VERIFYING QUALITY OF SERVICE OF ARCNET BASED ATOMOS COMMUNICATION SYSTEM FOR INTEGRATED SHIP CONTROL

Niels Nørgaard Nielsen, M.Sc.E.E., Stud Ph.D.
Jens Dalsgaard Nielsen, M.Sc.E.E., Associate Professor
Henrik Schiøler, M.Sc.E.E., Associate Professor
Ole Brun Madsen, Professor
Department of Control Engineering
Institute of Electronic Systems
Aalborg University, Denmark

Abstract: As part of the ATOMOS project (Funded by EU, DG VII) a reliable communication system with predictable behaviour has been designed. The selected solution is a network based on redundant ARCnet segments extended with an EN50170 compliant fieldbus based layer on top of an ARCnet SAP (service access point) layer. An important characteristic of the communication system is that the functionality and timing must be verifiable in order to satisfy requirements from classification companies like Lloyds and Norsk Veritas. By including Service Categories, Traffic Descriptors and Quality of Service concepts in the system it is made possible to measure network performance and hereby obtain verifiable methods for traffic planning and network surveillance. This paper describes the design of a simulation system to be used for testing and verifying the communication on a single ARCnet segment throughout the development phase and to verify real-world integration of the communication system and the applications connected to the network.

1. INTRODUCTION

As the automation of control and information systems onboard ships has evolved the dependence on reliable information flow between distributed components of the system has increased. The requirements to the network characteristics in control applications are primarily temporal in contrast to existing administrative networks where the primary requirement is reliable data transfer. In order to handle the temporal requirements there has been a growing demand from the industry for conformance classes as a design aid and from classification societies for classification and verification of the communication infrastructure onboard ships.

The EU funded ATOMOS project (subtask 2312) has been dealing with the development of a network to be used as the basic building block for a reliable communication link between system components in an Integrated Ship Control (ISC) system. This work has resulted in the selection of an dual-ARCnet (ARCNET Local Area Network Standard, 1992) based network and the construction of a conceptual framework for the communication system. The background for the selection of ARCnet and the modelling paradigms used in the ATOMOS project is described in (Nielsen, 1998).

The research area of computer network analysis has spread over a wide range of disciplines starting out with Erlangs queuing theory for telephone networking early this century. These methods yields a stochastic approach for determining queue behaviour in queuing systems with stochastic traffic arrival and stochastic service. The queuing theory has been used as the fundament for network research during the last decades. The stochastic methods have a weakness when applied to network systems where deterministic traffic characteristics are required.

In order to minimise this problem several network standards have been designed that support predictable traffic behaviour. One of these is the connection oriented Asynchronous Transfer Mode (ATM) protocol where a host requires a virtual connection with a given set of properties. If the network is capable of offering the requested properties it accepts the connection otherwise it rejects it. Another approach has been taken in the Time Triggered Protocol (TTP) where the bandwidth is divided between the nodes in predefined time slots (which requires all traffic to be scheduled in advance). TTP is primarily intended for usage in process control environments where the traffic is typically periodic formed by e.g. sampled signals.

The task of verifying ISC communication systems implies methodologies for network as well as application verification due to the fact that applications have a direct impact on network functionality and vice versa. Therefore the verification methods must integrate the network and the applications in order to ensure that mutual influence is covered sufficiently.

As the analytic methods especially for verification of temporal behaviour in networked systems are limited to less-realistic special cases it has been chosen to use simulation as a verification methodology.

This paper contains a description of the basic communication system that has been chosen in the ATOMOS project. A hierarchical model covering the system components from the applications to the network in the communication system is described. Furthermore companion standards for the traffic characteristics of each component in the communication system are defined by defining a set of standardised service categories, traffic descriptors and quality of service parameters.

A framework for a scalable simulation system for the ATOMOS communication system is described and some simulation examples are shown.

2. TECHNOLOGY

2.1 The ATOMOS network system

ATOMOS is an EU funded series of projects originated in 1992. In broad terms the goal is to increase the competitiveness of the European fleet by use of modern technology. The ATOMOS deals with a large number of different tasks to reach goals like: wheel house innovation, infrastructure development, conceptual standards for SCC design (see www.atomos.org).
AAU has been engaged in developing of high efficient, reliable, verifiable infrastructure. As a result an EN 50 190 vol. 1 API based on a dual ARCnet has been developed and used in several mock-ups.

The ATOMOS network protocol is divided in two blocks: An EN 50 190 protocol layer (layer 5-7) on top of a service layer. The service layer provides seamless communication based on a reliable channel service (Read/Write Service Access Points). The service layer keeps track of life-lists and reliable communication paths. It handles redundant information reduction so a reliable datagram communication is provided for the upper layers. A major advantage is the verifiable communication due to the use of a token based MAC network as well as early detection of changes in network layout and state of attached controllers.

