SELF-CONFIGURABLE DISTRIBUTED CONTROL NETWORKS

ON NAVAL SHIPS

J.A.A.J. Janssen, M.G. Maris
Netherlands Organization for Applied Scientific Research (TNO)
Physics and Electronics Laboratory
The Hague, The Netherlands

ABSTRACT

One significant challenge the Royal Netherlands Navy is facing is how to increase the ship's response capabilities to calamities. In our view, self-configuring distributed control networks are required to reach this goal. TNO-FEL, in cooperation with the Royal Netherlands Navy researches such an automated robust ship control system. The researched system consists of autonomous control clusters of sensors and actuators. This novel system makes decisions autonomously, independent of a human operator, based on the information it gathers about its environment. In case of a calamity, it reconfigures itself. For example, when leaks are detected in a fluid system, the flow is automatically rerouted and if needed additional pumps are activated. Furthermore, our approach does not depend on a centralized Ship Control Center. Consequently, it is robust against both Ship Control Center and communication infrastructure failures. Clusters isolated from the rest of the system will still be able to limit autonomously the impact of a calamity. Hence, a distributed control network increases the robustness of ship control systems, improves the reaction time in case of calamities and reduces the required manpower for emergency recovery. This paper focuses on the technology required to realize robust self-configurable distributed control networks for naval ships.

KEY WORDS

Distributed Control Networks, Ship Control Systems, Distributed Intelligence

1. INTRODUCTION

Today already, operators in the Ship Control Center are faced with too many complex and time-critical tasks. This problem will be even larger for tomorrow's ships with reduced manning [1]. A solution to reduce the workload is to add more intelligence in onboard systems. These systems must be capable of making decisions autonomously even in the presence of calamities [2]. In addition, these systems must be robust against failures of parts of the system. The traditional method to solve this is the use of redundant systems. However, this increases the building and maintenance cost. A better solution is to use methods that are more intelligent: the devices and systems are able to reconfigure themselves and exploit the remaining operational resources to perform a given task. In this paper, we will give our view on how to create such robust, autonomous, self-configurable systems.

This paper is organized as follows: in Section 2, the NID (Networked Intelligent Devices) concept is presented. This concept forms the basis of the self-configurable distributed control network as introduced in this paper. In order to validate the robustness of the NID-concept a reliability analysis is performed. This is the topic of Section 3. Before introducing the distributed intelligence methods, a case is introduced in Section 4 with which the operation of our approach is demonstrated. The distributed intelligence methods are presented in Section 5, including their evaluation. In Section 6, the technology required for the implementation of robust, autonomous, self-configurable systems is discussed. Based on today's current technology level an implementation technology is selected. Finally, Section 7 states the conclusions.

2. THE NID-CONCEPT

The basis of the distributed control network described in this paper is the so-called Networked Intelligent Devices (NID) application concept, developed at TNO-FEL[4]. NID-applications comprise interconnected devices, such as sensors, actuators and displays. The devices communicate continuously with each other. They inform the other devices on the network about their status. They do not send just data, but the information concealed in their observations. The devices are standalone units that cooperate with other devices. Together, they negotiate about the state of the system in order to make decisions collectively [5].

This approach has the crucial property that it does not rely on a central control processor. Each individual device performs its actions based on the information it retrieves from the network. In this manner, there is no single point of failure; i.e. the devices do not rely on the presence or absence of other devices. However, the presence of other devices may make the coordinated decisions more reliable.

To create an NID, it is necessary that local computing power is available. The current technology level makes it feasible to integrate sensor technology, microprocessors and network connections in small low cost devices [6].

3. RELIABILITY OF DISTRIBUTED CONTROL NETWORKS

In this section, a distributed approach such as the NID-concept is compared with a centralized approach. Traditionally, most control systems are constructed as central control systems. All sensor data is collected at one central computing node. There it is analyzed and if necessary actuators are activated. This approach was valid because computing power was expensive and it was not feasible to equip each sensor and actuator with its own computing device. However, it has several shortcomings:

- The operation of the system depends on the availability of the central computing node *and* the availability of the network.
- Because data is transported over the network, the network load can be high. When using a microprocessor locally at the sensor the data can be transformed into information, this reduces the network load considerably.
- Because all computation is done at a central computing node, the software for interpretation of the collected data is complex. This may lead to software that is difficult to maintain.
- Adding extra sensors or actuators without using a network requires extra (hardware and software) provisions at the central computing node.

The centralized approach is traditionally also used on naval ships. A missile hit on the central computing node may lead to complete system failure. The system also fails when the connection between the individual sensors and actuators and the central computer node is lost.

More robustness against calamities such as missile hits and fires can be created by distributing the intelligence over the ship. The basic idea behind this concept is that when a part of the system fails, the remainder of the system is still able (with reduced functionality) to perform its task.

In order to quantify that a distributed system is inherently more robust than a centralized approach we use Markov modeling [7]. In Figure 3.1 the Markov state diagram for a centralized approach with *N* devices is given. Figure 3.2 shows the equivalent for the distributed approach. The states indicate the number of available devices while the transitions represent the fail probabilities. Initially all devices are available and the system is in state *N*. When a device fails, the system state is changed to state *N-1*. Each time a device fails, the system functionality is degraded until it fails completely. When the central computing node fails in the centralized approach, the entire system fails immediately.

