on Reliable Computing of International Journal of Automation and Computing (Springer, Vol.4, No.4, Oct.2007). He has been conference chairman of various International Conferences like International Conference on Quality, Reliability and Control 2001, International Conference on Multimedia and Design 2002 and International Conference on Quality Reliability and Information Technology 2003 and the Editor of the conference proceedings. He is a recipient of Leadership in Reliability Engineering Education & Research award by Society of Reliability Engineering, Quality & Operations Management.

He is a Senior Member of IEEE and Life Fellow of IETE. His areas of research in Reliability Engineering are interdisciplinary applications in Software Engineering, Computing, Maintenance and Power system.

Manoj Kumar graduated in Electronics & Communication Engg. from Faculty of Engineering & Technology, Jamia Millia Islamia, New Delhi in 1998. Since 1998, he is working as Scientific Officer at Control Instrumentation Division of Bhabha Atomic Research Centre (BARC), Mumbai. Currently he is pursuing Ph.D. degree at Indian Institute of Technology Bombay (IITB), India.

His research interest includes design and development of embedded/computer based control systems, performance, reliability and safety analysis.

A. Srividya joined IIT Bombay in 1988 and is currently Professor in Civil Engineering with research focus in the areas of Reliability Engineering and Quality Management. She has supervised/co supervised 12 PhD's and 50 Masters thesis in the area of Structural Reliability, Reliability Based Optimisation, Simulation Studies for Reliability Estimation, Quality Benchmarking Studies for Service Industries, Quality Systems and Accelerated Life Testing. She has executed various research projects to promote industry interface and has been course co-ordinator for industry CEPs like Reliability Engineering, Quality Management and Six Sigma for IT industries. She has jointly edited books on "Reliability Engineering and Quality Management", "Quality, Reliability and Information Technology", and is also an author of book titled "Fuzzy Reliability Engineering-Concepts and Applications". She has over 50 publications in various international and national journals. She has published over 100 papers in conferences. She is a life member of ISTE and ISPS. She has been conference chairperson of International Conference on Reliability, Safety & Hazard 2005 (Advances in Risk Informed Technology) at Hotel the Leela, Mumbai, Dec 01-03, 2005. She was instrumental in editing and reviewing the proceedings of various International Conferences like International Conference on Quality Reliability and Control 2001, International Conference on Multimedia and Design 2002 and International Conference on Quality Reliability and Information Technology 2003 and International Conference on Reliability, Safety & Hazard 2005 and a Editor of the proceedings (Narosa). She was also the Conference chair for International Conference on Quality, Reliability and Infocom Technology held at INSA, New Delhi (Dec.2-4), 2006. She was a Conference Chair of the International Conference on Reliability, Safety and Quality Engineering, held at NPCIL, Mumbai from Jan. 5-7, 2008 and Editor of the proceedings (Macmillan). She is a recipient of "Leadership in Reliability Engineering Education & Research award" by Society of Reliability Engineering, Quality & Operations Management.