# Integrated Communications Demonstrator

Tom Nabarro[a], Andrew Dunlop[a], George Purcell[a], Gary Francis[b] and Peter Smith[b]
[a] Roke Manor Research Limited, Old Salisbury Lane, Romsey, Hampshire. SO51 0ZN
[b] BAE Systems Advanced Technology Centre, West Hanningfield Road, Great Baddow, Chelmsford, Essex. CM2 8HN.

## Abstract

*The SEAS DTC CC022 integrated communications demonstrator has brought together four previous DTC communications research projects, targeted at ad-hoc network environments, into a collaborative software architecture. A face recognition application was developed to run over the network to demonstrate the beneficial features of the contributing projects (CC001, CC004 and CC007 (Roke) and CC005 (BAE Systems ATC)).*

Keywords: Resilient networks, ad-hoc networks, clique based pricing, capability awareness

## Introduction

The objective of the integrated demonstrator project (CC022) was to show the interworking of a number of previously completed SEAS DTC CC theme projects, via a demonstration scenario with which they all have compatibility.

The original projects were:

- *CC001 Resilient Network (ResNet)* [1]. This project culminated in a demonstrator for an algorithm that selects dual redundant paths through the Active Data Network (ADN), allowing fast recovery from link and node failures. The demonstrator contains a number of network nodes connecting the streamed data terminal nodes and a graphical monitoring system to show the links selected for use. Hardware links connecting the nodes are broken and made by manual intervention in the cabling at a patch panel.
- *CC005 Dynamic Agent-Based Communications Management* [2]. This project based path pricing on an algorithm for discovering interfering link sets in a single waveband wireless ad-hoc network. Interfering link sets are maximal cliques derived from a graph of interfering links using the Bron-Kerbosch maximal clique algorithm. Path pricing is used for application bandwidth management in order to eliminate congestion in the wireless network once the pricing algorithm has converged.
- *CC004 Resilient Distributed Control in the Presence of Unreliable Communication* [3]. This project produced a communications manager that discovers network connectivity and provided a resilient mechanism for sharing information
- *CC007 Capability Awareness* [4]. This project proposed techniques for defining and distributing capability information. A simulator containing an urban wireless propagation model and mobile agents was created and was used to demonstrate an improvement over simpler data sharing mechanisms in large networks of agents using broadcast wireless techniques.

The common factor in all of these Communications and Control research theme projects was multi-node ad-hoc network models and this remained the chosen substrate for this demonstrator.

This paper attempts to convey a flavour of the work involved, full details being contained in the final project report [5].

## Architecture

The CC001 (ResNet) demonstrator consisted of a physical network of interconnected nodes in a partial mesh configuration. The configuration of the mesh was chosen to demonstrate the selection of disjoint link sets in the dual paths and also to demonstrate the additional protection of fast link repair mechanisms which are part of the algorithm. ResNet configuration allows the network links to be allocated notional bandwidths, which are used in the selection of links by the ResNet path selection algorithm. The ResNet algorithm for link selection assumes independence of the links, that is, it assumes that they do not interfere with each other, and that the selection of one link does not affect the bandwidth available on another. In reality, in a meshed network with a single carrier frequency, the wireless bandwidth between (at least) adjacent links is shared.

The CC005 demonstrator in contrast, consisted of a network of nodes with simulated wireless connections, in which nodes (and consequently topology) can move and disconnect or reconnect from other nodes. The links are treated as homogeneous wireless links, all with the same basic bandwidth. The software is designed to operate within networks of any configuration but the previous demonstration used a sparse branching network. This type of topology was used due to the limited number of nodes available in the demonstrator.

Thus one challenge for the CC022 project was to create network behaviour that contained the desirable elements from both CC001 and CC005.

The requirements of the demonstrator architecture were:

- That the network support 2 data sources (at least) and a single data sink node, see Figure 1.
- That all the terminal nodes are connected to two or more internal network nodes (so that the action of the disjoint path algorithm may be demonstrated).
- That the resulting cliques allow for differential pricing to affect application traffic reservations made on the source to sink paths. This requires that the source nodes each be connected to a different pair of internal network nodes in Figure .

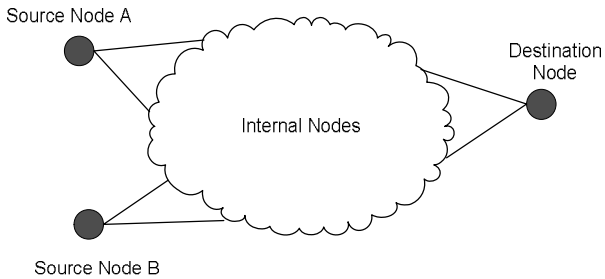


**Figure 1: Logical architecture of the CC022 demonstrator**

A demonstrator topology with as good a compromise as possible within the constraints of resources was decided upon. The topology selected was aimed at being disparate enough to distinguish independent cliques whilst maintaining sufficient regularity of neighbouring nodes to allow for bypass creation. The chosen topology is an example of what is possible, the topology configuration is arbitrary and should not adversely affect the performance of the individual component processes.