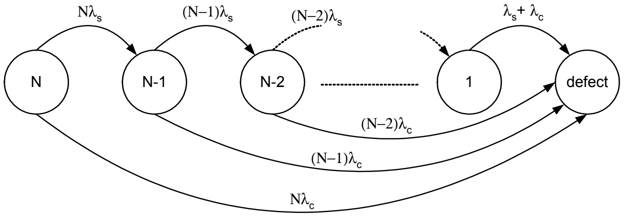


Figure 3.1 Markov model for the centralized approach

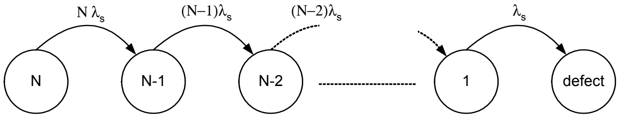


Figure 3.2 Markov model for the distributed approach

In order to solve the Markov equations we have to define the failure rates associated with the state transitions. In this example we define the device failure rate as $\lambda_s = 0.01$, this means one failure in 100 combat hours as well for the centralized as the distributed approach. The failure rates in both systems are equal because the probability that an NID device (in the distributed approach) fails as a result of an missile attack or fire is equal to the probability that a sensor or actuator fails under the same circumstances in a centralized approach. In addition to the device failure rate, we also have to define the failure rate of the central computing node. In our example, we define this failure rate at $\lambda_c = 0.005$. The reason why it is smaller than the device failure rate is that the central computing node is in most cases better protected and has probably a backup facility. Based on these failure rates we computed for a system with five devices the reliability of the system given the number of still active devices. The results for the centralized approach are given in Figure 3.3 and the results for the distributed approach are shown in Figure 3.4. These figures show the probability that *N* devices are operational as a function of the operational hours. For example, the line *N=5* represents the probability that all five devices are operational. The line *defect* represents the probability that the complete system has failed. The values *P* are the probabilities on a certain state after 50 combat hours. As these figures indicate, the reliability of a distributed approach is larger.

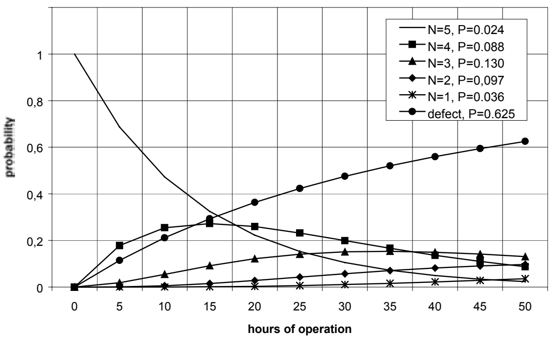


Figure 3.3 Results reliability analysis of the centralized approach over 50 combat hours

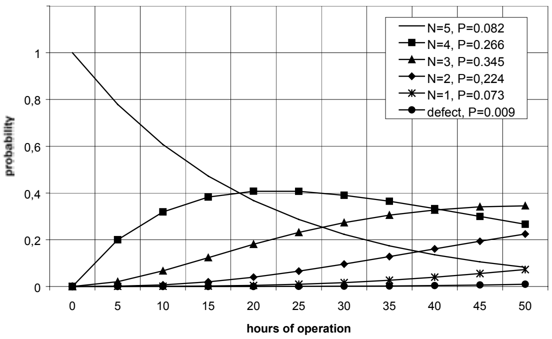


Figure 3.4 Results reliability analysis of the distributed approach over 50 combat hours

It should be noted that the used failure rates are only illustrative. The NID devices contain normally more components, which may increase slightly the probability of failure over the years. Furthermore, network failures are not considered.

One should keep in mind that a Markov model is a simplification of reality and that not all aspects can be accounted for. For example, the ability that a group of smart NID devices is still able to perform a task when they are isolated from the system is not included. However, we believe that it is still useful to make this comparison and that it strongly indicates that a distributed approach is more robust. This observation only holds of course when the distributed approach can offer the same functionality as the centralized approach, which is the subject of the upcoming sections.

4. CASE: CHILLED WATER SYSTEM

In order to demonstrate self-configurable distributed control networks, a representative case study was selected: the chilled water system of a frigate. This system is part of support SEWACO (Sensors, Weapons and Command). Its task is to distribute cooling throughout the ship. Cooling consists of three steps: seawater is used to cool a cooling fluid, the cooling fluid is used to cool the cooling water, and finally, the cooling water is used to cool the users. Two user types are distinguished: vital users and non-vital users. In principle, the cooling for the vital users can not be interrupted. It, however, is allowed to disconnect the non-vital users from cooling to favor cooling of the vital users. In our case, we consider two independent zones, each zone having its own three stage cooling system. Crossovers between the zones are added to implement redundancy. All pumps are redundantly implemented, e.g. they consists of two pumps, an active primary and a secondary. With the use of valves, some parts of the system can be isolated and crossovers can be opened to interconnect the two zones. When due to a calamity the system has a leaky pipe, this leak will be detected with the use of flow sensors and pressure sensors. Based on this information, valves will be opened or closed such that as many as possible users can still be cooled. In our approach, the decision as to close or open which valve or to activate what pump is made locally without the use of a central control system. In addition, temperatures are measured to ensure sufficient cooling. Figure 4.1 shows the schematic representation of the suggested chilled water system. For reasons of clarity, the sensors are not shown.

