 ISBN 978-3-8007-3343-9 © VDE VERLAG GMBH

Demonstration of an OLA-based Cooperative Routing Protocol in an Indoor Environment

Yong Jun Chang, Haejoon Jung, and Mary Ann Ingram
School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, Georgia 30332-0250, USA
Email: {yongjun.chang, hjung35, mai}@gatech.edu

***Abstract*—Concurrent cooperative transmission (CCT) is a cooperative transmission (CT) technique, also known as distributed transmit diversity, where a collection of single-antenna nodes induce a signal-to-noise-ratio advantage in a receiver by transmitting multiple copies of the same message, at the same time, in multiple diversity channels. CCT has been considered in many physical layer system studies including realistic demonstrations, which presented cooperative diversity gain in terms of error rate improvement or range extension. However, there have been very few attempts to evaluate CCT-based network routing schemes on a testbed with consideration of the overheads and challenges that CCT poses to the higher layers.**

In this paper, we compare the OLAROAD cooperative routing protocol with the conventional Ad hoc On Demand Distance Vector (AODV) protocol, on an ad hoc network of software defined radios. OLAROAD is based on a physical layer that forms opportunistic large arrays (OLAs). We evaluate the performance of the protocols on a linear network topology in a typical office building in terms of hop count, round trip time, packet delivery ratio and route discovery time, so that the advantages and disadvantages of CCT-based routing algorithm may be observed.

I. Introduction

Cooperative transmission (CT) is a physical layer wireless communication scheme in which spatially separated wireless nodes collaborate to form a virtual array antenna for the purpose of increased reliability. In this paper, we define Concurrent Cooperative Transmission (CCT) as a special non-coherent form of CT in which cooperating nodes transmit copies of the packet at approximately the same time through two or more orthogonal channels. A receiver that can combine these copies gets diversity gain in fading and shadowed environments, and if the transmit power of the cooperators is not reduced, then there is also array gain. Recent physical demonstrations of CCT in indoor environments include preamble-based transmit synchronization [1] and range extension [2]. This paper reports the first experimental comparison between a network-layer or routing protocol designed for the CCT physical layer and the Ad hoc On-demand Distance Vector Routing (AODV) protocol, which operates on a conventional non-CT physical layer.

The CCT routing protocol is based on the OLA Routing On-Demand (OLAROAD) protocol, previously proposed by the authors [3] [4], which uses a particular form of CCT called the Opportunistic Large Array (OLA) [5]. In the basic decode-and-forward OLA scheme, all relays that decode a packet from a source or another OLA and which have not transmitted the packet before, will shortly thereafter relay the packet. The group of relays that transmit the packet together forms the OLA. The opportunistic aspect is that OLA members are generally not known a priori, since membership depends on which nodes pass cyclic redundancy check (CRC). The definition of OLA does not imply any particular type of modulation or method for achieving diversity, for example, orthogonal frequency division multiplexing (OFDM) with distributed space-time block code (STBC) could be used [6]. The implementation of OLAs in this paper uses binary frequency shift keying (BFSK), orthogonal subcarriers for diversity, and non-coherent equal gain combining. OLAs provide fast and contention-free broadcasts with a low overhead in implementation [7]. OLAs are also effective for unicast since OLA-based cooperative routes have lower latency, higher delivery rates, and mitigate the network partition problem [3]. Moreover, in dense networks, OLA-based routing schemes are highly resistant to mobility [4].

CCT is not compatible with conventional routing and medium access control (MAC) protocols or any off-the-shelf radios. Therefore, to compare the protocols we implement both protocols on a network of SDRs. In this paper, we focus on the network layer as we compare OLAROAD with AODV on an ad hoc network of SDRs in a practical indoor environment. Also, this paper shows the need of a second handshake, which does not exist in the original version of OLAROAD in [3], and clarifies the challenges in the CCT-based routing.

