

Adaptive and resilient systems for emergency response

R. Robinson

*Queensland Research Laboratory
National ICT Australia
Email: ricky.robinson@nicta.com.au*

J. Indulska

*School of Information Technology and Electrical Engineering
The University of Queensland
and
Queensland Research Laboratory
National ICT Australia
Email: jaga@itee.uq.edu.au*

ABSTRACT: Events of recent years have brought to light the difficulties involved in responding to major emergencies, particularly with respect to the on-site coordination of the many responding agencies. This paper presents an overview of the work being conducted by a team within National ICT Australia (NICTA) on rapidly deployable, resilient and adaptive networks and applications for emergency response. Our work is focused on creating reconfigurable, self-healing hybrid wireless mesh networks and enabling context-aware applications for emergency response teams that can be deployed over such networks.

BIOGRAPHY: Dr Ricky Robinson is a researcher in the SAFE Networks work package of the Smart Applications for Emergencies (SAFE) Project at NICTA's Queensland Research Laboratory. He is conducting research into context-aware systems, and self-configuring and self-healing networks, with an aim to applying the research outcomes to the domain of emergency response.

Assoc. Prof. Indulska leads the SAFE Networks work package. Prof Indulska's research interests in the last 12 years are interoperability of distributed systems and issues related to design and development of context-aware/pervasive/ubiquitous systems with the emphasis on seamless computing and intelligent support of user tasks.

1 Introduction

Emergency response is difficult. Sometimes it is also extremely dangerous (Jiang et al. 2004). The difficulties and dangers in emergency response arise not only because of hostile environmental conditions, but also due to management, coordination and logistical problems. Therefore any technology that can provide warnings about dangers in the immediate vicinity of emergency response personnel or alleviate the problems related to the management and coordination of emergency response teams is a welcome addition to the emergency response toolkit, with the following caveat: the technology must not introduce more dangers and difficulties than it removes.

With these things in mind, it is clear that there is an opportunity to develop systems that can support emergency response teams on the ground and the "operations" teams that help to coordinate them. Our research is geared towards providing the flexible, reconfigurable, self-healing technology solutions that provide the foundations for the kinds of adaptive, autonomic

applications required by emergency response personnel. This research is divided into two parts: reconfigurable, autonomic networks and resilient context management systems. Context management systems gather, store, reason over and distribute facts about the environment that allow applications to adapt themselves to the current situation. Autonomic networks can reconfigure themselves in the event of link and node failures. Together, these components enable communication and networked applications to operate with a higher degree of robustness and autonomy in often hostile environments.

The remainder of this paper gives a very high-level overview of our research in this area and is organised as follows. Section 2 discusses the overall architecture of a self-healing, self-configuring, adaptive infrastructure for supporting emergency response applications. Sections 3 and 4 discuss the two major components of the architecture: autonomic networks and context management. Section 5 concludes the paper and suggests some areas of future research.

2 System Overview

Our architecture can be seen as a two-layer platform; one layer deals with network issues and the other layer provides support to adaptive applications. Although each layer handles different problems, they are both concerned with reducing the need for human intervention within the system as a whole.

The bottom layer, the autonomic network, is concerned with packet delivery and communication. In emergency response scenarios, it is important to be able to deploy networks quickly. It is therefore advantageous to make use of existing network infrastructure, some of which may have been damaged during the emergency incident, and to link the existing infrastructure to any ad hoc network components that might be deployed as part of the emergency response for communication and to enable networked applications such as mobile video feeds and *danger notification* systems. However, networks that can be quickly deployed are often characterised by other challenges as a result of the environments within which they are typically deployed and technological limitations. These challenges are usually related to the mobility of the nodes in the network and interference from the environment. These challenges manifest themselves as node disconnections. Autonomic networks seek to provide a greater level of robustness and resilience to disconnections by utilising specialised routing protocols that can quickly find alternative routes through a network of mobile and stationary nodes.