EN 50 190 is made of the P-NET (vol.1), PROFIBUS (vol.2) and WORLD-FIP (vol.3) standards which gives a reasonable guarantee for a well structured interface. The interface include creation, read, write, diagnose etc.

Because of the divisions between the transport layer (SAP layer) and the EN 50 190 vol. 1 layer it should be "easy" to substitute the transport mechanism if necessary.

2.2 System overview

The entities of a node in the ATOMOS ISC system is arranged as shown in Figure 1. As illustrated several

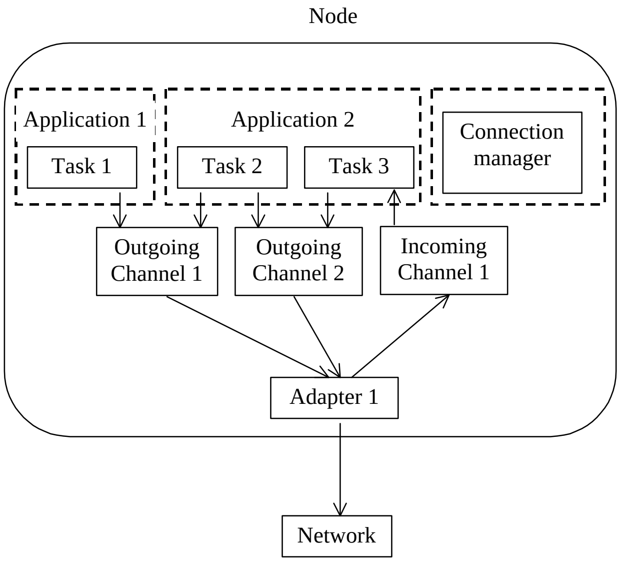

Figure 1. The arrangement of entities in the communication system

applications can be executed on a node. Each application can in turn contain several tasks where tasks can be threads or in some other way a logically isolated entity.

A task can generate traffic to be transported on the network. This is done through channels that are logical connections from a traffic source to a traffic destination. The channels are simplex connections which means that a duplex connection must be implemented by two channels in opposite directions. The channels use adapters to access the network.

The set-up for investigation in this paper is a single segment system due to the fact that time critical communication often will occur on a single segment so routing delays and traffic bottle-necks in bridges are avoided. In (Schiøler, 1998) problems related to multiple segment ARCnets are treated.

Furthermore the system described in this paper requires that the communication patterns between the network entities are statically defined (during system design) in order to avoid dynamic channel allocation. This reduces the functionality of the connection manager shown in Figure 1. The intention of the connection manager is bandwidth allocation, connection establishment, channel maintenance and supervision of communication

Table 1. Traffic characteristics for various application groups. (+) means always required, (-) means never, (*) means optional

Application type	Periodic msg generation	Sporadic msg generation	Stochastic msg generation	Transmission delay constraints	Delay variation constraints	Fixed msg size	Variable msg size	Stochastic msg size	Data rate constraints	Priority
Periodic message transmission (PMT)	+	-	-	*	*	+	-	-	-	*
Minimum Period Message Transmission (MPMT)	-	+	-	+	-	+	-	-	-	*
Available Bit rate (ABR)	-	-	+	-	-	*	*	*	+	*
Unspecified Bit Rate (UBR)	-	-	+	-	-	*	*	*	-	-

behaviour in the system. As the communication patterns are static the bandwidth allocation and connection establishment is predefined and can therefore be removed from the connection manager.

The tasks and the channels in the network system uses Traffic Definition Records (TDRs) and Channel Definition Records (CDRs) respectively for defining the communication they require. These records are kept by the connection manager making it able to maintain the channels and to supervise that the actual network usage conforms to the requested resources.

3. NETWORK SERVICE CATEGORIES

In order to control the traffic flow on the network it is necessary to have a methodology for description and measurement of the traffic. For this purpose a number of service categories has been defined based on the concept of service categories known from the ATM network standard (Stallings, 1998).

Real-time services:

- Periodic Message Transmission (PMT)
- Minimum Period Message Transmission with temporal constraints (MPMT)

Non-real-time service:

- Available Bit Rate (ABR)
- Unspecified Bit Rate (UBR)

The PMT service can be used for servicing periodic traffic sources with a fixed period *T* between messages. The term minimum period indicates that the source might generate messages with a lower rate (higher period *T*) than specified but that the period between two messages must newer be lower than *T*.

