2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Robust Routing and Multi-Confirmation Transmission Protocol for Connectivity Management of Mobile Robotic Teams

James Stephan, Jonathan Fink, Benjamin Charrow, Alejandro Ribeiro, and Vijay Kumar

Abstract— **Providing reliable end-to-end communication for teams of robots requires the integration of novel routing techniques, motion planning algorithms, and transport level communication protocols. In this paper we look at existing robust routing solutions that provide redundancy at the routing layer and develop the Multi-Confirmation Transmission Protocol (MCTP) to take advantage of that redundancy at the transport level. The resulting system that integrates robust routing and MCTP is evaluated in experiments performed in complex environments. The integrated system is observed to provide a robust architecture that allows for near lossless communication while operating in a complex environment with less traffic than standard confirmation protocols.**

I. Introduction

The usefulness of autonomous robot teams depends greatly on their ability to work together. The further we are able to extend their abilities, particularly their communication skills, the more useful they are in a variety of scenarios. For example, robot teams would be ideal for situations that are dangerous for humans. One typical example is a search and rescue mission where currently we rely on humans to enter into unsafe territories. Should a team of autonomous robots be able to better communicate with themselves, as well as with a human operator, the more successful they will be in replacing human rescuers. This paper aims to demonstrate the ability of a team to have reliable communication without the need of existing infrastructure by combining intelligent network routing with a novel transmission protocol.

Prior works in motion control for networked teams have focused on constraining or modifying motion trajectories that preserve communication links or recover from disconnected topologies [1]–[6]. There are two main drawbacks of these works. The first drawback is that these papers focus on point-to-point links, instead of end-to-end rates. The second drawback is that the link between two robots is purely a function of distance. However, it has been shown that proximity alone is a poor indicator of channel reliability due to the effects of shadowing and fading [7], [8].

In attempting to resolve these deficiencies prior works [9], [10] leveraged an existing stochastic model of point-to-point communication rates, based on received signal strength (RSSI), and incorporated it into a robust solution to the concurrent routing and mobility problem in order to maintain end-to-end data rate guarantees. Further, this work experimentally demonstrated that by using coarse predictions of point-to-point communication capability, long-term motion and network routing plans could be computed that were robust to instantaneous fluctuations in signal strength due to fading and maintain desired end-to-end rates. It has to be noted that these experimental demonstrations are conceptual in that the ability to maintain desired rates is shown to be available at the routing layer but not actually realized at the transport layer. Indeed, existing transport-level algorithm such as TCP and UDP are designed to run on reliable physical substrates and as such inherently incapable of exploiting the provided redundancy.

Regarding that inability, there has also been considerable research in the area of communication over wireless links. The previous literature has detailed the limitations of using the standard Transmission Control Protocol (TCP) algorithm over a wireless link, [11]–[13]. The fundamental flaw in TCP is the assumption that any error in the communication is the result of congestion. As such, there have been modifications to TCP to mitigate the misclassification of random losses as congestion associated with wireless links [14], [15]. These methods perform well in a point-to-point static configuration but are sub-optimal when links disappear and reappear as robots move through an environment.

The contributions of these work are: (i) The development of the multi-confirmation transmission protocol (MCTP) to exploit the routing-layer redundancy provided by the robust routing solutions of [9], [10]. (ii) Integration of MCTP and robust routing into a system that provides robust communication capabilities for teams of robots. (iii) An experimental demonstration of the advantages of the integrated system with respect to regular (non-robust) routing layers and conventional transport protocols.

MCTP utilizes multiple point-to-point packet receive confirmations as opposed to the single end-to-end confirmations provided by conventional transport protocols. This makes MCTP better suited to be a OSI transport layer replacement in wireless ad-hoc networks in general and to integration with robust routing, an OSI network layer replacement, in particular. We validate experimentally that our protocol outperforms UDP and has equivalent performance to a single acknowledgement based protocol while having a smaller communication overhead. Upon integration of MCTP and robust routing we perform experiments to serve as a comparison between robust routing and traditional network routing solutions when utilized within the scope of a mobile communication-maintenance setting. This comparison

J. Stephan (jstephan@seas.upenn.edu) A. Ribeiro (aribeiro@seas.upenn.edu) are with the Department of Electrical & Systems Engineering, University of Pennsylvania, Philadelphia, PA. J. Fink (jonathan.r.fink3.civ@mail.mil) is with the US Army Research Laboratory, Adelphi, MD. B. Charrow (bcharrow@seas.upenn.edu) and V. Kumar (kumar@ seas.upenn.edu) are with the Department of Mechanical Engineering& Applied Mechanics, University of Pennsylvania, Philadelphia, PA.

978-1-4799-6934-0/14/$31.00 ©2014 IEEE

is demonstrated through a series of experiments where data must be communicated from a sensing robot back to an access point through a multi-hop network while the sensing and support robots in the network are autonomously deployed into a complex environment. These demonstrations show the promise of MCTP and robust routing as a communication maintenance architecture that allows for near lossless communication while operating in a complex environment with less traffic than standard confirmation protocols.

The rest of the paper is organized as follows. In Section II we describe the robust routing solution that will be later integrated with MCTP. MCTP is introduced in Section III where we explain the transmitter and receiver side state machines that result in its implementation. We further discuss the integration of MCTP and robust routing. In section IV, we compare the performance of the integrated robust routing and MCTP system versus non-robust routing when deploying a team of mobile robots tasked with servicing the end-to-end data-rate requirements of a sensing robot.

