Soldiers, Agents and Wireless Networks: A Report on a Military Application[1]

Technical Contact: Bob Gray

Dartmouth College
Hanover, NH 03755
robert.s.gray@dartmouth.edu

Abstract

Ideally soldiers in the field would have portable computing devices, which, connected with a wireless network, would provide access to military databases, terrain maps, and other soldiers. In this report and the associated invited talk, the author presents some routing and mobile-code technologies that have been developed to support soldiers in the field, as well as the highly intelligent network-sensing and planning agents that will be needed to fully solve the networking problems.

1. Introduction

In current military operations, soldiers typically have voice communication only, which makes it difficult to access needed information and coordinate mission activities. Ideally each soldier would have a portable computing device through which they could query military databases, access maps of the surrounding terrain, view the positions of their fellow soldiers, and send complex observations to the mission planners at headquarters. Providing such computing capabilities to soldiers in the field involves many technical challenges at both the hardware and software levels.

The Active Communications (ActComm) project,[2] which is a Multi-University Research Initiative (MURI) funded under AFOSR Contract F49620-97-1-03821, focuses on two pieces of the software[3] level: (1) wireless-routing systems that route traffic from one soldier to another (and back to headquarters), and (2) mobile-code systems that allow the soldiers to efficiently access databases in the main military network. The underlying assumption in the ActComm work, consistent with military needs, is that all soldiers have short-range, high-bandwidth wireless hardware to communicate with each other, while a few soldiers also have long-range, low-bandwidth hardware to serve as gateways to the main network. Due to the short range of the soldier-to-soldier hardware, data going from one soldier to another might need to be routed through several intermediate soldiers. Moreover, the soldiers are continually moving relative to each other, so the available routes change from one moment to the next. The soldiers might even move out of range of each other, requiring the routing system to queue messages until the network disconnection goes away.

---

[1] This report appeared at PAAM 2000 in Manchester, England.
[2] http://actcomm.dartmouth.edu/
[3] The hardware level is equally interesting. The devices must withstand the dirt, noise and vibration of a military environment, must use *jamming-resistant* wireless technology, must not require the soldiers to shift their attention away from the battlefield, and must become inoperative if separated from the soldiers.

Under this assumption, the ActComm participants – Dartmouth College, Harvard University, Rensselaer Polytechnic Institute, the University of Illinois, Lockheed Martin, and ALPHATECH – have developed a range of routing and mobile-code technologies. These technologies have been integrated into a single testbed. In this short paper, the author describes the testbed, its component technologies, one of the military scenarios to which the testbed has been applied, the simple agents that make the current testbed work, and how more intelligent agents will play a key role in future versions. In the invited talk, the author also covers performance results from recent test runs. Together the paper and talk present one of the most challenging network environments in which agents need to operate, an environment that arises not only in military operations, but also in civil search-and-rescue, disaster relief, and refugee management.

2. Scenario and Testbed

The current test scenario for the ActComm project is shown in Figure 1. An intelligence team at headquarters has intercepted one or more phone calls and determined that a terrorist faction is likely to meet in a particular building. Headquarters dispatches a team of soldiers, who take up observation posts (probably hidden) around the building. If the terrorists come to the building, the soldiers will secure the building and arrest the terrorists.

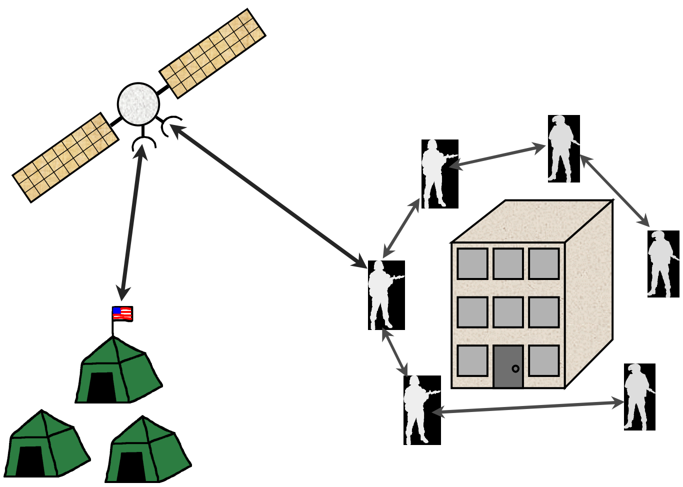

Figure 1: The counter-terrorism scenario.