The top layer, the context manager, is a distributed middleware that provides applications that run on top of the autonomic network with information about the environment that can allow them to adapt to the current situation. The context manager gathers data from sensors within the autonomic network and makes it available to applications so that they can adapt themselves without user intervention and provide relevant information to users. For example, a context-aware application can use information about a user's location to tailor the information it shows to a user *and* to adjust its own behaviour in terms of the manner in which it chooses to provide that information to the user.

This two-layer architecture is described in more detail in the following two sections.

3 Autonomic Networks

Autonomic networks are those networks that can automatically reconfigure themselves in the event of disconnections and failed nodes. By this definition, most present-day networks could be classed as autonomic, since most routing protocols are designed to discover alternative routes through a network. However, autonomic networks can adapt themselves for a range of other reasons including:

- socio-economic factors, such as network cost and political concerns;

- the users' working environment, which may limit or extend the available routing options due to device characteristics; and
- application requirements, which may include specific Quality of Service needs.

Furthermore, autonomic networks are expected to operate with little or no input from network administrators. Their properties can thus be summarised as follows:

- self-managing;
- self-optimising;
- self-monitoring;
- self-repairing; and
- self-protecting.

Our goal is to endow these properties to *hybrid wireless mesh networks*. These networks contain stationary nodes (also known as mesh routers) that form a wireless mesh backbone, as well as mobile nodes and mobile network segments that attach themselves to the backbone in an ad hoc fashion. The approach we are taking is to modify the existing Ad Hoc On-Demand Distance Vector (AODV) routing protocol (Perkins & Royer 1999), which was originally designed for ad hoc mobile networks. Our initial focus is on the self-repairing characteristic of autonomic networks. This characteristic tries to ensure that if a network failure occurs, an alternative route can be found quickly to minimise disruption to applications that execute within the network.

AODV discovers routes between source and destination nodes as they are needed rather than in an upfront manner. Once a route has been established, it is maintained as long as traffic is propagated along that route, otherwise the routing entries at each intermediate node will expire. AODV uses Route Request (RREQ) and Route Reply (RREP) packets to establish routes between nodes. RREQ packets are broadcast by source nodes wishing to establish a route to another node in the network. The destination node and nodes that have a route to the destination will respond with an RREP packet, which is propagated back to the source via intermediate nodes. The intermediate nodes create a routing entry in their routing tables that records the next hop node towards the destination. In this way, the entire route is established between a source and destination node. If a network link along the route fails, then the node closest to the failure on the source's side of the failure propagates a Route Error (RERR) message back to the source. If the route is still required, the source can repeat the above process to discover another route to the destination.

AODV assumes that all nodes are equal. In a hybrid wireless mesh network, this is not the case. Some nodes, usually the stationary nodes, have a higher capacity than others. In addition, link failures are less likely between two stationary nodes. Therefore, it is advantageous to bias the routing protocol to favour those stationary nodes that are part of the mesh backbone (i.e., the mesh routers). Our research group has modified AODV to use the heuristic that backbone nodes should be favoured in the routing process (Pirzada et al. 2006). This is accomplished by augmenting the AODV RREQ packet with a field that indicates the number of mesh routers that have been traversed on the way to the destination. The destination will then propagate a RREP packet along the route with the highest ratio of mesh routers to non-mesh router nodes. Our experimental results show that our modifications to AODV result in an improvement to the packet delivery rate of up to 15% compared with unmodified AODV (see Figure 1).