II. Related Works

Most existing studies on CT-based ad hoc routing use a conventional (non-CT) multi-hop route as the "primary" route, where the nodes in the primary route, which are called the "primary" relays, recruit their cooperators [8] [9] [10]. The CT routing techniques in this category can limit the additional overhead to use CT and maintain a certain level of compatibility with the conventional system structure, but they must first perform conventional routing in a given topology. Therefore, the range extension strategy of CT cannot be exploited as a solution to overcome the network partition problem.

The authors gratefully acknowledge partial support for this research from the National Science Foundation, grant number CNS-1017984.

In contrast, OLAROAD is not limited by the existence of the conventional non-CT route, so it fully utilizes the wireless broadcast advantage and exploits cooperative range extension [3]. A similar cooperative transport architecture is previously proposed by Ramanathan [11], but the specific routing method was not provided. The controlled broadcast region (CBR) is a cooperative route based on the barrage relay network (BRN) [12] [13] [14]. As an OLA-based method, BRN is also an autonomous cooperative communication scheme, however BRN differs from our OLA implementation in that it uses a certain phase dithering method in its physical layer, has a different cooperator selection mechanism, and relies on network time synchronization or GPS [13]. Also, the OLAROAD implementation in this paper is different from the CBR construction, because OLAROAD has a second handshake for the route confirmation, which CBR construction lacks. Even though some demonstration of CT-based broadcasts and unicasts have been reported in [15] [1] [2], to the best of our knowledge, there has not been an experimental study comparing an OLA-based routing protocol with a conventional routing protocol such as AODV on a testbed.

III. OLAROAD PROTOCOL

In this section, we describe the implementation of OLAROAD, including the modifications of its original design.

A. Original Protocol Description

We now explain how OLAROAD builds the cooperative route for a given source and destination pair. The original version of OLAROAD [4] builds the route with a single control message handshake consisting of a route request (RREQ) and a route reply (RREP) for one unicast flow. The simulation result in Fig. 1 illustrates how OLAROAD creates the cooperative route, where the source is indicated by the red square in the center and the destination is denoted by the yellow star on the right of the source. Also, the gray dots and the black dots indicate the nodes not in the route and the nodes in the route, respectively. As described in [4], the source node initiates the route discovery by the OLA broadcast of a RREQ. This broadcast covers the network in successive OLA hops. The nodes that decode the ith hop form the ith Downstream Level (DL^i) indicated by the thick black concentric circles in Fig. 1. When the relay nodes forward the RREQ message, they increment the hop-count field (the downstream level) in the RREQ, so that each node identifies its downstream level. In this way, the destination also achieves the cooperative hop-counts from the source to itself, e.g., the destination downstream level (DDL) is '3' in Fig. 1.

Upon reception of the RREQ, the Destination triggers an OLA-broadcast of a RREP message that contains the destination downstream level (DDL) and the upstream hop-count field (UL^j), which is incremented by the nodes relaying the RREP message similar to the RREQ transmission. However, unlike the forward path set-up through the RREQ, the nodes can forward the received RREP only when the sum of its downstream level (i) and upstream level (j) is less than or equal to the destination downstream level (DDL) plus one as given by