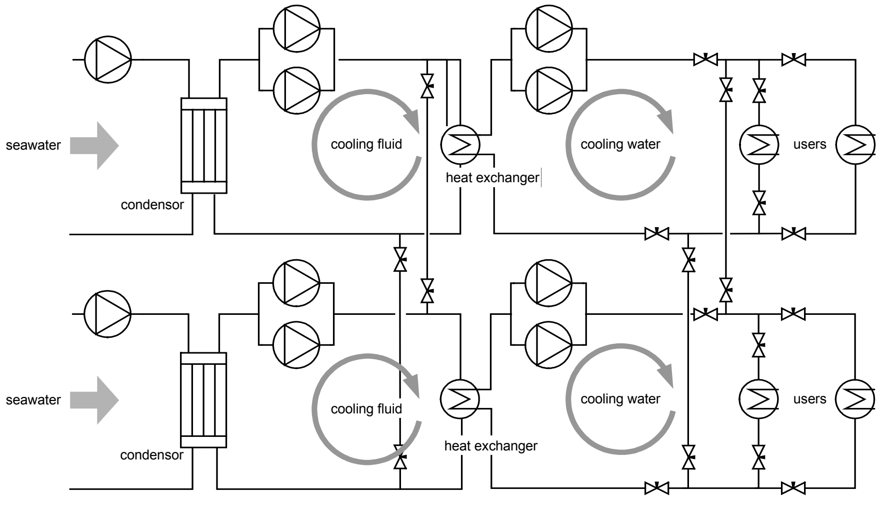


Figure 4.1 Chilled water system

5. AUTOMATIC RECONFIGURATION

In this section, various control methods for the implementation of self-configurable distributed control systems are presented in the context of the chilled water system. The advantages and drawbacks of the methods are inventoried and a solution is proposed.

5.1 Rule-based control method

Rules are a simple and powerful technique to implement responses of a system to certain stimuli [8]. A rule has the following syntax:

<*rule name*>: **if** <*condition*> **then** <*action*>

where <*rule name*> is the name of the rule and <*condition*> the condition under which the <*action*> is performed. The condition is based on sensor data while actions are performed by the actuators. The <*condition*> as well as the <*action*> can be compound statements of various conditions respectively actions.

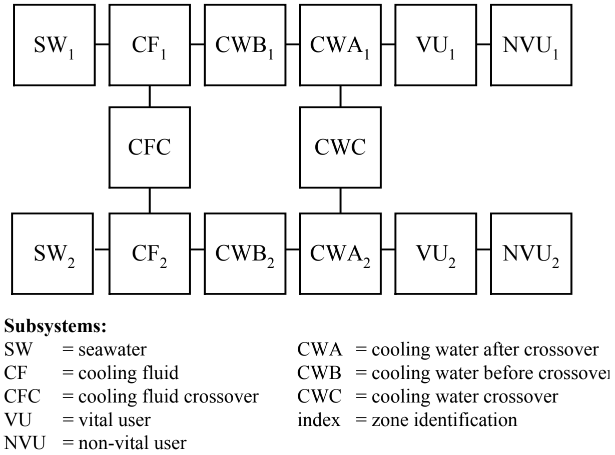

SW1
CF1
CWB1
CWA1
VU1
NVU1
CFC
CWC
SW2
CF2
CWB2
CWA2
VU2
NVU2
Subsystems:
SW = seawater
CF = cooling fluid
CFC = cooling fluid crossover
VU = vital user
NVU = non-vital user
CWA = cooling water after crossover
CWB = cooling water before crossovei
CWC = cooling water crossover
index = zone identification

Figure 5.1 Division in subsystems

In order to limit the amount of rules we propose to split the chilled water system in subsystems. This is depicted in Figure 5.1. The subsystems are defined such that a defect in the subsystem makes the complete subsystem useless. For example, a leak in subsystem "Cooling water before crossover" makes the pumps as well as the heat exchanger useless. Therefore, they can all be isolated from the system by closing the valves.

The partitioning of the system in subsystems results in two types of rules: global and local ones.

Global rules

With each subsystem a number of rules and states (see Table 5.1) are associated. The state **S** of a subsystem depends on the states of the NIDs within this subsystem. A state change of a subsystem is communicated to all other subsystems. Based on this information the subsystems evaluate the following rules:

Rule CFC1: **if** **S**(SW$_x$)[1] == *defect* AND **S**(CF$_1$) != *defect* AND **S**(CF$_2$) != *defect* AND **S**(CFC) != *defect* **then** open CFC

Rule CFC2: **if** **S**(CF$_x$) == *pumps defect* AND **S**(CF$_y$) != *defect* AND **S**(CFC) != *defect* **then** open CFC

Rule CWC: **if** **S**(CWB$_x$) == *defect* AND **S**(CWA$_x$) != *defect* AND **S**(CWA$_y$) != *defect* AND **S**(CWC) != *defect* **then** open CWC

Rule CF: **if** **S**(CFC) == *open* **then** activate secondary pump

Rule CWB: **if** **S**(CWC) == *open* **then** activate secondary pump

Rule NVU: **if** **S**(SW$_1$) == *defect* OR **S**(SW$_2$) == *defect* OR **S**(CF$_1$) == *defect* OR **S**(CF$_2$) == *defect* OR **S**(CWB$_1$) == *defect* OR **S**(CWB$_2$) == *defect* **then** deactivate NVUs

The last rule shows that a conservative assumption is made. In some situations, it is still possible to cool the NVUs by activating the secondary pumps and opening the crossovers.

