An Ad-hoc Network for Teams of Autonomous Vehicles[1]

Bhargav R. Bellur, Mark G. Lewis and Fred L. Templin
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025.
{bhargav,lewis,templin}@erg.sri.com

Abstract—This paper presents the study of a mobile ad-hoc network for teams of autonomous vehicles. We discuss the special challenges presented by the autonomous team model, and we present the self-sustaining, self-configuring dynamic network architecture we have developed to address them. We further discuss actual field experiments in which elements of the architecture have been proven through realistic test scenarios using surrogate unmanned aerial and ground vehicles. We conclude the paper by presenting lessons learned through the field experiments, performance analysis results and plans for future work.

1. INTRODUCTION

Autonomous teams of unmanned aerial vehicles (UAVs) present a challenging scenario for tactical information operations. Since these teams must operate in remote regions with little/no infrastructure, we assume that networks are formed in an **ad-hoc** fashion and that information exchanges occur only via the wireless networking equipment carried aloft by the individual UAVs. While certain autonomous team configurations (such as close formation flying, shown in figure 1) result in relatively stable topologies, UAVs are fast moving, agile and in constant motion. As such, rapid fluctuations in the network topology may occur when individual vehicles suddenly veer away from one another or when wireless transmissions are blocked by terrain features, atmospheric conditions, signal jamming, etc.

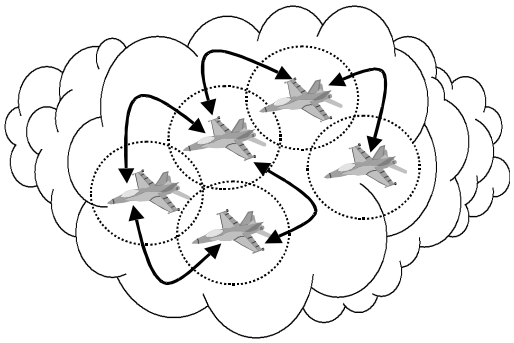

Figure 1. An autonomous team of UAVs in formation flight.

In spite of such dynamically changing conditions, vehicles in an autonomous team must maintain close communications with one another in order to avoid mid-air collisions and facilitate collaborative team mission execution. Additionally, they also must remain in communication with other forward deployments as well as remote command posts for command-and-control and situation awareness. We therefore anticipate a requirement for self-configuring, self-sustaining dynamic networks coupled with a location-independent flexible addressing architecture for effective information operations in forward power projections.

Since May 2000, we have conducted experiments through a program sponsored by the United States Office of Naval Research (ONR). This has led to the development and evaluation of a self-configuring, self- sustaining dynamic ad-hoc network architecture for information operations in autonomous teams. While typical research test-beds include simulation environments and laboratory configurations, our work has studied practical information operations in actual fielded deployments using remote-controlled aerial and ground vehicles as surrogate UAVs.

2. NETWORK ARCHITECTURE

The network architecture for information operations in UAV autonomous teams must support communications at the **intra-team**, **inter-team** and **global internetworking** levels, with individual UAVs acting as **network nodes** at all levels.

Intra-team Communications

UAV teams are highly collaborative in nature with a requirement for time-critical communications. Recall that UAVs in an autonomous team communicate amongst themselves via the wireless networking equipment carried aloft the individual vehicles. However, the transmission range of each UAV is limited in order to preserve its battery power. Hence, an autonomous team of UAVs is organized into a **Mobile Ad-hoc Network (MANET)**, wherein messages between UAVs may be forwarded via other members of the autonomous team. Since communications bandwidth is a scarce resource in a MANET, it is important that the routing protocol be efficient in terms of overhead.

SRI has developed a protocol called **Topology Broadcast based on Reverse-Path Forwarding (TBRPF)** [1,2] for efficiently disseminating link-state updates. TBRPF

[1] This work was funded by the United States Office of Naval Research (ONR) under Contract Number N00014-00-C-305.

provides a complete topology link-state routing protocol in that each node is provided with the state of each link in the network. TBRPF is extremely agile in that a change in the up/down status of links is quickly detected, and alternate routes are immediately computed. The proof of correctness and pseudo-code for TBRPF as well as examples illustrating its operation can be found in [1,2].