$$i + j \leq DDL + 1. \tag{1}$$

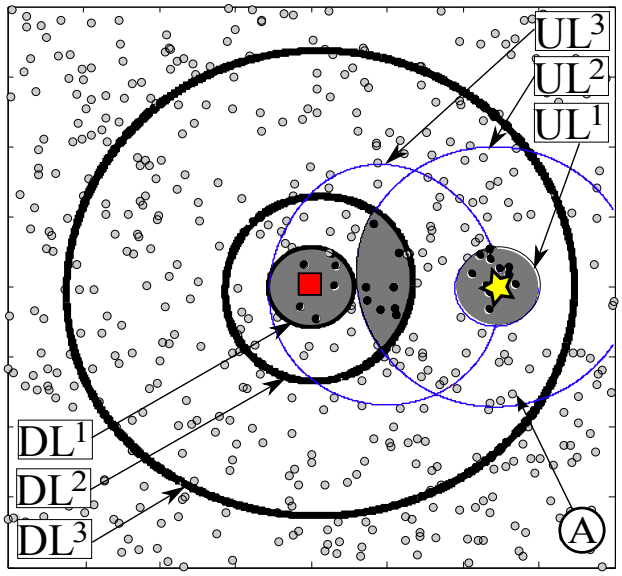


Fig. 1. An example of the OLAROAD routing

In Fig. 1, for example, the node Ⓐ is in DL^3 and in UL^2, then the sum is 5, which is greater than 4 (=DDL+1=3+1), then it cannot forward the received RREP. On the other hand, the nodes that satisfy the condition, indicated by the black dots, forward the RREP message. Following this rule, this RREP broadcast also creates the upstream level boundaries shown by the thin blue contours in Fig. 1, but unlike the downstream level boundaries, they are not concentric because of the cooperator selection. The nodes that satisfy (1) compose the "cooperative route." In the DATA phase, the source sends the data as an OLA broadcast along the cooperative route. In other words, only nodes that participated in the RREP are allowed to relay the data packets.

B. Protocol Modification - Second Handshake

The previous version of OLAROAD before this paper finishes the route construction when the RREP reaches the source. Therefore, the data transmission phase begins right after the single handshake (RREQ-RREP) has been completed successfully. Similar to OLAROAD, the BRN unicast routing also consists of a single handshake process (RTS-CTS in their terms). However, the cooperative route built through the single handshake might not work in some cases because OLAs of the RREQ phase (concentric rings) are not the same as the OLAs of the cooperative route (limited to the gray shaded areas in Fig. 1, for example), and therefore the ranges of the OLAs are different. For example, in Fig. 1 the number of the cooperators at the second downstream level in the DATA phase is 8, which is about 25% of the cooperators in the RREQ phase. For the

same reason, the data transmission takes 4 hops, while the RREQ takes only 3 hops to reach the destination. Moreover, it is also possible that the data transmission fails even if the RREQ-RREP handshake successfully finishes. Therefore, we added one more handshake process consisting of two control messages - Route Confirm (RC) and Route Confirm Acknowledgement (RCACK). The second handshake process is triggered by the source through the RC message broadcast when it receives the RREP in the first handshake process. Because the purpose of the second handshake is just to confirm that the cooperative route actually works, this packet has a simple format including Packet type, Source address, Destination address, and Broadcast ID. Only the nodes in the cooperative route can forward the RC packet. After the destination receives the RC packet, it sends a RCACK message back to the source. This message has the same fields in the RC, but its Broadcast ID is managed by the destination node, not the source node. This is a similar way to handle the destination sequence number in the control packets of AODV.

If the source receives the "RCACK" packet, then it finishes the route discovery process and starts the data packet transmission. If the source cannot receive the RREP in the first phase after sending RREQ or cannot receive the RCACK in the second phase after sending the RC packet during a certain amount of time, then source will resend a new RREQ message with different parameters such as enabling some techniques such as ganging of levels, transmit power control, and step-size control proposed in [3].

C. Issues in OLAROAD

Because the cooperative transmission has an SNR advantage through array and diversity gains, OLAROAD can guarantee the better link quality, shorten the number of hops, and mitigate the network partition problem in the conventional single-input-single-output (SISO)-based routing by the range extension property. However, there is a trade-off between the improvement and the interference increased by the CT. Aside from the interference to other flows, OLAROAD has the following issues caused by its range extension property in a dense network, which occur in BRN unicast routing as well.