Table 5.1 Subsystem states

Subsystem	State
SW$_1$ and SW$_2$	normal, defect
CF$_1$ and CF$_2$	normal, defect, pumps defect
CWB$_1$ and CWB$_2$	normal, defect
CWA$_1$ and CWA$_2$	-
VU$_1$ and VU$_2$	-
NVU$_1$ and NVU$_2$	on, off
CFC	open, closed, defect
CWC	open, closed, defect

Local rules

In addition, for each NID within a subsystem, states and rules are defined. The state of the subsystem depends on the states of the NIDs within this subsystem. A NID state change is immediately communicated to all other NIDs within the same subsystem. These NIDs will evaluate the following rules (if applicable):

Rule 1: **if** **S**(primary pump) == *defect* **then** activate secondary pump

Rule 2: **if** **S**(primary pump in CF$_x$) == *defect* AND **S**(secondary pump in CF$_x$) == *defect* **then** **S**(CF$_x$) = *pumps defect*

Rule 3: **if** **S**(primary pump in CWB$_x$) == *defect* AND **S**(secondary pump in CWB$_x$) == *defect* **then** **S**(CWB$_x$) = *defect*

[1] The index represents the zone of the subsystem.

Rule 4: **if S**(pipe$_x$) == *leak* **then S**(subsystem$_x$) = *defect*

Rule 5: **if S**(subsystem$_x$) == *defect* **then** isolate subsystem$_x$

If the state of a subsystem becomes *defect,* the valves at the boundaries of this subsystem will be closed to avoid the propagation of a leakage. In addition, all NIDs in the defect subsystem will shutdown.

Evaluation

The chilled water system can easily be described with a few simple rules. However, the given rules have some limitations. For example, we cannot activate extra cooling measures when for some reason insufficient cooling reaches the users.

In general, a rule-based system is successful when all possible situations can be identified and are captured in rules. However, when situations arise that were not anticipated at design time, the system is not capable to respond. For example, when the seawater subsystem is defect, the cooling fluid crossover is opened. When due to some other defect this is impossible, there is no rule to open the cooling water crossover. In addition, when defining the rules, one must be careful not define rules that contradict. In systems that are more complex, it may be hard to avoid this because of the vast amount of rules. This problem can partly be avoided, if one uses a hierarchy of rules as we did by splitting the system in subsystems.

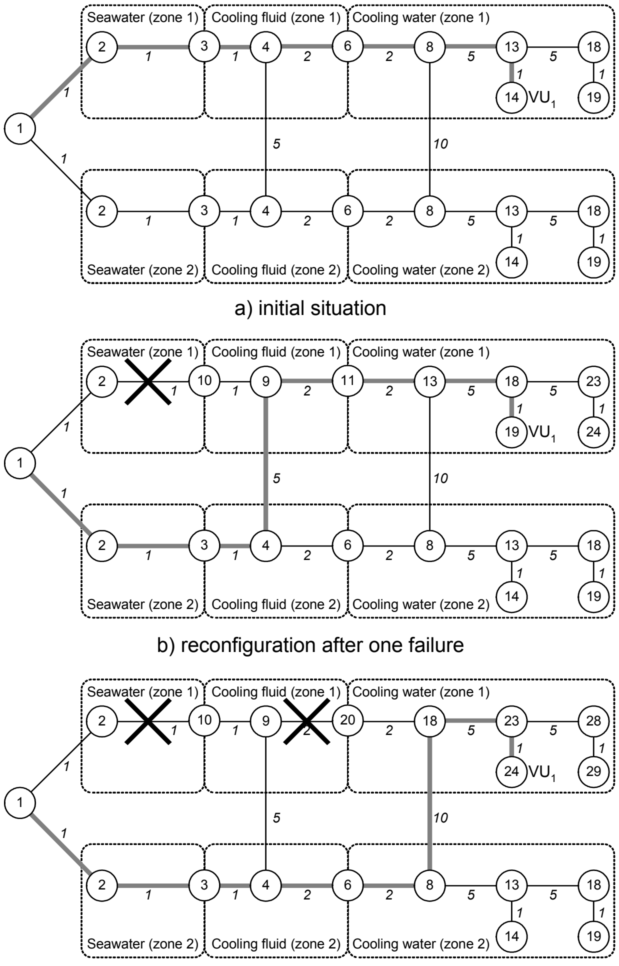
A major drawback of a rule-based system is scalability. For every new function or device added to the system, a new set of rules must be defined for both the added devices and the already present devices. At some point there are simply too many rules to avoid conflicts thus reliability wanes.

5.2 Gradient control method

The gradient method is often used to determine the shortest path in a network [9]. This method is also useful to determine a path through a system of pipes. The route that is autonomously selected depends on the status and configuration of the system. Possible routes can for example be lost as a result of a calamity or system failure. In that case, the control network must be capable of finding an alternative route.

