Tactical Information Operations for Autonomous Teams of Unmanned Aerial Vehicles (UAVs)[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 a study of tactical information operations in autonomous teams of Unmanned Aerial Vehicles (UAVs). We discuss the special challenges presented by the autonomous UAV team model, and we present the self-sustaining, self-configuring dynamic network architecture we have developed to address these challenges. We further discuss actual fielded 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 fielded experiments, performance analysis results and plans for future work.

Table of Contents

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.

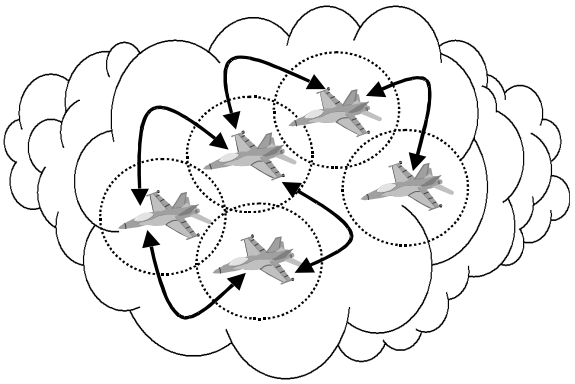

Figure 1: An autonomous team of UAVs in formation flight.

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 behavior. Additionally, regardless of their current tactical situation and location within the theater, autonomous teams must remain in close communication with other forward deployments as well as remote command posts for command-and-control and situation awareness information exchanges. 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 actual fielded 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 network architecture for information operations in autonomous UAV 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. We continue to use the test-bed to gain valuable practical experience with mobile, ad-hoc network routing protocols for intra-team communications and advanced internetworking protocols for flexible addressing and global

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

mobility support as autonomous teams move about within the theater of operation.

This paper describes our test-bed consisting of aerial vehicles and ground robots capable of carrying wireless communications equipment. Section 2 describes elements of the dynamic networking architecture we have developed to support information operations for autonomous UAV teams. The test-bed platforms on which we have deployed our technologies are described in Section 3. Section 4 describes ongoing experiments of the mobile internetworking technologies in actual fielded deployments. Section 5 presents the results of experiments to evaluate the performance of the Mobile Ad-hoc Network. Section 6 concludes with lessons learned from past experiments and plans for future work.

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. At the intra-team level, each node must maintain reliable, time-critical communications with the rest of the team to collaboratively plan and execute tactical objectives as well as ensure safe flight parameters. At the inter-team level, information operations typically involve higher-level mission coordination between team leaders or other specialized nodes within the individual teams. Finally, internetworking solutions are required to provide global command/control and situation awareness access to individual nodes. In the following subsections, we discuss the components of our dynamic mobile network architecture that address information operations requirements at each of these 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 limited 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 consumed in transmitting control traffic. We now describe a pro-active link-state routing protocol that is well suited for this purpose.

SRI has developed a protocol called **Topology Broadcast based on Reverse-Path Forwarding (TBRPF)** [1,2,3] for efficiently disseminating link-state updates in mobile ad-hoc wireless networks. TBRPF is 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 of TBRPF as well as examples illustrating its operation can be found in [1]. Pseudo-code for the TBRPF protocol as well as the formats of the messages used by the protocol appears in [2].

The TBRPF protocol consists of the following two mechanisms: (I) Neighbor Discovery, and (II) Broadcasting of link-state updates. The purpose of the neighbor discovery protocol [2] is to allow each node in the network to quickly detect the neighboring nodes with which the node has a bi-directional link. It also detects when a bi-directional link to some neighbor node no longer exists.

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)**.

The broadcast trees are updated dynamically using the topology information that is received along the trees themselves, thus requiring very little additional overhead for maintaining the trees. Minimum hop-path trees are used because they change less frequently than shortest-path trees based on a metric such as delay.

Based on the information received along the broadcast trees, each node computes its *parent* and *children* for the broadcast tree rooted at each source u. At node i, the parent and children for the broadcast tree rooted at source node u are denoted by p_i(u) and children_i(u). Each node then forwards updates originating from source u to its children on the tree rooted at source u in the following manner (see Figure 2). Any link-state update originating from node u is accepted by node i if (1) it is received from the parent node p_i(u), and (2) it has a larger sequence number than the corresponding link-state entry in the topology table at node i. If accepted, the link-state update is entered into the topology table of node i, and then broadcast to neighbors of node i if children_i(u) is nonempty.

TBRPF uses both positive and negative acknowledgments to reliably transmit control messages to neighboring nodes. When the neighbor discovery protocol detects a new neighbor or the loss of an existing neighbor, link-state updates are triggered. However, the protocol has built-in parameters to prevent the frequent generation and forwarding of link-state updates. Finally, TBRPF uses

infrequent periodic updates to correct rare errors that may occur.

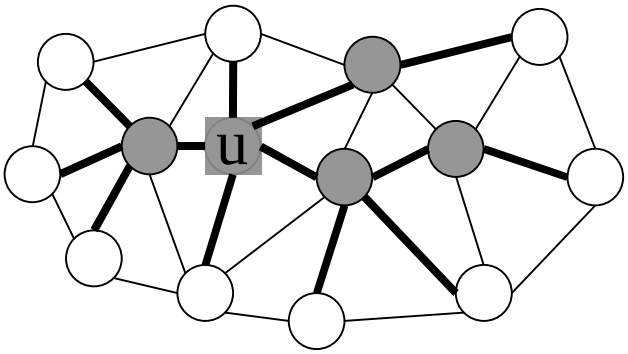

Figure 2: A Mobile Ad-hoc Network (MANET) amongst UAVs in an autonomous team.

In figure 2 above, the circles denote UAVs, and lines between them denote UAVs in direct wireless transmission range of one another. Bold lines indicate the min-hop path spanning tree rooted at node **u**. Note that a link-state update originating at node u is disseminated though the network via only five transmissions (indicated by the shaded nodes).

Inter-Team Communications

In large-scale deployments, multiple autonomous UAV 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.