II. Robust Routing

We use robust routing to mitigate the effects of fading by leveraging the spatial diversity of the robot's configuration. This allows for a more sustained and less variable end-to-end data rate. The key concept behind robust routing is that fading can be modeled as a Gaussian random variable with a specific mean and variance that is determined by the environment. If the mean and variance are known, then network integrity and a minimum data rate can be achieved in a probabilistic sense. This section presents the problem statement, the mathematical background, and the implementation of the robust routing System.

We are given a team of N robots, with positions x_i, for $i = 1, \ldots, N$ and a human operator at an access point, $i = 0$. We define the vector $\mathbf{x} = (x_0, \ldots, x_N) \in \mathbb{R}^{2(N+1)}$. The robots are kinematic and fully controllable. Therefore we assume a simple control model of the form $\dot{x}_i(t) = u_i(t)$, where $u_i(t)$ is the control input for robot i. The goal is to have a sensing robot, $i = N$, move to a specific point x_g, while maintaining a minimum end-to-end data rate, a_{min}, with the access point.

We define the routing variables, α_{ij}, to indicate the percentage of time robot i will send data to robot j. We collapse across both i and j into the vector $\boldsymbol{\alpha} \in \mathbb{R}^{(N+1)^2}$. Similarly, we define a point-to-point rate function, $R_{ij}(\mathbf{x})$, which indicates the amount of data that robot i can send to robot j, given the spatial configuration $\mathbf{x}$. Therefore the amount of data that robot i can inject into the network, given $\boldsymbol{\alpha}$ and $\mathbf{x}$, is the difference between incoming and outgoing packets,

$$a_i(\boldsymbol{\alpha}, \mathbf{x}) = \sum_{j=0}^{N} \alpha_{ij} R_{ij}(\mathbf{x}) - \sum_{j=1}^{N} \alpha_{ji} R_{ji}(\mathbf{x}). \tag{1}$$

In order to maintain network integrity, we require that $a_N \geq a_{min}$ and $a_i \geq 0$ for all other i. Since the value of a_i is a function of both $\boldsymbol{\alpha}$ and $\mathbf{x}$, the solutions to the motion planning and the routing problems are dependent on each other. The random nature of wireless channels, e.g., small-scale fading caused by multipath propagation, causes the rate function, $R_{ij}(\mathbf{x})$, to also be a random value. This prevents us from making deterministic guarantees on the end-to-end rates a_i. As such, we solve for an $\boldsymbol{\alpha}$ for which the probability of achieving the input data rate, a_i, for all robots is above a reliability tolerance ϵ,

$$\mathrm{P}[a_i(\boldsymbol{\alpha}, \mathbf{x}) \geq a_i] \geq \epsilon, \forall i. \tag{2}$$

To optimally achieve probabilistic satisfaction of the end-to-end rate constraints, we formulate an optimization problem,

$$\begin{aligned} \boldsymbol{\alpha}(\mathbf{x}) = \arg\max_{\boldsymbol{\alpha}, a_\Delta} \quad & a_\Delta \\ \text{st} \quad & \mathrm{P}[a_i(\boldsymbol{\alpha}, \mathbf{x}) \geq a_i + a_\Delta] \geq \epsilon, \ \forall i. \end{aligned} \tag{3}$$

The objective is to maximize the slack a_Δ which represents a rate margin that is probabilistically satisfied for every robot in the network. This optimization problem belongs to a class of convex optimization problems known as Second Order Cone Problems (SOCP), and can thus be solved optimally [9]. We write (3) as a $\boldsymbol{\alpha}(\mathbf{x}) = \arg\max$ to emphasize that for any fixed spatial configuration of robots, we can efficiently solve for the optimal routing variables $\boldsymbol{\alpha}$. Also note that if there exists a solution to (3) for a configuration $\mathbf{x}$, with $a_\Delta \geq 0$, then the end-to-end data rate guarantee in (2) is satisfied and we see that network integrity is preserved.

This leads to the formation of a hybrid system where the motion control is performed in continuous time, and the routing is solved in discrete time. Decomposing the system in this fashion allows us to solve for the routing variables, $\boldsymbol{\alpha}(\mathbf{x}(t_m))$, for a given configuration $\mathbf{x}(t_m)$, and use them for $t \in [t_m, t_m + \Delta)$, where $\frac{1}{\Delta}$ is the rate at which (3) is solved. Since configurations that are close to each other exhibit similar wireless behavior, we can assume the routing variables for t_m are valid for Δ seconds afterwards. This is reinforced by the addition of the slack variable, a_Δ, which can be related to the set of configurations for which $\boldsymbol{\alpha}(\mathbf{x}(t_m))$ are valid. Accordingly we can decouple the system into two problems: a motion control portion operating in continuous time, and a network integrity preserving path planning portion operating in discrete time.