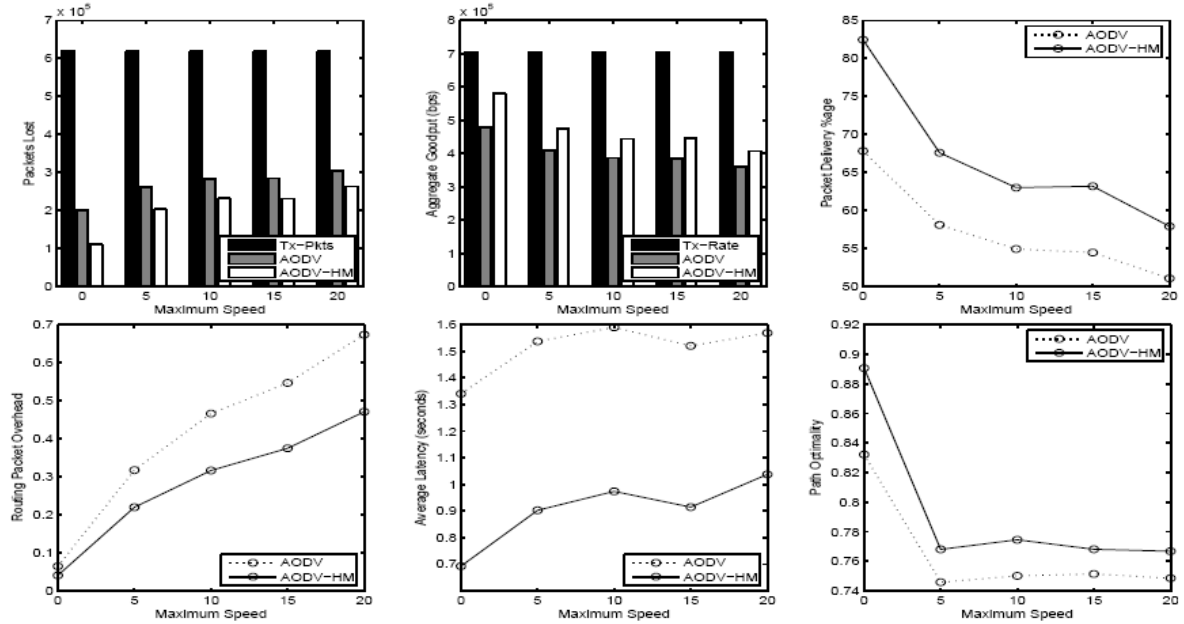


Figure 1: Preliminary experimental results showing standard AODV compared to our modified version for hybrid mesh networks (AODV-HM).

4 Context Modelling and Management

While an adaptive network layer improves resilience to link and node failures by quickly discovering alternative routes through the network, the context manager facilitates application adaptation.

Modern context management systems gather, store and disseminate context information that conforms to a particular *context model*. These models can be expressed in a range of different languages, which each have their advantages and disadvantages. We have chosen to use the Context Modelling Language (CML) (Henricksen & Indulska 2004), which is derived from the Object-Role Modelling language (ORM) (Halpin 2001). CML describes the objects and relationships between those objects that are of interest to a context-aware application. For example, CML models often describe the relationship between people and their current location. Depending upon the application, the model might also contain relationships between a location and its temperature, users and their devices, devices and the communication modes that they offer and so on. Each of these relationships is known as a *fact type*. Particular instances of objects compose *fact instances*, which conform to a fact type. For example, the person *Bob* may be located at *300 Adelaide Street* (conforming to the *Person has Location* fact type), and the outside temperature at that location might be *23 degrees Celsius* (conforming to the *Location has Temperature* fact type). The model provides a formal mechanism for defining within the context manager the kinds of context information that are relevant to the context-aware applications that retrieve context facts from it. In addition, CML enables each fact type to be classified as either *static*, *profiled* (user-defined), *sensed* or *derived*, which enables applications to make more informed decisions about the data it is using to adapt itself. Furthermore, fact types can be annotated with *quality metrics*. These quality metrics are used to indicate various dimensions of data quality. Examples include accuracy, freshness, confidence, resolution and credibility.

Different context-aware applications will have different tolerance levels for the various quality dimensions. A fire-fighter might only need to know the temperature of a room he/she is about to enter to the nearest 5 degrees, but an application that monitors a nuclear reactor might require a more accurate reading of the reactor core and is therefore less tolerant of errors.

We are developing a logically centralised, physically distributed context manager – the NICTA Context Engine (NICE) – that provides applications with the ability to retrieve context facts by querying, and to subscribe for notifications when context facts change or when a context fact contains particular values (see Figure 2). NICE stores context information that conforms to the set of CML models that it is given by a user or administrator. CML is a graphical language, so we have developed an XML-based serialisation of CML, called XCML, that can be used by NICE at runtime to represent CML models as well as fact instances (Robinson et al. 2006).