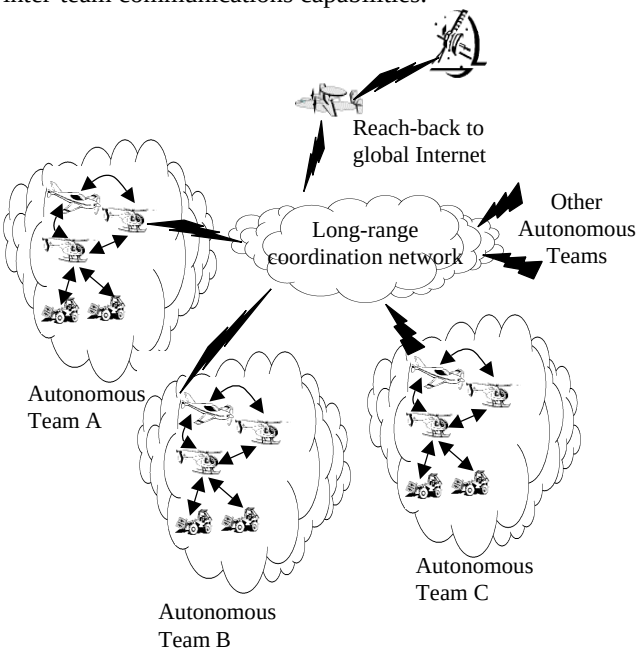

Figure 3 A large-scale deployment of autonomous teams with inter-team communications capabilities.

As described in the previous subsection, SRI's TBRPF MANET routing protocol provides an efficient, agile and scalable solution for information operations within a single autonomous team. We extend this architecture by incorporating a **router affiliation** protocol for the formation of self-configuring, self-sustaining dynamic networks. In our architectural model, at least one UAV in each autonomous team carries wireless communication devices capable of operating on both the low-power, short-range intra-team network and the 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.

The router affiliation mechanism we have employed in our architecture is a core component of the Internet Protocol, Version 6 Stateless Address Auto-configuration mechanism as specified in [4,5]. This mechanism provides periodic **router advertisement** messages that serve as *beacons* for UAVs to locate an inter-network router for their autonomous team. The mechanism additionally 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. As described in the next section, these properties provide the necessary preconditions for **global internetworking**.

Global Internetworking

Autonomous teams of UAVs must perform missions such as surveillance, intelligence gathering, and coordinated tactical strikes without risking human lives in dangerous environments. Yet, long-range communications capabilities are required to provide human observers in distant command posts with command and control or situation awareness access to both autonomous teams and the individual UAVs in the forward deployment. 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. Our scheme uses the Internet Protocol Version 6 (IPv6) addressing architecture [6] 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 and identifies the home network from which the node originates.

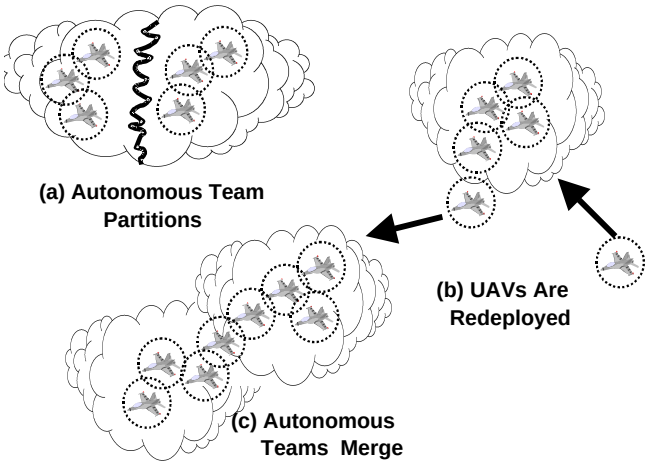

Figure 4: Node mobility events.

As nodes move throughout the network (see figure 4), 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. In this way, nodes in the forward deployment can be accessed and tracked even as they move throughout the theater of operation.

A key element of our architecture is the **Internet Engineering Task Force (IETF) Mobile IPv6** protocol [7] that manages the **binding** between a node's home address and current care-of address. Using Mobile IPv6, **correspondent nodes** may access mobile nodes regardless of their current location in the forward deployment by maintaining **binding cache entries**. 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 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 deployments using the global Internet in its current manifestation.

3. Practical Test-bed Elements

In this section, we describe the computer and network hardware elements, software integration architecture, and surrogate UAV nodes 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.

Computer/Communications Hardware Elements

We are currently using commercial, off-the-shelf *Lucent WaveLAN/IEEE Turbo 11 Mb PC Cards* as the radio-frequency wireless network interfaces in our test-bed (www.wavelan.com). This radio is compliant with the IEEE 802.11 and IEEE 802.11b Standards [9]. It operates in the 2.4 GHz frequency band, using the Direct Sequence Spread Spectrum (DSSS) modulation technique, and provides up to 11 Mbps data transfer rates with a maximum range of approximately 1000m line-of-sight. We configure the WaveLAN cards to use the IEEE 802.11b **Ad-hoc** mode so that communications are **peer-to-peer**, rather than 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 which were chosen for their small form factor and light weight.


Figure 5 Toshiba Libretto sub-notebook and Compaq iPAQ Pocket PC with wireless network interface card.

The Toshiba Libretto model 110ct is a sub-notebook computer that uses a 233 MHz Intel Pentium processor, has a 5 GB hard drive, a 800x480 color display, and 2 type II PCMCIA card slots. The entire unit weighs 2.2lbs. The Compaq iPAQ H3650 uses a 206 MHz Intel Strongarm processor, has 32 MB RAM, 16 MB ROM, no hard drive, a outdoor readable TFT 240x320 LCD display with 4096 colors. An optional PCMCIA sleeve is used to accept the WaveLAN IEEE 802.11b PC Cards. With the PCMCIA sleeve, the iPAQ weighs less than 1lb.

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. The current port runs on RedHat Version 7.0 and has been tested under multiple Linux kernel releases including the most recent release 2.4.9 from (www.kernel.org). We have also integrated the USAGI Project IPv6 distribution (www.linux-ipv6.org) that includes the most up-to-date standards compliant IPv6 implementation available for Linux. The USAGI distribution also includes an advanced implementation of Mobile IPv6 for Linux (http://www.mipl.mediapoli.com/) which we use in our testing. Finally, we have developed and integrated an early implementation of the **ISATAP** protocol in our custom Linux kernel and use it extensively to support our flexible addressing scheme.