Once a calamity is detected, a gradient is setup from the seawater to the users. The source of cooling is the seawater, therefore the seawater serves as the logical starting point. From there, a message that holds a counter is sent to its functional neighbors. The neighbors increase the counter with a number that reflects the distance between the sender and the receiver. The increased counter value *C* is locally stored and forwarded to its other neighbors. When a node receives multiple values, it only stores the lowest one. This process repeats until the user node is reached. Each device (pumps, valves, sensors, users, heat exchangers, etc.) can function as node in the gradient network. After the numbers have been assigned to all nodes, the shortest path is found by following the steepest (most negative) gradient from user to seawater node by taking into account the distance with its neighbors, e.g. *($C_{neighbor}$ – C)/distance*. The path that minimizes this criterion is selected. A message is propagated along this path. The valves on this path are opened and pumps are activated.

The process is visualized with the gradient network of Figure 5.2. This figure only shows the distribution of cooling. The heat drain off is not shown because it is similar to the gradient network for distributing cooling. The initial situation is shown in Figure 5.2a. The distances between the nodes are given on the arcs. The counter values are listed within the nodes. The shortest distance (C=14) between the seawater and VU$_1$ is highlighted. When due a calamity a path is unavailable, the system must find an alternative route. This is shown in Figure 5.2b for a single failure. Figure 5.2c shows the rerouting when two failures occur.



a) initial situation

b) reconfiguration after one failure

c) reconfiguration after two failures

Figure 5.2 Gradient method

Evaluation

The main advantage of this method is that only a very limited model of the environment needs to be present in every NID. Adding a new device only involves the knowledge of who are the neighbors. In contrast to the rule-based method, it creates a path between the source (seawater) and the users instead of locally fixing a problem. Identical to the rule based method, this method is not capable to handle the situation when the temperature remains too high despite of a path between source and the user.

5.3 Demand-supply control method

A demand-supply control system is inspired by the mechanism observed in free market trade [10]. There is a product that is desired by the demanding party (in this case cooling) and there are suppliers of this product. The demander submits its demand over the network to all suppliers. Those suppliers interested in offering their service will respond with an offer. Suppliers that observe a better offer than their own will not respond. The supplier with the best offer is activated and supplies cooling. The criterion for the best offer is determined by the system designers. In our approach, we use a priority value. This value depends on the location of the supplier with respect to the demander (the users). A short distance results in a higher priority. The suppliers respond with their offers to supply cooling including their priority value.

The distribution of the available cooling products will continuously change over time as demander's needs vary. When more than sufficient cooling is available at the user, the supplier with the lowest priority will be disconnected. Hence, a dynamic equilibrium for cooling will emerge.

Evaluation

Dynamic demand-supply systems are particularly suited for control systems for which it is difficult to determine the way to control the system. Especially, in case of a damaged supplier the system automatically picks a new supplier(s) to meet the cooling demand. A possible drawback of this method is that it relies on the temperature measured at the users. Due to the inertia of water, the lack of cooling may be discovered rather late. This problem can be solved by introducing a slack between the critical temperature and the temperature for activating another supplier.

This system can be expanded relatively easy with additional suppliers. To these suppliers an unused priority value is assigned. Consequently, they can join the demand-supply control system.

5.4 Proposed solution

All three methods have their own specific advantages and drawbacks. The most important ones are summarized in Table 5.2. The table clearly indicates that none of the methods described can solve the problem of reconfiguration and redistribution of cooling alone. Therefore, we propose to use a hybrid approach.

Table 5.2 Method comparison

Method	Advantage	Drawback
Rule based	Simple	Not scalable, cannot exploit multiple sources
Gradient	Creates a path between a source and a sink, scalable	cannot exploit multiple sources
Demand-supply	Can exploit multiple sources, scalable	May be too slow

The first action of the system in case of a calamity is to identify the calamity and to take appropriate measures to limit the impact of the calamity. For example, when a pipe leaks, the valves that can isolate the leaky pipe must be closed immediately. This kind of fast first level reaction can be implemented very well with the *local* rules. These rules can be fairly simple which limits the drawback of scalability.

The second action is to create a path between the seawater and the user. The gradient method is fit for this task. When the user is insufficiently cooled while there is a path from the seawater to the user, the demand-supply method can be used in addition to activate multiple sources (e.g. additional pumps or multiple paths).

With such a hybrid approach, we believe that all situations are covered. The responses to calamities and the decision-making are done locally without the need for a central computing node. This improves the robustness.