The path planning problem, which has dimension $\mathbb{R}^{2N}$, is complicated by the combination of geometric constraints, as well as network integrity constraints. The geometric constraints are imposed by features in the environment, such as walls and obstacles, while the network integrity constraints are imposed by the end-to-end rate requirements. Thus the space of feasible configurations for the team is the intersection of the obstacle free space, $\mathcal{X}_{free}$, and the set of network integrity preserving configurations, $\mathcal{X}_{net}$. The goal is to find a path, $\mathcal{P}$, through $\mathcal{X}_{feasible}$ from the initial configuration, $\mathbf{x}_{init}$, to any point in the set of configurations that have $x_N = x_g$. To find such a path we employ a Rapidly Exploring Random Tree (RRT), which allows us to efficiently search for a path that lies completely in the obstacle free space, $\mathcal{P} \subset \mathcal{X}_{free}$. The RRT algorithm also allows us to check for network integrity during the exploration process, guaranteeing that the waypoints of $\mathcal{P}$ preserve network integrity. This results in a path, $\mathcal{P}$, that is feasible in the physical domain, and also preserves network integrity at each waypoint.

III. Multi-Confirmation Transmission Protocol

The robust routing algorithm described in Section II enables redundancy at the routing layer. Existing transport-level algorithm such as TCP and UDP are designed to run on reliable physical substrates and as such inherently incapable of exploiting the provided redundancy. In order to leverage the benefits of robust routing and support implementation on simple radios available to power-constrained platforms, we develop the multi-confirmation transmission protocol (MCTP).

MCTP is a modification of Nagle's algorithm for small packets [16] and takes advantage of the spatial redundancy by allowing a packet that failed over one link to be retransmitted over a different link. This reduces the likelihood that a packet will be lost when one link is removed, as subsequent retransmissions of the packet will not use that particular link. This approach mitigates the random losses that occur over wireless channels due to link failure. Therefore, this communication protocol combines the benefits of both TCP and UDP protocols, to allow for efficient and reliable communication over a multi-hop wireless network.

The standard TCP algorithm is optimized for operation over wired networks where packet loss is assumed to be the result of congestion. As such, the TCP algorithm dynamically adjusts the number of unconfirmed packets on a link at any given time in order to achieve the highest data rate over that link. Operating under this assumption in a wireless context causes the link to be underutilized due to unnecessary reduction in the number of unconfirmed packets caused by random link failures. In the standard UDP algorithm the utility of the link is maximized, but has no confirmation of successful transmission. The protocol we propose mimics the utilization of UDP while providing the transmission confirmation of TCP. This protocol is agnostic to the determination of the routing variables and is therefore suitable to many applications.

The protocol operates in the following fashion. Initially, each robot is assigned a unique identifier, s, for this implementation we used, $s \in \{1 \ldots N\}$. When there is data to transmit, we uniquely label and send the packet according to the routing solution C_i. The routing solution is expressed as a unique CDF for each robot, i, with the mass on j proportional to the percentage of data i should send to j. Therefore to determine the destination, t, that robot i should use for each packet a random variable, x, is drawn from a uniform distribution from 0 to 1 and t is determined such that $x \in C_{i,t}$. We then insert the label into a data structure, $\mathbb{A}$, to record that the packet has been sent, but successful transmission has not been confirmed. We repeat this process for all outgoing packets. Upon successful reception of a

Algorithm 1 Algorithm for receive

Require: number of robots N, maximum number of packets in confirm message P, number of new message to respond to M, and maximum time between responses T

Initialize N queues $\{Q_i\}$ and timers $\{T_i\}$
while System running **do**
  if Packet p successfully received from robot s **then**
    add p_{id} to Q_s
    if size of Q_s mod M equals 0 **then**
      Send the contents of Q_s to s, reset T_s
    end if
    if size of $Q_s > P$ **then**
      Pop oldest element off of Q_s
    end if
  end if
  for i = 1 to N **do**
    if $T_i > T$ **then**
      Send the contents of Q_i to i, reset T_i
      Pop off Q_i mod M oldest elements of Q_i
    end if
  end for
end while

packet, we add the label to a data structure, $\mathbb{B}$, and respond with the last P labels, but only if there have been M new packets since last response. Upon receiving a confirmation message, we compare the labels to those in $\mathbb{A}$. If a label in the confirmation response matches one in $\mathbb{A}$, the label is removed and successful transmission of the packet is confirmed. If, however, after a specified T seconds transmission is not confirmed, the packet is re-sent. This process can be seen in Algorithms 1 and 2. The values of T, P and M can be changed to provide the desired level of delivery guarantee depending on the scenario.

In order to allow for successful reassembly at the receiving side, each message, of arbitrary length, that is to be transferred is assigned a unique 16-bit identifier. Then each message is broken up in to 85-byte chunks, suitable for transmission, and assigned a number. The message identifier as well as the chunk number are added to the payload of the transmitted packet. These packets are then provided as input to the MCTP algorithm, specifically line 3 for Algorithm 2. This information is then used to reassemble the message at the final destination as well as prevent re-transmission of duplicate packets inside the mulit-hop network. MCTP also leverages the redundancy of the routing layer by allowing a packet to travel multiple paths to the final destination with only minimal overhead. This is due to the random selection of the link to use during re-transmission. Therefore, as the amount of redundancy in the routing protocol increases the more diverse set of paths a packet is able to traverse and the more robust the system becomes.