The MPMT service is used as a service for transporting sporadically generated messages with transmission delay constraints. This means that the messages will be generated within random periods greater than *T* but the delay constraint can be significantly lower than *T*.

The ABR and UBR services are non real-time services intended for stochastic traffic. The difference between the two categories is that the ABR service guarantees a minimum data rate whereas the UBR service does not.

It should be noted that the service categories do not describe the traffic as it is transmitted on the physical network rather than a set of services that are used by the applications. Traffic of all types will be converted in the channels to conform to the channel transmission method (see section 4.2).

In table 1 the service categories are listed together with a set of traffic characteristics that can be used to describe the services and the descriptors relevant to each service category are marked. Below the meaning of the characteristics is explained.

Periodic msg generation: Messages are generated periodically with period *T*.

Sporadic msg generation: Messages are generated at random time. The period between message generation will always be larger than a minimum time T_{min}.

Stochastic msg generation: Messages are generated at random time. The period between message generation can be infinitely low.

Transmission delay constraints: The message must be delivered to the receiving network entity within a specified delay. The transmission delay is the time from the last bit of the message is delivered to the transmitting network entity until the last bit of the message is received at the corresponding network entity on the receiving node.

Delay variation constraints: The delay variation constraint time is the maximum deviation from expected arrival time. This constraint is used to ensure that the deviation from the sample rate in sampled systems is minimised.

Fixed msg size: The size of messages are fixed.

Variable msg size: The size of messages can vary but follows a deterministic pattern.

Stochastic msg size: The message size is random.

Data rate constraints: The service has some minimum requirements to the data rate offered by the network.

Priority: The messages has a priority at a given level. Priorities are not implemented in the current communication system.

3.1 Traffic descriptors and quality of service parameters

The traffic descriptors are used as a quantitative description of the service category that they are attached to. The traffic descriptors are a blend of the ATM traffic descriptors (Stallings) and an extract of the traffic characteristics shown in table 1.

- Peak Message Rate (PMR)
- Message Size (MS)
- Minimum Average Message Rate (MAMR)

The PMR is used to specify the period T of periodic traffic like the PMT and MPMT service categories. MS specifies the required payload size of messages and MAMR specifies the minimum period T between messages measured over an amount of time $T_{average}$ significantly greater than T. The MAMR descriptor is primarily used to specify a desired minimum data rate for ABR service category traffic.

The usage of the traffic descriptors is described in chapter 4.

A Quality of Service (QoS) parameter is used as a quantitative measurement of network performance. Therefore several of the QoS parameters can be derived as being the quantitative representation of the traffic descriptors defined in the previous section.

The QoS parameters defined for the communication system are inspired by the QoS parameters and traffic descriptors used in the ATM network (Stallings, 1998).

The QoS parameters are:

- Maximum Transfer Delay (MTD)
- Message Delay Variation (MDV)
- Message Loss Ratio (MLR)
- Average Data Rate (ADR)

MTD is the maximum time a message must be on the way measured in the interval from the last bit is delivered to the transmitting network entity and until the last bit is delivered to the receiving network entity.

The MDV defines the maximum allowed deviation from the message rate PMR. This means that when the periodic message stream has settled (within a few message transmissions) the reception of messages at the receiving node must be within MDV from the expected reception time.

The MLR defines the maximum ratio of messages that must be lost due to transmission errors and out of time delivery. For the UBR service category this QoS parameter is used for measurement of the data throughput.

The ADR is primarily used to measure the data rate on ABR service category traffic.

Table 2 shows the relevance of the traffic descriptors and QoS parameters to each service category.

Table 2. The relevance of traffic descriptors and QoS parameters to service categories (+) required, (-) not required

Application type	PMR	MS	MAMR	ADR	MTD	MDV	MLR
PMT	+	+	-	-	+	+	+
MPMT	+	+	-	-	+	-	+
ABR	-	-	+	+	-	-	+
UBR	-	-	-	+	-	-	+

4. CONFORMANCE CLASSES

In order to standardise the communication systems in ISC systems a set of conformance classes has to be defined. The conformance classes must cover the communication system itself as well as the applications that utilises the communication system. In order to satisfy the requirements for a conformance class an entity must satisfy standardised interfaces and internal behaviour.

Each entity of the ISC system (shown in Figure 1) can be placed in one of the following functional groups

- Network
- Network administration entities
- Applications

The network group contains entities that conforms to one of the traditional network standards which in this case is the ARCnet, and p-net compliant entities. The network administration group contains e.g. routers, bridges, traffic administration entities, network surveillance entities etc. The application group contains entities that can be regarded as network users.