1) Intra-flow interference: Suppose that the source has multiple packets to send to the destination node. If the number of the packets is very large, as in a video file transfer, then, the key factor that decides the throughput at the destination is the inter-packet delay or the packet insertion period between two consecutive data packets. For example, in a conventional TDMA-based multi-hop network using a single non-CT path comprising half-duplex nodes, the minimum inter-packet delay is three time slots, since each node can overhear both the previous hop and next hop transmissions. However, in CT network, it is possible that there are many nodes in a certain OLA. For example, in Fig. 2, the cooperative data transmission at DL^3 which is desired to reach DL^4, goes beyond the neighboring hop DL^2 boundaries and interferes with the next data packets reception at DL^1. In this case, the

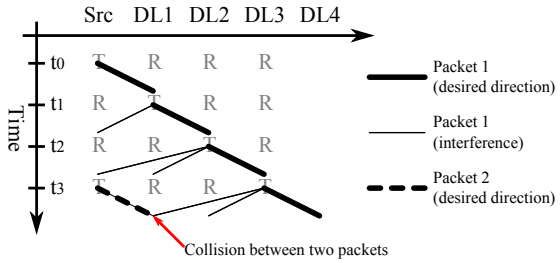


Fig. 2. The intra-flow interference TDMA time diagram

inter-packet delay should be extended to more than or equal to four TDMA time slots. This is the simplest way to solve the intra-flow interference problem, however, it is not a good solution because it lowers the data rate.

2) RREP failure: Another challenge created by the range extension of CT is RREP failure. In a dense network, the RREQ message is forwarded by an OLA-broadcast, which forms OLAs according to concentric rings, as shown in Fig 1. These rings expand their size rapidly as the hop-count increases, if the node degree (the number of nodes in the decoding range of a single node) is high enough. In that situation, the backward RREP transmission cannot go over the very long last downlink OLA-hop and cannot reach the nodes in the next uplink level. BRN unicast solves this problem by increasing the right-hand side in (1) thereby creating much larger upstream (RREP) OLAs, but this solution might cause more intra-flow interference and very low throughput if longer inter-packet delay must be used. Also, it can cause more interference to other flows (other source and destination pairs). Alternatively, step-size control in the RREQ phase has been proposed in [3]. However, because our network is relatively small, and our network topology is nearly linear (step size does not grow in an unbounded fashion in linear networks [16], [17]), we have not implemented step-size control in this paper.

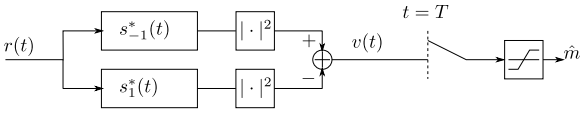
IV. Network Model

The OLAROAD routing protocol requires a different type of physical and data/link layer protocol from the SISO-based ad hoc routing protocol (i.e., AODV). In this section we describe the model for each layer to implement AODV and OLAROAD. For both routing protocols we assume that all wireless nodes have only a single antenna and have limited transmit power.

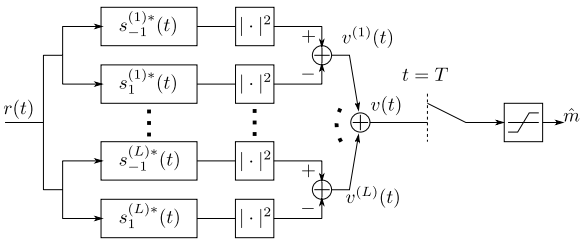
A. Physical Layer

For the SISO communication link used in the AODV protocol, BFSK with non-coherent reception (i.e., envelope detection) are used in this paper. FSK modulation enables a power efficient transmitter and a simple, low-cost receiver. The baseband signal of BFSK for single node transmission is represented as

$$s_m(t) = \sqrt{\frac{2\mathcal{E}}{T}} e^{j2\pi m(\triangle f/2)t},\ m \in \{-1, 1\},\ 0 \le t \le T, \quad (2)$$