By allowing the sender to continue transmitting packets without waiting for a response, similar to TCP, the channel

Algorithm 2 Algorithm for transmit

Require: number of robots N, timeout value S, number of retransmission allowed R, and routing solution CDF, C_i

Initialize map T and A
while System running **do**
  if Packet p to send **then**
    Draw random value x from $U[0, 1]$
    Transmit p to robot t for which $x \in C_{i,t}$
    Add t_{now} to T and 1 to A for id of p
  end if
  if Confirmation packet received **then**
    for Each id in packet payload **do**
      Remove id from T and A
    end for
  end if
  for all (j, t) in T **do**
    if $t <$ current time - S **then**
      if $A[j] < R$ **then**
        Draw random value x from $U[0, 1]$
        Transmit p to robot t for which $x \in C_{i,t}$
        Update $T[j]$ with t_{now} and increment $A[j]$
      else
        Remove mapping for j from T and A
      end if
    end if
  end for
end while

utilization is higher than if the sender waited for transmission confirmation for each packet. However, by using a fixed response window size of M and not a dynamic window size, such as in TCP, channel utilization again increases.

IV. Experimental Validation

We present a series of experiments in order to compare the performance of regular (non-robust) and robust routing when applied to the maintenance of network integrity while supporting a single sensing robot that is moving through an environment and transmitting data to a fixed base station. While multiple scenarios are presented, a common feature is that the sensing robot is not able to communicate reliably with the base station over a single communication channel. Our experimental scenarios take place in two unique environments, the 5th floor of the Levine building and the 5th floor of the Graduate Research Wing of the Moore building, both at the University of Pennsylvania.

We begin by comparing two routing solutions, a non-robust and a robust routing solution. The non-robust routing solution is derived from a Linear Problem (LP) and is given by,

$$\begin{aligned} \boldsymbol{\alpha}(\mathbf{x}) = \arg\max_{\boldsymbol{\alpha}, a_\Delta} \quad & a_\Delta \\ \text{st} \quad & a_i(\boldsymbol{\alpha}, \mathbf{x}) \geq a_i + a_\Delta, \end{aligned} \tag{4}$$

where $R_{ij}(\mathbf{x})$ is replaced with the $\mathrm{E}[R_{ij}(\mathbf{x})]$ in the calculation of $a_i(\boldsymbol{\alpha}, \mathbf{x})$. Notice that the value of ϵ and the channel variance do not appear in the problem because the solution is only concerned with the expected value of the channel reliability, while the robust routing solution we use is given by the solution to (3) and maximizes the probability of satisfying end-to-end rate requirements. The main drawback of only using $\mathrm{E}[R_{ij}(\mathbf{x})]$ in the LP is that if fading or another phenomenon cause the realized channel reliability for a single link to be much less than the expected value the entire network is likely to suffer. This leads to a routing solution that is not resilient to single link failures and as such is considered non-robust

Due to the limited number of *Scarab*s available the experiments were limited to a maximum of 5 robots. This is enough robots to demonstrate the benefits of robust routing over non-robust routing. In order to more fully show the benefits of MCTP we have also included simulations showing the benefits of MCTP over a Simple ACK algorithm as the number of robots grows or the input data rate increases.

A. Scarab Platform

For this paper we use *Scarab*s [17], a custom built robot designed at the University of Pennsylvania, as our robotic platform. The newest version of the *Scarab* consists of a 30 meter Hokuyo Scanning Laser Rangefinder, two Robo Claw 5 amp Motor Controllers, and a computer containing an Intel i5 3.8 GHz processor, 4 GB of RAM, and a 60 GB SSD hard drive with a full installation of Ubuntu 12.04 LTS. The ROS development environment, specifically Fuerte, is also installed. An image of a standard *Scarab* can be seen in Fig. 1. For wireless communication between *Scarab*s we use the Digi International XBee transceivers. These modules allow the user to control frequency and power. The XBee radios are capable of transmission on 16 evenly spaced channels in the 2.4 GHz spectrum. The XBee radio also allows for 5 discrete power levels, ranging from -10 dBm to 0 dBm. The XBee transmits data via a fixed packet size of 100 bytes, with a preamble the result is an effective payload size of 90 bytes for each transmission.

As shown in Fig. 1 each *Scarab* in these experiments contains 4 XBees. Each Xbee is configured to transmit at 0 dBm to allow for maximum distance between robots. Additionally, each XBee is responsible for communication on a different frequency. The frequencies chosen are evenly spaced to allow for maximum signal isolation between radios. This allows for the communication between one pair *Scarab*s to not interfere with communication between another pair of *Scarab*s, which is important as neither routing solution takes into account packet collisions.

B. 4 Robot Experiment in Levine

In this section we detail an experiment that compares robust routing to non-robust routing. The purpose of this experiment is to show that a system using the robust routing solution experiences a more reliable and consistent end-to-end reliability than a non-robust routing solution.

The setting for this experiment is the Levine building. This experiment consists of three runs for each routing solution,

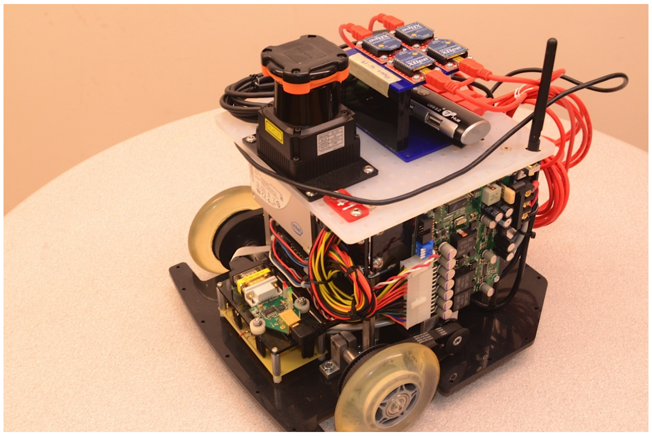


Fig. 1: The newest generation of the *Scarab*s. The XBees are mount on top of the platform behind the Hokuyo.