The TBRPF protocol consists of: (I) Neighbor Discovery, and (II) Broadcasting of link-state updates. The purpose of the neighbor discovery protocol is to allow each node in the network to quickly detect the neighboring nodes with which the node has a bi-directional link.

TBRPF achieves its efficiency by sending topology updates along min-hop path spanning trees rooted at the source of the update. TBRPF uses the concept of *reverse-path forwarding* to reliably broadcast each topology update in the reverse direction along the dynamically changing broadcast tree formed by the min-hop paths from all nodes to the source of the update. Since the *leaves* of the broadcast tree rooted at a particular source do not forward updates originating from that source, a dramatic reduction in control traffic is achieved compared to link-state flooding protocols such as **Open Shortest-Path First (OSPF)**.

Inter-Team Communications

In large-scale deployments, multiple autonomous teams may engage in coordinated missions spread across arbitrarily wide geographic regions. We envision that such deployments will entail a hierarchical arrangement with inter-team communications capabilities.

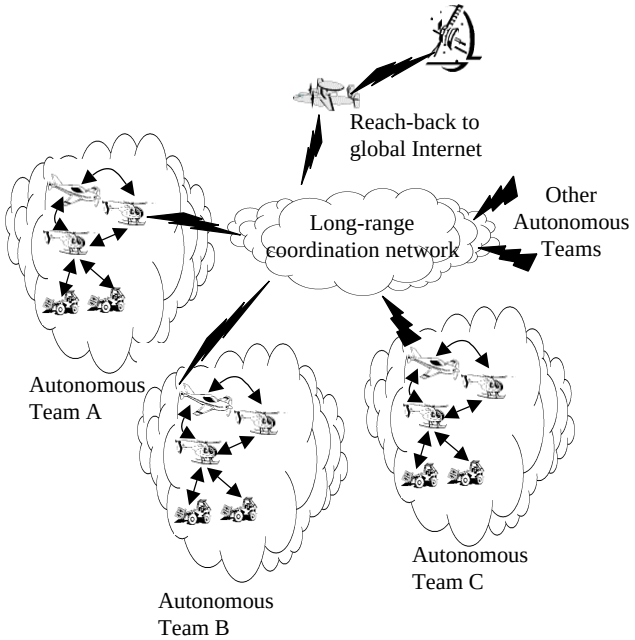

Figure 2. A large-scale deployment of autonomous teams with inter-team communications capabilities.

In our model, at least one UAV in each autonomous team carries wireless communication devices capable of operating on both a low-power, short-range intra-team network and a higher-power, longer-range coordination network. One such suitably equipped UAV is chosen as the **inter-team router** (or **cluster head**) for the autonomous team through a dynamic router election process. (The election process automatically selects a new router in the event of failure.) This router provides a gateway through which other nodes in the autonomous team may access the **long-range coordination network** thus achieving resource sharing and economies of scale through aggregation.

We incorporate a router affiliation which is a core component of the Internet Protocol, Version 6 (IPv6) Stateless Address Auto-configuration mechanism as specified in [4,5]. Periodic **router advertisement** messages serve as *beacons* for UAVs to locate an inter-network router for their autonomous team and provides **stateless address auto-configuration** whereby UAVs automatically form layer-3 network addresses that are both *globally unique* and *topologically correct* for their affiliated router.

Global Internetworking

Autonomous teams of UAVs must perform missions such as surveillance, intelligence gathering, and coordinated tactical strikes, using long-range communications capabilities are required to provide human observers in distant command posts. Our architecture addresses this requirement by organizing the network as a seamless, mobile extension to the global Internet.

Since autonomous teams and individual UAVs may move about rapidly throughout the theater of operation, we require a **flexible addressing scheme** capable of tracking nodes as they move. Again, we use the IPv6 addressing architecture as the basis for flexible addressing. In our model, IPv6 addresses combine distinct **location** and **identity** components and are uniquely assigned to each node in the dynamic network. Nodes are initially assigned a unique **home address** that never changes.