(a) BFSK demodulator with envelope detection



(b) BFSK demodulator with EGC combining

Fig. 3. Block diagram of single BFSK envelope detector and Equal Gain Combiner

where m denotes the symbol index, $\mathcal{E}$ is transmit symbol energy, and T denotes the symbol duration. The signal received through a complex flat fading channel, whose gain and propagation delay are denoted by h and τ_l respectively, is given by

$$r(t) = h \cdot s_m(t-\tau) + w(t), \tag{3}$$

where $w(t)$ is complex white Gaussian noise with power spectral density N_0. In general, the decision rule of an envelope detector can be written as

$$\hat{m} = \underset{m\in\{-1,1\}}{\arg\max} \left| \int r(t) s^*_m(t) dt \right|. \tag{4}$$

In BFSK signal, the decision rule also can be expressed using a soft decision value $v = |\int r(t)s^*_1(t)dt| - |\int r(t)s^*_2(t)dt|$. The decision rule can be rewritten as $\hat{m} = \text{sgn}(v) = v/|v|$. The SISO BFSK receiver model is shown in Fig. 3(a).

In our implementation, to achieve transmit diversity, each node is allocated one sub-frequency among L orthogonal sub-frequencies. For a while, let us assume that L different nodes choose L different sub-frequencies and they can synchronize their transmission times. The baseband signal of the node that uses sub-frequency l is written as

$$s^{(l)}_m(t) = \sqrt{\frac{2\mathcal{E}}{T}} e^{j2\pi\{m(\triangle f/2)+(l-1)\cdot\delta\}t}, \; 1 \le l \le L, \tag{5}$$

where δ is frequency separation of two adjacent sub-frequencies. The received signal superimposing L orthogonal transmissions through complex flat fading channel gains $h^{(l)}$ and propagation delays $t^{(l)}$ is given by

$$r(t) = \sum_{l=1}^{L} h^{(l)} \cdot s^{(l)}_m(t-\tau^{(l)}) + w(t). \tag{6}$$

Assuming that δ is selected such that all sub-channels are orthogonal, equal gain combining (EGC) can be achieved using multiple pairs of BFSK envelope detectors as shown in 3(b). The decision rule of EGC combiner can be written as

$$\hat{m} = \underset{m\in\{-1,1\}}{\arg\max} \sum_{l=1}^{L} \left| \int r(t) s^{(l)*}_m(t) dt \right|. \tag{7}$$

Similar to single transmission case, the decision rule also can be expressed using a soft decision value in which $v = \sum_l v^{(l)} = \sum_l |\int r(t)s^{(l)*}_1(t)dt| - \sum_l |\int r(t)s^{(l)*}_2(t)dt|$.

In order to synchronize transmission times of the cooperators in the absence of a globally synchronized clock, the cooperators use the source message as a trigger signal [18]. The cooperators estimate the start of the packet (SOP) time and add a fixed amount of time, "T_{proc}," to schedule their transmission time. In [1] the authors showed that the transmission time errors of this scheme are on the order of 100 nanoseconds in an indoor environment and their rms values converge with hop index.

B. Data Link Layer

The data Link Layer consists of the Media Access Control (MAC) and Logical Link Control (LLC) sub-layers. In a SISO communication system, simultaneous transmissions from neighboring wireless nodes constitute a collision and decoding is not possible. Typically, most MAC layer standards rely on physical carrier sensing (e.g., carrier sense multiple access with collision avoidance (CSMA/CA) in IEEE 802.11 and IEEE 802.15.4) while a virtual carrier sensing (e.g., RTS/CTS in IEEE 802.11) is optional. In IEEE 802.11, RTS/CTS is used only when a data packet size is larger than "RTS/CTS threshold." The IEEE 802.11 specification [19] shows that the peak throughput can be achieved when the RTS/CTS threshold is around 250∼500 octets. In this paper we avoid implementing RTS/CTS by using a small, 100 bytes-sized data packet. For link reliability, which is a function of LLC, a data acknowledgment (ACK) scheme is used. After a receiver correctly receives a data packet, it sends an acknowledgment (ACK) to the transmitter. If an ACK is not received by the transmitter within a short timeout period, the transmitter attempts to retransmit the data packet up to seven times [20]. If no ACK is received after seven retries, the data packet is abandoned and an error is reported to the network layer.