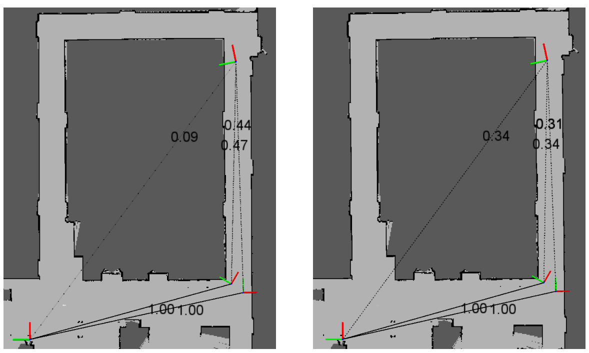
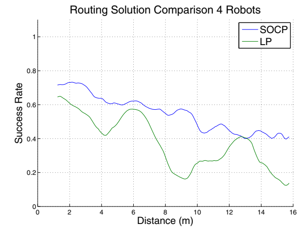


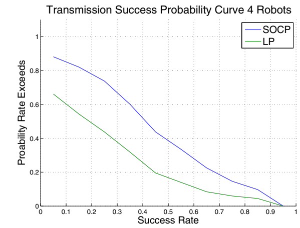
Fig. 2: LP (left) and SOCP (right) routing Solution for the 5^{th} floor Levine. The LP solution relies heavily on the line-of-sight paths, this allows for maximum data rates but does not mitigate fading.

that results in six distinct runs. For the path planning portion of each run, we use the system described in section II with the network integrity check derived from the solution to (3), for robust routing or (4), for non-robust routing. Therefore, we know only that the paths lie inside the feasible space for the given problem formulation. In these experiments a_{min} is interpreted as the probability of successful end-to-end transmission between the sensing *Scarab* and the access point. We set the values for the minimum end-to-end reliability and probability tolerance for the routing problems to $a_{min} = 0.2$ and $\epsilon = 0.8$. Also, for this building, we model fading as having zero mean and variance of 32 dB^2.

As the *Scarab*s move through the environment, the routing solutions are computed in real time and are transmitted to the *Scarab*s. This is done in order to obtain the optimal routing solution for the current configuration, as well as to remove the need for inter-robot coordination. During the execution of the runs, we use UDP to show the performance of robust routing compared to the performance of non-robust routing when using an unreliable protocol. This helps us interpret the benefits of our protocol when it is added to the system, as discussed in further detail in section IV-E. Additionally, the sensing *Scarab* outputs data at a constant rate of 1 kilobyte



(a) Percentage of successfully received packets using UDP. The robust solution provides more reliable performance while the non-robust solution varies greatly.



(b) The probability that the routing solution will provide a data rate greater than or equal to the requested value.

Fig. 3: Levine 5^{th} floor experiments, $N = 4$.

per second, or 10 packets per second. The speed of the *Scarab*s is limited to 10 cm/sec to allow for the collection of at least one data point at every centimeter along their path.

Figure 2 shows the environment on the 5^{th} floor of Levine, as well as a final configuration for one of the experimental runs. There are 3 *Scarab*s initially near the access point in the lower left corner of the building. As the experiment unfolds, the sensing *Scarab* moves to the right along the 8 meter hallway until reaching the intersection. The *Scarab* then turns left and proceeds through the 15 meter hallway, where line-of-sight with the access point is broken. The *Scarab* eventually reaches the top right corner of the building. Meanwhile, the other 2 *Scarab*s move in order to support the minimum data rate requirement.

In Fig. 2 we can see that for the same configuration the non-robust and the robust routing solutions differ. Specifically, the ratio of the data being sent back directly to the access point is 0.09 for the non-robust solution and 0.34 for the robust solution. This vast difference is due to the fact that the robust solution takes into account that there is an equal probability of a successful transmission when using the link that connects directly back to the access point, as there is when using the two-hop link that connects through either support *Scarab*.

Figure 3a shows the average end-to-end successful packet transmission using UDP for non-robust routing and robust routing as a function of distance. In Fig. 3a we can clearly see that the robust routing solution consistently outperforms the non-robust solution. The benefits of the robust routing solution are also evident in the absence of large variations, which are seen in the non-robust routing data. Specifically, the standard deviation of the robust solution is 0.1081 while the standard deviation of the non-robust solution is 0.1536 over the path.