Normally entities from different classes will be running on the same physical node. This means that the conformance classes represents logically rather than physically separated entities. The statistical reliability dependency that this yields is ignored in the conformance class description as it is rather a matter of reliability analysis.

4.1 Network group

The network group consists of the Adapters and the Network itself. The companion standards for these entities are already available through the network standards. This means that the behaviour of the entities and the interfaces towards the surrounding systems are clearly defined by standards provided by official standardisation organisations.

4.2 Network administration group

As the network described in this paper is limited to a single segment case with statically defined traffic patterns the entities belonging to the network administration class reduces to

- Channels that are used to control the traffic flow between applications and the network.
- Connection managers that handles the allocation of channels on each node in the system and supervises the traffic flow through the channels.

The channels are used to shape the traffic transmitted by the applications in order to have a controlled and homogenous flow on the network. The traffic shaping is implemented by designing the channels as token buckets. Figure 2 illustrates the principle of the token bucket. The credit generator (the tokens will be called credits in order to avoid confusion with the ARCnet tokens) produces credits at a given rate ρ of credits pr second. The credits are placed in the credit bucket that has a maximum capacity β. Every time a message is added to the message queue it can be transmitted only if there is a credit in the credit bucket. By setting the size of the credit bucket to $\beta=1$ the channel will not allow transmission of messages at a higher rate than the credit rate ρ but it allows the applications to generate messages with jitter on the time period.

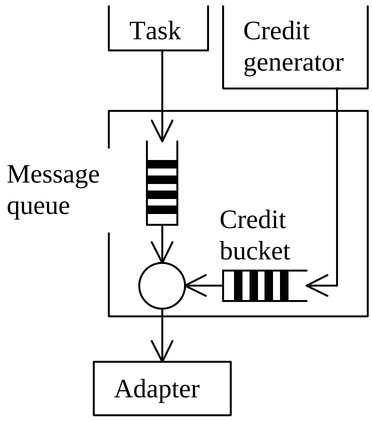

Figure 2. A token bucket

The traffic shaping performed by the channels yields a number of advantages to letting applications access the network directly.

- The flow on the network is deterministic and homogeneous
- Low priority traffic can be controlled in order not to block high priority traffic
- Queuing behaviour analysis seen from the applications is simplified as the token bucket has a predictable behaviour.

The requirements to the channel are set by the application through a Channel Definition Record (CDR) when it requests the channel. In the current implementation where all application traffic sources must be defined during the system design phase the channel characteristics needed for a given traffic source are calculated preliminary.

A CDR will have the contents as shown in Figure 3. The PMR field of the CDR is used to define the credit generation rate ρ of the token bucket. The QoS parameters are derived from the requirements that the application has to the channel. During system verification the values of these parameters will be used as a benchmark for the channel conformance.

Channel Description Record (CDR)	
Traffic descriptors:	Peak Message Period (PMR) Message Size (MS)
QoS parameters:	Maximum Transfer Delay (MTD) Message Delay Variation (MDV) Message Loss Ratio (MLR)

Figure 3. Channel description record

The CDRs are kept by the connection manager and are used during operation to monitor whether the channels operate according to the settings. As conformance to the traffic descriptors can only be supervised on the outgoing channels whereas conformance to the QoS parameters can only be supervised on the incoming channels the connection manager must be in possession of CDRs for incoming as well as outgoing channels.

4.3 Application group

The application group contains the ISC applications. Each application consists of one or more tasks that can be described based on their communication requirements. These requirements are divided into a variety of basic traffic types as follows (From Hede, 1996):

1. Open loop control applications (e.g. sensors giving logging information for process surveillance) that must conform to the PMT service category
2. Closed loop control applications (e.g. engine control, steering control) that must conform to the PMT service category
3. Security sensors (e.g. smoke sensors, heat sensors) that must conform to the MPMT service category
4. Event sensors (e.g. opening and closing of watertight doors) that must conform to the MPMT service category
5. Administrative applications that must conform to the ABR or UBR service category

For each traffic type a Traffic Definition Record (TDR) can be defined. Like the CDR the TDR is kept by the connection managers in order to monitor if the traffic requirements are satisfied. Likewise the connection manager must have TDRs for incoming as well as outgoing traffic.

The TDRs for the five basic traffic types are shown in Figure 4.