As nodes move throughout the network, they affiliate with new routers (as described in the previous subsection) and adopt **care-of addresses** in which the location component of the address identifies their current autonomous team affiliation while the identity component remains the same.

Finally, our architecture includes a transition mechanism devised by SRI [8] known as the **Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)** that allows seamless interoperation between our IPv6 flexible addressing model and the addressing scheme used in the existing global Internet. This mechanism allows flexible addressing between remote command posts and forward autonomous team.

3. TEST-BED ELEMENTS

In this section, we describe the elements used in our test-bed environment. Since May 2000, we have used this test-bed as a realistic model for experimentation with our network architecture in autonomous UAV teams.

Software Integration Architecture

SRI's TBRPF protocol was originally implemented [3] in the **FreeBSD** operating system (www.freebsd.org) with the Merit Multi-Threaded Routing Toolkit (MRT) daemon (www.mrtd.net). This implementation has been in use for laboratory and fielded experiments since June 1999. As of January 2001, TBRPF has been ported to **Linux** (www.linux.org) and enhanced to include a number of protocol improvements. We have developed and integrated an implementation of the **ISATAP** protocol in our custom Linux kernel and use it extensively to support our flexible addressing scheme.

During test bed experiments and demonstrations, we use topology display to show nodes join and leave and links form and break.

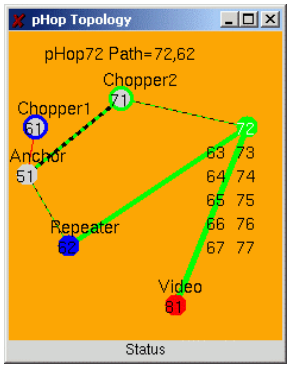

Figure 3. Dynamic topology display screen shows current nodes and links on iPAQ.

Computer/Communications Hardware Elements

We are currently using commercial IEEE 802.11b pccards operating in the 2.4 GHz frequency band, using the Direct Sequence Spread Spectrum (DSSS) modulation, and provides up to 11 Mbps data transfer rates with a maximum range of approximately 1000m line-of-sight. We configure the cards to use **Ad-hoc** mode, rather than be dependent on fixed infrastructure elements.

We currently employ a number of portable laptop and Pocket PC computers for our experiments, including the Toshiba Libretto and Compaq iPAQ chosen for their small form factor. The Toshiba Libretto model 110ct is a sub-notebook computer with 2 type II PCMCIA card slots weighs 2.2lbs. The Compaq iPAQ has an optional PCMCIA sleeve used to accept the 802.11b Cards and weighs less than 1 lb.

Figure 4. Toshiba Libretto sub-notebook and Compaq iPAQ Pocket PC with wireless network interface card.

Surrogate UAVs

For outdoor field experiments, we are using several unmanned helicopters and ground robots that run our ad-hoc network software. They are made available through partner researchers at SRI and the University of California, Berkeley. Typical of the helicopters we use are the Yamaha RMAX Aero Robot and Yamaha R-50 (www.yamaha-motor.co.jp/sky-e/index.html). The RMAX Aero Robot has a 122-inch main rotor, empty weight of 128 pounds, a payload capacity of 66 pounds, and a flight duration of 60 minutes.

Figure 5. Yamaha R-50 and RMAX Aero Robot helicopters.

The ground robot platform is an ActivMedia Pioneer intelligent robot (www.activrobots.com) with an embedded computer. It can move at 0.8 meters per second and carry a payload of up to 30 kg.

Figure 6. ActivVision Pioneer All Terrain Robot with Toshiba Libretto node.

4. FLIGHT EXPERIMENTS

Flight experiments have entailed increasingly more complex test cases as we gain experience with new elements of our network architecture and correct flaws discovered through experimentation. In the following section, we summarize a few typical experiments.

Multiple Autonomous Teams

In this experiment, we had two autonomous teams with each team operating on a separate WaveLAN RF channel. Wireless gateway nodes provide inter-autonomous team routing capabilities over a common (third) RF channel.