Another interesting outcome of these experiments is shown in Fig. 3b, which plots the probability of exceeding a given data rate based on the empirical data collected. For this figure, the closer the line is to the upper right corner the better. The first important item from this plot is that the robust routing solution has a 0.78 probability of exceeding a data rate of 0.2. This closely matches the input parameters

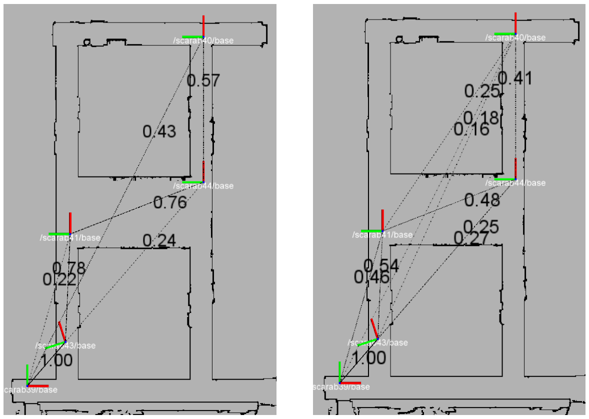
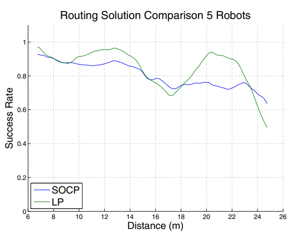
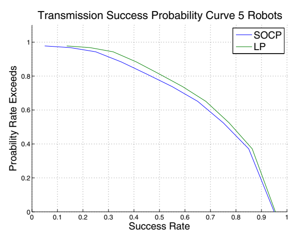


Fig. 4: LP (left) and SOCP (right) routing Solution for the 5^{th} floor GRW. The LP solution relies heavily on the line-of-sight paths, this allows for maximum data rates but does not mitigate fading.



(a) Average ratio of successfully received packets using UDP. The robust solution provides more reliable performance while the non-robust solution varies greatly.



(b) The probability that the routing solution will provide a data rate greater than or equal to the requested value.

Fig. 5: GRW 5^{th} floor experiments, $N = 5$.

of $a_{min} = 0.2$ and $\epsilon = 0.8$. Also note that, even though the formulations were done with $a_{min} = 0.2$, the resulting robust and non-robust routing solutions provide data rates that exceed 0.4 with probability 0.52 and 0.25, respectively. In this scenario it is obvious that the robust routing formulation greatly outperforms the non-robust routing.

C. 5 Robot Experiment in GRW

In this section again we detail an experiment that compares robust routing to non-robust routing. The purpose of this experiment is to confirm the benefits of robust routing and show that they are not limited to one environment.

This experiment takes place in the Graduate Research Wing and consists of two runs for each routing solution. The path planning portion of this experiment is not the RRT method, but instead performed manually in order to optimize the number of line-of-sight links. The RRT method is not used in these experiments because it only guarantees feasibility, rather than any form of optimality. This experiment is modeled after the experiment described in section IV-B. The only differences are that $a_{min} = 0.3$ and $\epsilon = 0.75$ are used for the routing problem parameters.

As the *Scarab*s move through the environment, the routing solutions are again computed in real time and are transmitted to the *Scarab*s. UDP is used as the communication protocol. This will help us interpret the benefits of our protocol when it is added to the system.

Figure 4 shows the environment on the 5^{th} floor of the Graduate Research Wing, as well as a final configuration for one of the experimental runs. There are 4 *Scarab*s initially near the access point in the lower left corner of the building. For this experiment, the sensing *Scarab* moves to the right along an 11 meter hallway until reaching the intersection. It then turns left and travels down the 25 meter hallway, eventually reaching the top right corner of the building. Meanwhile, 3 *Scarab*s move in order to support the minimum data rate requirement.

In Fig. 4 we can see that for the same configuration the non-robust and the robust routing solutions again differ. The differences are similar to those in Fig. 2, most notably the increased number of links out of the sensing *Scarab* in the robust solution compared to the non-robust solution.

Figure 5a shows the average end-to-end successful packet transmission using UDP for non-robust routing and robust routing as a function of distance. In Fig. 5a we can see that the non-robust routing solution consistently outperforms the robust solution. This is in contrast to the results in section IV-B, but again the rapid fluctuations are present in the non-robust routing solution, but not in the robust routing solution.Additionally, the robust routing solution again has a lower standard deviation in comparison to the non-robust solution, which are 0.0837 and 0.120, respectively.

The over performance of the non-robust solution in this experiment set, when compared with the results in Fig. 3a, is due to the *Scarab*s following the same path for both routing systems. As mentioned above, one drawback of the method outlined in section II is that the resulting path and routing solution only require feasibility. That fact, in conjunction with the larger feasible space for the non-robust solution, implies that the resulting solutions for the non-robust system will be sub-optimal to those found via the robust routing system. One remedy is to increase the value of a_{min} for the non-robust system. Unfortunately the points that are removed from the feasible space will not be the same as those removed by the SOCP constraints.

In Fig. 5b we again plot the probability of exceeding a given data rate based on the empirical data collected. The first important result from this plot is that the robust routing solution has a 0.96 probability of exceeding a data rate of 0.3. This far exceeds the input parameters of $a_{min} = 0.3$ and $\epsilon = 0.75$. Also note that the robust routing solution provides a higher probability for a given data rate, up until 0.65, at which point the large fluctuations assist the non-robust solutions data rates.

