11TH ICCRTS
COALITION COMMAND AND CONTROL IN THE NETWORKED ERA
TITLE: On coherence intervals and the control paradox

Jens-Olof Lindh
(Student paper)
Kockums AB
SE-371 82 KARLSKRONA
SWEDEN
+46 455 68 3043
jens-olof.lindh@kockums.se

11TH ICCRTS
COALITION COMMAND AND CONTROL IN THE NETWORKED ERA

TITLE: On coherence intervals and the control paradox

Jens-Olof Lindh
(Student paper)
Kockums AB
SE-371 82 KARLSKRONA
SWEDEN
+46 455 68 3043
jens-olof.lindh@kockums.se

Abstract

Based on explorative empirical studies, this paper suggests that data coherence is a necessary but not sufficient prerequisite for shared information and shared situation awareness in cooperating but physically distributed asynchronous cognitive systems (DAC-systems). A temporal model of DAC system interaction has been developed and a corresponding method for post operations analysis of temporal data has been tested.

Results show that interaction between nodes shall occur within a temporal interval, if internal coherence and external relevance shall be maintained: We call this the coherence interval. Loss of coherence (decoherence) may be caused by overuse of available means for communication and/or lack of common domains and/or technologies. Also, DAC systems exhibit an observability horizon, at which attempts to increase control by increasing the amount of observations in fact cause a loss of control. We call this the control paradox.

It also appears that our findings about DAC systems correlate well with aspects such as information-time vectors, relative simultaneity and the information-system uncertainty principle, normally discussed in modern physics. The coherence interval can also be used to indicate if cooperating groups are able to share a common and relevant dataset.

1 Introduction

The current international interest in information technology support for realizing systems based on network enabled capabilities can be traced at least back to Cebrowsky and Gartska (1998). [6] They suggested that focus of development is shifting from platforms to the network; that the view of actors is shifting from being independent to becoming parts of a continuously changing ecosystem; and emphasized the increasing importance for actors in this ecosystem to make strategic choices in order to adapt and/or survive.

Alberts, Gartska and Stein [1] defined Network Centric Warfare (NCW) as the application of new ways of thinking about human and organizational behavior to military operations. Their emphasis on key concepts for Network Centric Warfare includes (i) the use of a geographically dispersed force that is (ii) knowledgeable and (iii) effectively linked with all other entities of the battlespace. In forces adapted to NCW, synergy is to

be generated by dispersed and distributed entities among which responsibility and work can be dynamically reallocated with respect to the situation. In 2001, Alberts et al [2] extends the discussion by describing the value chain of NCW as a business model dependant on the reach and richness of the conveyed information. Reach is described in terms of the amount of sharing made possible (i.e., by a network). Richness is described in terms of information quality attributes such as e.g., completeness, correctness, currency, accuracy and consistency across different command centers. Other qualities preferred in systems for network centric operations (NCO) are flexibility, adaptability, sustainability and interaction in terms of feedback processes (i.e., battlespace monitoring) and dynamic planning and execution of operations (i.e., battlespace management).

Systemic properties of NCW/NCO-systems such as e.g., interaction, complexity and latency have been addressed in simulations. A networked command situation where a task force needs to get a weapon on target was discussed by Perry in 2002. [18] Moffat addressed NCW from a complex adaptive and open systems perspective, reporting models for simulation and control of swarming and clustering systems. [17] In their subsequent analysis of methods for assessment of information sharing, collaboration and decision making in military headquarters, some focus is put on the amount, age and uncertainty of distributed information. [19, p.72] They suggest that information freshness is concerned with the age of information and that timeliness relates to the perceived need of the information; both are functions of the information rate of change over time. They also note that it takes longer time for a cluster of cooperating nodes to get a coherent view of a situation, if there are more sources of information and more frequent reporting. This effect is attributed to information overload, in the sense that unneeded and/or redundant information saturates the information handling processes without contributing to the quality of estimates.

Part of the argumentation for NCO is also based on the application of the Boyd OODA-loop in order to achieve maneuver warfare abilities. Boyd himself borrowed from Sun Tzu when argumenting that good command may include making the adversary collapse before combat through making him loose contact between his lines and shattering the coherence within his armies. Also, somewhat contrary to its naïve interpretation, we see that the OODA-loop is a *time*-based model. *Speed* in itself is little appreciated. [4, 7, 20]

While often implicitly subsumed, coherence appears to be requisite for cooperation. The purpose of this paper is to investigate findings from an empirical study of data transfer in systems of distributed asynchronous cognitive (DAC) nodes. A time based interaction model is introduced for assessment of some temporal aspects of data transfer and some fundamental limitations to achieving coherence and maintaining control in such systems.

In the following *Section 2 Background*, we introduce some results from our work on systems for Network Enabled Capabilities (NEC), including the model of Open Computational Systems (OCS) and a series of demonstrators on different aspects of NEC-based systems. *Section 3 Problem formulation* introduces the research questions at hand and *Section 4 Definitions* includes our working definition on temporal aspects of the context of NEC-based systems, data transfer and coherence. In the following *Section 5 Experiments* we introduce a couple of experiments that have been done exploring aspects of coherence in open distributed systems. In *Section 6 Analysis* we reason about of temporal intervals and phenomena such as saturation, latency, observability and the

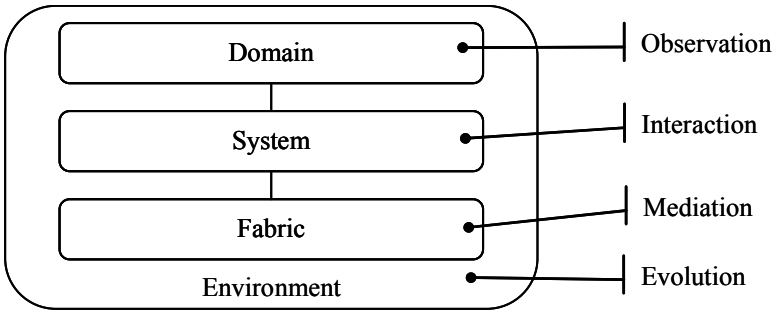

Figure 1: The open computational systems (OCS) model.

consequential control problems. The paper ends with *Section 7 Discussion and related work* in which we outline some relations to e.g., modern physics and suggest some requisites for maintaining control in DAC systems. Finally *Section 8 Summary and future works* concludes the presentation.

Experiments and calculations are detailed in Appendix A.

2 Background

In this section we give an introduction to some key concepts of Open Computational Systems (OCS); a series of OCS-based demonstrators and some findings from assessments of the latter, which are the basis for the subsequent problem formulation.

2.1 Open computational systems

The anticipated flexibility of network enabled systems (NEC) implies that they are to be open, in a systemic meaning. Openness as a systemic criterion can be defined in several different ways. In 2001, Fredriksson and Gustavsson introduced the notions of sustainable multi-agent societies and open computational systems (OCS). [9] In such systems, information is a substantial part of the (generalized) matter that is exchanged across the borders of a system (see Figure 1). Open systems can have the ability to anticipate and support interactions and structures that evolve over time. They strive to establish a state of dynamic equilibrium and are in this context to be understood as systems evolving over time in such a way that (i) there is no real starting or stopping the system, (ii) it is not possible to know in advance, what all operating circumstances will be and (iii) there is a significant amount of computers and software in the loop [10].

According to the OCS model, control in open systems requires that all units addressing a specific task (mission, activity) can make use of a shared and common description of the operational domain. This Domain description is preferably formulated as a relevant subset of the operational environment. The OCS System level corresponds to the processes available in people and/or as software that provide behaviuors. The Fabric level relates to all hardware used for computation and/or communication. The relationship between the OCS model and the layered physical-information-cognition model of NCW [2] was discussed in Lindh (2004). [15]

From a general systems perspective, Bertallanfy defined openness as *"... a system in exchange of matter with its environment, presenting import and export, building-up and breaking-down of its material components."*. [3, p.141] The overall system is

sometimes referred to as a socio-technical system. Our socio-technical systems differ, however, in two respects from the systems addressed in [3] in being both information and user centric. A group (system) is in our setting a collection of nodes (subsystems) and links. The links connects the nodes according to some given connectivity model. The group can include human users and is embedded in an environment. Figure 1 also indicates how the levels of OCS relates to the cognitive nature of our system view.

2.2 Demonstrator experiences

A series of systems demonstrating different aspects of OCS have been developed over the last few years. [16] Observations from e.g. the TWOSOME, SCOPE and BGS 1 demonstrators indicate that different sessions that run with exactly the same initial parameter settings sometimes lead to unexpected behaviours. It appears that the systems at some level of temporal and computational resolution loose the ability to establish a consistent and repeatable behavior. Variations in message handling and computational performance (not related to software bugs) seems to affect relations between cooperating nodes in such ways that a composed and predictable behavior cannot be assured with deterministic confidence below some level of detail. We believe these phenomena are related to (i) balance between inter-node communication and intra-node computational load, (ii) variance caused by e.g., the order of discrete events in a high temporal resolution environment and (iii) to the levels of abstraction modelled in to the system.

One example of this is when the simulated minesweepers of TWOSOME once in a while do not distribute themselves to provide the signature of a large ship – even though the simulations are "cold started" every time. Another is a lagging experience of SCOPE. When three dimensional user interfaces are started and using large percentages of available processing resources, a concurring lack of coordination can be observed in the representations of simultaneously simulated units. In BGS 1, simulated vessels were given specific signature and sensor characteristics, and then allocated to competing groups. Tactical information was distributed in group specific "front end" fabric components while (simulated) physical and environmental properties were distributed in a physically separated "backbone" fabric (i.e., other cables and switches) and connected to a data base that provides a shared and common arena for all participators. It has proven difficult to repeat a scenario and reach a specific end state by manual control of the nodes. Or to be more precise, specific end states can be reached in a general manner, but differences can always be found at some level of detail. Obviously, interaction in the system of human users, software based simulations and graphical interfaces provide variability at a level below the human capabilities to detect and handle.

While the systems referred to above provide valid representations of network enabled capabilites, they are all based on simulated entities. In order to concretize real world capabilities and cut away simulated aspects of the demonstrations, we subsequently developed the Blue Force Tracker (BFT). The purpose of BFT is to distribute node specific state information within a group of cooperating units (i.e., a "blue force") so that a common set of data can be created, shared and maintained, thus supporting information- and cognition based actions. As a consequence of building on the OCS model, the BFT has no specific commanding and controlling unit. Rather, it is designed to keep all members equally informed about the current state of the environment, without depending on any single unit or node. BFT is the test bench for our studies of coherence.