For testing and debugging, we use **tcpdump (**www.tcpdump.org**)** to collect packet traces to verify

protocol correctness. The collected packet traces are viewed using **Ethereal** (www.ethereal.com), a network protocol analyzer tool with a graphical user interface. We have modified Ethereal (version 0.8.15) to enable a user to examine different fields of TBRPF protocol packets. We use several other standard network applications to exercise the network, including the MASH (www.openmash.org) **vic** and **vat** conferencing tools, web browsers, and the **netperf (**www.netperf.org**)** performance measurement utility.

During test bed experiments and demonstrations, we use topology display software we have developed to graphically display the network as nodes join and leave and links are formed and break. This tool is implemented in X-windows using the GTK toolkit under FreeBSD and Linux (there is also a Windows 95 version). Thick green and thin red lines indicate solid and transient links, respectively.

Figure 6. Dynamic topology display screen shows current nodes and links on iPAQ.

Surrogate UAVs

For outdoor field experiments, we are using several unmanned helicopters and ground robots that are made available through partner research groups at SRI International and the University of California, Berkeley as part of the Office of Naval Research autonomous vehicle research project. 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, and a payload capacity of 66 pounds. The RMAX has a maximum flight duration of 60 minutes, altitude limitation of 328 feet, a control range of 500 feet, and is equipped with a 246cc water cooled 2-stroke engine.

Figure 7. Yamaha R-50 and RMAX Aero Robot helicopters.

The ground robot platform is an ActivMedia Pioneer intelligent robot (www.activrobots.com) which has an embedded computer running Linux, four all-terrain wheels that can move at 0.8 meters per second and carry a payload of up to 30 kg.

Figure 8. ActivVision Pioneer All Terrain Robot with Toshiba Libretto node.

Note that portable computers equipped with wireless network interface cards and loaded with the software elements that comprise our network architecture are used as nodes in our autonomous team test networks. On the Pioneer robots, we currently affix Toshiba Librettos with Velcro tape, and a crossover cable connects to the Pioneer robot's control computer (see figure 8). In the near future we plan to move our software onto the Pioneer's control computer eliminating the Libretto and the crossover cable. On the helicopters, we place nodes in a padded payload box

and use an external antenna that is positioned below the landing gear during flight for reliable RF signal range.

Figure 9. Mounting node in payload box onto the helicopter.

4. FLIGHT EXPERIMENTS

Flight experiments have been conducted at the Richmond Field Station along with members of the UC Berkeley BEAR and SRI Autonomous Control research teams since May 2000. Our 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 sub-sections, we describe the experiments we have conducted to date.

Deployment of TBRPF routing protocol in Surrogate UAVs

The primary objectives for the first flight test were to deploy the Toshiba Libretto communications processor on the helicopter, and to verify the correct working of the TBRPF MANET routing protocol. First, a Toshiba Libretto communications processor was mounted on the Yamaha helicopter to be used for the test. A metal enclosure was devised to accommodate the Libretto along with foam packing material to dampen in-flight vibration. The enclosure was securely fastened to the underside of the helicopter and an external antenna was mounted on the helicopter's tail for the communication processor's WaveLAN wireless network interface. Two persons, denoted A and B, each operated a handheld Toshiba Libretto communications processor (nodes t2 and t4, respectively) while the pilot flew the helicopter-mounted Libretto (node t3). Each node ran the TBRPF routing protocol over a WaveLAN IEEE 802.11 RF wireless network interface to form a three node MANET. The nodes used standard Internet (IPv4) data delivery protocols (ICMP/IP, RTP/UDP/IP) to exchange unicast messages while the TBRPF routing protocol provided dynamic route adaptations in response to topology changes. The two flight test scenarios that were conducted are described below.

Scenario I: In this test (see figure 10), nodes t2 and t4 were positioned at the rear left and right corners (respectively) of a building adjacent to the flight test field while the pilot deployed the helicopter (t3) in the flight test field in front of the building. During the test, t2 and t4 remained stationary (and the bi-directional communications link t2↔t4 remained stable) while t3 moved between a series of aerial waypoints. Nodes t2 and t4 each ran persistent "ping" sessions directed at t3 with ICMP_ECHOREQUEST messages sent at 1 second intervals. The test proceeded through four phases of about 60 seconds per phase as follows:

a) With the helicopter on the ground, both the links t2↔t3 and t4↔t3 are broken since the building blocks RF transmissions. As there is no path to t3, no ICMP_ECHOREQUEST messages sent from t2 or t4.

b) When the helicopter assumes "high center" hovering position above the level of the building with line-of-sight to both t2 and t4, both the links t2↔t3 and t4↔t3 are established. Now, both t2 and t4 send ICMP_ECHOREQUEST messages to t3 at 1sec intervals over their respective links to t3. Both t2 and t4 receive ICMP_ECHOREPLY messages from t3 at ~3msec round-trip delay.

c) When the helicopter assumes "low right" hovering position with line-of-sight to t4, the building blocks RF transmissions to t2. Hence, the link t2↔t3 link breaks while t4↔t3 link remains. t4 continues to send ICMP_ECHOREQUESTs over t4↔t3 link, and t2 sends ICMP_ECHOREQUESTs over newly-formed t2↔t4↔t3 multi-hop relay path. We observed that t4 continues to see ~3msec round-trip delays for t3's ICMP_ECHOREPLYs while t2 sees ~6msec round-trip delays due to multi-hop relay via t4.

d) When the helicopter assumes "low left" hovering position with line-of-sight to t2, the building blocks RF transmissions to t4. Hence, the link t4↔t3 link breaks while t2↔t3 link remains. t2 continues to send ICMP_ECHOREQUESTs over t↔t3 link, and t4 sends ICMP_ECHOREQUESTs over newly-formed t4↔t2↔t3

multi-hop relay path. We observed that t2 continues to see ~3msec round-trip delays for t3's ICMP_ECHOREPLYs while t4 sees ~6msec round-trip delays due to multi-hop relay via t2.

continuity was robust when links were either "solid up" or "solid down" but degraded in "marginal" link state instances. This was expected since the initial TBRPF implementation did not incorporate link state metrics such