This scenario has been run several times on the Dartmouth campus with students playing the role of soldiers. Since the ActComm project does not consider hardware issues, each "soldier" has a standard laptop computer. Each laptop has a GPS unit to determine the soldier's position and a wireless Ethernet card to communicate with other soldiers.[4] Each gateway soldier simply has a second wireless Ethernet card set to a different transmission frequency (as opposed to the satellite connection that might be available in real life).

[4] Each laptop is a 200 MHz Pentium II running Linux, the GPS units are Motorola Oncore units, and the wireless Ethernet cards are Lucent 802.11 Turbo Bronze cards with a maximum throughput of 2 Mb/s and an outdoor range of around 400 meters.

Each soldier has a map of the urban area where the building is located. The current positions of the other soldiers are displayed on the map. The soldiers enter descriptions of the people that they observe entering the building. Each observation is sent to headquarters so that the mission planners can determine whether the person is one of the suspects. As the planners make their determination, they might send a series of pictures to the soldiers, and ask them to identify whether the person is in one of the pictures. The soldiers also can query military databases that live in the main network. The testbed has three available databases: (1) news articles arriving on a military news feed, (2) transcripts of intercepted phone calls, and (3) descriptions of people relevant to the mission at hand and the operational area. The database containing the descriptions is called a black-gray-white (BGW) database, since each person is marked as bad, neutral or good. The soldiers in the field primarily search the BGW database, while the team at headquarters searches all three databases.

3. Technology Components

The ActComm project is concerned with two challenges arising from this scenario, namely, maintaining network connectivity and minimizing the use of the network links that connect the soldiers to the main network. The first challenge is addressed with ad-hoc routing and active messaging, while the second is addressed with mobile code or mobile agents.

3.1. Ad-Hoc Routing

The APRL[5] algorithm (Karp, 1998) from Harvard University provides the basic routing functionality in the testbed. Each laptop continually broadcasts "ping" packets. A laptop that receives a "ping" packet knows that it is in transmission range of the sender and marks the sender as an immediate neighbor. The ping packet includes the complete routing table of the sender, so the receiving laptop also adds routes for those laptops that are reachable via the sending laptop. As the laptops move relative to each other, they receive ping packets from different sets of neighbor laptops and update their routing tables accordingly.

The APRL algorithm has the benefit of simplicity, but it generates more control traffic than desired in some situations. The ActComm project is exploring several other routing algorithms. Under the GPSR algorithm, the sending laptop looks up the physical location of the target laptop, and then sends the message to the neighboring laptop that is physically closest to the target's location (Karp, 2000). Due to variations in laptop distribution, there might be times when there is no neighbor closer to the target. To handle this, GPSR sends out probe packets to map the boundaries of regions that contain no laptops. Packets are routed around the edges of these "void" regions when necessary. GPSR generates less control traffic than APRL, but does require a robust lookup service. Other routing algorithms under consideration include STARA (Gupta, 1997), which takes the delay along each network path into account, and several mobile-code approaches, such as (Minar, 1999), in which routing agents move from machine to machine, updating the routing tables as they move.

3.2. Active Messaging

[5] APRL stands for Any Path Routing without Loops; GPSR, Greedy Perimeter Stateless Routing, and STARA, System and Traffic Dependent Adaptive Routing Algorithm.

The routing algorithms above identify a route between two computers, but of course, only if a route *physically* exists. In a dynamic wireless network, there might never be a moment at which there exists a complete route from the source to the target computer. Instead the message must be sent partway through the network, and then sent the rest of the way once the changing connectivity opens up a path to the target. In other words, there must be another layer on top of the basic routing system to handle network disconnections. The goal of this layer is to queue messages as "close" as possible to the critical disconnection, so that the data can be forwarded as soon as the disconnection goes away.