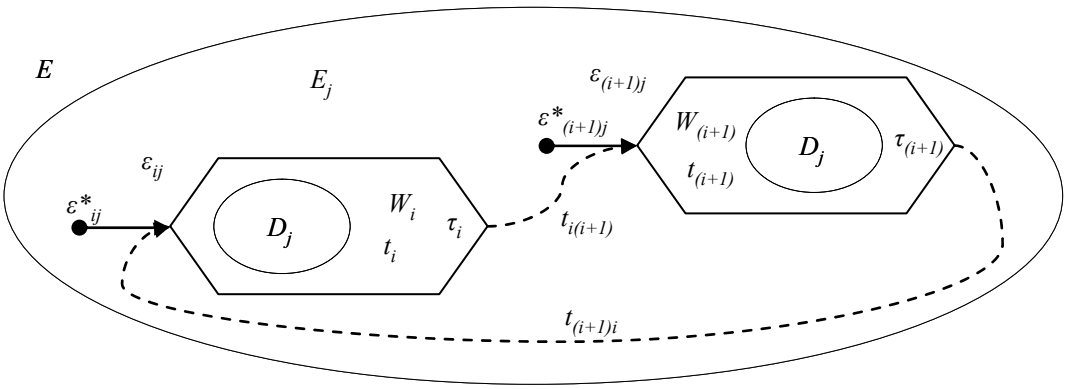

Figure 2: General node interaction model.

3 Problem formulation

As we saw in the introduction, improved and shared battlespace awareness is regarded as a prerequisite for improving military operations. An important goal to that end is to enable e.g., self–synchronization which in turn has to be based on a common understanding of the situation and shared information. We have also seen that temporal aspects of data transfer are important, and that coherence is (at least implicitly) regarded necessary for correct situation assessment and control of interacting clusters. We claim that the existence and availability of a coherent set of data such as e.g., a common set of attributes and values is a necessary but not sufficient criterion for sharing any information in a cooperating group of cognitive units among which it is to be used. The common data set is incoherent if different units have different values associated with the same attributes. Further processing of data will in that case be of limited value as far as shared operational aspects are concerned. The general problem addressed in this paper is under what interoperational circumstances data coherence can or cannot be established and maintained in distributed asynchronous cognitive (DAC) systems. The specific problem addressed is how to establish means for experimentation, measurement and analysis with/of temporal aspects of distributed asynchronous systems.

In order to explore the general aspects of coherence, we establish suitable formal representations and models of the data transfer and management at hand. To that end we need to model and measure relevant metrics during experimentation and evaluate the results accordingly. However, since we specifically are addressing the relations between data flow and emergent system characteristics, we also have some specific measurement challenges. Collecting data about the interaction between nodes in distributed systems as those addressed here is in itself an operation of interaction and data transfer. To that end, a comprehensive experimental environment, including a basic methodology, has been developed for the purpose of post operations analysis of node interactions. The methodology allows temporal comparison of interaction events, based on the local data sets as registered in databases at different nodes regardless of the settings of their internal clocks. We assume data to be correct and appropriate. Therefore, uncertainties with regard to data quality as such are disregarded in this work. Also, we do not specifically address higher level group behaviours such as e.g., decision processes and internal conflicts. The following sections present relevant definitions, models and experiments.

4 Definitions

We make the following assumptions: All nodes have cognitive capabilities, are physically separated and temporally asynchronous. There is no central control function, so all nodes establish their world models based on local cognitive capabilities and data transfer. In this modelling section we define a general framework for coherence metrics. However, the experiments are done in modelling a network with only one group of nodes and at one organizational level. A general model of interaction between two nodes with a shared data set embedded in different contexts is presented in Figure 2. See also Appendix A.

Nodes: The contents and graphics of nodes are modelled with the representation suggested by Eric Hollnagel in the Functional resonance model (FRAM, [12]). This also allows each node to contribute to a joint cognitive system. [13] Let each node i have cognitive capabilities manifested as e.g., a local controller consisting of some processing unit with appropriate algorithms and means for creating and maintaining node specific representations of current states in a local world model W_i,. The local world model consists of the corresponding computational context C_i, environment E_i and domains D:

$$W_i(t) = \langle C_i(t), E_i(t), D(t) \rangle \tag{1}$$

Domains: Let each domain D_j be represented by a time dependant data set. Let each node i have a local representation D_{ij} of D_j, and let D_{ij} be a subset of the local world model W_i of node i. The domain D is represented at each node with a shared data set corresponding to some set of attribute-value tuples $<a_n, v_n>$:

$$D_j(t) = \left\langle \left\langle a_{j1}, v_{j1} \right\rangle, \left\langle a_{j2}, v_{j2} \right\rangle, \ldots, \left\langle a_{jn}, v_{jn} \right\rangle, t \right\rangle \tag{2}$$

$$D(t) = \left\langle D_1(t), D_2(t), \ldots, D_m(t) \right\rangle \tag{3}$$

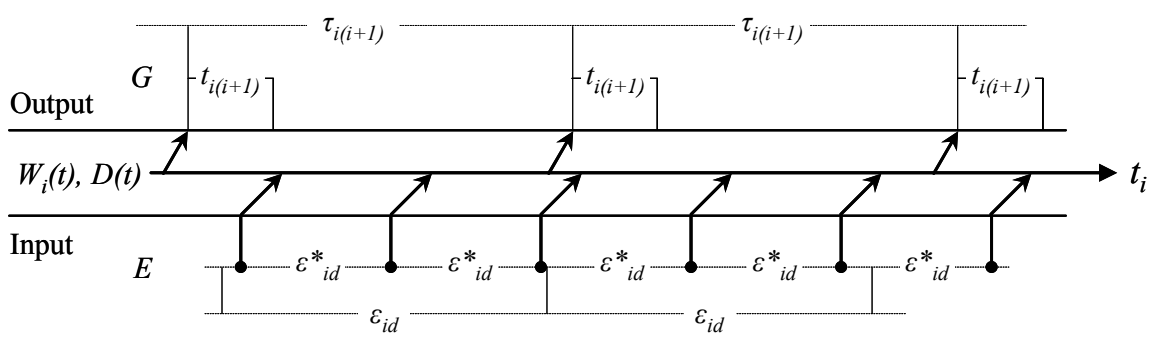

Figure 3: Data input-output model of a node with respect to environment E and group G.

Environment: Different domains typically belongs to different abstraction levels of the system at hand. Let E be the unbounded environment (the real world context) were all nodes are embedded. Part of the environment E can be described in a domain D_j. Let E_j be a relevant subset of the environment E which is described in domain D_j. Let E_{ij} be a subset of E_j that is accessible by node i and described in domain j of i. Let $E_{ij}dt$ be the change rate of E_{ij} over time so that the interval ε_{ij} between relevant changes in E_{ij} is

$$\varepsilon_{ij} = \frac{1}{E_{ij} dt} \tag{4}$$

Let ε^*_{ij} be the selected parsing interval of node i with respect to ε_{ij}. See also Figure 3.

Groups: Let a group G be defined as all nodes i having local representations of the same domain D at time t. The dynamics of group G is restricted to the abilities enabled by (system level) algorithms residing at each node N_i $(i=1...n)$ and the temporal dynamics of the shared data set of domain $D(t)$:

$$G(t) = \langle\langle N_1, D(t)\rangle, \langle N_2, D(t)\rangle, \ldots, \langle N_n, D(t)\rangle\rangle \tag{5}$$

Messages: Let data be exchanged between nodes as messages m_i containing node identification N_i, node time t_i when the message was created and a data content $<a_{ji}, v_{ji}>$ from $D_j(t_i)$ (eq. 2):

$$m_i = \langle N_i, D_j(t)\rangle \cong \langle N_i, \langle a_{ij}, v_{ij}\rangle, t_i\rangle \tag{6}$$

Each message also holds a size σ_m.

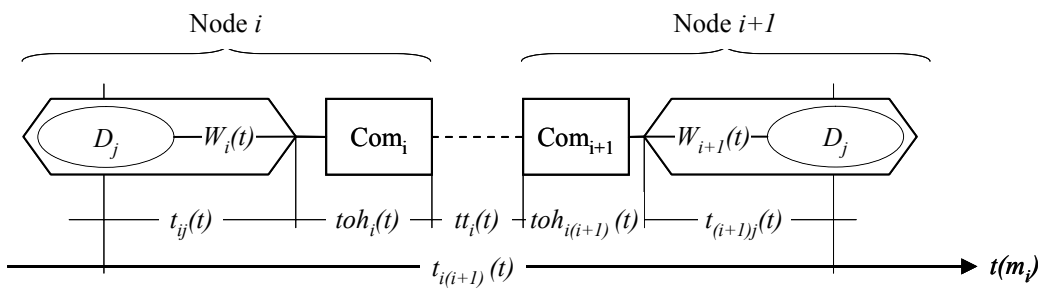

Figure 4: Message transfer model with respect to nodes i and i+1.

Message transfer models: Figure 4 presents a two-node message transfer model based on the Shannon-Weaver communication model (1949, as related in [13] but noiseless). We have distinguished the nodes with cognitive capabilites (e.g., computers) and their local world models from their corresponding means for communication (e.g., radios). The significance of this is that a message is created and cognitively "sent" before it is distributed on a technical level. Same but reverse reasoning applies to the receiving node.

Let t_{ij} be the time needed for handling and transferring data from/to the local world representation W_i in node i to/from its internal data set D_j. Let $t_{ij}(t)$ be related to the quotient between computational context $C_i(t)$ and the processing capability $proc_i(t)$ at each node i and each instant t:

$$t_{ij}(t) = \frac{C_i(t)}{proc_i(t)} \tag{7}$$

Let the message transfer time $tt_{i(i+1)}(m_i,t)$ of message m_i from node i to node $i+1$ be the quotient of message size σ_m and used link speed $v(t)_{i(i+1)}$:

$$tt_i(m_i, t) = \frac{\sigma_m}{v(t)_{i(i+1)}} \tag{8}$$

Let the total time $t_{i(i+1)}(t)$ needed for transferring a message m_i between the representations of domain D_j at nodes i and $i+1$ at time t be the sum of node specific data handling times t_{ij}, node specific transfer overhead times $toh_i(t)$ and message transfer time:

$$t_{i(i+1)}(m_i, t) = t_{ij}(t) + toh_i(t) + tt_i(t) + toh_{i+1}(t) + t_{(i+1)j}(t) \tag{9}$$

The node specific transfer overhead times toh_i depends on the characteristics of communication equipment used such as e.g., its internal control mechanisms and data transfer capacity. Also note that at some level below our current focus of analysis, the communication and computation equipment (such as e.g., a radio and a computer) can be regarded as two separate nodes.

Coherence: A data set $D(t)$ is coherent, $K(D(t))$ at time t if and only if its attribute-value pairs are identical at all nodes. Let coherence at level j, $K_j(t)$, be True iff the set $D_j(t)$ of equation (2) exists and are identical for all nodes N_i. The (global) coherence is defined as:

$$\mathrm{K}(D, t) = True \quad iff \quad \forall j \left(\mathrm{K}_j(t) = True\right) \tag{10}$$

Let coherence K_j be False for the set D_j if not all shared attributes and their respective values are identical. We have:

$$\mathrm{K}(D, t) = False \quad iff \quad \exists j \left(\mathrm{K}_j(t) = False\right) \tag{11}$$

Coherence fraction: Let the coherence fraction κ_j at level j be defined as a quotient between time intervals $K_j(t) = True$ (notation $dt_{Kj=T}$) and the total time elapsed:

$$\kappa_j = \frac{dt_{\mathrm{K}j=T}}{dt_{\mathrm{K}j=T} + dt_{\mathrm{K}j=F}}, \quad 0 \le \kappa_j \le 1 \tag{12}$$

Corresponding notions holds for the (global) coherence fraction κ.