Open loop control:

Service category:	PMT
Traffic descriptors:	Peak Message Period (PMR) Message Size (MS)
QoS parameters:	Maximum Transfer Delay (MTD) Message Delay Variation (MDV) Message Loss Ratio (MLR)

Closed loop control:

Service category:	PMT
Traffic descriptors:	Peak Message Period (PMR) Message Size (MS)
QoS parameters:	Maximum Transfer Delay (MTD) Message Delay Variation (MDV) Message Loss Ratio (MLR)

Security/Alarms

Service category:	MPMT
Traffic descriptors:	Peak Message Period (PMR) Message Size (MS)
QoS parameters:	Maximum Transfer Delay (MTD) Message Loss Ratio (MLR)

Events:

Service category:	MPMT
Traffic descriptors:	Peak Message Period (PMR) Message Size (MS)
QoS parameters:	Maximum Transfer Delay (MTD) Message Loss Ratio (MLR)

Administrative traffic:

Service category:	ABR or UBR
Traffic descriptors:	Minimum Average Message Rate (MAMR)
QoS parameters:	Message Loss Ratio (MLR) Average Data Rate (ADR)

Figure 4. Traffic Definition Records for the basic traffic types

5. SIMULATOR

The simulation system must be structured for openness, modularity and scalability. The openness is required in order to make it possible for third party developers to integrate e.g. applications or special network managers in the simulation system. The modularity gives the opportunity of taking specific parts of the simulation system for integration in e.g. a network test bench. The scalability is required because the simulator will be integrated with a larger system containing more sophisticated network managers, networks of other types, network connectors like bridges and gateways etc.

5.1 Simulator structure

The basic framework of the simulator is derived from the system structure shown in Figure 1. This structure directly satisfies the requirement of modularity where specific entities can be extracted and applied in other systems. In order to maintain an autonomous character of the entities an object oriented design is preferred. This requirement might not be satisfied by some applications but these applications should then be encapsulated in an object structure. Figure 5 shows the framework of the object system used for the basic simulation system. The timing manager is associated with all objects in the system which is not shown on the figure in order to keep the number of lines down.

The object structuring is used to simulate the system entities as tasks (threads) in a real-time computing environment because the simulation will be executed on a single computer where the real-time behaviour will not be accurate. Therefore the simulator contains the timing manager which keeps track of the time in the system and distributes computing resources for the system entities.

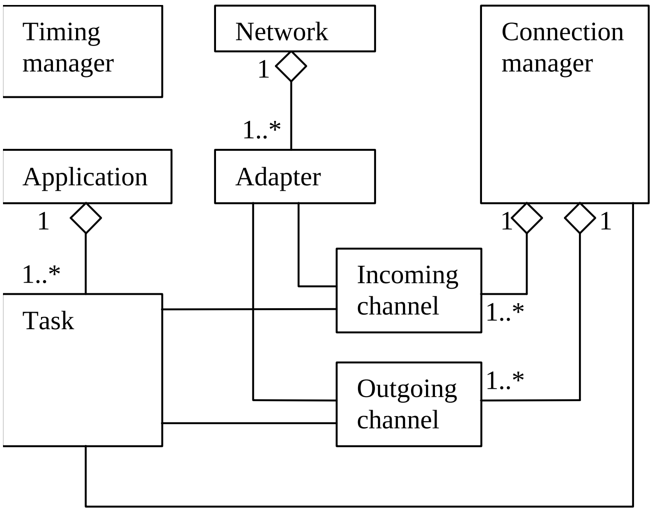

Figure 5. Object framework for the network simulator. Straight lines are associations, lines terminated with a rhomb are aggregations. The timing manager is associated with all objects which is not shown in the figure.

5.2 Simulator timing

In order to obtain optimal granularity of time in the simulations the timing in the system is based on event timing where time is advanced in variable amounts. The time increment is obtained by letting the timing manager find the next chronological event to happen in the system. The event period relevant to each object is maintained by having the object calculate a new event period every time the last event timeout expired.

The timing manager requires a standardised interface to all the objects in the system in order to be able to perform the timing control. The standardised interface has been designed as a basic class that must be the ancestor for all objects in the system (except for the timing manager itself). Figure 6 shows a Java implementation of the abstract basic timed object class.

The constructor of this class registers the object with the timing manager. The timerEvent method is called by the timing manager when the object is in turn for being trigged. The timerEvent routine is responsible for performing the proper functionality (which is defined in the object instance) and calculates the time period till the next event.

The TimedObject class is important because it can be used as an ancestor to system objects as well as an encapsulation for non-object routines. For example a time triggered interrupt routine should be implemented in the timerEvent method. Upon execution the nextEventTimer should be set to the period for the timer interrupt.

```
public abstract class TimedObject{
  private TimingManager timingManager;
  protected double nextEventTimer=0;

  public TimedObject(TimingManager t){
    timingManager=t;

   timingManager.addTimedElement(this);
  }

  public abstract void timerEvent(
        double AbsoluteTime);

  public double getTimer(){
    return nextEventTimer;
  }

  public double decTimer(double
        decValue){
    nextEventTimer -=decValue;
    if (nextEventTimer <0)
      nextEventTimer =0;
    return nextEventTimer;
  }
}
```

Figure 6. Java implementation of the class TimedObject

5.3 Topology description

The topology description is a complete system description containing details on each entity contained in the system for simulation. Below is a list of the network entities that must be included in the topology description as well as the details that must be included on each entity.