We configured both the autonomous teams with IPv6 routing between them. This experiment succeeded in forwarding ICMP and TCP messages between the two teams. The two helicopters belonged to different teams, and were operating on different frequencies. We used ground-based dual-interface gateway nodes to provide inter-autonomous team routing capabilities over a common (third) RF channel.

Class Based Queuing

The goal of the flight test was to gain experience with traffic management policies for fair media sharing. These include Class-Based Queuing (CBQ) and the IETF Differentiated Services paradigm to replace simple FIFO queuing.

To this end, we have integrated the ALTQ (Alternative Queuing) implementation into the FreeBSD kernel on our MANET nodes. More information can be obtained at http://www.csl.sony.co.jp/person/kjc/software.html

We enabled the CBQ queuing model for wireless interfaces and defined traffic management policies for several different traffic classes, including:

- TBRPF neighbor discovery protocol messages
- TBRPF protocol messages
- Bulk TCP, bulk UDP, or bulk HTML traffic.

By implementing these fairness policies on all MANET nodes in the experiment, we were able to ensure that none of the traffic classes faced starvation in the face of packet loss and network congestion. This was borne out by an experiment in which a stationary traffic generator node directed bulk TCP, bulk UDP and HTML traffic toward a mobile node. The mobile node encircled the building next to the flight test field while the helicopter and an additional hand-held node were dynamically discovered by TBRPF. All of the traffic streams experienced degradation as packet loss, delay variance, and multi-hop forwarding delays came into play. However, we observed that none of the traffic streams faced starvation.

TCP window sizing for multi-hop networks

We observed that choosing an appropriate TCP window size makes a significant difference for multi-hop TCP performance. The ***netperf*** traffic generator tool allows configuration of the maximum send and receive socket buffer sizes for the experiments, which results in upper bounds being set for the TCP window size. We found that maximum performance was achieved with a much smaller TCP window size than occurs in "standard" Internet TCP sessions, since this eliminates channel access starvation on IEEE 802.11.

Innovative Scheme to Mount the Antenna on the Helicopter

During our experiments, we observed some odd performance variations relating to the helicopter's mobility. When the helicopter was stationary, ping round-trip times (RTTs) were in the neighborhood of 6-8 msec for a pair of nodes using the helicopter as a multi-point relay. But, when the pilot maneuvered the helicopter through a series of flight patterns, the ping RTTs varied wildly; often reaching 1.2 seconds or more. Our hypothesis is that the antenna mounted on the helicopter (which is intended for non-mobile indoor applications) performs very poorly when the helicopter's landing skids, fuselage, rotor, etc. block the signal between the helicopter and the ground station

We lowered the antenna's mounting on the helicopter's undercarriage. This has practically eliminated the delay variance. To verify this, we had a ground node "ping" the helicopter while the pilot flew it through a variety of maneuvers. Ping RTTs were on the order of 3-4msec (with some rare outliers in the neighborhood of 10-15msec) regardless of the orientation of the helicopter with respect to the ground node and regardless of the helicopter's speed.

Multi-hop IP Multicast; Collaborative Communications

Autonomous teams of UAVs require robust multicast services to support collaborative communications; even when multiple hops are necessary to distribute multicast messages to all members of the autonomous team. We developed a simple multicast extension to the TBRPF protocol for this purpose and staged experiments using IP multicast information. In these experiments, we verified the multi-hop multicast capabilities using the MASH **vat** voice-over-IP application.

Researchers carrying ground nodes around the flight test field with the helicopter as the central hub in a star topology. We then verified that the helicopter was correctly providing a multi-hop multicast relay service by issuing repetitive "roll-calls" over the voice-over-IP multicast session. We found that the voice-over-IP transmissions were clear as long as each ground node maintained a solid link to the helicopter regardless of how fast the helicopter flew or the helicopter’s physical orientation with respect to the ground node.