5 Experiments

The purpose of these experiments is to verify the coherence fraction model. As point in case, and also as a basis for hardware and software selection, the experiments are done on the BFT demonstrator. In doing this we also have to develop and test a method for measuring and analyzing temporal properties of distributed asynchronous cognitive (DAC) systems. Therefore the experiment is divided into two parts. Experiment 1 tests the method as such at a presumably constant and stable system. This is done in order to assess the general soundness of this empirical approach. Experiment 2 addresses a changing interval τ^*_i for node i distribution of its current state to the other members of the same group. This is done in order to investigate and assess relations between coherence in a group and the data update intervals of the respective nodes. The experimental setting is designed with guidance of the CCRP Code of best practice for Experimentation. [5] Detailed calculations and results are provided in Appendix A.

Method: Three computers 1, 2 and 3 serve as nodes. Each computer maintains a current state in its primary memory. It also stores e.g., incoming and outgoing messages with e.g., identity, content and local node-time of its registration in a local database. The data collection conflict described in the problem formulation (above) is addressed with local data storage and post operations analysis. As a consequence of this, the resolution provided by the registering ability of the databases at the different computers is also the lower limit of the measurements resolution. As this registration has a millisecond granularity, it is coarse compared to transfer times expected in e.g., current local area networks. We also face the problem of synchronizing different node-internal clocks to an extent sufficient for comparing the database contents.

The measurement resolution problem is addressed by selecting a combination of message size and communication link speed so that the message transfer time is significantly larger than the minimal measurement precision. The post operations node time correction problem is addressed by making the assumption that messages sent via radio from one node over a short distance arrives at all other nodes simultaneously. Given this and that at least $n-1$ nodes send data which is received at the other nodes, clock time correction factors can be solved after the tests and applied to the data sets before comparisons (see Appendix A eq. A.1-A.15). Both design parameters suggested above were implemented by selecting VHF radio at 9600 bit/second as message transfer media.

Setting: Both experiments were conducted at a stationary, indoors setting. All data was collected automatically and analyzed after completed experiments. The independent variables are for both experiments (i) the total message transfer time $t_{i(i+1)}$ defined as the time passing from that a message is defined in the domain of one node until the same message has been recognized and registered in the domain of another node, and (ii) the coherence fractions κ_3 for node 3 and κ for the whole group. The dependent variables are

- the message distribution interval $\tau_{i,i+1}$, which in experiment 1 is constant at 5 seconds for all nodes and in experiment 2 is decreasing towards zero and raising back to 5 seconds for node *3*,
- message size σ_m which is set to 100 bytes for each and all messages,
- radio link data transfer speed which is 9600 bit/second, and
- the equipment as such which is constant for these experiments.

The computational context was kept as stationary as possible through not interfering with and not running any unrelated software on the computers during the experiments.

5.1 Results from experiment 1

Experiment 1 is designed to investigate the coherence of a stable system with large temporal margins. As a 100 byte (800 bits) message over a 9600 bit/s link would take a nominal transfer time $tt_i = 800/9600 = 0.083$ seconds, setting the message interval $\tau_i = 5$ seconds would provide plenty of margin and good coherence. The number of messages sent was chosen to provide approximately 30 minutes of data, corresponding to about 1050 measurements.

The total message transfer times turned out to be about 0.20 seconds for nodes *1* and *2*, and 0.25 seconds for node *3*. The coherence fractions between the nodes of experiment 1 are presented in Figure 5. Node *3* is represented with its singular coherence

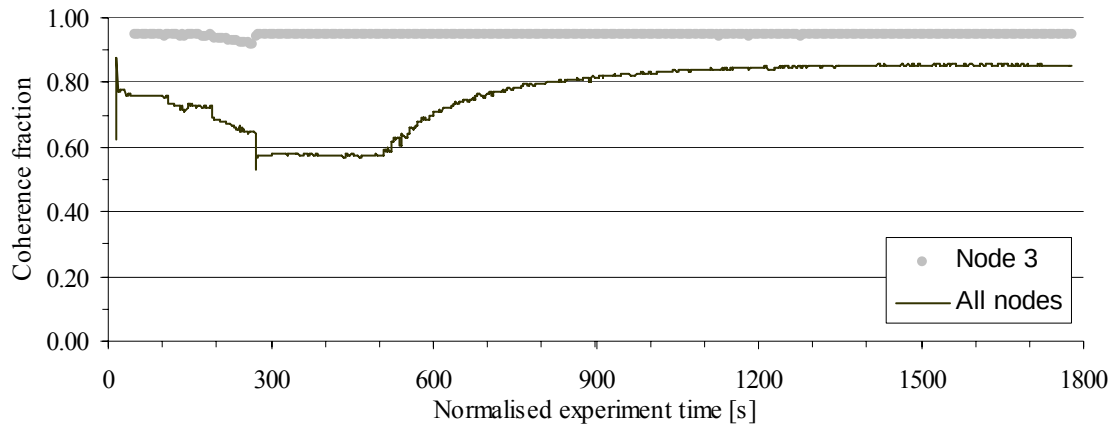

Figure 5: Coherence fractions; result from experiment 1.

fraction; i.e., for how large part of total time nodes *1* and *2* have the same data as node 3, with respect to changes in the data originating from node *3*. The group of all nodes is represented with a moving average over 9 consecutive data points.

Even though this test was designed to provide a stable baseline, a significant loss of coherence occurred at about $T_N = 280$ s. This dip is attributed to race phenomena in the asynchronous message sending processes. Such phenomena has to be regarded as normal in this type of system; reasons for this are discussed in some detail in Appendix A. We can also note that (i) the slight dip in local coherence κ_3 of node *3* caused a large disturbance in the overall coherence and that (ii) a seemingly asymptotic coherence of approximately $\kappa = 0.85$ not is achieved until about 700 seconds after the relatively short disturbance period of node 3.

5.2 Results from experiment 2

The purpose of experiment 2 is to investigate the effect of decreasing the interval τ

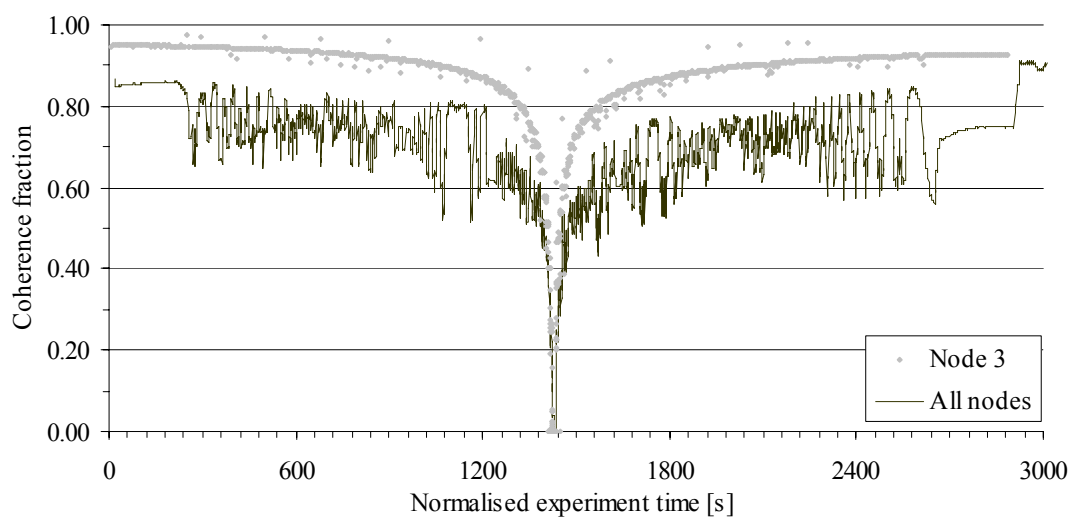

Figure 6: Coherence fractions; result from experiment 2.

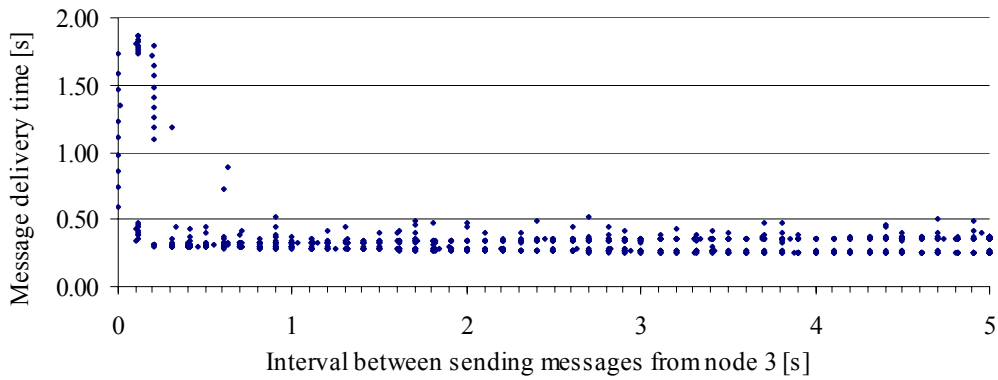

Figure 7: Sending node 3 perspective on message delivery times t_{31} *to node 1.*

between messages sent from one of the nodes (node *3*), with respect to the coherence fraction of that node and the overall coherence of the group.

To that end, the second experimental setting is the same as the first, with the difference that the interval between sending messages from node *3*, τ_3, is decreasing to zero and then increased again. Ten messages are sent for each value of the interval τ_3, which changes in decrements of 0.1 second in each step from 5 seconds in the beginning to zero after about 1400 seconds elapsed experiment time. Thereafter intervals grew back to 5 seconds again, still in increments of 0.1 seconds in each step. In order to establish stable start and end states, 40 messages were sent at $\tau_3 = 5$ seconds in the beginning and end of the experiment. The total duration of the data collection was approximately 2800 seconds, resulting in a total of about 2400 measurements in the joint data base.

The coherence fractions of experiment 2 are presented in Figure 6. We can see that κ for the first and last periods are comparable to those of experiment 1. This is reassuring, since the conditions are set to the same for these periods. When τ_3 begin its decrease, we see two major effects. The first is that the variability of the moving average over the coherence fraction of all three nodes is increasing significantly. The second is that the coherence fractions begin sloping downward until they suddenly drop to zero, after which they recover again in an almost mirrored pattern. We can also note (i) a significant raise of group coherence after node *3* has stopped transmitting data (at about $T_N = 2800$ seconds) and the group therefore shrank from three to two nodes, and that (ii) both experiments indicate that the coherence fraction for the group always is lower than the coherence for a singular node. This indicates that coherence both decrease and become less predictable with increasing number of nodes.

A node 3 perspective: Figure 7 shows a time corrected plot of the delivery times measured for messages created at node *3* and delivered to node 1. The x-axis shows the interval τ_3 between messages sent from node *3*. The y-axis shows the message delivery time t_{31} defined as in eq. 9. We can see that message delivery times from node *3* is plotted at 0.25-0.30 s for the first half of the run (the lower series of horizontal plots). This is about the same delivery times as measured in experiment 1 (see appendix A).