Our design focus is on robustness and resilience of the context manager. This robustness and resilience is achieved in two ways. First, we are developing reconfigurable context sources. Context sources are those entities that provide raw facts about the environment (Indulska et al. 2006). Usually, these will be sensors such as thermometers, cameras, location badges and so on. These sensors are deployed within the mesh network, and they register themselves with the context manager. Often, the sensors are battery powered, so failure due to depletion of the battery is a common occurrence. To improve resilience to this kind of failure, we are investigating ways of dynamically binding sensors that provide the required types of raw data to the context manager so that if one sensor fails, an alternative source of context information can be found. Second, we are developing “smart” caches at the application (client) side to improve robustness in cases where the client device is disconnected from the network. As mentioned above, CML enables context facts to be classified as static, profiled, sensed or derived and to be annotated with quality metrics. Caching and purging algorithms can use this metadata to make more informed decisions about what to cache and when to purge. The freshness metric, for example, could be used to purge stale facts from the cache, while the resolution metric, which could measure least noticeable difference or update frequency, might be used to decide whether or not to proactively cache a particular fact.

The ability to reconfigure context sources and to cache context information in clever ways is intended to minimise the disruption caused to context-aware applications when failures occur. This is crucial to the kinds of applications that are increasingly deployed by emergency response teams. A context-aware communication application, for example, can choose an appropriate method of communication (e.g., audio via VoIP, text via e-mail or SMS) depending upon the present situation of the intended recipient of that communication (e.g., devices available, noise in the environment).

We envisage a “top-to-bottom” context-aware application that can be used by all emergency response personnel, which adapts to the role of the person using the application and the device on which the application is being used. Emergency rescue personnel on the ground would use the application to provide them with warnings of imminent danger and fine-grained situational awareness. When a rescuer locates a survivor underneath rubble, with a single click of a button on a GPS-enabled PDA or wearable computer, the rescuer can indicate the discovery of a survivor and his/her position. This information could be stored in the context manager and disseminated to other nearby rescuers who are then alerted about the discovery of the survivor. On the other hand, a rescue coordinator would see a different view of the context-aware application on their laptop. The coordinator needs higher level information such as a map indicating the whereabouts of all located survivors and the status of their rescue (located, under way, and completed). A Police Commissioner or a member of government might only require information about the number of casualties and survivors to report to the general public via the media.

The context manager therefore plays several roles: it enables situational information to flow quickly to where it is required, and it provides applications with a basis for adaptation so that users are not required to provide direct input to the application on a regular basis.