6. IMPLEMENTATION ISSUES: ROBUST CONTROL NETWORKS

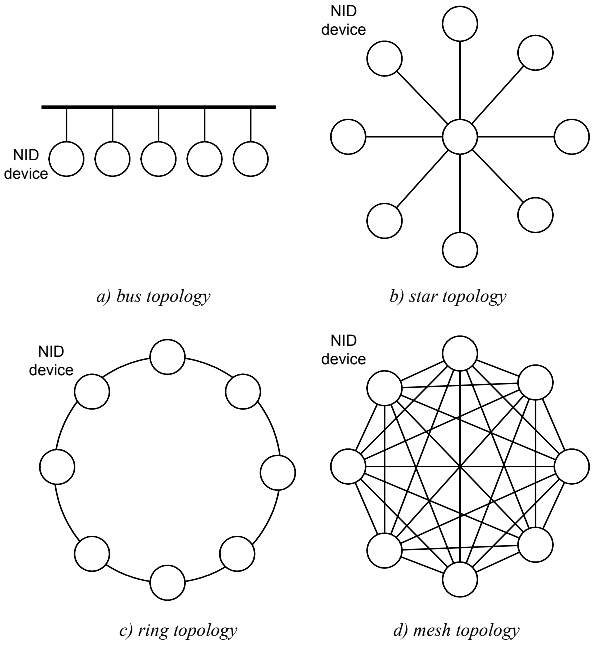
In this section, we look into implementation issues regarding robust control networks. First, we identify the most important characteristics for robust control networks on naval ships. Secondly, selected commercial candidate systems are compared. Finally, a system architecture is proposed.

6.1 Characteristics

Below, typical robust control network characteristics are presented [11][12][13][14].

Topology

The network topology determines how the devices are physically connected with each other. A topology must be such that a network breach does not lead to complete failure of the system. The most commonly used network topologies are given in Figure 6.1.



a) bus topology b) star topology

c) ring topology d) mesh topology

Figure 6.1 Network topologies

In a bus topology, every device is connected to a single network. A network failure results in a splitting of the network into two parts. When using a star topology, only the device connected to the malfunctioned wire is disconnected from the rest of the network. When using a ring topology at least two network breaches are necessary to isolate devices from each other. The most general topology is the mesh. This topology is robust against many network failures, however it requires a huge amount of cabling.

Single point of failure
To be robust, a network may not rely on a single point of failure. Although a device may be extremely simple in nature, it can still be the one whose failure makes the complete system useless. For example, in a star topology the central hub only copies the messages form one branch to the other. It has no intelligence and can be fairly simple. However, failure will result in complete system failure.

Graceful degradation
When parts of the system fail, it must still be possible to communicate over the network in order to diminish the effect of the calamity.

Hot-swap replacement
In a dynamic environment, such as a naval ship, it must be possible to replace devices in a running system, i.e. a device can be replaced without shutting the complete system down.

Scalability
In order to cope with future upgrades or extensions of the system, it must be relatively easy to add new devices. For example, a mesh topology does not scale well because for every new device a huge amount of new cabling is required.

Address space
The address space defines how many devices can be connected to the network. Future naval ships will be equipped with a few thousand sensors and actuators; therefore, the address space should be large.

Transport media
EMC threats (intended or unintended) can have a large impact on the performance of a network-connected system. To cope with these problems isolation is required.

Peer-to-peer
In peer-to-peer communication, all devices are able to communicate with all other devices. In this way, the communication does not depend on a single master (e.g. MIL-STD-1553B [15]). This improves the robustness.

Deterministic and real-time behavior
The network should deliver messages on time. Delays introduced by the network should be minimized and predictable. Two types of network protocols can be identified [16]:

- Time-Triggered Protocols: a timetable is used to decide when a device is allowed to communicate. This ensures that a message is always send at specific time intervals and that a device has access to the medium.
- Event-Triggered Protocols: a device tries to send a message upon an event (for example fire). There is no predefined timetable. Consequently, two devices may try to send a message at the same time. This may lead to collisions and consequently some delay.

Although a time-triggered protocol seems to be favorable, it has some major drawbacks. First, it is difficult to design because the timetable must be such that each event will be delivered on time. Second, adding a new device results in the construction of a new timetable. Third, a device must wait with sending its alarm messages until a time slot is available.

When using an event-triggered protocol a device does not have to wait on a time slot. Furthermore, the network can easily be expanded and designed. Considering our application we do not expect that the amount of messages is such that delay due to collisions will have a major impact on the performance of the system.

Priorities
Some messages have higher priorities than others do. Therefore, there should be a mechanism that enables the use of priorities.

Message acknowledgment
In order to be sure that messages have arrived message acknowledgement should be supported.

Routers
The use of routers (or hubs) can be important. They can deliver the following services:

- Increase the physical length of the network.
- Regulate the message flow such that the network load reduces.
- Reconfigure the network when calamities are detected.
- Serve as buffers between parts of the network such that faults do not propagate.

Standardization
Using an internationally recognized standard decreases the development and replacement costs. It also ensures that products will be available for a longer period.

6.2 Control network technologies

In this section, we compare three candidate network technologies that we consider for distributed control networks. They are DeviceNet, Ethernet/TCP and LonWorks.

Communication between devices using *DeviceNet* [17] is based on the CAN (Controller Area Network) standard. CAN is a communication protocol developed for application in the automotive industry. The CAN specification describes the first two layers of the OSI [18] model (physical layer and data link layer). In addition DeviceNet specifies the application layer (layer seven of the OSI model) and some additions to the physical layer.

Ethernet with the TCP/IP protocol is used mainly to interconnect PCs and workstations. Nevertheless, in industrial automation it is gaining popularity [19]. Ethernet specifies the physical and datalink layer, IP specifies the network layer while the transport layer is described by TCP. The remaining three OSI layers are not filled in.