A slight increase in lowest t_{31} can be seen at the interval between messages $\tau_3 = 0.1$ seconds (decreasing). At $\tau_3 = 0$ seconds, t_{31} climbs up the y-axis for the ten consecutive messages, each taking longer time to deliver. During the following ten messages sent at

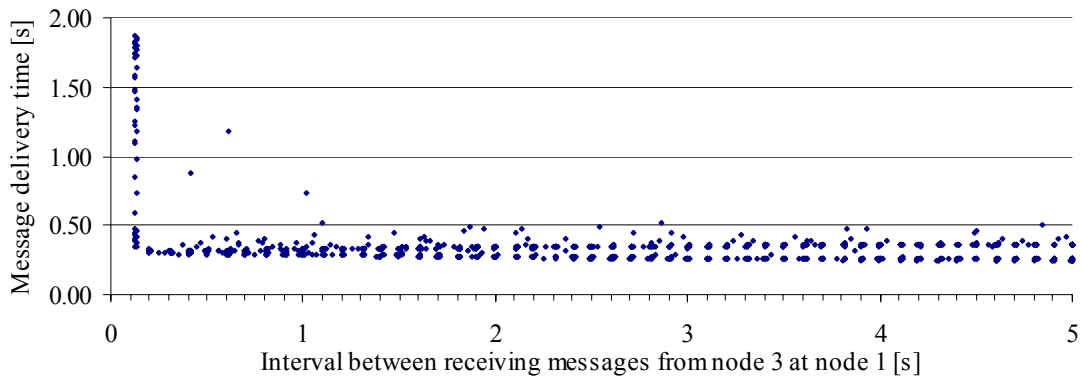

Figure 8: Receiving node 1 perspective on message delivery times from node 3.

$\tau_3 = 0.1$ (increasing) we have a maximum t_{31} of about 1.85 seconds. As τ_3 increases again, the message delivery times return to the previous, for a slow ascend from 0.30-0.35 s.

We have reason to believe that the drift of the average t_{31} from 0.25 to 0.35 seconds for $\tau_3 \geq 0.2$ s is a result of the correctional calculations and that the actual message transfer time is about 0.25 s as in experiment 1 (see appendix A). However, we have to give some extra consideration to the relatively large message delivery times of low τ_3:

A node 1 perspective: Figure 8 shows a plot similar to the above but with the x-axis showing the interval τ between messages sent from node *3 arriving at node 1*. The most significant difference from Figure 7 is the steeply increasing message transfer times appearing at $\tau_1 \approx 0.1$ s. This interval is slightly above the nominal technical transfer time $tt_{31} = 0.083$ s needed for transferring the 100 bytes of data over the selected radio link.

6 Analysis

The results from the experiments indicate that the suggested models and methods can be used to measure temporal characteristics of in a group of distributed asynchronous units. We have shown that data coherence can be related both to *ad hoc* interference between nodes and to the intervals between sending messages from a specific node.

Experiment 1: The fact that message transfer times of 0.20-0.25 seconds are significantly larger than the nominal 0.083 seconds lead to the conclusion that node internal data handling adds significant time to the nominal message size—bandwidth calculation . We also have reason to believe that different hardware matters, at least for the relations between dependent variables selected here. Though counterintuitive, computer *3*, which is the fastest, is associated with the longest message distribution times.

Experiment 2: Figure 8 shows that while messages are created at node *3* in accordance with our predefined pace τ_3, they can not arrive at node *1* with a τ lower than allowed by the maximum technical data transfer capacity. When this capacity is reached, messages are queued and handled as fast as possible. This indicates that regardless of how often messages are sent, there exists a shortest interval between which messages can be received. It also indicates that analysis of actual network behavior is more accurately described from the perspective of the receiving node, than from the perspective of the sending node. The further analysis is based on the perspective of the receiving node.

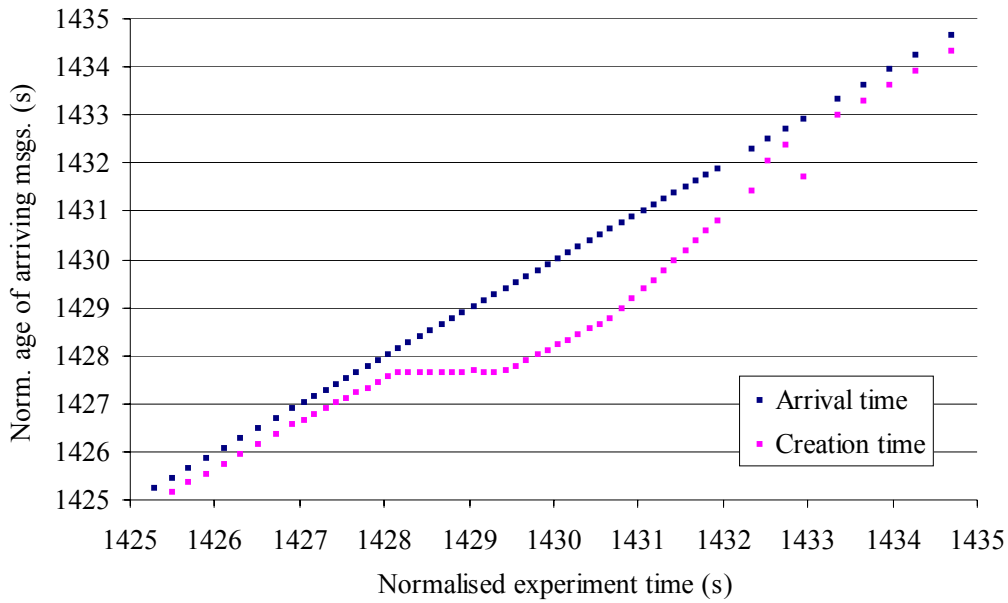

Figure 9: Receiving node 1 perspective on age of messages from node 3.

An alternative plot of the same data is presented in Figure 9 Reception and creation times of all messages created around the $\tau_3 = 0$ interval are represented in a normalized time-time plot in which message reception times are distributed along the x-axis and message creation times along the y-axis. Each message is represented by a vertical plot-pair. The lower plot (red) denotes the time a message was created at node *3*, and the corresponding upper plot (blue, vertically above) denotes the time when the same message was recognized at *the latest* of the other two nodes *1* and *2*.

Horizontal distances between plot-pairs indicate elapsed time between receptions of messages. Vertical distances between red plots indicate elapsed time between the creations of a message. Vertical distances between a red and its corresponding blue plot

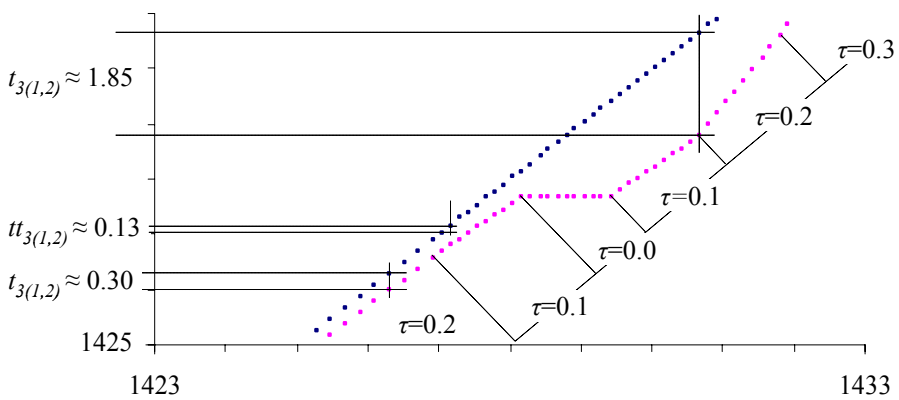

Figure 10: Analysis of the message transfer time.

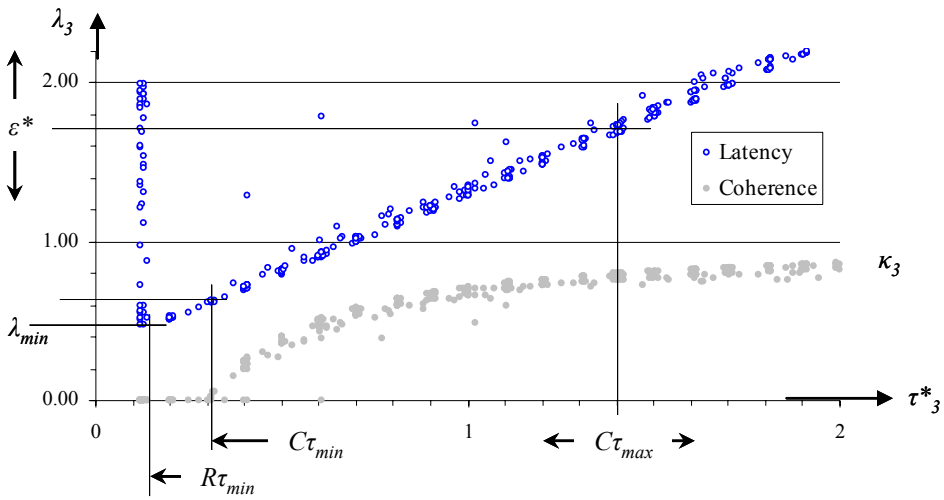

Figure 11: Receiving node 1 plots of coherence and latency as functions of τ^*_3.

indicate the time passed between the creation of a message and its recognition at the domain representations at the last of the other nodes.

At $t_N \approx 1428$ the sequence of red plots turn horizontal. This corresponds to $\tau_3 = 0$ when 10 messages are created with as small temporal interval as allowed by the computational context. We can also see that we have a quite constant distance between the diagonal blue plots. This indicates that messages are arriving at constant temporal intervals. The "depth" of the bulge below the diagonal tell us something about the delay of messages having been created but not transferred to the anticipated receptors.

Saturation: A detailed analysis of the relations between the interval τ_3, and the message transfer times $t_{i(i+1)}$ is presented in Figure 10. This show that the latter are stable at approx. 0.25-0.30 seconds until the interval between new messages becomes shorter than the nominal transfer time. For $\tau_3 = 0.1$, the transfer time is growing slowly, as actual $tt_{i\,min} \approx 0.13$ seconds. When $\tau_3 = 0$ the messages are just queued at the sender, arriving at the other nodes as fast as practically possible given the message size and transfer speed. The message transfer time is not decreasing until the sending interval has regrown to $\tau_3 = 0.2$, during which the transfer time has grown to approx. 1.85 seconds. Constant transfer times are reestablished as the interval raise above $\tau_3 \geq 0.3$.

This analysis show that (i) as when the temporal interval between messages sent gets smaller, the data transfer system becomes saturated, (ii) message transfer times grows quickly in a saturated system, but that (iii) messages keep arriving at a steady (maximum) pace in the saturated communication system.

Let us now add a couple of related observations: Each node can observe the arrival rate of incoming data and it can set its own message sending interval. But it can not assess the time it takes for a message sent to arrive at its destinations. Also, each node can set the intervals between assessments of the relevant changes ε^*_{ij} of its environment, and when changes in ε_{ij} are recognized, these can (or not) be distributed as extra messages. But such messages may cause the message sending interval τ to become so small that the system is saturated and actually thus delaying the arrival of both the extra and all other messages. This indicates that that the measured time $t_{ij}(t)$ rather than representing *the message*

transfer time as such, actually should be regarded as *the age* of a message when it arrives at the receiving node. A change message may be created and distributed e.g., when a relevant change of the environment is recognized. The age of a message corresponds to the sum of delays from changing a value in a domain representation at one node until this change has worked through to all other nodes of the same group. We call this latency.