C. Network Layer

We assume AODV for our non-CT routing protocol [21]. We follow all the requirements in [21], such as the message formats. The AODV control messages are sent to port 654 using the User Datagram Protocol (UDP). Also, the configuration parameters are shown in Table I, most of which are the default values in *GTNetS* [22]. However, we change "NODE TRAVERSAL TIME" corresponding to the processing time of the testbed, and "NETWORK DIAMETER" corresponding to the network size in the experiments. Also we implement local route repair, but only for the nodes that are in the route [20].

OLAROAD handles the routing process using UDP which has the port number of 655. As in AODV, the originating node in OLAROAD sets the timeout for receiving a RREP after sending the RREQ. However, unlike AODV, OLAROAD has another timer for the second handshake process (RC-RCACK). If the second handshake fails, it will try the route discovery again with a new RREQ. For lack of space, we do not provide the experimental results with a single handshake in this paper,

TABLE I
CONFIGURATION PARAMETERS IN AODV IMPLEMENTATION

Parameter	Value
MY ROUTE TIMEOUT	10 sec
ACTIVE ROUTE TIMEOUT	10 sec
HELLO INTERVAL	1 sec
ALLOWED HELLO LOSS	3
RREQ RETRIES	3
TTL START	7
TTL THRESHOLD	13
TTL INCREMENT	2
NETWORK DIAMETER	13
NODE TRAVERSAL TIME	0.15 sec

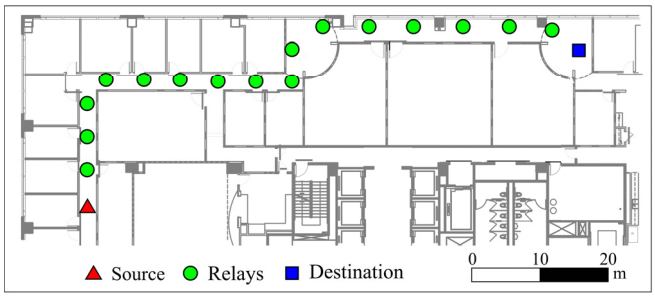


Fig. 4. The floor plan and node placement

which results in the frequent failure in the data transmission phase.

V. IMPLEMENTATION USING GNU RADIO TESTBED

Since there is no off-the-shelf radio that supports CCT, we implemented AODV and OLAROAD routing protocols using Universal Software Radio Peripheral 1 (USRP1) and GNU Radio Software-Defined Radio. The USRP1 is a hardware platform that allows a software radio to be implemented in a general purpose processor [23]. The USRP1 system consists of a radio frequency (RF) daughter board that functions as an RF front-end and a main board that has an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC), a field-programmable gate array (FPGA), and universal serial bus (USB) interfaces. The data streamed over the USB interface is in the form of 32-bit I/Q samples that consist of a 16-bit in-phase component and a 16-bit quadrature component. GNU Radio is used as a base-band signal processing module, which operates on a PC that is attached to the USRP1 via the USB interface [24]. GNU Radio is an open-source software package that is usually run on a Linux operating system. It has various signal processing blocks written in the C++ language and protocol stacks written in the Python language.

A. Time-tagging Method and CSMA/CA

In a USRP1 and GNU Radio system, a large variance of processing times caused by several random latencies makes implementation of CCT transmit time synchronization and CSMA/CA challenging [25] [18]. In [18], the authors proposed and implemented on the FPGA a time-tagging method to support transmit time synchronization. In order to implement time-sensitive MAC functions in GNU Radio systems, we adopted the approach in [25], which splits core functions of a MAC layer and implements some parts of functions on the FPGA of the USRP1.