The identifier for each entity is used to associate the network objects to each other and should not be confused with the addresses of the network entities. The node identifier is used to indicate which node each entity belongs to. This gives the opportunity of hierarchical identification instead of using consecutive numbering through all the nodes.

In the current system the channels are treated as independent entities. In a “real” system channels are created by the connection manager on request from the tasks.

The format shown below for describing the system topology is rather awkward for manual usage and could in the future be supported with a design tool, preferably graphical. The contents of the topology description is inspired by (Hede, 1996).

Network

Identifier: Network identifier
Address: The address that can be used by routers to find the network segment (Not relevant in the single segment case)
Bandwidth: The transmission capacity of the current network.

Adapter

Identifier: Unique adapter identifier
Node: Identifies the node holding the entity
Address: The address the network must used to address the adapter
Network: Identifies the segment to which the adapter is connected
Buffer size: The available space for buffering of frames for transmission.
Connection Manager: Identifies the connection manager that the adapter must deliver incoming traffic to

Connection Manager

Identifier: Unique connection manager identifier
Node: Identifies the node holding the entity
Incoming channels: List of incoming channels the connection manger is managing
Outgoing channels: List of outgoing channels the connection manger is managing

Outgoing Channel

Identifier: Unique channel identifier
Node: Identifies the node holding the entity
Connection Manager: Identifies the connection manager
Port: The port number of this channel
Adapter: Identifies the adapter to be used for transmission of traffic
Destination address: Address of the adapter that is the destination for the traffic from this channel
Destination port: The port number of the channel that is the destination for the traffic from this channel
Buffer size: The available space for buffering of frames for transmission.
CDR: Channel description record

Incoming Channel

Identifier: Unique channel identifier
Node: Identifies the node holding the entity
Connection Manager: Identifies the connection manager
Port: The port number of this channel
Task: Identifies the task to receive the traffic
Source address: Address of the adapter that is the source of the incoming traffic
Source port: The port number of the channel that is the source of the incoming traffic
Buffer size: The available space for buffering of incoming frames.
CDR: Channel description record

Task

Identifier: Unique task identifier
Node: Identifies the node holding the entity
Application: Identifies the application the task is attached to
Connection manager: Identifies the connection manager
Outgoing channel: A list of outgoing channels used to transmit the traffic
TDR: A list of Traffic Description Records defining the traffic for each channel

Application

Identifier: Unique application identifier
Node: Identifies the node holding the entity
Task: A list of tasks belonging to this application

5.4 Test scenario

In addition to the topology description the simulation can be described by a test scenario which is a description of the sequence of events that the system must go through during the simulation. The events that can happen during the simulation are

- Initialisation control
- External input
- Errors

The initialisation control can be used to control e.g. the time for the nodes to be ready to attach to the network, initial behaviour of applications etc. The external inputs can be e.g. the temperature value on a temperature detector, human interaction like changing the rotation speed of the engine etc. Errors can be severe errors as e.g. breaking network wires or less severe errors as lost messages on the network.

Preferably the test scenario will be described in an existing simulation language, textual or graphical, which has not yet been chosen.

6. EXAMPLE SYSTEM

A typical system that the ATOMOS network could be applied to is a nine cylinder camshaft less main engine running between 0 and 300 rounds per min. Each cylinder has its own controller which perform valve control, injection control and lubrication as well as security supervision. Each cylinder computer acts as a hot standby (secondary controller) for the next cylinder. Two main computers distribute dynamic calculated valve profiles and adjustment of fuel injection. All though this is a time varying system due to variation in RPM it is sufficient to examine the worst case situation of full speed. Because shaft position is distributed by the network communication is very crucial. The communication is:

1) From each main controller to each cylinder for each rotation: new valve profiles, new fuel message and new status messages. Furthermore each node must receive the same information for the next cylinder which it acts as a standby for
2) Each cylinder controller communicates to the main controllers the actual dynamic information.

The characteristics of the communication must be predictable so the cylinder computers can make a decision in proper time when information is not received, so an emergency routine can take place before the engine is damaged. Some traffic must be shifted forward in time so detection of malfunction can be taken in correct time. The spare time on the network can to a certain degree be used for transport of log information, and other maintenance traffic.