Latency: Let the latency $\lambda(m_i)$ for message i, where the interval between sending messages from node i is $\tau_i(m_i)$ and the time for transferring this message to the other nodes is $t_{i(i+1)}$ (as of eq.9) be defined as:

$$\lambda(m_i) = \tau_i(m_i) + \max\left(t_{i(i+1)}(m_i)\right) \tag{13}$$

In Figure 11, the coherence fraction κ_3 and latency λ_3 are plotted against the selected message sending interval τ^*_3 and related to the selected environment parsing interval ε^*. Note that while τ, λ and ε are measured in seconds, κ is unit less. That all metrics are plotted in the same numerical range at the y-axis is just a consequence of the selection of variables and values in the experiments. While the plot of Figure 11 is based on a specific set of data, it here serves as basis for discussion about general principles.

Following the x-axis from the right to the left we can see that the coherence drops to a stable zero at $\tau_3 \approx 0.3$ s, corresponding to $\lambda_3 \approx 0.63$ s. We can also see that the latency decrease linearly with τ_3 until it suddenly goes vertical at $\tau_3 \approx 0.1$ s, corresponding to a minimal latency of $\lambda_{min} \approx 0.45$ s. When τ_3 goes below the interval τ corresponding to λ_{min}, saturation causes the latency to grow quickly. In between these intervals of τ_3 we have the lowest recorded latency (and thus relatively fresh information) but not a coherent data set. In this case, *internal coherence* relative to the available data at the nodes is lost.

Following the y-axis upwards we find the selected parsing interval ε^* of the environment, that (i) may change according to preferences, and (ii) have two intersections with the data latency plot. The left intersection indicates that saturation (too short τ^*) may cause latency to rise above ε^*: As a consequence of trying to send data faster than the current means for data transfer can handle at a given time, data becomes too old. The right intersection corresponds to a system in which a node does not send updates to its peers as often as it recognizes relevant changes in its environment: In both of these cases e*xternal coherence* relative to the environment is lost.

This means that there are situations when either of or both internal and external coherence criteria can be violated. Data coherence in a DAC system can only be maintained for a specific set of message sending intervals.

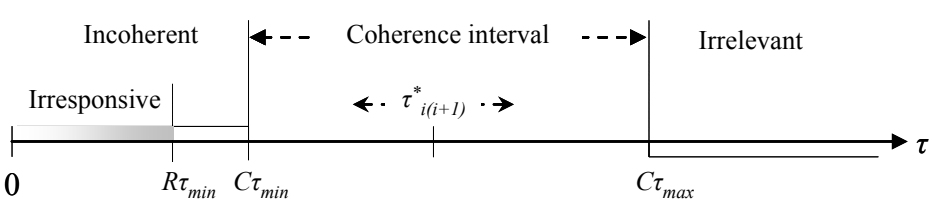

Figure 12: The coherence interval.

The coherence interval: Let the *minimal response interval* $R\tau_{min}$ be the shortest time allowed between updates to avoid a saturated communication system. Let the *minimal*

coherence interval $C\tau_{min}$ be the shortest time allowed for a coherent data set to restore coherence after a local change of a shared value, before the next change takes place. Let the *maximal coherence interval* $C\tau_{max}$ be the longest time between two consecutive updates of contextually relevant data necessary allowed for being able to recognize relevant changes in the environment. Now, take ε^*_{ij} (eq. 4 ff.), and take a look at Figures 11 and 12. Let the coherence interval be all values of the interaction interval τ^*_i for which

$$0 < R\tau_{\min} < C\tau_{\min} < \tau^*_i < C\tau_{\max} < \varepsilon^*_{ij} \tag{14}$$

If $\varepsilon^*_{ij} \leq C\tau_{max} < \tau^*_i$, external coherence criteria are violated and the group becomes irrelevant. If $\tau^*_i < C\tau_{min}$, internal coherence criteria are violated and the group will develop an incoherent data set D_j. If $\tau^*_i < R\tau_{min}$, the data transfer system becomes saturated and irresponsive as the latency grows while data is put on queue. Or, as shown in experiment 1, coherence may be lost (and even saturation introduced) even though $C\tau_{min} < \tau^*_i$, if "normal" race conditions occur in the Fabric level data distribution. Figure 11 show several coherence fraction data points below the main distribution. In a given system, coherence may always be zero (see also Figure 17 in appendix A). However, we shall recognize that the actual values of these temporal limits for every implemented system depends on hardware, logic and data characteristics such as i.e., processor and message transfer capacity, local computational context and the size of the message to be transferred (see eqs. 7 and 8 above).

The control paradox: A typical control loop (such as e.g., the OODA loop, [7] p.344) involves maintaining a good understanding of the current situation. Data relevant to a situation may be acquired through e.g., one node asking for feedback from the other nodes of a group. If however the means for communication are working above but close to the minimum coherence interval, such a query for more (or more frequent) data may cause the system to loose coherence and/or even become saturated and irresponsive.

Let us define the control paradox as the inverse relation between the amount of data we can gather about a system and the amount of behavior that can be created in the same system at the same time ("behavior" is here to be understood as a general performance criteria). As the processing capabilities at the cognitive units are to take care of both creating behavior (such as e.g., maintaining local/internal control and coherence) and handling messages with respect to other units and the environment (such as e.g., maintaining group wise/external control and coherence), there is a trade-off. The more information we ask for from a cognitive unit, the less the control of behavior while handling queries and generating data. Or putting it the other way; the more resources that is needed to control a node, the less information about the internal states of that node can be distributed to other nodes.

Also, since the responder in a systems-of-systems situation at some level typically itself is a controller at a lower system level, the corresponding lower level control mode [13] is likely to change for the worse. An attempt to improve high level control by increasing situation awareness through asking for more data from members of a cooperating group may therefore lead to an actual reduction of control at each responding node and a deterioration of behavior at the group at large. The control paradox for distributed cognitive systems suggests that, if being relevant in the first place, the more a controller works to improve understanding and increase situation awareness by gathering more information, the less actual performance is provided by the system under control.

7 Discussion and related work

The models and analyses handled above admittedly relates to a wide array of topics. For the purposes of this paper, we will focus on two areas of special interest. First are relations to phenomena such as observability and simultaneity normally discussed in modern physics. Second are implications of extending this reasoning to larger groups and/or multi-level organizations. A third theme relating to cognitive control and system models [12, 13] is recognized and presumed but left to some other presentation.

Relations to modern physics: This paper presents a thoroughly time based model for interaction between distributed asynchronous systems. From the principles of coherence intervals, data latency and saturation follows that there is an upper limit as to how much data can be distributed in a given system at a given time. As the amount of data also increase with increasing precision and granularity of the world model (eq.1) we can see that there is an upper limit to the amount of details that can be coherently maintained in the same. Or, to put it another way, there is a lowest level of detail beneath which the system can not be observed in a timely and coherent manner. We may call this the observability horizon. Data about events taking place beneath this horizon will not be made available to the observer until so allowed by the available data transfer capacity.

This reasoning is analogous to the theory of special relativity that was put forward by Einstein in 1905. [8] The semi-rigid body assumption put forward by Einstein may compare to our assumptions of a constant and specified computational context *C* and environment model *E*. If so, the change of the domain $D_j(t)$ over time as expressed in eqs. 1 and 2 above can be reduced to an n-dimensional information-time vector…

$$W_i(t) = D_j(t) = \left\langle \left\langle a_{j1}, v_{j1} \right\rangle, \left\langle a_{j2}, v_{j2} \right\rangle, \ldots, \left\langle a_{jn}, v_{jn} \right\rangle, t \right\rangle \tag{15}$$

…which is comparable to Minkowski´s four dimensional space-time vector:

$$"world" = \langle x, y, z, t \rangle \tag{16}$$

Openly admitting the lack of mathematical rigor, the argument for the relevance of this comparison is based on two facts. All DAC systems have a maximum data transfer speed (e.g., the $v(t)_{i(i+1)}$ of eq.8). The receiving node perspective in Figure 9 shows that for a saturated system, there need not be any correlation between arrival of data and the age of the same. These notions correspond to the maximum speed of light and the relativistic simultaneity proposed in the special theory of relativity. The consequence of a special information-relativistic theory on DAC systems is to begin with that a strong notion of shared data (and thus shared awareness) does not hold. As a data receiver, simultaneous events are those that are described with data arriving within a short enough interval Δt at my place. The actual relation between the occurrences of those events is difficult to assess without post operations analysis e.g., as presented in this paper.

Furthermore, when observing quantum particles, the observation itself affects the particle in an irreversible way. The Heisenberg uncertainty principle tells us that it is not possible to measure both position and direction of such a particle at the same time. If one aspect is known, the other is uncertain. [11] Reasons for these phenomena are related to the discrete properties of quantum particles and the fact that our means for observation (optics) are of the same scale as the subject observed (light).

Now, digital computational systems are inherently discrete. Quantum properties and uncertainties of information, time and energy with relation to the behavior of competing organizations is under investigation by e.g., Lawless et al. [14] However, not focusing at an organizational decision problem., this paper address a data transfer measurement problem analogous to the particle measurement problem, in that we need to move data to learn about the movement of data. The reasoning about the coherence interval and the control paradox presented above states that for a group constituting a DAC system, it is not possible to get both maximum behavior and distribute large amounts of data at the same time. Therefore we also suggest that DAC systems are subject for an information-system uncertainty principle: The more state information that is to be distributed among the units of a DAC-system (position) the less control the DAC is able to impose on its joint behavior (direction), and *vice versa*.

Relations to multi-node and multi-level interaction: As stated in the definitions above, groups may consist of several nodes (eq.5) each having a local world model (eq.1) that may contain a number of domains with a number of attribute-value pairs (eqs. 2 and 3). These definitions does not in any way restrict e.g., contents of the models, interaction topologies or instantiations of organizational structures. This paper refers to a three-node single level group. What can be predicted about DAC-systems that contains a larger number of nodes and/or multiple organizational levels?

When increasing the number of nodes of a group, new means for computation and communication are added as well. Assuming the Domain and Environment to be constant, the coherence interval (the span between being incoherent and irrelevant) is a function of possible changes in communication topology and capacity. It may increase or decrease depending on changes in e.g., broadcasting and/or routing capabilities. Experiment 1 indicates, though, that all other aspects equal, the coherence will decrease with increasing number of nodes. The reason for this is the spontaneous race phenomenon caused by slightly uncoordinated message distribution (Figure 5).

New organizational levels can be added by introducing new domains to existing nodes. These new domains can model the world at higher or lower abstraction levels, thus allowing for more general or specific deliberations about the world without requiring a substantial increase in the sheer size of the domains.

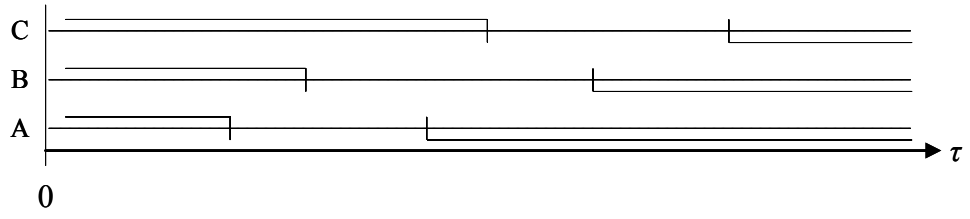

Figure 13: Coherence intervals at multiple organizational levels.