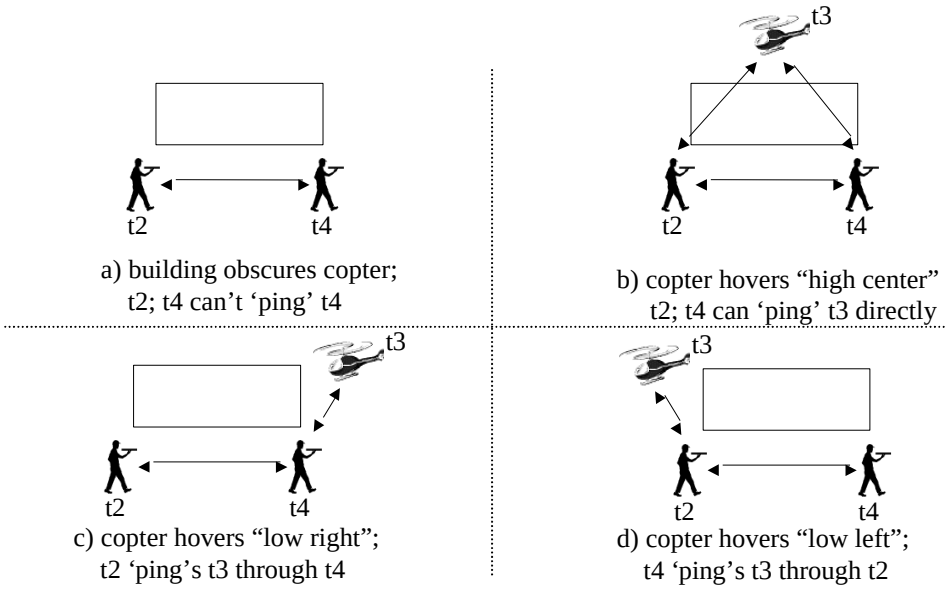

Figure 10 - TBRPF protocol verification through multi-hop flight

Scenario II: In the second test nodes t2, t4 and the helicopter (t3) began in close proximity with one another in the flight test field such that all bi-directional links (t2↔t3, t2↔t4, t3↔t4) were established. During the test, nodes t2 and t4 were mobile while node t3 (the helicopter) assumed a stationary "high center" hovering position. Nodes t2 and t4 engaged in a bi-directional "vat" voice-over-IP session by periodically sending digitally-encoded audio samples to one another via unicast RTP/UDP/IP messages.

This test entailed a qualitative evaluation of voice-over-IP session continuity as the t2↔t4 link was broken and re-established due to signal fading and terrain features as nodes t2 and t4 moved about independently of one another. Links t2↔t3 and t3↔t4 were maintained throughout the test due to continual line of sight contact from the ground nodes to the high hovering helicopter. The voice-over-IP session was preserved across several transitions in which messages were in some instances carried across the direct link from t2↔t4 and in other instanced relayed through the helicopter via the multi-hop path t2↔t3↔t4. The test was run for approximately 10 minutes.

In both test scenarios, session continuity was preserved across dynamic topology changes as links were established and broken. From a qualitative perspective, session

as signal strength, message loss percentage, etc. Methods for dampening routing table oscillations due to marginal link states will be addressed through proposed TBRPF extensions in subsequent efforts. All the objectives for the first flight test were achieved. In terms of hardware robustness, the mounting enclosure for the Libretto provided an ideal solution that carried the airborne Libretto through the flight tests with no damage.

We also agreed on the high-level architectural strategy of using an Ethernet hub as the processor interconnect on board the helicopter, with the communications processor serving as the gateway from the helicopter's Ethernet to the wireless interface.

Deployment of TBRPF routing protocol in mobile robots

We have integrated the TBRPF MANET routing protocol with the Saphira robotics control environment. The Saphira environment allows a human pilot operating a graphical user interface at a workstation to control one or more mobile Pioneer ground robots. Control messages between the workstation and ground robots are carried over IEEE 802.11 wireless network devices with single-hop connectivity only. (i.e., the robots must always be within range of either the workstation itself or another fixed-infrastructure communications access point element.) Through our

integration efforts, however, we have shown that the Saphira robotics control environment can also be supported in a MANET.

On both the Saphira workstation and Pioneer robot, the native (single-hop) wireless network interface is replaced by an Ethernet interface for point-to-point communications with an external MANET communications processor. The communication processors are Toshiba Libretto's that run the TBRPF routing protocol and provide two network interfaces; an Ethernet interface for point-to-point communications with the Saphira workstation/Pioneer robot and an IEEE 802.11 Lucent WaveLAN wireless interface for MANET support.

The first task was to mount a Toshiba Libretto on the Pioneer robot. The small form factor of the Toshiba Libretto as well a flat surface on the Pioneer robot allowed us to simply tape the Libretto securely onto the robot. We then interfaced the Pioneer robot's on-board Linux-based processor with the Libretto via a simple 10BaseT Ethernet "crossover" cable that enables point-to-point Ethernet communications between the two processors.

Next, IPv4 addresses were assigned such that the Saphira workstation and accompanying Libretto communications processor form their own logical IPv4 subnet, the Pioneer robot and it's accompanying communications processor form a second IPv4 subnet and the mobile ad hoc wireless network forms a third subnet. In this way, the communications processors serve as IPv4 routers between the MANET and Ethernet subnets on the Saphira workstation and Pioneer robot.

A simple hallway exploration mission was used to test the integration (see figure 11). In this mission, the pilot at the Saphira workstation directed the Pioneer robot through a round-trip tour of a hallway. As the pilot directed the robot to travel beyond single-hop wireless transmission range from the workstation, the TBRPF routing protocol dynamically established a multi-hop route across a mobile relay node physically positioned between the workstation and robot. This allowed the pilot to continue to direct the robot with no perceptible interruption in service as the multi-hop route was established (and subsequently torn down when the robot returned to its base). In other words, dynamic MANET topology changes were transparent from the perspective of the Saphira robotics control session.

Autonomous Team of UAVs and Mobile Robots

The goal was to conduct integrated MANET experiments with nodes on mobile airborne and ground-based vehicles. Our experiments focused on a single autonomous team, with all nodes running the TBRPF routing protocol on the same WaveLAN RF channel.