The information that the network has to transport is:

- Valve profiles, fuel message and status information from each main controller to the each primary and secondary cylinder controller each 200 msec
- Dynamic information from the cylinder controllers to the main controllers each 200 msec

The system consists of the following entities (the redundant ARCnet is omitted:

- 1 ARCnet network
- 2 main controllers
- 9 cylinder controllers

Each main controller contains the following network entities:

- 1 adapter
- 1 connection manager
- 9 outgoing channels for transmission of traffic to the “primary” cylinder controller
- 9 outgoing channels for transmission of traffic to the “secondary” cylinder controller
- 9 incoming channels receiving traffic from the “primary” cylinder controller
- 9 incoming channels receiving traffic from the “secondary” cylinder controller
- 1 application
- 9 tasks, each responsible for one cylinder. Each task transmits the traffic to (and receives traffic from) both the primary and secondary cylinder controller

Each cylinder controller contains

- 1 adapter
- 1 connection manager
- 2 outgoing channels (one for the primary control traffic and one for the secondary)
- 2 incoming channels (one for the primary control traffic and one for the secondary)
- 1 application
- 2 tasks

In the following examples of the topology description for the main engine will be given.

Network

Identifier = 1
Address = 1
Bandwidth = 312500 (bit/sec)

Adapter (On main controller 1)

Identifier = 1
Node = 1
Address = 255
Network = 1
Buffer size = 1000 (bytes)
Connection manager = 1

Connection manager (On main controller 1)

Identifier = 1
Node = 1
Incoming channels =
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18
Outgoing channels =
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18

Outgoing Channel (On main controller 1)

Identifier: 1
Node = 1
Connection Manager: 1
Port: 1
Adapter: 1
Destination address: 249
Destination port: 2
Buffer size: 1000 (Bytes)
CDR:

PMR	5 (msg/sec)
MS	40 (bytes)
MTD	20 (msec)
MDV	10 (msec)
MLR	5 (%)

Incoming Channel (On main controller 1)

Identifier: 1
Node = 1
Connection Manager = 1
Port = 3
Task = 4
Source address = 250
Source port = 1
Buffer size = 1000 (bytes)
CDR:

PMR	5 (msg/sec)
MS	40 (bytes)
MTD	20 (msec)
MDV	10 (msec)
MLR	5 (%)

Task (On main controller 1)

Identifier = 1
Node = 1
Application = 2
Connection manager = 1
Outgoing channel = 1
TDR:
Service category: PMT (Periodic Msg Transmission)

PMR	5 (msg/sec)
MS	40 (bytes)
MTD	20 (msec)
MDV	10 (msec)
MLR	5 (%)

Application (On main controller 1)

Identifier = 1
Node = 1
Task = 1,2,3,4,5,6,7,8,9

7. RESULTS

The simulations described in this chapter is based on the example system given in the previous chapter. The simulations has been performed in the worst case situation where the engine runs at full speed (300 RPM). In this case the applications produce the maximal amount of traffic due to the proportionality between message intensity and engine speed.

The scenarios that has been simulated are rather alike except for the phase pattern of the periodic packet generation on each channel. The phase φ_i of the i'th channel is defined as the time for the first packet relative to the simulation start time T_{start}. The period of packet generation on each channel i is T_i=1/PMR.

A channel i therefore generates the n'th packet at time

$$t_i^n = T_i \cdot n + \varphi_i \text{, } n = 1,2,.....$$

The following three types of simulations has been performed:

Sim1: The phase of all channels in the system is set to φ_i=0.

Sim2: The phase of the channels i on each main controller is calculated as $\varphi_i = i \cdot 0.01$msec, i=1..18. The phase of the channels on each cylinder controller k is calculated as φ_i=((k-1) ·2+i) ·0.01msec, k=1..9, i=1..2..

Sim3: The phase of all channels in the system is random in the range $0 \le \varphi_i < 0.02$.

Table 3 shows the maximum delay measured on the packet stream send from each node in the system. The utilisation value (Util) is calculated based on the theoretical maximum number of packets with a payload of 40 bytes that can be transmitted on the network.