Figure 13 models a three level organization and its corresponding coherence intervals. Under the assumption that lower levels are working more closely in a geographical sense than higher levels do, and that short range means for communication enables higher capacity communication, the lower levels are also assumed to have lower minimum coherence intervals; coherence is more easily maintained. The assumption that these

groups also work closer to the field and thus are exposed to faster changes in the relevant environment, accounts for the corresponding decrease in maximum coherence interval; these units are more prone to become irrelevant. Hence we stipulate the following criteria for establishing and maintaining a coherent set of data within the organizations:

1. Bridging Domains: At least one node needs local representations of each of the adjacent domain levels. This means that if some node represents domains A and B, some other have to represent domains B and C.
2. Bridging Fabric: Each of the different means for communication that are used need to be co-located with some other such means at at least one node. This means that relaying nodes are needed between all applied communications technologies have to be used.
3. Overlapping coherence intervals: All adjacent levels in the organization need to establish and maintain coherence intervals that are to some extent overlapping.

A consequence of 3 above is that command will be irrelevant, if a higher level cannot react (maintain internal coherence) within the intervals needed to be relevant at a level below (meeting its external coherence criteria). If e.g., some of the networking criteria above are lost for level B of Figure 13, level C is not likely to be able to generate information with a pace that is relevant at level A. This is in effect also a detachment of the A-level unit into enforced self-organized and/or autonomous behavior. It shall also be recognized that an apparent reason for loss of coherence (decoherence) is that the rate of change in the relevant environment suddenly increases. Or, in other words, that we are surprised. This is a strong connection between this work and e.g., manouvre warfare. Also, the findings presented here support and formalize the temporal statements put forward by Perry and Moffat in 2004 ([19], see the latter part of the introduction).

An obvious brute force approach to maintaining coherence in DAC-systems would also be to make sure that there always will be “sufficient” capacity to meet “all” requirements on e.g., response time and redundant resources. This approach is unsatisfying in at least two aspects. *First* is that the relations to e.g., computational load in terms of intervals between requests to/from various nodes and the resulting computational complexity are unaccounted for (and probably also non-linear). More work is needed to understand the principles of scaling up DAC systems. *Second* is that, at least when used in some kind of competitive context, systems are likely pushed to their maximum performance limits. And even if the initial intention may be to keep the system within a safe and known performance envelope, situations are likely to develop when actual maximum performance is asked for. This is likely to push a DAC system into irresponsiveness and loss of coherence, stability and eventually also control. So, regardless of technical performance, we need to learn to control the emergent behaviours of DAC systems.

8 Summary and future works

In order to explore the characteristics of distributed asynchronous cognitive systems (DAC-systems), we have developed temporal models and experimental methods to perform empirical investigations of non-simulated systems. Our results show that coherence fractions can be calculated from time stamped distributed messages, and that the coherence of a group decrease with (i) decreasing the temporal intervals between sent

messages and (ii) increasing number of members in the group. The interval between messages arriving at a node is an important indicator of communication system saturation and increasing data latency. Large latency indicates data may be too old to be relevant.

The coherence interval model improves the understanding of how the amount of distributed data relates to the internal coherence and external relevance of a system. The control paradox tells us that there is a conflict between interaction and performance, leading to situations where increasing the rate of observations (as in e.g., an OODA-loop) of the states of a cooperating group may cause loss of coherence and irresponsiveness.

Or, to put it slightly more straightforward; the more we try the worse we may do.

As a consequence of the observability horizon imposed by the maximum data transfer capacity of a system, we have also identified strong analogies between phenomena appearing in DAC systems and phenomena first described in the theory of special relativity and by the Heisenberg uncertainty principle. It appears that our findings about DAC systems correlate well with aspects such as information-time vectors, relative simultaneity and the information-system uncertainty principle.

On a more concrete level, loss of coherence situations (decoherency) may be caused by e.g., lack of common domain descriptions, lack of common communication technology and/or of overuse of the available means for communication and computation. Decoherency may lead to organizational detachment, which in turn raise needs for autonomous behaviours and self-synchronization. While models provided here may describe the phenomena as such, more work is needed to understand and find support for decoherency prediction and avoidance.

Accepting the coherence interval as a necessary criterion for meaningful cooperation, and the control paradox as a valid model for system behavior, we also need to continue with investigating some of its other implications. One is the assumptions on the qualities, characteristics and phenomena associated with DAC systems as such and related metrics. Another relates to continued investigation of consequences of increasing the number of nodes in multi-level organizations. Finally, and maybe also the most intriguing, is the continued exploration of the identified analogies with theories and models of modern physics. The consequences of a relativistic information model need further investigation.

Acknowledgements: This work is funded by Kockums AB as part of the post graduate student program in which the author is participating. The author's supervisor, professor Rune Gustavsson of Blekinge institute of technology (BTH), has provided crucial support in discussing and formulating the mathematical models of this paper. Jonas Rosquist, Markus Andersson and Tomas Sareklint has provided both constructive critique and the technical support necessary for the experiments. Kockums AB is a part of ThyssenKrupp Marine Systems.

References

1. Alberts, D.S., J.J. Gartska, and F.P. Stein, *Network Centric Warfare - Developing and Leveraging Information Superiority*. 2nd revised ed. 2000, Washington D.C.: CCRP. 284
2. Alberts, D.S., J.J. Gartska, R.E. Hayes, and D.A. Signori, *Understanding Information Age Warfare*. 2001, CCRP. August 2001. ISBN 1-893-723-04-6
3. Bertalanffy, L.v., *General System Theory; Foundations, Development, Applications*. Revised ed. 1969, New York, NY: George Braziller, Inc.
4. Boyd, J.R., *Patterns of Conflict*. 1986, USAF. December 1986.
5. CCRP, *Code of Best Practice for Experimentation*. 2002, CCRP, Information Age Transformation Series. p.308. 2
6. Cebrowski, A.K. and J.J. Gartska, *Network-Centric Warfare: Its Origin and Future.* Naval Insitute Proceedings, 1998. **124/1/1, 139**(January 1998): p. 28-35.
7. Coram, R., *Boyd. The Fighter Pilot Who Changed The Art of War*. 2002: Little, Brown and Company. 485
8. Einstein, A., *Relativity: The special and general theory*. 1920: Elegant E-books. R.W. Lawson
9. Fredriksson, M. and R. Gustavsson. *A Methodological Perspective on Engineering of Agent Societies*. in *Proceedings from ESAW 2001*. 2001. Prague Springer-Verlag Berlin Heidelberg.
10. Fredriksson, M., Online Engineering; On the nature of open computational systems, 2004, Dissertation series no. 2004:05, Department of interaction and system design, School of engineering, Blekinge institute of technology, Ronneby, Sweden
11. Heisenberg, W.v., *Physics and Philosophy*. 1958: Penguin Classics. 144
12. Hollnagel, E., *Barriers and Accident prevention*. 2004: Ashgate. 226
13. Hollnagel, E. and D.D. Woods, *Joint Cognitive Systems*. 2005, Boca Raton, FL: CRC Press Taylor and Francis Group. 223
14. Lawless, W.F., L. Chaudron, and C.P. Abubucker. *Metrics for uncertainty in organisational decision-making*. in *2006 Command and Control Research and Technology Symposium (CCRTS-06)*. 2006. Coronado, CA June 20-22, 2006: US DoD CCRP.
15. Lindh, J.-O. *The Ecological Layer of Warfare*. in *9th Intl. Command and Control Research and Technology Symposium 2004 (ICCRTS-04)*. 2004. Copenhagen, Denmark Sept. 14-16 2004: CCRP.
16. Lindh, J.-O. *Tools and Technologies for Interaction in Open Systems*. in *Military Technology 2*. 2005. Stockholm, Sweden October 25-26, 2005.
17. Moffat, J., *Complexity Theory and Network Centric Warfare*. DoD Command and Control Research Program. 2003, Washington D.C.: US DoD CCRP. 161
18. Perry, W.L., R.W. Button, J. Bracken, T. Sullivan, and J. Mitchell, *Measures of Effectiveness for the Information-Age Navy: The Effects of Network-Centric Operations on Combat Outcomes*. 2002, RAND. p.190. MR-1449-NAVY
19. Perry, W.L. and J. Moffat, *Information Sharing Among Military Headquarters*. 2004, RAND. p.158. MG-226-UK
20. Tzu, S., *The Art of War: Complete Texts and Commentaries*. 2003, Boston & London: Shambhala Publicatons Inc. 457 p. T. Cleary

Appendix A: Solving the temporal equations for message time corrections

We are sending messages between three asynchronous nodes. They do not have synchronized internal clocks. We want to investigate the relation between how long it takes to send a message, how often these messages are sent (from each node) and to what extent the nodes have a coherent set of data. The following temporal notions are used in this appendix:

t_i Actual time-point; measured at node *i*.

t_{ic} Corrected time-point at node *i*.

$t_{i(i+1)}$ Message transfer time; interval between send-time t_i at node *i* and reception time $t_{(i+1)}$ at node *(i+1)*.

$\delta_{i(i+1)}$ The measured interval between t_i and $t_{(i+1)}$ for one event occurring simultaneously at both locations.

τ_i The interval at node *i* between initializations of sending messages m_i and m_{i+1} to the group.

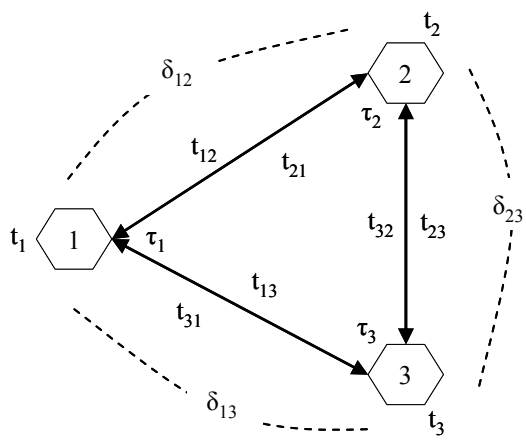

Figure 14: Temporal definitions as used in the experiment.

A.1 Experiment method overview.

Each node collected a message identity (MsgId) and its corresponding local time (t_i) at a local data base in each node. The data base is in this case representing the domain D of the collaborating group. The message identity is a composed six digit integer representing the node identity (1, 2 or 3) and a sequentially increasing integer. The time stamp is the local computer clock time at creating the MsgId-time post in the local data base. After the experiment, the set of sent messages that had arrived at both other nodes was selected and exported as text can be seen in Table 1.

The Microsoft Excel™ time format with 1.00 corresponding to 24 hours is used. 1 second is thus corresponding to $1/24/60/60 = 1/86400 \approx 1.15741\text{E-}05$. However, the measurement precision at the databases in the nodes is 1 millisecond, which corresponds to 1.15741E-8 in the used format. These calculations were done in two steps. The first

Table 1: Sample raw data from test series 1.