Specifically, we modified the FGPA code to handle timing information of received and transmitted samples as well as carrier sensing and back-off. First of all, a 32-bit hardware counter coupled with a sampling clock was built on the FPGA, and multiplexer inserts the counter value, called "rx-time-tags", into the data streams. Using these rx-time-tags, the SOP time estimator records the time of the SOP. In order to transmit a packet at a specific time (i.e., SOP time + T_{proc}), a tx-time-tag is attached at the start of a transmit packet. When the FPGA code detects the time-tag in a tx data stream, it holds transmission until the counter value matches to the time-tag value. If the time-tag value is outdated, all samples are flushed until a new time-tag is detected.

A carrier sensing block is also implemented in the FPGA. If a packet should be transmitted with CSMA/CA, a CSMA-tag is attached at the start of a transmit packet similar to the tx-time-tag. When the FPGA code detects the CSMA-tag in a tx data stream, it holds transmission and performs a back-off operation. Since a random number generator in a FPGA takes large amount of logic elements (LEs), the back-off durations are randomly generated by a PC at each packet transmission and reported to the FPGA prior to sending the packet. If a medium is busy until all back-off timers are expired, the FPGA sets a register to let the PC know transmission fails.

B. Protocol Stack

All protocols except for signal processing blocks in the physical layer are written in the Python language. All protocol stack layers are clearly and distinctly separated in this implementation. A Python class of a packet consists of a set of protocol data units (PDUs) that are appended and removed as the packet moves up and down the protocol stack. The packet class also has a metadata that contains the information related to reception (e.g., received signal strength indicator (RSSI) value, SOP time, etc.) and transmission (e.g., orthogonal sub-channel, scheduled transmit time, etc.). End-to-end connections between two nodes at the transport layer are specified using a tuple of source IP, source port, destination IP, and destination port similar to the ubiquitous sockets API in Linux/Unix environments.

VI. EXPERIMENTAL RESULTS

This section presents the experiment design and the results of the two routing protocols, AODV and OLAROAD. The metrics used in the routing performance comparison are packet delivery ratio (PDR), end-to-end hop count, end-to-end round trip time, and route discovery time.

The experiment in this paper was conducted using 18 of the nodes that were described in Section V. The nodes were deployed on the fifth floor of the Centergy building of the Georgia Institute of Technology, as shown in Fig. 4. This building consists of offices and research laboratories. As shown in the figure, we placed all the eighteen nodes in the

corridors, with office rooms on the one side and laboratories on the other side, so that they form a linear network with relatively favorable propagation without much attenuation by walls. For the sake of comparison, we let the source and destination nodes, indicated by the red triangle and the blue square in Fig. 4, be at the extreme ends of the linear network. The relay nodes, represented by the green circles, are placed with uniform spacing.

We used a 2.482GHz carrier frequency and 64kbps data rate. The packet for the experiment is 128 bytes long consisting of 100 bytes of binary data, 26 bytes of data header, and 2 bytes of CRC.

We manually assign one of the four diversity channels to each node in incremental fashion, i.e., $\{1, 2, 3, 4, 1, 2, 3, \ldots\}$ from end to end[1]. With this arrangement, a cooperating set might contain two or more nodes with the same diversity channel assignment. In this case, self-fading of the same-diversity-channel transmissions is possible. However, the self-fading signals still undergo flat multipath fading, and the path losses of the self-fading components will generally be quite different, so as long as the OLA transmission has multiple diversity channels, self-fading is not expected to be a problem.

To reduce network overhead, and to allow OLA transmission to achieve range extension, we fixed the transmit power of all nodes in the network, for both protocols, to be the same. However, we considered 7 different values for the fixed transmit power, from 18dBm to 0dBm, with 3dBm intervals. Increasing transmit power increases the node degree of the network as shown in Fig. 5, thereby decreasing hop count from the source to the destination. Also, the transmit power impacts the interference level in AODV, because all the nodes send the periodic beacon (HELLO message).