Large-scale Deployment; Multiple Autonomous Teams

In our largest fielded experiment to date, we demonstrated the operation of a multiple-team deployment combining all aspects of our network architecture (see Figure 3). Fourteen nodes organized into two autonomous teams were demonstrated; each autonomous team employed an actual UAV as a team leader that sent periodic router advertisements to provide stateless address auto-configuration for ground nodes. Each ground node was given a priori assignment to a preferred team, and remained affiliated with that team as long as it continued to receive router advertisements from its team leader. An additional node served as an aggregation point to link the entire forward deployment to the global Internet. The autonomous teams were given a mission to locate a robot evader that was concealed on the test field premises.

The experiment showed that our network architecture supported the functionality necessary for multiple autonomous teams to engage in coordinated missions. The TBRPF protocol maintained multi-hop routes through the network as ground nodes moved beyond single-hop range of their team leaders. The router affiliation protocol allowed ground nodes to re-affiliate with a different team leader when router advertisements from their preferred team leaders ceased. Finally, the flexible addressing scheme allowed ground nodes to maintain global interoperability even as they switched affiliation between their preferred team and alternate team based on router affiliation.

5. PERFORMANCE EVALUATION

We now present the results of the experiments that were conducted to measure the throughput under different traffic configurations in our mobile ad-hoc network. The experiments were conducted indoors with fixed nodes. We used the ***netperf*** tool to perform a TCP stream test for the duration of 30 seconds. The stream tests were conducted simultaneously if more than one unidirectional stream of traffic was present within the network.

First, eight mobile hosts (nodes A—G) were placed on a table close to each other such that all of them were within transmission range of one another. We then had one, two, three, and four concurrent unidirectional streams of traffic. The throughput was measured for each traffic stream, and averaged over five runs. From the experimental results, we observed that the wireless transmission medium is shared between the different traffic streams in a nearly fair manner. In particular, no individual traffic stream is starved by the presence of other traffic streams.

Next, we measured the throughput for a multihop string where only adjacent nodes on the line A—B—C—D—E are within communication range of one another.

For this configuration, we had one unidirectional stream of traffic from node A. The measured throughput for this 2-, 3-, and 4- hop traffic stream is shown in Table 1. As indicated in the table, a number of runs were conducted using ***netperf*** for each case. A number of experiments were also conducted to transfer large files (file sizes of 1MB to 10MB) using the ***ftp*** protocol.

For the 2-hop case, an effective throughput of up to 2 Mbps was measured. A large portion of the experimental runs consistently yielded results in the 1.4 to 1.6 Mbps range.

Unidirectional Traffic Stream	Throughput (Mbps)	Number of experimental runs
A → B (1-hop)	5.03 Mbps	10
A → C (2-hops)	Up to 2 Mbps (1.4 – 1.6 Mbps)	50
A → D (3-hops)	0.40 Mbps	10
A → E (4-hops)	0.30 Mbps	10

Table 1 Multi-hop throughput performance for unidirectional TCP streams.

For the multi-hop string configuration, we conducted ***ping*** tests from node A to node B (and later to nodes C, D, and E). In our experiments, the traffic rate was increased by increasing the frequency of ping packets of fixed size (1400 bytes) from 10 packets/sec to 100 packets/sec. In addition, the size of the ping packets was also varied from 1400 bytes to 4200 bytes. In each run of the experiments, about 200-500 ICMP echo request messages were transmitted. The measured performance metrics were the average delay and the packet loss rate. A summary of the experimental results appears in Table 2.

Traffic Stream	Saturation Throughput (Mbps)	Steady-state delay below saturation (ms)
A → B	2.80 Mbps	8 ms
A → C	1.0 Mbps	18 ms
A → D	0.43 Mbps	25 ms
A → E	0.36 Mbps	35 ms

Table 2. Multi-hop delay performance using the ping tool.

6. ARCHITECTURAL IMPROVEMENTS

The initial version of the TBRPF routing algorithm employed a hop-by-hop routing mechanism, where the next hop was computed based on the minimum-hop path to the destination node. However, it was observed during the flight tests that minimum-hop paths are not always be desirable. For example, if the minimum-hop path includes "weak" links, then (see Figure 7) data transmitted along this path may incur a significant amount of packet loss. Moreover, computing minimum-hop paths when a certain

link is marginal (i.e., oscillating between the up and down states) will lead to route oscillation. The subsequent out-of-order arrival of packets of a particular session at the destination node results in inefficiencies in certain higher-layer protocols.