D. Protocol Experiment

The previous two experiments rely on an unreliable communication protocol to compare the performance of non-robust and robust solutions to the network routing problem

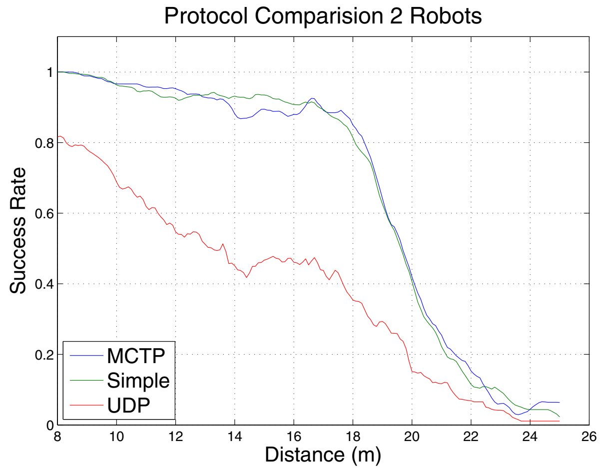


Fig. 6: Plot for a single *Scarab* moving away from the access point and turning a corner at 11 meters using three transmission schemes. As expected the two schemes that utilize confirmations show much more reliable packet transmission. Note that MCTP achieves the same performance as Simple ACK with 5 times less confirmation packets transmitted.

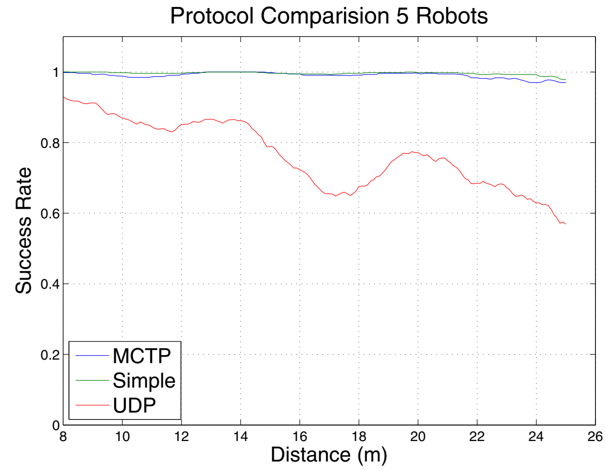


Fig. 7: Plot for a team of 4 *Scarab*s moving out while following the same paths as in Fig. 5.

by examining the reliability of end-to-end communication across a mobile network. However, an underlying assumption of the robust routing algorithm is that point-to-point communication is reliable but the rate of communication varies with channel quality. As discussed above, traditional methods to enable reliable point-to-point communication such as TCP are not suitable for highly variable and unreliable channels. This led us to develop the lightweight MCTP protocol which is experimentally demonstrated here.

The setup for this experiment involves a sensing *Scarab* moving away from the access point and traveling through the environment where direct line-of-sight is not always possible. Specifically the *Scarab* moves out from the access point and turns a corner, 11 meters away, before continuing to the end of the hallway. It is the same environment as the 5 robot experiments in section IV-C. While traveling the *Scarab* uses one of three transmission protocols: UDP, Simple ACK, and MCTP. Simple ACK is a lightweight protocol in which a response is given for every packet received. The meaningful parameters are T and R, which have the same meaning as the MCTP parameters.The parameters in this experiment are $N = 1$, $M = 5$, $T = 0.5$ sec, $S = 1.0$ sec, $R = 4$, and $C_1 = 1.0$ with data generated at 10 Hz.

It can be seen in Fig. 6 that for point-to-point connection, the performance of the UDP protocol degrades very quickly as the *Scarab* moves away from the access point. This is in contrast to the Simple ACK and MCTP protocols which work very well out to a reasonable distance. Particularly we see the rate for the confirmation protocols start at 1.0 and slowly drop to 0.85, at 18 meters. At this point the performance of the confirmation protocols begin to noticeably degrade with a precipitous drop from 0.85 to less than 0.1 in 4 meters.

E. Full System Validation

In our final experiment we incorporate our communication protocol, MCTP, with robust routing for a full system validation. This experiment seeks to show that by adding our communication protocol to the robust routing solutions we can achieve near loss-less end-to-end communication between the sensing *Scarab* and the access point, even when direct communication is not possible. For this experiment the same parameters and paths are used as in section IV-C, with only the communication protocol changing. The parameters used for the comfimation protocols are the same as those used in section IV-D.

When we incorporate the MCTP protocol the results immediately show improvement. As it can be seen in Fig. 7, when a confirmation protocol is used, the success rates increase dramatically. Using the Simple ACK and the MCTP protocols we see almost loss-less communication, even beyond 24 meters. The key result of this experiment is that the MCTP protocol, which is only sending confirmation messages for every 5 packets, has approximately the same reliability as the Single ACK protocol, with less than 0.025 maximum deviation between the two. By using MCTP with these parameters, compared to the Simple ACK, we can allow up to 5 times as many robots on the same confirmation channel. This means that for every robot added to the team, only 1.2 effective channels are required compared to 2 channels for Simple ACK.