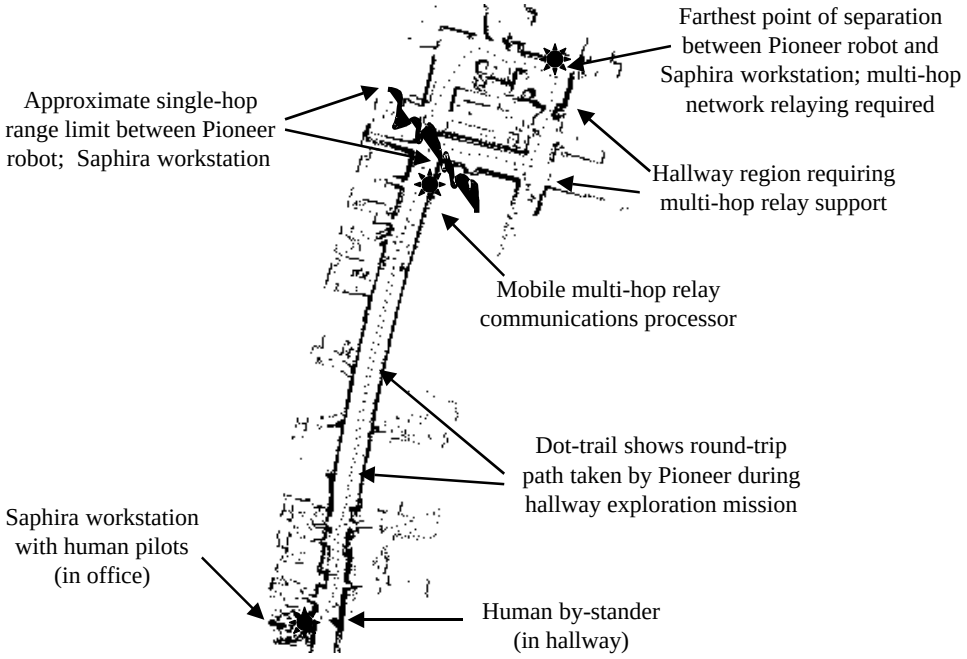

Figure 11 - Hallway exploration mission with Pioneer robot sensor data over multiple wireless hops

The first test involved two helicopters, a Pioneer ground robot, a robot controller workstation and a handheld node as members of a single autonomous team. To begin with, the Pioneer robot and robot controller were close to one another. The objective was to demonstrate the ability of the robot operator to move far away from the robot while still maintaining a control interface capability. However, the

robot control application was being run over a TCP-based remote X-windows session to the robot. For ease of navigation, the robot was transmitting live video feedback to the robot controller via this session. During the experiment, the Pioneer pilot also maintained a voice-over-IP session with an observer stationed near the robot.

We observed that the TCP robot control session worked well when there was a direct link between the robot and the pilot, and also worked fairly well when a single multi-hop relay was required. But, the TCP session performance degraded poorly when link states became marginal and significant packet loss occurred. To make matters worse, the traffic from the live video camera was saturating the channel, thus causing the huge delay variances and packet loss we were experiencing. This was not the case for the UDP-based voice-over-IP session, which continued to deliver intelligible audio in the presence of packet loss. From this, we learned that:

1. Quality-of-service based routing through the incorporation of intelligent link-state metrics within TBRPF will be beneficial to TCP's performance.

2. Real-time situation awareness updates from the robot should use UDP instead of TCP on MANETs.

Improvements to robot control architecture

The objective was to experiment with advancements in the Pioneer robot control architecture which de-couple real-time sensor data transmission from the TCP robot control session.

The experiment included one helicopter plus a number of additional ground-based nodes (both hand-held and mounted on a Pioneer robot). In this test, the Pioneer robot remained on one side of the large building adjacent to the flight test field while the robot control workstation moved to the other side of the building with the helicopter hovering in position to act as a relay. When the direct link between the robot and control workstation was broken, both nodes automatically discovered the helicopter as a multi-hop relay via the TBRPF routing protocol. The robot control workstation was able to completely encircle the building while still controlling the robot by using the helicopter and another hand-held node as dynamically discovered multi-hop relays. This was the first time we have been able to do this, and was made possible by the changes to the robot control architecture.

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 the autonomous teams. This experiment succeeded in forwarding ICMP and TCP messages between the two autonomous 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. This was the first successful attempt with communications between multiple autonomous teams.

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 the Lucent WaveLAN interfaces on the network processors 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 as multi-hop relays by TBRPF as needed. 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 were also able to observe 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. These blockages effectively "chop up" the data bits which the IEEE 802.11 protocol tries to repair via MAC-level retransmissions, thus leading to unacceptable delays.

Members of UC Berkeley's BEAR team were able to design an alternate mounting scheme for the helicopter's antenna. This used the same antenna, but changed the antenna's physical orientation with relation to the helicopter's undercarriage. This has made an enormous difference in the performance of the antenna. It has practically eliminated the delay variance issue we have been encountering since the beginning. 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. In other words, mobility within the operational parameters of the Yamaha helicopter now seems to be eliminated as a factor for single-hop IEEE 802.11 performance. Elimination of the delay variance issue will allow us to introduce complex flight patterns instead of just simple hovering in future experiments

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. Our test network employed four ground nodes deployed in a "star" configuration with the helicopter as the hub.

In this experiment, researchers carrying ground nodes dispersed around the flight test field until the helicopter became the central hub in a star topology. In other words, the ground nodes were either far enough apart from one another or shielded from one another by terrain features (e.g. buildings, trees, parked cars, etc) such that the helicopter was required to provide a multi-hop relay service for all multicast messages. The researchers 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. In this scenario, the researchers operating the four ground nodes took turns keying up their push-to-talk microphones and speaking to prove that they were still "all present and accounted for". 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. But, when the helicopter flew too far away from individual ground nodes or became shielded by terrain features, multicast messages to/from those ground nodes were lost as expected, leading to "garbled" voice-over-IP sessions.

Reliable Transport Protocol Performance

As discussed in previous subsections, the TCP reliable transport protocol has been shown to suffer severe performance limitations in multi-hop wireless networking environments due to assumptions made by TCP's congestion control algorithm. A recent enhancement to the TCP protocol known as **selective acknowledgement (SACK)** augments the standard TCP congestion control strategy. We performed a series of experiments to verify whether the TCP SACK scheme improve reliable transport protocol performance in a MANET.

Our experiment measured throughput improvement for TCP sessions using Selective Acknowledgement (SACK) vs. non-SACK based sessions. UAV flight patterns used in the experiment included, simple hover, tight circling above a stationary waypoint and slow movement between multiple waypoints.

Our goal was to determine whether the TCP SACK protocol feature could improve TCP performance for MANETs in cases where packets are lost due to temporary link outages.