We have, therefore, made enhancements to the routing algorithm to compute minimum-cost paths, where the cost of a link is inversely related to the "quality" of the link. In the current implementation, the device driver of the wireless network interface card is queried upon reception of Hello packets. The device driver responds with the Signal to Noise ratio (SNR) of the received Hello packet. This signal strength metric is maintained by the protocol for each neighbor node. Based on the signal strength metric, the protocol assigns a discrete-valued quality (or cost) to each link. The quality of each link is disseminated throughout the network via TBRPF. Minimum-cost paths are then computed, where the link cost is the maximum of the link cost reported in both the directions.

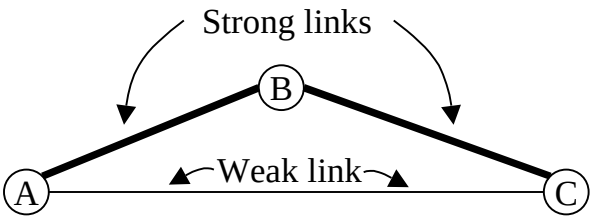

Figure 7. The link between nodes A and C is weak, and can incur a significant amount of packet loss.

7. Conclusions and Future Work

In this document, we present our insights into information operations for autonomous teams of unmanned aerial vehicles based on actual fielded experiments with surrogate UAV nodes. We present elements of our network architecture and describe our experiences learned in the process of actual fielded experiments. In future work, we will examine the experimental results in a more quantitative manner. We will additionally continue to evolve our network architecture as we gain more insights into the needs of autonomous teams.

References

[1] Bellur, B., and R.G. Ogier. 1999. "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *Proc. IEEE INFOCOMM '99*, pp. 178–186.

[2] M.G., Lewis, B. Bellur, R.G. Ogier, and F. L. Templin, "Topology Broadcast based on Reverse Path Forwarding (TBRPF)," draft-ietf-manet-tbrpf-05.txt, March 2002 (work in progress).

[4] Deering, S. and R. Hinden. 1998, "Internet Protocol, Version 6 (IPv6) Specification," RFC-2460.

[5] Thomson, S. and T. Narten. 1998. "IPv6 Stateless Address Autoconfiguration," RFC-2462

[6] Hinden, R. and S. Deering. 1998. "IP Version 6 Addressing Architecture," RFC-2373 (July).

[7] IETF. 1999c. "IP Routing for Wireless/Mobile Hosts (mobileip)," IETF Routing Area Working Group, www.ietf.org/html.charters/mobileip-charter.html

[8] Templin, F. 2001, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", draft-ietf-ngtrans-isatap-01.txt, May 2001 (work-in-progress).

[9] IEEE. 1985. IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE, New York, New York.

Authors

Bhargav Bellur *received his B.Tech degree in electrical engineering from I.I.T. Mumbai in 1988, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin in 1990 and 1995, respectively. Since June 1995, he has been working as a researcher at SRI International. His research interests are in the field of communication networks, distributed algorithms, and performance evaluation.*

Mark G. Lewis *received his B.S. and M.S. in Computing Science from the University of California Davis in 1979. He started programming in Fortran at the age 9, having learn Basic two years earlier. He has worked as a scientific program for UC Davis and the US Army Corps of Engineers. He is currently a research engineer at SRI International where he has been since 1979. There he has been involved the research and development of communication protocols, packet radio wireless networks, and has served as a Program Manager for 5 years.*

Fred L. Templin *received his B.S. degree in computer science from the Pennsylvania State University in 1983 and his M.S. degree in computer science in 1986 - also from Penn State. Following his M.S. degree, he worked for ten years as a network software engineer with Digital Equipment Corporation; two years of which were spent as a visiting researcher in the computer science department at the University of California, Berkeley. Since May 1997, he has worked as a research engineer at SRI International. His research interests include the Internet protocols, mobile wireless networks and network support for real-time multimedia applications.*