A 10-node wired network was created as shown in Figure , (which is actually the topology display associated with the monitoring tool). The source nodes are node 1 and node 10 and the destination node is node 2.

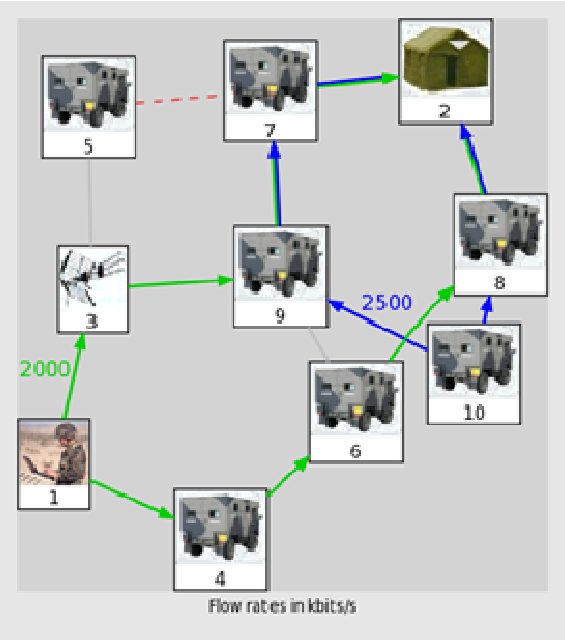


**Figure 2: Network monitoring tool display**

Each node has a fully functional instance of each of the contributing technology components. In addition, on separate nodes not displayed in Figure 2, are two centralised components; the Monitor and the Management Gateways which are used for administrative tasks.

The above diagram displays the default Multi Protocol Label Switching (MPLS) paths (before re-routing or bypass activation). The two chosen routes from node 1 to node 2 are shown in green while those from node 10 to node 2 are in blue. Thin grey lines between nodes indicate other available physical connections.

## Demonstrator Scenario

The data from each source is a video stream, with the importance of data and therefore its priority, deemed to be proportional to the number of faces detected at the video stream origin. At the destination, the frame size of each displayed video reflects the priority of the data, being varied in proportion to the transmitted bandwidth.

SEAS DTC CC004 / CC007 technologies allow for the normalisation of data importance into priority. This is calculable through the dissemination of the video stream's content relevance throughout the ad hoc network. The normalised priority of the data is then supplied to the CC005 software layer which converges to an optimal data rate for this traffic in the network. CC005's data rate calculations maximise the overall bandwidth utilisation throughout the network.

The resilience of the network is illustrated by the demonstrator when a link is lost. If the disabled link is part of a data flow, the data will be rerouted immediately. Even in the downtime during rerouting of this data path, flow data will still be being transmitted over the alternative independent path to avoid any potential data loss. The monitor will update the topology display to show the link loss and to represent the change in the flow route.

The CC001 component will discover new links and nodes and reroute to shorter and higher bandwidth routes if possible.

The CC005 component reacts to network changes by altering the bandwidth allocated to different application data flows in order to maximise the utility of the network according to priority and capability.

## Development

As well as defining several interfaces between the contributing CC00x components, various changes to the components themselves were needed. Some of the changes are indicated in the following paragraphs.

Modifications to the CC001 Ad-hoc On-demand Resilient Path (AORP) algorithm and implementation were needed, particularly the latter which previously had been rather restrictive e.g. a given node could only be on one MPLS path.

To establish dynamic links (i.e. facilitating multiple sources and / or multiple

destinations, enabling each node to potentially initiate or participate as a link endpoint), AORP had to be developed to derive endpoint subnet addresses and communicate these to other nodes.

The CC005 monitoring tool was essentially adopted although some features of the CC001 tool were incorporated. Various changes to the ResNet software had to be made to achieve compatibility. Several monitoring tool enhancements were made including improved responsiveness and readability of both the topology display and the bandwidth history display.

Updates were required to the COMMAND and Application Bandwidth Controller Daemon (ABCD) elements of the CC005 software for their integration into the CC022 demonstrator.

Bandwidth reservation messages are transmitted by ABCD and are *snooped* by instances of COMMAND on the nodes forming the data flow path. Routing the messages using MPLS and also tunnelling them, led to the requirement to parse the message taking into account the extra IP/GRE/MPLS headers in the packet. These additional headers had a knock-on effect which resulted in the messages not being able to be filtered on port numbers using the normal `pcap` filtering methods. (This is required so as not to snoop the actual data flows themselves, which would result in the COMMAND tool having to perform unwanted message parsing.) This was successfully resolved using a bespoke Berkley Packet Filter (BPF).