In our experiment, we arranged a three-node ad hoc network in which the helicopter served as a multi-hop relay between a pair of correspondent ground nodes using the 'netperf' traffic generator tool with TCP SACK both enabled and disabled. With a stable network configuration (helicopter hovering; ground nodes stationary; no packet loss) 'netperf' reported identical performance results for multi-hop TCP with or without SACK enabled. We then asked the pilot to maneuver he helicopter so that links were broken intermittently, and found that TCP SACK provided small performance gains over not using SACK. We found that, if links were broken for approximately 50% or more of the duration of the 'netperf' test (i.e. 50% or more packets lost), TCP performance dropped to nearly zero whether or not SACK was used. But when the packet loss rate was more moderate (e.g. 25% packet loss) TCP SACK sessions showed a definite performance improvement over non-SACK sessions. In one measurement, TCP SACK reported 750Kbps while non-SACK reported 540Kbps for identical packet loss rates.

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 12). 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 was configured as a surrogate AWACS node and 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.

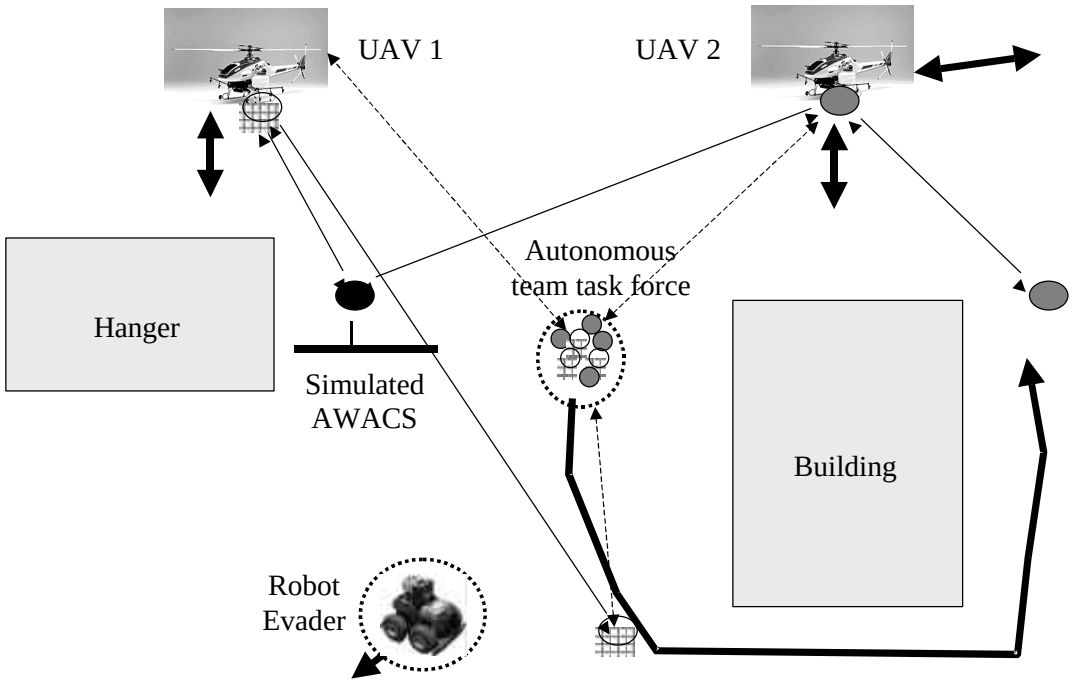

Figure 12 - Multiple autonomous teams engaged in joint tactical mission

This experiment was staged as a large-scale demonstration in conjunction with the annual review meeting for Office of Naval Research unmanned vehicle research project. Meeting attendees were recruited to carry nodes in the network even though they had no prior experience with our technologies. The purpose of the demonstration was to both familiarize the meeting attendees with our technologies and gain insights into the dynamics of larger network deployments.

The experiment proved 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

In this section, we describe the experiments that were conducted to measure the performance under various configurations in our mobile ad-hoc network. The performance metrics include throughput and average delay. The experiments described in this section were conducted indoors with fixed nodes. However, the results presented serve as a useful bound on the actual performance that can be expected in an autonomous team of UAVs.

The wireless network interface card used in our MANET test-bed is compliant with the IEEE 802.11 and IEEE 802.11b Standards. Before presenting our experimental results, we compute the minimum overhead incurred at the Medium Access Control (MAC) and physical (PHY) layers in the transmission of a single frame by the IEEE 802.11 MAC Protocol. This is presented in Subsection 5.1. The overhead traffic in the MANET that impacts its performance is described in Subsection 5.2. The tools used in the experiments are described in Subsection 5.3. Finally, the performance results are presented in Subsection 5.4.

Minimum Overhead in IEEE 802.11 MAC Protocol

We now compute the minimum overhead incurred at the MAC and PHY layers in the transmission of a single frame by the IEEE 802.11 MAC Protocol

The *basic access method* of the IEEE 802.11 MAC protocol is described next. When a station senses the channel is idle, the station waits for a Distributed Inter-frame Space (DIFS) period, and samples the channel again. If the channel is idle, the station transmits the frame. If the receiving station determines that the frame was received correctly, it waits a Short Inter-frame Space (SIFS) interval and transmits a positive acknowledgment (ACK) frame back to the source station.

However, if the station senses the channel is busy to begin with, the station must defer until (see Figure 13) the medium is idle for a DIFS period after the end of the current transmission. After deferral or prior to attempting to transmit again after a successful transmission, the station selects a random backoff interval and decrements the backoff interval counter while the medium is idle. Finally, a station begins to transmit a packet when the backoff timer expires.

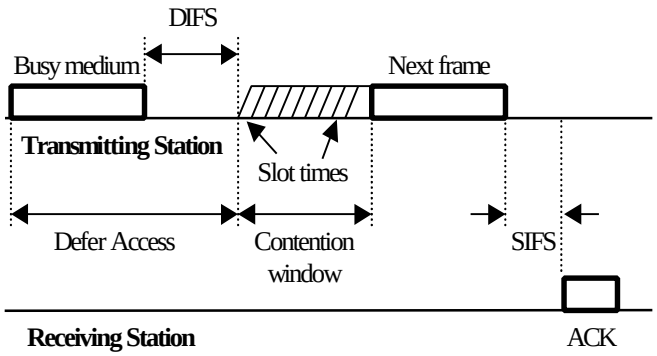

Figure 13: The basic access mechanism of the IEEE 802.11 Standard.