There are several approaches to constructing this layer, but one of the most promising is active messaging (Okino, 1999). In active messaging, each message specifies its own routing strategy, possibly in the form of procedural code that is attached to the message. The advantage of active messaging is that each application can apply an application-specific routing policy to its own messages. For example, an application might use a routing strategy that replicates high-priority messages at key network points and then sends the copies along different network paths. In the current testbed, the routing strategy is simple. Each message is sent to whichever reachable machine most recently had a connection to the ultimate target.[6] The message waits on that machine until the target is once again reachable. This simple strategy is sufficient but not as efficient as desired. In particular, since all the machines are moving, the target machine might re-connect to the network far away from the machine on which the message is waiting. The message then must be transmitted across several extra hops. One of the key goals during the remainder of the project is to identify the most appropriate routing and active-messaging strategies.

3.3. Mobile Code / Mobile Agents

A mobile agent is simply the most general form of mobile code, namely, an executing program that can move at times of its own choosing from one machine to another. A mobile agent often, but not always, displays some of the other characteristics associated with agents, such as autonomy and adaptivity. Mobile agents are used to move computation to more attractive network locations, often to avoid the use of unreliable or low-bandwidth network links. A more extensive discussion of mobile agents and the rationale behind their use can be found in (Gray, 2000). In the testbed, mobile agents are used for three purposes. First, the active-messaging system is implemented on top of the mobile-agent system. Each message is wrapped inside a mobile agent, which carries the message through the network. Although a production system would be implemented much closer to the network layer, implementing the system on top of the mobile-agent system allowed extremely rapid development.

Second, mobile agents that move from the soldiers' machines into the main network perform all multi-step queries.[7] The agents interact with the needed databases without using the unreliable, low-bandwidth link that connects the soldiers to the main network. The agent

---

[6] The current testbed uses TCP/IP as the lowest-level transport mechanism. TCP/IP performs poorly, however, when the connection spans multiple wireless links. Thus, the messages are actually transmitted hop by hop. The entire message is sent to an immediate network neighbor, from there to the next machine in the route, and so on. Later versions of the testbed will replace TCP/IP with one of the many proposed wireless transport protocols.

[7] Mobile agents also perform the queries entered by the mission planners at headquarters, since headquarters itself often has an unreliable connection to the main network.

completes the query faster, and does not waste bandwidth by sending intermediate results back to the soldiers. Of course, with sufficient *a priori* knowledge of a multi-step query, database or proxy developers could implement a single high-level operation that performs the query. These developers, however, can never have a priori knowledge of all queries. Mobile agents allow the queries to be performed efficiently even when developers have not provided query-specific support.

Finally, after a soldier sends an observation to headquarters, headquarters might send back a set of pictures. The soldier confirms whether the person she saw is in one of the pictures. In the testbed, headquarters sends not only the pictures, but also the *code* that displays the pictures and allows the soldier to browse them. The code and pictures are sent as a single mobile agent. As before, the picture-browsing code could be installed on the soldier's machine before the mission begins. The mobile agent, however, eliminates the need for the pre-installation step, something that is important if the mission is planned rapidly, and the soldier has never been involved in a "picture" mission before.

4. More Intelligent Agents

The agents in the current testbed are simple information and interface agents. Highly intelligent agents, however, are needed in the network layer itself to fully solve the routing and disconnection problems. In particular, if the network layer could detect an *impending* network disconnection, applications would have a brief time window in which to prepare for the disconnection. For example, a replicated server might pro-actively start a new replica (rather than waiting until a failure occurs *after* the disconnection). Major Lisa Shay at RPI, building on the work of (Thottan, 1998), is developing agents that learn the characteristics of the wireless network, and then detect impending disconnections by monitoring signal strengths, traffic levels and position (GPS) changes.