The major change for the ABCD tool for integration into the demonstration network was the implementation of a revised method of triggering a data flow. Originally, the ABCD tool would read its configuration file (containing a definition of the required data flow, in terms of data rate, priority, destination, etc) and immediately start transmitting bandwidth reservation messages. This in effect indicates the start of the data flow and the possible bandwidth limitation of that flow. In order for the integrated CC022 system to operate as required, ABCD needed the facility to be controlled from a separate application (in this instance CC007).Updates and additions to ABCD included:

- Operation over multiple interfaces rather than over a single interface.
- An acknowledge mechanism was introduced so that ABCD can acknowledge the receipt of control messages to the controlling application (CC007).
- The format of the bandwidth reservation message transmitted by ABCD was updated to include a generic header block, because the message is no longer just used internally to the CC005 software tools, but it used to communicate the requested data flow bandwidths to one or more other applications.
- The requirement to use two divergent paths through the network for each data flow led to the need to be able to define the flow source and destination addresses using interfaces not being used by the CC005 network COMMAND instances. This resulted in the price messages returned to ABCD containing information on links through which the flow passes being reported using different IP addresses.
- The original tool operated with a single path for each data flow. The total price for the flow was simply the sum of the prices for each link making up the flow. The CC022 demonstrator has two separate paths for each flow. The total aggregated price for each path is calculated separately, and the higher price is used to calculate the bandwidth allocated to the flow.

- The concept of *legacy* and *co-operative* applications at flow sources was introduced in CC005, but not fully implemented. All applications were assumed to be legacy and so bandwidth limitation was implemented by ABCD placing queuing disciplines (token bucket filters) on the appropriate interface(s). Co-operative applications are now specifically catered for. The queuing disciplines are not enforced and the application bandwidth is regulated via CC007.

CC005 topology discovery uses packet broadcasts, AORP uses packet multicast over designated interfaces for algorithm implementation and discovery purposes. The CC022 network had to be broadcast enabled and testing was performed to ensure that multicast and broadcast techniques did not interfere with each other.

The original CC004 / 007 simulator had a number of application modules, including capability information gathering and communications. New behaviour modules for CC022 were based on these modules. The capability module was expanded to maintain a knowledge database containing insights regarding CC022 application instances. These contained information about the applications; the source and destination IP addresses; the requested and allocated bandwidths; the number of faces detected and the priority that CC007 had determined.

Priority calculations are performed for the local application based on its knowledge of other applications on the network. It is triggered whenever the local application updates the number of faces detected or when, after a knowledge base update, the number of faces in another application changes. The priority value is then used to update the application capability information locally. The propagation of information between nodes is performed by CC004.

The original CC004 / 007 simulator comprised a single Java Virtual Machine (JVM) in which multiple Java Agent Development Environment (JADE) software agents ran the CC007 and CC004 behaviour modules. To avoid communication between these agents being completely reliable, CC004 simulated real world characteristics such as latency, jitter, propagation losses etc. In CC022, CC004 messages are sent across a real network and this necessitated replacing this code with a UDP socket and all that that entails (message fragmentation, buffering etc).

## Integration Issues

A selection of issues that occurred during integration are described in the following paragraphs.

Performance issues were encountered which caused topology packets, used by CC005 monitor software to represent current network topology, to be delayed. Excessive topology packet delay would result in node lifetime timeouts (configurable parameter within CC005 software) occurring. If a node lifetime timeout occurred the visualisation tool would incorrectly indicate a node having been disconnected from the network. Extending the length of the node timeout period allows for greater packet delays but reduces responsiveness of the visualisation tool to node or link loss.

Performance issues were resolved by a combination of traffic filtering and adding an additional node to perform monitoring functionality. The additional node was added due to the Gateway's processing constraints. The Gateway could not perform the monitor functionality whilst performing Gateway related tasks (mostly network related). A more powerful PC was introduced which was able to perform the graphical processing required by the

monitor and process messages in a timely manner.

The use of Virtual Private Network (VPN) tunnelling for generating the two divergent paths for data flows resulted in CC005 bandwidth reservation messages having Generic Route Encapsulation (GRE) headers that required parsing. It also involved data flows being defined with source and destination addresses on interfaces not being used by the COMMAND instances in the demonstrator network. Both issues were successfully resolved.

The application data is tunnelled across the network from and to splitter combiner endpoints, on a separate subnet to the AORP and CC005 discovery activity. In addition, the splitter combiner routes data for each path across separate subnets. This layered network led to some confusion with the CC005 software and price information was not being conveyed correctly to the flow source due to subnet boundaries. Modifications were made to ABCD to allow discovery of interfaces on different subnets.