F. Large scale system simulation

The performance advantage of MCTP relative to a simple ACK system is expected to become more marked as we increase the number of robots in the system. Due to the limited availability of Scarabs we perform simulations to quantify these advantage. In the simulations we have created a network of robots transmitting data over point-to-point links using the MCTP and simple ACK protocols for confirmation. In these simulations each robot generates

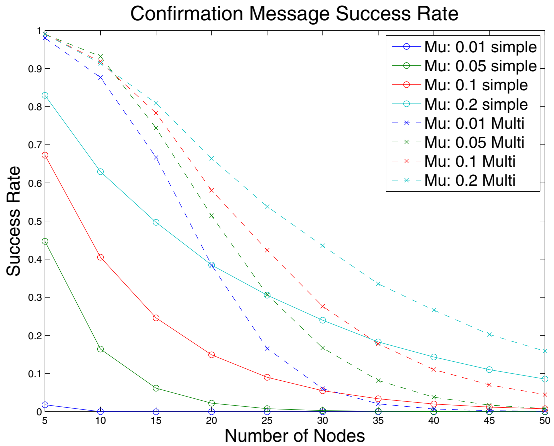


Fig. 8: Simulated evaluation of MCTP for large-scale systems. Notice how MCTP outperforms a Simple-ACK protocol as the number of robots grows or the data rates increases.

a packet of data every τ seconds, where τ is drawn from an exponential distribution with parameter μ. The destination for each packet is uniformly distributed over the other robots in the network. This allows us to vary both the number of robots in the network, as well as the average data input rate for each robot. These simulations assume that every data packet is successfully received and only the confirmation packets can be lost due to collision.

The results are shown in Fig. 8, which plots the average success rate of confirmation packets transmission as a function of the number of robots in the system, for a given value of μ. Since this is the average success rate the closer the value is to 1 the better the system is performing. The main item to notice is the wide gap in performance between Simple ACK and MCTP across all combinations of μ and the number of robots. This is directly attributable to fewer confirmation packets being sent, since a single MCTP message contains much more information than a Simple ACK message.

As a conclusion the MCTP protocol outperforms a Simple ACK system throughout our simulations. Specifically, as the number of robots increase the drop in performance is much more gradual for MCTP compared to Simple ACK, and the same relationship is seen when the average input data increases. This highlights the benefit of MCTP as the number of robots in the team grows or the input data rates increase.

V. Conclusions

We propose a software implementation that allows for communication between robots in an ad-hoc wireless network, while balancing reliability of transmission and effective speed. By building on the previous work we show the benefits derived by robust routing do extend beyond RSSI measurements, specifically the removal of wild fluctuations when teams move through an environment.

The results in this paper show that by using the robust routing system we are able to deliver a higher probability of successful packet transmission, which leads to less variable and therefore more reliable end-to-end data rates. Building on the reliability of robust routing we also develop a transmission protocol that provides the reliability of TCP while also providing the channel utilization of UDP. Providing a more reliable end-to-end data rate greatly increases the scenarios where autonomous robotic teams are useful. This is due to the ability of the team to move, while still providing a minimum quality of service.

Our future work will focus on an adaptive approach to the motion planning portion of the system in section II, in order to allow for a team of robots to operate in a more dynamic fashion. We will also explore decentralized motion planning algorithms to preserve operational efficiency as the team size grows.

References

[1] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 693–703, 2007.

[2] M. Zavlanos and G. Pappas, "Potential fields for maintaining connectivity of mobile networks," *Robotics, IEEE Transactions on*, vol. 23, no. 4, pp. 812–816, 2007.

[3] K. Savla, G. Notarstefano, and F. Bullo, "Maintaining limited-range connectivity among second-order agents," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 187–205, 2009.

[4] M. Schuresko and J. Cortes, "Distributed motion constraints for algebraic connectivity of robotic networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pp. 5482–5487, 2008.

[5] D. Spanos and R. Murray, "Motion planning with wireless network constraints," in *American Control Conference, 2005. Proceedings of the 2005*, pp. 87–92, 2005.

[6] O. Tekdas, W. Yang, and V. Isler, "Robotic routers: Algorithms and implementation," *Int. J. Rob. Res.*, vol. 29, pp. 110–126, Jan. 2010.

[7] H. Lundgren, E. Nordstrom, and C. Tschudin, "The gray zone problem in ieee 802.11b based ad hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, pp. 104–105, June 2002.

[8] J. Fink, N. Michael, A. Kushleyev, and V. Kumar, "Experimental characterization of radio signal propagation in indoor environments with application to estimation and control," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2834–2839, 2009.

[9] J. Fink, A. Ribeiro, and V. Kumar, "Robust control of mobility and communications in autonomous robot teams," *Access, IEEE*, vol. 1, pp. 290–309, 2013.

[10] J. Fink, *Communication for teams of networked robots*. PhD thesis, University of Pennsylvania, 2011.

[11] A. Ewerlid, "Reliable communication over wireless links," in *Nordic Radio Symposium 2001 (NRS 01*, pp. 3–5, 2001.

[12] G. Xylomenos and G. Polyzos, "Tcp and udp performance over a wireless lan," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 439–446 vol.2, 1999.

[13] R. Bruno, M. Conti, and E. Gregori, "Throughput analysis and measurements in ieee 802.11 wlans with tcp and udp traffic flows," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 2, pp. 171–186, 2008.

[14] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "Tcp westwood: end-to-end congestion control for wired/wireless networks," *Wirel. Netw.*, vol. 8, pp. 467–479, Sept. 2002.

[15] S. M. ElRakabawy, A. Klemm, and C. Lindemann, "Tcp with adaptive pacing for multihop wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05, (New York, NY, USA), pp. 288–299, ACM, 2005.

[16] J. Nagle, "Congestion control in ip/tcp internetworks." `http://tools.ietf.org/html/rfc896`. Accessed: 2014-06-16.

[17] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multirobot teams," *Robotics Automation Magazine, IEEE*, vol. 15, no. 1, pp. 53–61, 2008.