Table 3. Maximum delay (in milliseconds) on traffic from each node. Node 1 and 2 are the main controllers, 3-11 the cylinder controllers. The "Util" row shows the utilisation of the network,

Node	Sim1	Sim2	Sim3:1	Sim3:2
1	171,33	9,52	16,33	40,79
2	171,32	9,52	30,13	36,37
3	56,56	9,52	11,75	10,93
4	56,56	9,52	11,85	11,36
5	56,56	9,52	8,35	8,11
6	56,56	9,52	6,46	9,05
7	56,56	9,52	11,87	5,82
8	56,56	9,52	7,84	11,62
9	56,56	9,52	11,86	9,78
10	56,56	9,52	11,65	7,17
11	56,56	9,52	7,71	9,51
Util %	70	70	70	70

The results in Table 3 shows that the packet delays in the worst case situation (sim1), where all nodes start transmitting the periodic traffic simultaneously, are substantially larger than for the other simulations where the phase of the channels is distributed. The fact that the maximum observed delay is large can be intuitively understood because the last packets added to the queues will be waiting until the queues are emptied. Less obvious are the results in Table 4 showing that the maximum delay can be decreased in a degree where the MLR goes from 87% to 0% as in sim2 (the MLR naturally depends on the MTD value). In this simulation the phase of the channels has been distributed over the packet period of 200 milliseconds due to a very simple algorithm. The simulations of the third type, sim3:1 and sim3:2, will probably be a normal situation if the tasks on the different controllers are started without any mutual synchronisation of the communication. These simulations show a rather large variation in maximum delay, especially on the traffic from the main controllers, that are the only nodes loosing packets.

Table 4. Message loss ratio due to message delay exceeding MTD (20 msec). Node 1 and 2 are the main controllers, 3-11 the cylinder controllers

Node	Sim1	Sim2	Sim3:1	Sim3:2
1	96	0	0	26
2	96	0	24	58
3	68	0	0	0
4	68	0	0	0
5	68	0	0	0
6	68	0	0	0
7	68	0	0	0
8	68	0	0	0
9	68	0	0	0
10	68	0	0	0
11	68	0	0	0
Total	87	0	8	28

Table 5. Message loss ratio due to message delay variation exceeding MDV (±6 msec). Node 1 and 2 are the main controllers, 3-11 the cylinder controllers.

Node	Sim1	Sim2	Sim3:1	Sim3:2
1	90	0	9	60
2	92	0	61	65
3	82	0	0	0
4	82	0	9	0
5	82	0	0	0
6	82	0	0	0
7	82	0	10	0
8	82	0	0	0
9	82	0	1	0
10	82	0	0	0
11	82	0	0	0
Total	88	0	24	42

Table 5 shows the message loss ratio due to message delay variation which is calculated based on mean delay on each node (not shown). The message delay is compared to the mean delay. If the difference is larger than MDV the packet is discarded (lost).

The simulation results indicates that a vast improvement in network performance can be achieved if the connection managers in the ATOMOS system are extended with a mechanism for controlling the phase of the channels.

8. CONCLUSION

A design for an ARCnet based communication system to be applied in time critical process control systems has been presented. Conformance classes for the network relevant parts of the system entities has been defined. The conformance classes includes quantitative parameters like traffic descriptors and Quality of Service (QoS) parameters that can be used for traffic allocation and supervision in order to obtain reliability and verifiability in the communication system. Furthermore a framework for a scalable, modular and open simulation system close to the real world system has been described. Finally a series of simulations based on a main engine control system has been performed showing the behaviour of some of the QoS parameters.

The simulations has shown that the transmission delay through the network has a distinct dependency on the phase of the periodic traffic. In the applied example the transmission rate on all the periodic traffic sources is the same which gives a good intuitive impression of the behaviour of the system. In a wide range of applications the traffic rates varies considerable which reduces the clarity of the system behaviour and thereby enhances the requirement for an evaluation methodology. In the absence of analytical methods for evaluating network based systems like the presented the simulation approach gives a solid basis for evaluating system performance and thereby for evaluating the quality of the system design.

9. REFERENCES

ARCNET Local Area Network Standard (1992), ATA/ANSI 878.1. Technical report. ARCNET Trade Association, 3365 N. Arlington Hts. Rd. Suite J, Arlington Heigths, IL 60004

Hede, Mikkel, Jens Dalsgaard Nielsen (1996). Communication Pattern Analysis. Technical report ATOMOS II task 2.2/25, ID-code: A222.04.03.060.001.A.

Nielsen, Jens Dalsgaard, Henrik Schiøler and Niels Nørgaard Nielsen (1998). Trends in real-time communication networks onboard ships. In proceedings: CAMS98, Japan, pp. 323-326.

Schiøler, Henrik, Niels Nørgaard Nielsen, Jens Dalsgaard Nielsen and Niels Jørgensen (1998). Worst case queue length estimation in networks of multiple token bus segments. In proceedings: PDCS98, Chicago.

Stallings, William (1998). High-Speed Networks, TCP/IP and ATM Design Principles. Prentice Hall, 1998. ISBN: 0-13-525965-7.