LonWorks [20] is a network technology for the communication between various types of devices. LonWorks networks can be applied in home automation, industrial automation, aviation, transportation systems, etc. The communication protocol used is LonTalk. This communication protocol implements all seven layers of the OSI model.

Based on the characteristics listed in Section 6.1 we scored the characteristics of each of the three network technologies with the values: good (+), medium (+/-), bad (-). The results are shown in Table 6.1.

Table 6.1 Evaluation of network technologies

Characteristic	Control Network		
	DeviceNet	Ethernet	LonWorks
Topology	+/-	+/-	+
No single point of failure	-	-	+
Graceful degradation	+	+	+
Hot-swap replacement	+	+	+
Scalability	+	+	+
Address space	-	+	+
Transmission medium	+	+	+
Bandwidth	+	+	+
Peer-to-peer	-	+	+
Deterministic/real-time	+	-	+
Priorities	+	-	+
Acknowledgments	+	+	+
Standardization	+	+/-	+
Total	10/4	10/5	13/0

DeviceNet has a limited address space; only 64 devices in a single network are allowed. This is clearly insufficient to connect all sensors and actuators on naval ships. In addition, only the bus topology is available, this makes it less robust. DeviceNet is mainly used in practical implementations in master/slave applications, which introduces a (logical) single point of failure. For our robust concept, we need peer-to-peer communication. Based on these observations we did not choose DeviceNet.

The main drawback of Ethernet is its star topology. This introduces a single point of failure in the network, e.g. if the hub fails the complete network fails. This problem can be solved by implementing a small hub in each device. In this way, for example, a ring topology can be constructed. Unfortunately, no standard modules are available that implement this feature. In addition, the TCP/IP protocol is designed with data transport in mind and not control messages. Therefore, it is not real-time and deterministic.

LonWorks is designed as a control network. It has better real-time and deterministic characteristics than TCP/IP. In addition, it is possible to assign priorities to messages. The main advantage of LonWorks is that all seven OSI layers are defined. This improves interoperability. Various transmission media and network topologies can be used including the robust ring topology.

6.3 Generic network architecture

Based on the previous discussion we chose LonWorks as the basic network technology. This is a proven technology and has all characteristics to implement our ideas about self-configurable distributed control networks. Moreover, this technology is also used in US Navy applications [21][22].

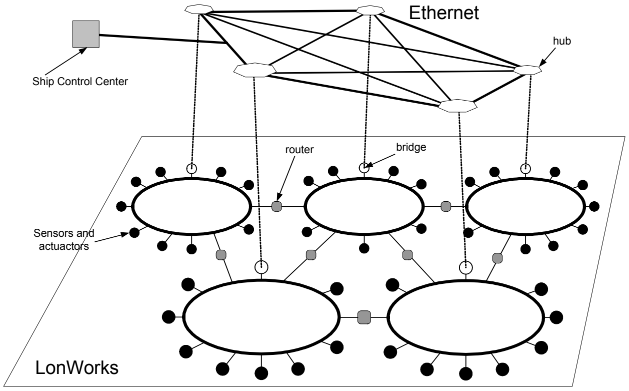


Figure 6.2 Proposed network architecture for future naval ship

In our view, devices in the same zone are to be interconnected with each other using a LonWorks network with a ring topology. In this way, various independent networks are created. These networks are interconnected using routers. This approach prevents that errors are propagated from one zone to another zone thus improving robustness. On a higher hierarchical level, we use Ethernet with TCP/IP. LonWorks and Ethernet can easily be interconnected with the use of bridges. LonWorks messages can even be transported on top of Ethernet. Using LonWorks has the advantage of exploiting interoperability between devices, while Ethernet ensures interoperability with alarm management and interconnection with other operational systems. Figure 6.2 shows the proposed network architecture.

7. CONCLUSION

In this paper an approach is presented that tackles some of the problems the Royal Netherlands Navy is facing: operator load and reduced manning. With the use of intelligent nodes in a control network, it is possible to autonomously reconfigure a system in case of a calamity without the use of a human operator. The proposed distributed approach has the advantage of an increased level of robustness compared to centrally controlled systems as we showed with reliability analysis. Three control methods are proposed for autonomous self-configuration: rule-based, gradient and demand-supply. We believe that by using these methods in a hybrid approach it is possible to create robust and reliable self-configurable control networks on naval ships. The introduced approach is presented in the context of the chilled water system, however, it can be applied to other systems as well, such as the distribution of water for fire fighting or the generation and distribution of electricity. Application of such a distributed control network increases the robustness of ship control systems, improves the reaction time in case of calamities and reduces the required manpower for emergency recovery. Based on the results gained from this research, a small-scale demonstrator will be built in cooperation with the Royal Netherlands Navy to validate the claims made in this paper.

8. ACKNOWLEDGEMENT

The authors would like to thank R.A. Logtmeijer and E. Westermeijer of the Royal Netherlands Navy and R.M. Neef, A. van Lieburg and S.P. van Gosliga of TNO-FEL for their fruitful discussions and their many productive suggestions.

9. REFERENCES

[1] R.A. Logtmeijer, E. Westermeijer, Automation Technology for Workload and Manning Reduction, *To be published in SCSS 2003*, Orlando, FL, April 2003.