Following the analysis presented in [10], we enumerate the overheads involved in the wireless transmission in Table 1. Note that all the frames in the MAC layer (data and ACK) require the PHY layer overhead. For every unicast data frame that is transmitted successfully, the following overheads are required (see Figure 13). The actual values given below were obtained from the IEEE 802.11 Standard.

- PHY layer overhead
- DIFS = 50 ms
- Length of Contention Window
- SIFS = 10 ms
- ACK frame (14 bytes)

However, for broadcast or multicast frames, overheads associated with the SIFS as well as the ACK frames do not apply. For the case of unicast transmissions Table 2 computes an upper bound on the throughput that can be achieved when the length of the payload of MAC data frames is 1500 bytes and 2300 bytes.

Overhead	Overhead in μs
DCF Inter-frame Space (DIFS)	50
Short Inter-frame Space (SIFS)	10
PHY layer Header (Long Preamble and Header)	192
PHY layer Header (Short Preamble and Header)	96
MAC layer Header	(34x8)/11 = 24.7
ACK	(14x8)/11 = 10.2
Minimum Total Overhead (Broadcast)	266.7

(Long Preamble and Header)	
Minimum Total Overhead (Unicast) (Long Preamble and Header)	478.9
Minimum Total Overhead (Broadcast) (Short Preamble and Header)	170.7
Minimum Total Overhead (Unicast) (Short Preamble and Header)	286.9

Table 1 Computation of the overhead involved in the transmission of a MAC frame for the DSSS system when the payload data rate is 11 Mbps. Each of the data and ACK frames incurs the PHY layer overhead. Since the computed overhead neglects the length of the contention window, we refer to it as the Minimum Total Overhead.

MAC Payload (bytes)	Payload Transmission Time (in μs)	Minimum Total Overhead (Short Preamble & Header)	Maximum Throughput (in Mbps)
1500	1090.1	286.9	11x1090.1/ (1090.1+286.9) = 8.71
2300	1672.7	286.9	11x1672.7/ (1672.7+286.9) = 9.39

MAC Payload (bytes)	Payload Transmission Time (in μs)	Minimum Total Overhead (Long Preamble & Header)	Maximum Throughput (in Mbps)
1500	1090.1	478.9	11x1090.1/ (1090.1+478.9) = 7.64
2300	1672.7	478.9	11x1672.7/ (1672.7+478.9) = 8.55

Table 2 Maximum achievable throughput for unicast transmissions.

Overhead Traffic

This subsection describes the overhead traffic in the mobile ad-hoc network that impacts its performance. Recall that Subsection 5.1 dealt with the overhead at the MAC and the physical layers in the transmission of a single data frame. In addition to this, the mobile ad-hoc network also incurs an overhead arising from other kinds of messages (see Figure 10). These include:

- Hello messages involved in the TBRPF neighbor discovery protocol. In the current implementation, TBRPF neighbor discovery replaces ARP for interfaces that support mobile ad-hoc networking capabilities.

- TBRPF protocol messages: In the current implementation, TBRPF messages are transmitted as UDP messages using the temporary port 5555.

- TCP level Acknowledgment messages. These are usually piggybacked along with data traffic in the reverse direction, if such traffic exists. Otherwise, the TCP acknowledgments are sent as standalone messages.

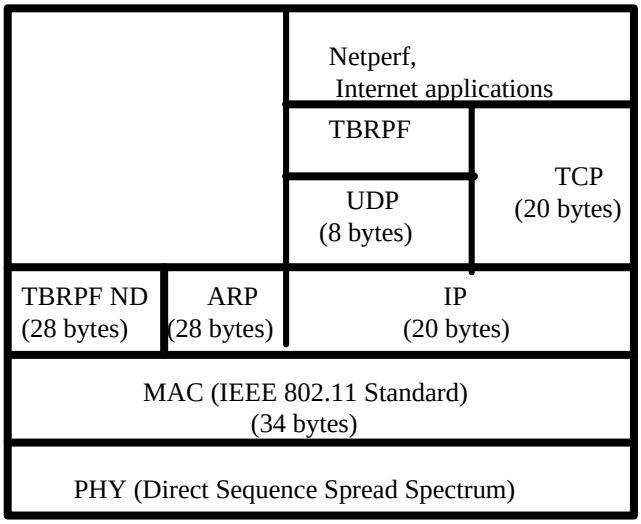

Figure 14 Protocol layers involved in the mobile ad-hoc network along with the overhead incurred at each layer.

Network Performance Tools

We used the ***netperf*** *(www.netperf.org)* and ***ping*** tools to measure the performance of the mobile ad-hoc network.

Netperf is a benchmark that can be used to measure various aspects of networking performance. The most common use of *netperf* is measuring bulk data transfer performance, where the test measures how fast one system can send data to another system, and how fast that other system can receive it. This is also referred to as "stream" or "unidirectional stream" performance. The *TCP stream performance* test is the simplest test type of the *netperf* program. It is invoked by entering the command
localHost% netperf -H remoteHost -l len
which performs a test of duration *len* seconds between the local system and the system identified by *remoteHost*. The socket buffers at either end will be sized according to the

systems' default and all TCP options will be at their default settings.

We have also conducted performance measurements using the ***ping*** tool. A ping session or test involves a sequence of ICMP echo request messages sent from the source of the ping session to its destination. The destination responds with echo reply messages, each containing a copy of the data sent in the corresponding echo request message. Hence, a ping test induces bi-directional traffic within the network.

Performance Results

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. As explained before, 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 measured for each traffic stream (averaged over five runs) is given below:

a) One unidirectional stream of traffic from node A to B.
   - *Throughput of A → B stream = 5.03 Mbps.*

b) Two concurrent unidirectional streams of traffic (A → B and C → D)
   - *Throughput of A → B stream = 2.33 Mbps.*
   - *Throughput of C → D stream = 2.98 Mbps.*

c) Three concurrent unidirectional streams of traffic (A → B , C → D, and E → F)
   - *Throughput of A → B stream = 1.79 Mbps.*
   - *Throughput of C → D stream = 1.67 Mbps.*
   - *Throughput of E → F stream = 1.29 Mbps.*

d) Four concurrent unidirectional streams of traffic (A → B , C → D, and E → F, and G → H)
   - *Throughput of A → B stream = 1.06 Mbps.*
   - *Throughput of C → D stream = 1.45 Mbps.*
   - *Throughput of E → F stream = 0.55 Mbps.*
   - *Throughput of G → H stream = 1.20 Mbps.*

From the experimental results, we observe that the wireless transmission medium is shared between the different traffic streams in a somewhat fair manner. In particular, no individual traffic stream is starved by the presence of other traffic streams.