MsgId	t_1	t_2	t_3
100001	38831.596175312500	38831.596264780100	38831.596182175900
100002	38831.596238807900	38831.596322604200	38831.596240972200
100003	38831.596296759300	38831.596380590300	38831.596298842600
…	…	…	…
300355	38831.616793252300	38831.616874965300	38831.616790891200

step served to establish the method and assess the relevance of the measurements. The second step served to investigate the consequences of decreasing the interval between sending messages of the same length.

Time correction calculations: The node-time difference $\delta_{i(i+1)}$ is the measured interval between t_i and $t_{(i+1)}$ for one event occurring simultaneously at both locations. While the time difference can be approximated as the temporal difference between e.g., nodes $i+1$ and $i+2$ receiving one message sent by node i, we do still not know the actual message transfer times

The nodes are sending messages with constant size with the interval τ_i between sending messages is also constant at 5 seconds. Let node 1 have the reference time.

$$t_1 = t_1 \tag{A.1}$$

$$t_2 = t_{2c} + \delta_{12} \Rightarrow t_{2c} = t_2 - \delta_{12} \tag{A.2}$$

$$t_3 = t_{3c} + \delta_{13} \Rightarrow t_{3c} = t_3 - \delta_{13} \tag{A.3}$$

The temporal interval t_{ij} needed for message transfer between nodes i and j are:

$$t_{12} = t_{2c} - t_1 \tag{A.4}$$

$$t_{13} = t_{3c} - t_1 \tag{A.5}$$

$$t_{21} = t_1 - t_{2c} \tag{A.6}$$

$$t_{23} = t_{3c} - t_{2c} \tag{A.7}$$

$$t_{31} = t_1 - t_{3c} \tag{A.8}$$

$$t_{32} = t_{2c} - t_{3c} \tag{A.9}$$

Since we use radio based communication we assume the message transfer times tt_i to be equal for the vector of all messages m_i originating from the same node. In order to address equations (A.4) to (A.9) we also assume that the average of the total message transfer times $t_{i(i+1)}$ for all messages sent from node i are equal:

$$t_{12}(\overline{m}_1) = t_{13}(\overline{m}_1) \tag{A.10}$$

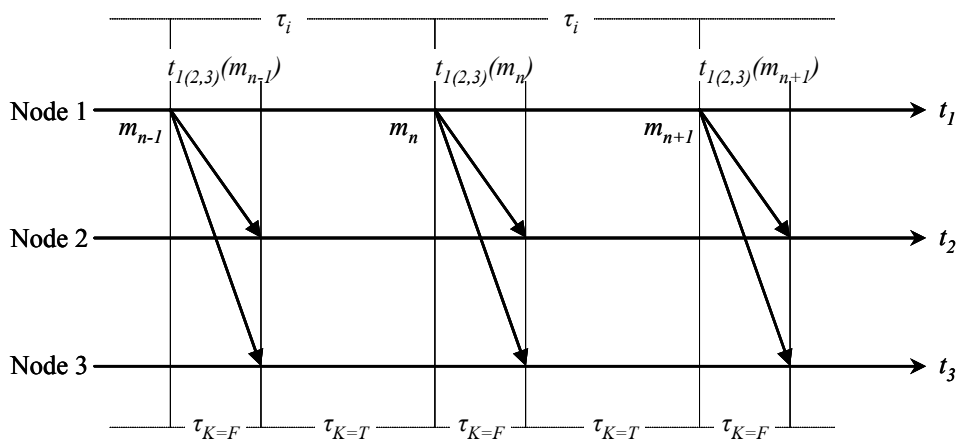

Figure 15: Temporal message distribution model for node 1.

$$t_{21}(\overline{m}_2) = t_{23}(\overline{m}_2) \tag{A.11}$$

$$t_{31}(\overline{m}_3) = t_{32}(\overline{m}_3) \tag{A.12}$$

This and the experimental design give the following three correction times:

$$\delta_{12} = t_{31}(\overline{m}_3) - t_{32}(\overline{m}_3) \tag{A.13}$$

$$\delta_{13} = t_{23}(\overline{m}_2) - t_{21}(\overline{m}_2) \tag{A.14}$$

$$\delta_{23} = t_{13}(\overline{m}_1) - t_{12}(\overline{m}_1) \tag{A.15}$$

Coherence calculations: Coherence K is True when all nodes holding representations of domain D have the same values. Coherence K is False when at least some node that holds a representation of domain D have differing values.

$$\mathrm{K}(D,t) = True \quad iff \quad \forall j\,(\mathrm{K}_j(t) = True) \tag{A.16}$$

$$\mathrm{K}(D,t) = False \quad iff \quad \exists j(\mathrm{K}_j(t) = False) \tag{A.17}$$

For *a one sending node system* this translates to K becoming False whenever a value v is changing at a node i and staying False until the new value v_n is transferred to all other nodes $i+1$ of the same group. K becomes True again when the last of the receiving nodes $i+1$ has been updated. K stays True until the next value v_{n+1} is defined.

The value v_n is part of the message m_n which is created at the same time as v_n is changed (eq. 6). We then get the following definitions.

From this follows that K=T for the duration $dt_{Kj=T}$ between the last update of the previous message until the next value is changed and a new message is created and that K=F for the duration $dt_{Kj=F}$ of message transfer from node i to all nodes j. Note that $dt_{Kj=T}$ is zero if a value of some attribute is changed before its previous value has been distributed to all other nodes of the group. So, the coherence fraction κ_j is

$$\kappa_j = \frac{dt_{\mathrm{K}j=T}}{dt_{\mathrm{K}j=T} + dt_{\mathrm{K}j=F}}, \quad 0 \le \kappa_j \le 1 \tag{A.18}$$

… where …

$$dt_{\mathrm{K}j=T} = t_i(m_n) - \left[t_i(m_{n-1}) + \max\left(t_{i(i+1)}(m_{n-1})\right)\right] \quad \begin{cases} t_i(m_n) > \left[t_i(m_{n-1}) + \max\left(t_{i(i+1)}(m_{n-1})\right)\right] \\ Otherwise\ 0 \end{cases} \tag{A.19}$$

… and …

$$dt_{\mathrm{K}j=F} = \max\left(t_{i(i+1)}(m_n)\right) \tag{A.20}$$

A.2. Experiment 1 - The time correction method

The purpose of experiment 1 is to develop and validate the time correction method an calculations. The duration of the data collection of step one is approximately 1800 seconds and the joint data base contains about 350 posts per node, totalling at about 1050 points of data. .

As clock times t_i are both set differently and have varying speeds, $\delta_{i(i+1)}$ was calculated as a linear correction on the form $y=mx+k$ where x is the i'th message and k is the average time difference. In this instance we get the following correction equations.

$$\delta_{12} = 6.414449\mathrm{E}\text{-}10\mathrm{x} + 8.146306\mathrm{E}\text{-}05 \quad \mathrm{R}^2 \approx 0.58 \tag{A.21}$$

$$\delta_{13} = 2.297506\mathrm{E}\text{-}09\mathrm{x} - 2.751824\mathrm{E}\text{-}07 \quad \mathrm{R}^2 \approx 0.84 \tag{A.22}$$

$$\delta_{23} = 1.676894\mathrm{E}\text{-}09\mathrm{x} - 8.174180\mathrm{E}\text{-}05 \quad \mathrm{R}^2 \approx 0.76 \tag{A.23}$$

Equations (A.2), (A.13), (A.16) and (A.3), (A.4) and (A.17) gives

$$t_{2C} = t_2 - (6.414449\mathrm{E}\text{-}10)(\text{Message number}) + 8.146306\mathrm{E}\text{-}05 \tag{A.24}$$

$$t_{3C} = \mathrm{t}_3 - (2.297506\mathrm{E}\text{-}09)(\text{Message number}) - 2.751824\mathrm{E}\text{-}07 \tag{A.25}$$

These operations yield a time difference corrected set of data as exemplified in Table 2. Equations (A.4) to (A.9) are now used to calculate the message transfer times. The message creation times $t_i(m_i)$ for all messages m_i of node i were converted from the MS Excel™ format to seconds and normalized to t_N using the following formula.

$$t_N = \left(t_i(m_i) - t_1(m_1)\right) * 86400 = \left(t_i(m_i) - 38831.6688141204\right) * 86400 \tag{A.26}$$

Table 2: Corrected data from test series 1.

MsgId	t_1	t_{2C}	t_{3C}
100001	38831.596175312500	38831.59618331640	38831.59618244880
100002	38831.596238807900	38831.59624113980	38831.59624124280
100003	38831.596296759300	38831.59629912530	38831.59629911090
…	…	…	…
300355	38831.616793252300	38831.61679327450	38831.61679035080

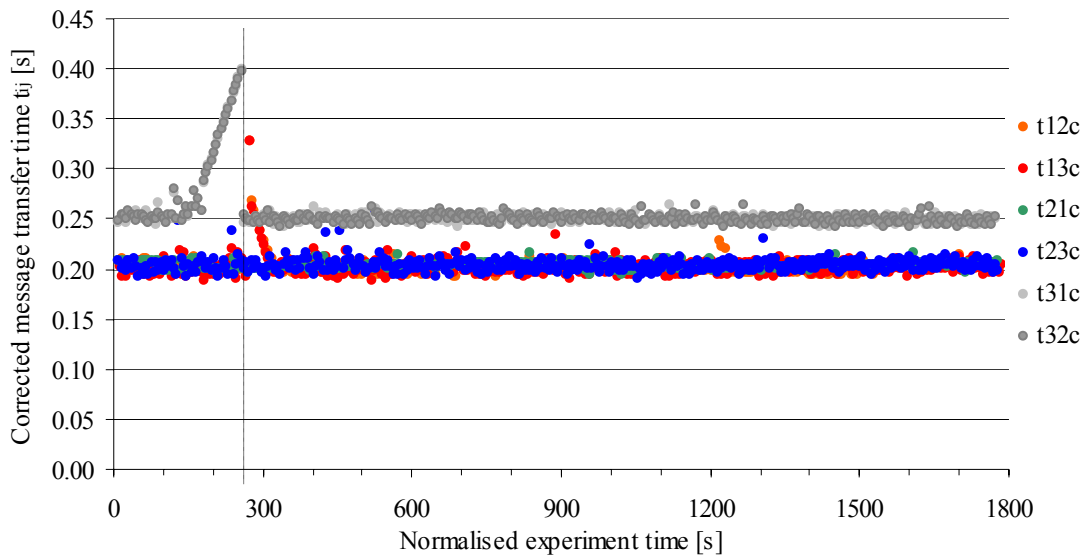

Figure 16: Message transfer times; result from test 1.

The results of these calculations are plotted against the normalised experiment time t_N in Figure 16. We can see an anomaly in distribution times of nodes *1* and *3* occurring at about $t_N = 270$ seconds. This is caused by interference between the respective radio systems of the two nodes. The VHF radios used have a carrier detection mechanism and a buffering function. If the radio frequency is already in use, this function allows a message to be queued until the frequency is free again.