[2] R.M. Neef, A. van Lieburg, S.P. van Gosliga, A Layered and Distributed Approach to Platform Systems Control, *To be published in SCSS 2003*, Orlando, FL, April 2003.

[3] T. McCoy, H. Hegner, D. Desai, Distributed Intelligent Sense and Control System for Integrated Power System, *In Proceedings of the 12th Ship Control Systems Symposium*, The Hague, The Netherlands, Oct. 1999.

[4] M.G. Maris, J.A.A.J. Janssen, Networked Intelligent Devices, *flyer TNO-FEL*, The Hague, 1999.

[5] J.A.A.J. Janssen, M.G.Maris, Cooperative Decision Making using Networked Intelligent Devices, *JavaOne Conference*, San Francisco, CA, June 2001.

[6] T.V. Nguyen, A Survey of Smart Sensors and Actuator Technology and Potential Applications to Ship Control Systems, *In Proceedings of the 12th Ship Control Systems Symposium*, The Hague, The Netherlands, Oct. 1999.

[7] J. Pukite, P. Pukite, *Modeling for Reliability Analysis*, IEEE Press, New York, 1998.

[8] P. Katz, A Multiple Rule Engine-Based Agent Control Architecture, *In Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems*, Opatija, Croatia, May 2002.

[9] E.W. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische Mathematik, 1, 1959, 267-271.

[10] M. Woolridge, *Introduction to Multiagent Systems*, John Wiley and Sons, New York, 2002.

[11] P. Madan, Overview of Control Networking Technology, *Echelon Corporation*, Palo Alta, CA.

[12] M. R. Tennefoss, Technology Comparison: LonWorks Systems versus DeviceNet, *Echelon Corporation*, Palo Alto, CA, 1999.

[13] F. Lian, J.R. Moyne, D.M. Tilbury, Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet, *IEEE Control Systems Magazine*, Feb. 2001, 66-83.

[14] J.S. Pascoe, N. Nissanke, R.J. Loader, A Generic Safety-Critical Network Technology Preliminary Study, *The University of Reading*, U.K., Jan. 2000.

[15] SBS Avionics Technologies, An Interpretation of MIL-STD-1553B, *SBS Avionics Technologies*, Albuquerque, NM, May 1998.

[16] H. Sivencrona, J. Hedberg, H. Röcklinger, Comparative Analysis of Dependability Properties of Communication Protocols in Distributed Control Systems, *SP Swedisch National Testing and Research Institute*; Borås, Sweden, April 2001.

[17] Open DeviceNet Vendor Association, Inc., DeviceNet Technical Overview, *Open DeviceNet Vendor Association, Inc.*, Boca Raton, FL, 2001.

[18] A.S. Tanenbaum, *Computer Networks*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[19] B. Moss, Real-time Control on Ethernet, *Dedicated Systems Magazine*, Brussels, Belgium, 2000.

[20] Echelon Corp., Introduction to the LonWorks System, *Echelon Corporation*, Palo Alto, CA, 1999.

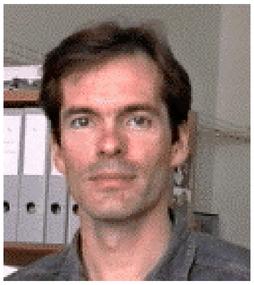
[21] D. Dalessandro, Survivable Automation Technology Demonstrations for Reduced Manning, *Naval Surface Warfare Center, in Wavelengths*, West Bethesda, MD, June 2000.

[22] A.J. Tucker, Opportunities & Challenges in Ship Systems & Control at ONR, *ONR Presentation at the IEEE Combined Decision and Control Conference*, Orlando, FL, December 2001.

BIOGRAPHY AND CONTACT INFORMATION

Dr. Johan Janssen received his B.Sc. degree in Technical Computer Science from the Hogeschool Arnhem, the Netherlands. Afterward, he studied Electrical Engineering at the Delft University of technology, where he joined the Information Theory group. In 1993, he earned cum laude a M.Sc. in electrical engineering. He has been a Ph.D. student at the Computer Engineering group at the Delft University of Technology where he performed research in the area of Application Specific Processors. In 2001, he received this Ph.D. degree. Early 1998 he joined TNO's Physics and electronics laboratory in The Hague, where he co-developed the Networked Intelligent Devices (NID) concept. Currently his research interests are in embedded system design and research towards networked intelligent devices.

Johan Janssen can be reached at j.a.a.j.janssen@fel.tno.nl



Dr. Marinus Maris earned a M.Sc. in electrical engineering from the Delft University of Technology, the Netherlands. During his studies, he developed an extreme low-power instrumentation amplifier. After several years of developing memory chips at Sony in Japan, he pursued a Ph.D. in intelligent robotics at the University of Zurich in Switzerland. During this time, his focus was on robot peripheral intelligence for which he developed highly integrated smart sensors. He then moved on to the Netherlands Organization for Applied Scientific Research (TNO) where he co-developed the Networked Intelligent Devices (NID) concept. Currently his research interests are in applying NIDs for Cooperative Decision-Making.

Marinus Maris can be reached at maris@fel.tno.nl

Presented at the Thirteenth International Ship Control Systems Symposium (SCSS) in Orlando, Florida, on 7-9 April 2003.