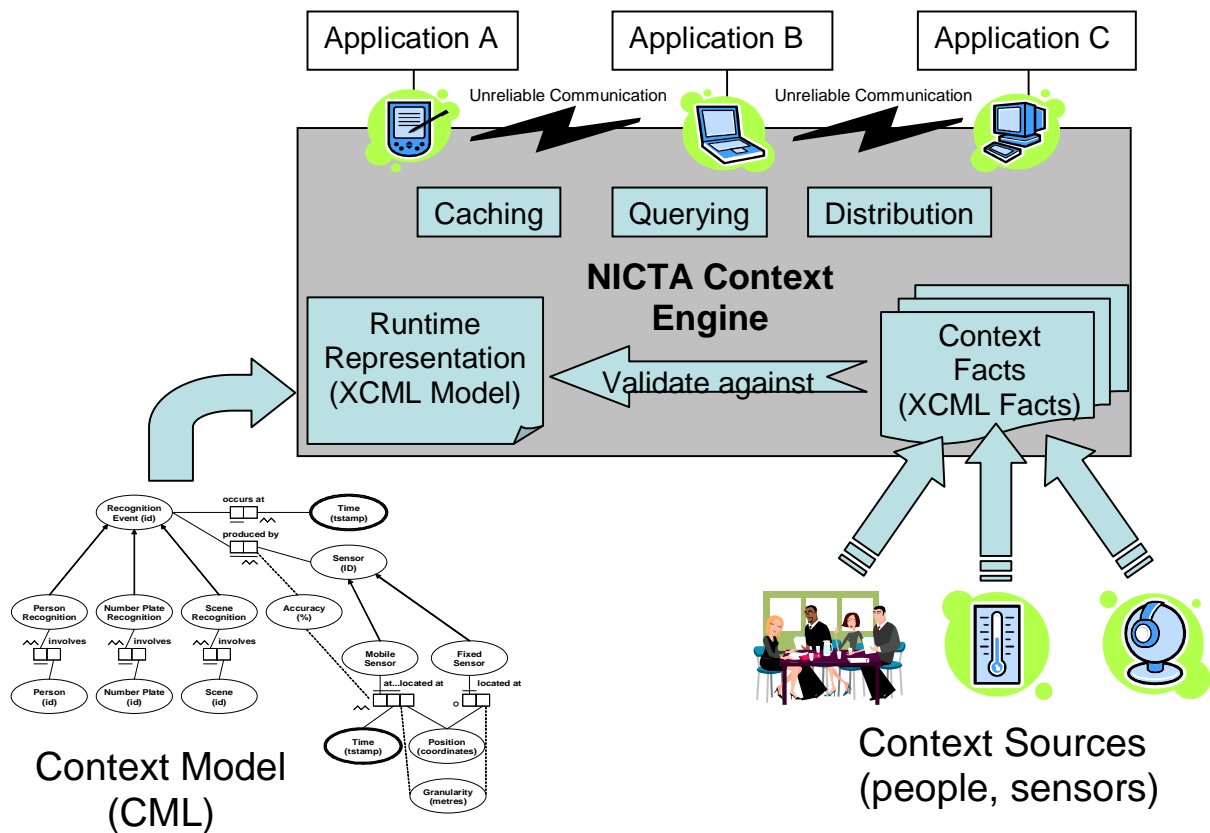


Figure 2: The NICTA Context Engine (NICE) is logically centralised but physically distributed.

5 Conclusions

This paper provided a high-level overview of the work being carried out by National ICT Australia in the area of autonomic networks and resilient context management. Our work in this area is being applied to the domain of emergency and disaster response. The adaptive and resilient systems that we are working towards will augment the existing applications and communication media available to emergency response workers. One of the goals of these systems is to reduce the dangers faced by emergency workers, and to facilitate faster and more efficient coordination among the various groups of people involved in emergency response by providing the right information to the right people at the right time.

Acknowledgements

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology, and the Arts; the Australian Research Council through Backing Australia's Ability and the ICT Research Centre of Excellence programs; and the Queensland Government.

References

- Halpin, T. A. 2001. *Information Modeling and Relational Databases: From Conceptual Analysis to Conceptual Design*. San Francisco, USA. Morgan Kaufman.
- Henricksen, K. & Indulska, J. 2004. A Software Engineering Framework for Context-Aware Pervasive Computing. In *Proceedings of the Second IEEE Conference on Pervasive Computing and Communications (PerCom 2004)*. Orlando, USA. March 2004. IEEE Computer Society.
- Indulska, J., Henricksen, K. & Hu, P. 2006. Towards a Standards-Based Autonomic Context Management System. In *Proceedings of the 3rd International Conference on Autonomic and Trusted Computing (ATC-06)*. To Appear. Wuhan and Three Gorges, China. 3-6 September 2006. Springer Verlag.
- Jiang, X., Chen, N. Y., Hong, J. I., Wang, K., Takayama, L. A. & Landay, J. A. 2004. Siren: Context-aware Computing for Firefighting. In *Proceedings of Second International Conference on Pervasive Computing (Pervasive 2004)*. Vienna, Austria, 18-23 April 2004. Springer Verlag.
- Perkins, C. E. & Royer, E. M. 1999. Ad Hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, USA, February 1999*. IEEE Computer Society.
- Pirzada, A. A., Portmann, M. & Indulska, J. 2006. *Hybrid Mesh Ad Hoc On-Demand Distance Vector Routing Protocol*. Technical Report. National ICT Australia.
- Robinson, R., Henricksen, K. & Indulska, J. 2006. *An XML Representation of ORM-Based Context Models*. Technical Report. National ICT Australia.