Also important are planning agents that decide how to allocate scarce network resources to competing applications. (Bredin, 2000) develops market-based algorithms for system-level allocation of bandwidth and other resources. At a higher level, (Mozumi, 1998) presents several algorithms that allow a mobile agent to plan the best route through the network (i.e., which copy of a replicated database should it use, which database should it visit first if the task is not strictly sequential, and so on). Although these network-sensing and planning agents are not used in the current testbed, they will play a key role in future versions.

5. Conclusion

The ActComm project aims to provide computer access to soldier in the field. Its initial focus has been the routing algorithms needed to maintain network connectivity and the mobile agents that move query processing into the main network (or interface code onto the soldiers' machines). Its next focus will be the network-sensing and planning agents. When all the components are eventually integrated, they will form an effective infrastructure for supporting applications in dynamic wireless networks. With the current proliferation of such networks, the components and infrastructure will be applicable far beyond the military domain.

6. Acknowledgements

The ActComm project involves many people. Sue McGrath at Lockheed Martin, Eileen Entin at ALPHATECH, and Major Lisa Shay at RPI were instrumental in developing the counter-terrorism scenario. Brad Karp developed both the APRL and GPSR routing algorithms. The other lead researchers are George Cybenko, David Kotz and Daniela Rus at Dartmouth College, H. T. Kung at Harvard Univesity, Ken Vastola at RPI, P. R. Kumar, Tamer Basar and Gul Agha at the University of Illinois, and Ken Whitebread at Lockheed Martin. Arne Grimstrup and Ron Peterson, two staff members at Dartmouth, provided significant programming support. In addition, an army of students at the academic institutions helped to integrate the components into the testbed. Finally, many thanks to the Department of Defense and AFOSR, who fund the project under AFOSR contract F49620-97-1-03821.

References

(Bredin, 2000) Jonathan Bredin, Rajiv T. Maheswaran, Cagri Imer, Tamer Basar, David Kotz and Daniela Rus. “A Game-Theoretic Formulation of Multi-Agent Resource Allocation.” In *Proceedings of the Fourth International Conference on Autonomous Agents*, Barcelona, May, 2000. To appear.

(Gray, 2000) Robert S. Gray, George Cybenko, David Kotz and Daniela Rus. “Mobile Agents: Motivations and State-of-the-Art Systems.” In Jeffrey M. Bradshaw, editor, *Handbook of Agent Technology*, AAAI/MIT Press, 2000. In press.

(Gupta, 1997) Piyush Gupta and P. R. Kumar. ``A system and traffic dependent adaptive routing algorithm for ad hoc networks." In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 2375-2380, San Diego, December, 1997.

(Karp, 1998) Brad Karp and H. T. Kung. “Dynamic Neighbor Discovery and Loop-Free, Multi-Hop Routing for Wireless, Mobile Networks.” Harvard University, May, 1998. Draft. Available at http://www.eecs.harvard.edu/~karp/aprl.ps

(Karp, 2000) Brad Karp and H. T. Kung. “Greedy Perimeter Stateless Routing.” Harvard University, February 2000. Submitted to Mobicom 2000.

(Minar, 1999) Kwindla Hultman Kramer, Nelson Minar and Pattie Maes. “Tutorial: Mobile Software Agents for Dynamic Routing.” In Mobile Computing and Communications Review, 3(2), 1999, pages 12-16.

(Moizumi, 1998) Katsuhiro Moizumi. *The mobile-agent planning problem*. Ph.D. Thesis, Thayer School of Engineering, Dartmouth College, 1998.

(Okino, 1999) Clayton Okino and George Cybenko. “Modeling and analysis of active messages in volatile networks,” In *Proceedings of the 37th Annual Allerton Conference on Communications, Control and Computing*, Allerton, Illinois, September 22-24, 1999. Available at http://agent.cs.dartmouth.edu/papers/okino:active.ps.Z

(Thottan, 1998) Marina Thottan and Chuanyi Ji. "Proactive Anomaly Detection Using Distributed Intelligent Agents." In *IEEE Network*, Sep/Oct 1998, pages 21-27.