Next, we measured the throughput for the configuration shown in Figure 15, where the thick lines represent walls that can effectively block the propagation of wireless transmission in the 2.4 GHz frequency band. Hence, only adjacent nodes on the line A—B—C—D—E are within communication range of one another.

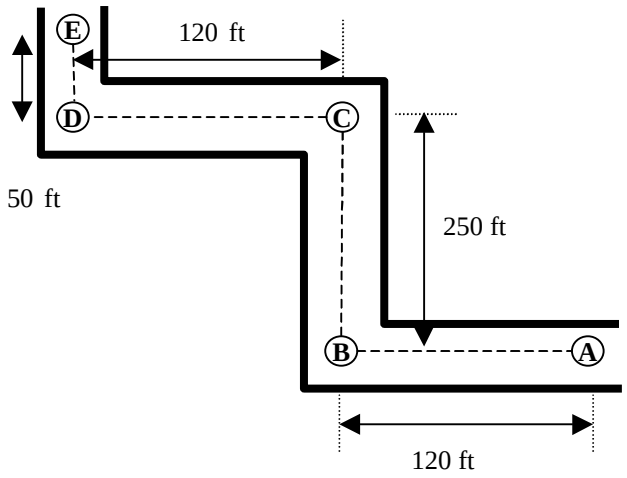

Figure 15 Performance results for multi-hop traffic streams originating from node A.

For this configuration shown in Figure 15, we had one unidirectional stream of traffic from node A to node C, which is routed via node B. The measured throughput for this 2-hop traffic stream is shown in Table 1. Table 1 also shows the measured throughput for a single 3-hop (respectively, 4-hop) traffic stream from node A to node D (respectively, node E). As indicated in the table, a number of runs were conducted using the ***netperf*** tool for each case. In order to verify the throughput results, 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 3 Multi-hop throughput performance for unidirectional TCP streams.

For the configuration of fixed nodes shown in Figure 15, 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 4.

Figure 16 plots the average delay as a function of the offered traffic as the number of hops in the ping session varied from 1 hop to 4 hops. Figure 17 likewise plots the packet loss rate as a function of the offered traffic.

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 4. Multi-hop delay performance using the ping tool.

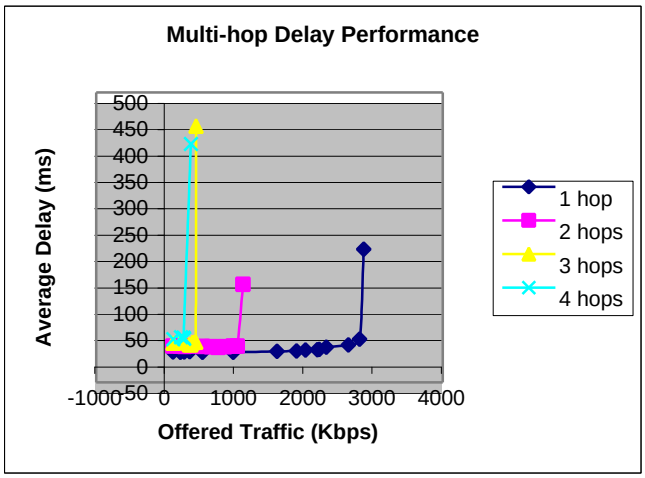

Figure 16 Variation of average delay with offered traffic for ping sessions of varying number of hops.

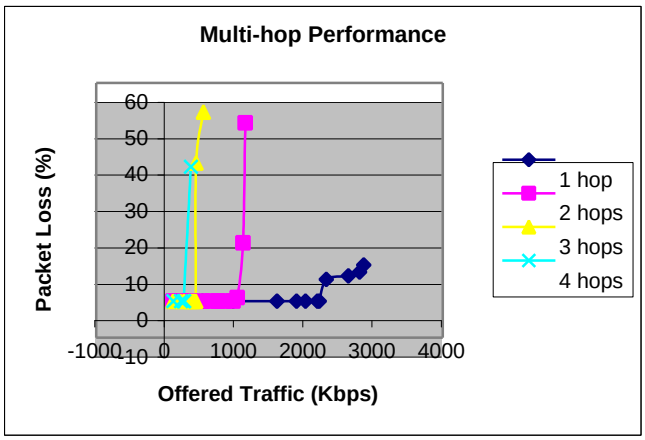

Figure 17 Variation of packet loss with offered traffic for ping sessions of varying number of hops.

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 18) 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 mobile ad-hoc network via TBRPF link-state updates. Minimum-cost paths are then computed, where the link cost is the maximum of the link cost reported in both the directions.

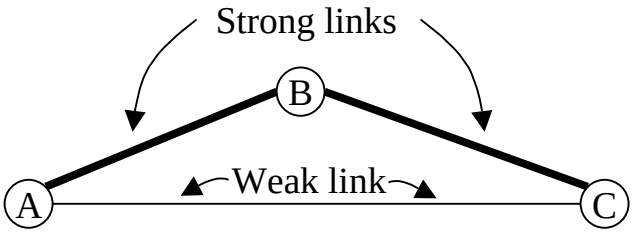

Figure 18
The link between nodes A and C is weak, and can incur a significant amount of packet loss.

6. 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.

Acknowledgments - The following people have made significant contributions to this ongoing effort:

Andrew Agno, Pauline Berry, Yonael Gorfu, J. Peter Marcotullio, Richard Ogier, Charles Ortiz, Peter Ray, David Shim, Ambatipudi Sastry, Ron Tal, Anne Urban, Regis Vincent, Julie Wong

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] Bellur, B., R.G. Ogier, and F. L. Templin, "Topology Broadcast based on Reverse Path Forwarding (TBRPF)," draft-ietf-manet-tbrpf-02.txt, September 2001 (work in progress).

[3] Templin, Fred "A Practical Implementation of a Mobile Ad-hoc Network Routing Protocol," *SRI International Technical Report*, ITAD-3527-TR-99-132, September 1999.

[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.

[10] Benny Bing, "Measured Performance of the IEEE 802.11 Wireless LAN," *Proceedings of the 24th Conference on Local Computer Networks*, 1998, pp. 34-42.

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.*