From time to time both nodes intend to send their messages at overlapping time intervals. In this particular case, the internal clocks of nodes *1* and *3* are drifting slightly. (see eq. A.17). The radio transmissions are beginning to interfere with each other. At $t_N = 150$ node *1* is still transmitting when node *3* wants to begin sending its message. So, *3* waits until *1* is done. The drift continues until at $t_N = 270$ node *3* eventually transmits

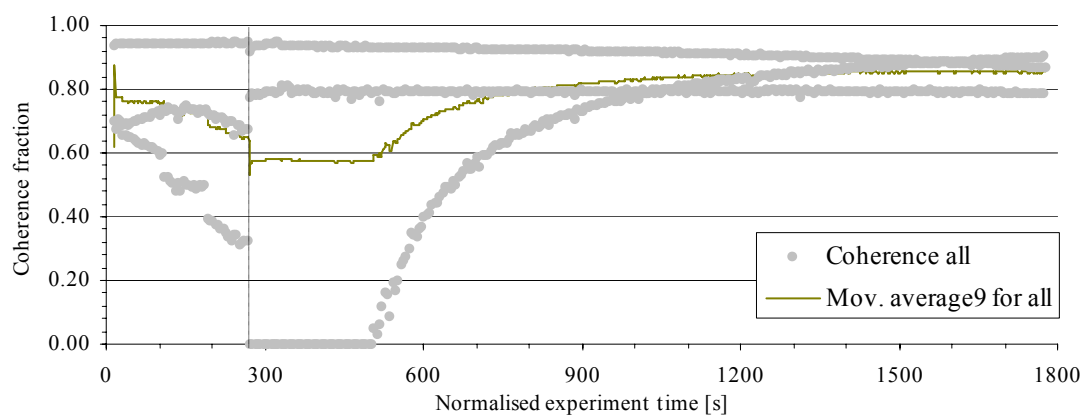

Figure 17: Test 1 coherence fractions with respect to all 3 nodes.

before node *1*, which thereafter will have to wait for its turn on the air. At this instance two messages are also lost as both nodes begin transmission with so small temporal difference that the radios carrier detection system does not detect the conflict. Therefore, both messages are transmitted but none of them resolved at the other nodes. At approximately $t_N = 330$ the conflict has disappeared.

As we also can see the overall impression is that messages from nodes 1 and 2 are distributed in similar times, while messages from node 3 are taking constantly longer time. Assuming that the corrected message distribution time t_{ijC} in seconds can be described with normal distributions N(x;σ), calculated after the anomaly ending at $t_N=330$, we get:

$$t_{1jC} = N(0.202;0,0054) \tag{A.27}$$

$$t_{2jC} = N(0.205;0,0054) \tag{A.28}$$

$$t_{3jC} = N(0.251;0,0038) \tag{A.29}$$

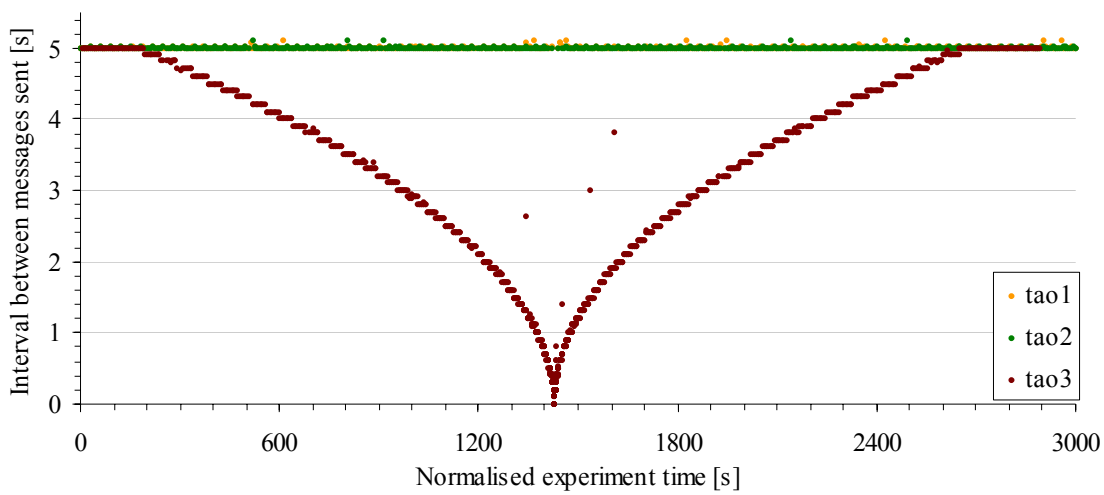

Figure 18: Changes in transmission interval τ in experiment 2.

A.3. Experiment 2 – Varying intervals *τ* between messages

The purpose of experiment 2 is to investigate the effect of decreasing the interval τ between messages sent with respect to system coherence and recovery times. The experimental setting is the same as in step one with the difference that the interval between sending messages from node 3, τ_3, is decreasing from 5 seconds to zero and then increasing to 5 again, in increments of 0.1 second. Ten messages are sent at each interval τ. In order to establish stable start and end states, 40 messages were sent at $\tau_3 = 5$ seconds in the beginning and end of the experiment respectively. Figure 18 presents the intervals τ_i between messages over the normalized experiment time t_N. This lead to a total duration of the data collection of approximately 3000 seconds and a total of about 2400 posts in the joint data base.

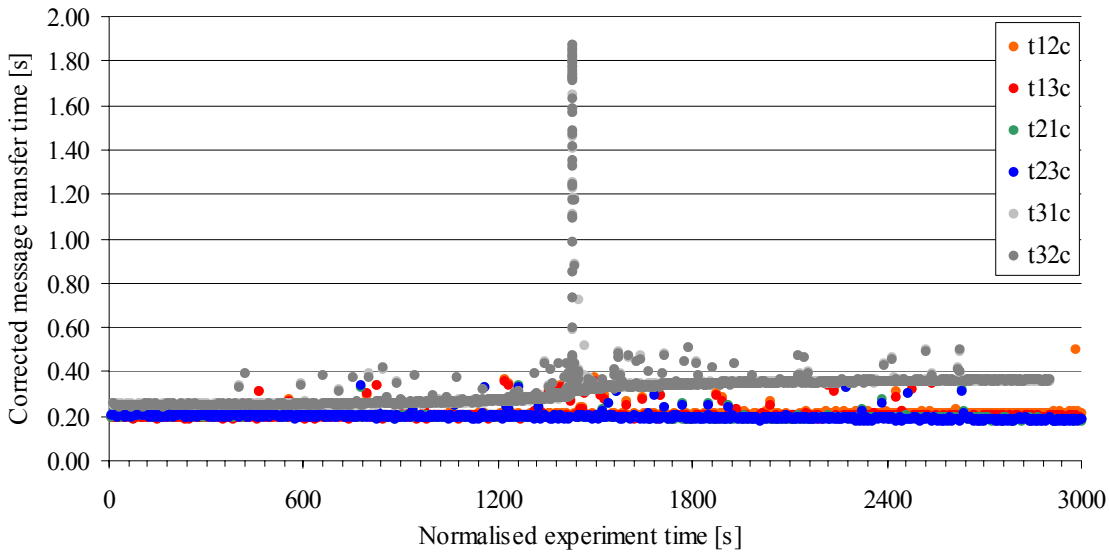

Figure 19: Message transfer times; result from experiment 2.

Calculations for temporal correlation of the node specific data sets as described in Equations (A.1) to (A.21) are also applied to the data set of step two. The resulting corrected message transfer times are presented in Figure 19.

For t_N >600 the resulting the message transfer times are quite similar to those of step one. One significant feature appears at t_N = 1440 where t_{3j} rise sharply from about 0.30s to approximately 1.90s. This coincides with the minimal interval $\tau_3 = 0$. We will look more thoroughly on this below.

Another significant feature is the vaguely s-shaped raise of t_{3j} from 0.25s to 0.35s over the duration of the experiment. This is an artefact of using a linear correction equation based on the number of messages rather than on elapsed time in the experiment. Since there are many more messages per time in the middle of the experiment, the message-linear correction gives a more sloped section in the time-linear representation of Figure 19 This correction factor misfit is likely also the reason for the relatively diverged measures on the right hand side of Figure 20.

For *a multiple sending node system* the coherence becomes more complicated to assess. As all the times of all logged events are normalized, we calculated the normalized coherence False (eq. A.22) and True (eq A.23) times for all messages and sorted the message transfers in ascending order on the t_{iN}. The coherence fraction κ was calculated for all messages and plotted against the normalized experiment time, as can be seen in Figure 20. A moving average over 9 posts was added to support the visual inspection of the results.

One significant feature of the plot is the loss of coherence at $t_N \approx 1400$ s. This corresponds to the sequence in which τ_3 approached 0.2 to 0 second intervals between messages. As the normalised message distribution time $t_{i(i+1)}$ is 0.20-0.25s, this is in accordance with our expectations following eq. A.27.

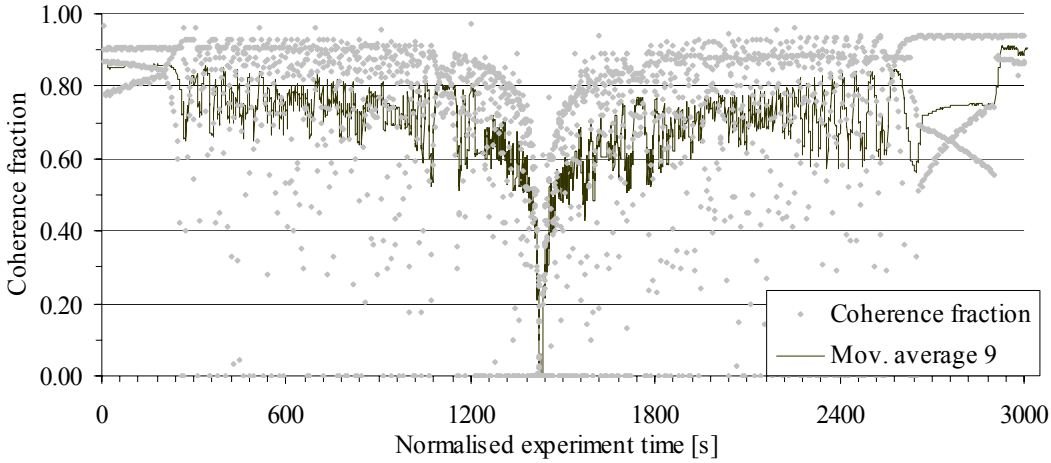

Figure 20: Experiment 2 coherence fractions with respect to all 3 nodes.

Another significant feature of the plot are the relative stability of κ in the beginning and end of the series, and the vague but apparent correlation of the moving average and the changing τ. We can also see that level of the moving average at all times is lower for the group of three nodes than for the coherence plotted for the single node 3.

A.4 Experiment 2 – the time-time plot calculations

The time-time plots presented in figures 9 and 10 are based on two calculations.

First, the message arrival times are plotted right-on as the normalised arrival time t_{N1} of messages sent from node 3 to node 1. This data is readily available by applying eq. A.26 to all rows under header t_1 in Table 2 that concerns messages originating at node 3.

Second, the corresponding message creation times are calculated as t_{N1} minus the largest of the message delivery times t_{2C} and t_{3C} associated with the same message, as presented in eqs. A.24-25. The resulting curves appear when these two data sets are scatter-plotted on two linear timelines.

/ End of Appendix A.