The interpretation of the MPLS tables on network nodes by the CC005 COMMAND tool was modified to enable the selection of the most recent MPLS entry for a specific label (label-path is a one-to-one mapping). The entry with the highest key for a specific label is used, this ensures that the MPLS route used to convey CC005 price and bandwidth utility messages (alongside application data) is always the most recent.

The network configuration of the ResNet nodes presented a problem as these are configured with each of the inter-node network interfaces having the same IP address (as they are modelling a single wireless interface). CC004 packets need to be broadcast on all the ResNet interfaces of each node but not on the COMMAND monitor interface. To do this all CC004 packets need to be sent from each of these interfaces in turn. It is however; not possible using a Java standard library datagram socket to send packets to specific interfaces if these can not be distinguished by IP address. Therefore a new software interface was introduced using Python sockets which can be bound to specific interfaces by name. This allowed the Java CC004 socket class to use a standard socket and remain independent of the node configuration.

In the original CC004 / 007 simulation each of the agents were in the same JVM and had similar levels of tasks to process. However in the demonstrator for this project each agent was on a different node and some of these were newer machines than others. In addition to this the amount of processing required on each node varies considerably. All nodes need to update their knowledge base and propagate the capability information through the network but in addition to this the source nodes are required to communicate with the ABCD and with the Application as well as working out priorities. Because of these mismatches, eventually the nodes on slower machines or with more to do would become overwhelmed with the number of CC004 messages waiting to be processed.

To solve this problem two approaches were taken. The first was to reduce the volume of CC004 messages which were being sent by each node as these were being sent much more often than was necessary. The second approach was to introduce a queue which messages would be added to as they were received but to limit its size. When it became full older messages could be dropped as newer messages will have superseded them anyway.

## Conclusions

The resilience of the network (i.e. the contribution of CC001) has been illustrated by the demonstrator's behaviour when a link is lost. If the disabled link is part of the

data flow, the data is rerouted immediately. In the downtime during rerouting of this data path, flow data is still transmitted over the alternative independent path (of the path pair) to avoid any potential data loss. The monitor updates the topology display to show the broken link, which was subsequently removed from the topology. The monitor represents changes in flow route and discovers new links and nodes added to the network. Shorter or higher bandwidth routes are switched to on discovery. The monitor updates the topology display to show the changes in connectivity and routing

The demonstrator reacts to network changes by modification of the bandwidth (policed by CC005) allocated to different application data flows in order to maximise the utility of the network. This is observable on the topology display (Figure 3), the bandwidth utilisation history display and in the frame sizes of the transmitted video streams. Similarly, when one data stream becomes more important than the other (because more faces are recognised), the equitable rearrangement of bandwidth is reflected in these displays, thereby also confirming that CC004 and CC007 are causing capability information to be propagated.

The implemented project architecture reflects the proposed project architecture closely, the main implementation problems encountered related to network traffic processing, addressing and interception. Performance issues in the parsing of user mode packets were mitigated using a custom data link layer filter to reduce the number of packets needing to be parsed. Dedicated hardware was employed for more computational intensive tasks such as monitor and visualisation tools. This reduced the processing demands on the active data network nodes.

Monitoring tools were developed to illustrate the network activity controlled by the integrated software components. The network related behaviour of the software component was tuned to maximise overall performance.

However, there remains a large area of overlap with respect to local topology discovery which is still executed essentially independently by CC001, CC004 and CC005, thereby generating a degree of excess traffic between nodes and thus requiring extra processor power to handle. This of course is a reflection of the fact that the original components were not designed from the outset to work together and illustrates the difficulty of deeply integrating disparate projects.

Nevertheless, cooperative interaction between software components that implement complex algorithms was achieved, resulting in an interesting demonstrator that illustrates the combined benefits of the contributing technologies.

## References


[1] Reeve A, Wilkinson T, *Resilient Networks*, SEAS DTC report CC001/01, September 2005

[2] Francis G, McCabe A, Smith P, *A Prototype Hardware Implementation of Dynamic Communications Management for Autonomous Vehicles*, 3rd SEAS DTC Technical Conference – Edinburgh 2008

[3] Long D W, *Resilient Distributed Control in Hostile Communications Environments*, SEAS DTC report CC004/03, March 2008

[4] Long, D W, *Communication of Capability Awareness Knowledge in Multi-Agent Systems,* 3rd SEAS DTC Technical Conference – Edinburgh 2008

[5] Dunlop A, Nabarro T, Smith P, *Integrated communications demonstrator,* SEAS DTC report CC022/02, January 2010


## Acknowledgements


The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.