Constraining the bandwidth of the SISO network to a quarter of the bandwidth of the CT network may seem an unfair disadvantage to AODV. However, because the spectral efficiency in the experiment (0.5b/s/Hz) places our physical layer in the power-limited regime [26], increasing the bandwidth of the SISO links by a factor of four, while keeping the transmit power fixed, does not change the link performance appreciably.

A. End-to-end Average Hop Count

The first experiment studies the delay performance in terms of the average hop count from the source to the destination. End-to-end latency is often approximated by the number of hops multiplied by the time spent on each hop, including processing, queuing, propagation, and transmission delays. Fig. 6 shows the hop count statistics, which are averaged over 100 runs of the 100 data packets transmission (the total number of transmitted packets=10,000=100×100). In the figure, the red and black curves represent the hop counts of AODV and OLAROAD, respectively. Also, the red and black "I-shape" error bars indicate the 95% confidence intervals of the statistical data. As we expect, the hop counts of both AODV and

[1]A distributed algorithm to assign diversity channels to nodes is deemed feasible by the authors, but beyond the scope of this paper.

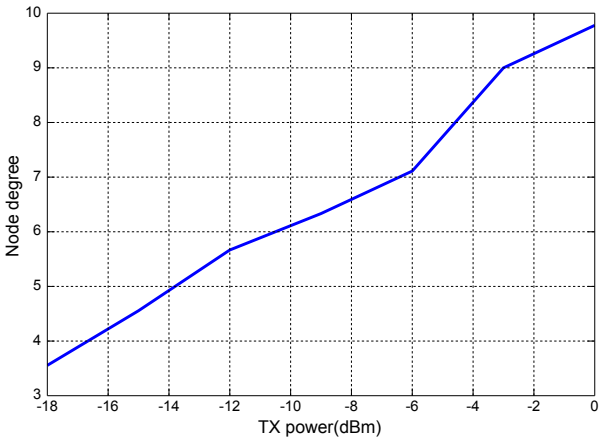


Fig. 5. Node degree depending on transmit power levels

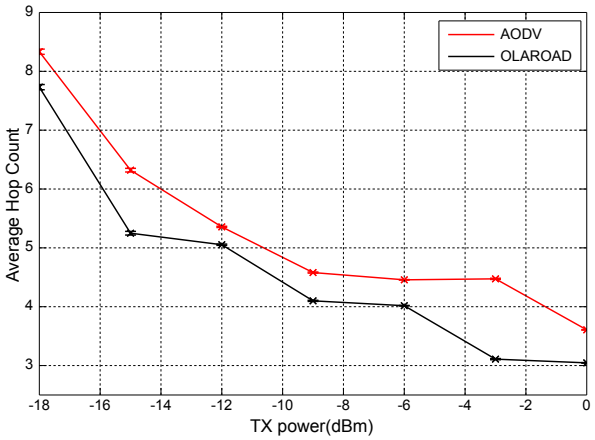


Fig. 6. End-to-end hop count

OLAROAD decrease, as the transmit power increases. When we compare the two, the hop counts of OLAROAD are always less than AODV. However, the difference is not as much as we would predict for CCT on Rayleigh fading channels, because the line of sight (LOS) channels in a corridor do not experience as much fading as non-LOS channels, and the linear topology does not give enough node degree for OLAROAD to jump over many hops. In this set-up, AODV also benefits the hop count saving by finding the shortest path as the transmit power increases. Moreover, AODV has the link layer error control, whereby it retransmits erroneous packets up to seven times in our implementation, which is triggered by the timeout of the link layer ACK message from the desired receiver. Therefore, AODV is able to successfully send the data packets even though the shortest path is unstable.

B. Round Trip Time (RTT)

We conducted separate experiments to measure the latency of the data transmission in terms of RTT. In this experiment, the source node first transmits a packet to the destination, then,

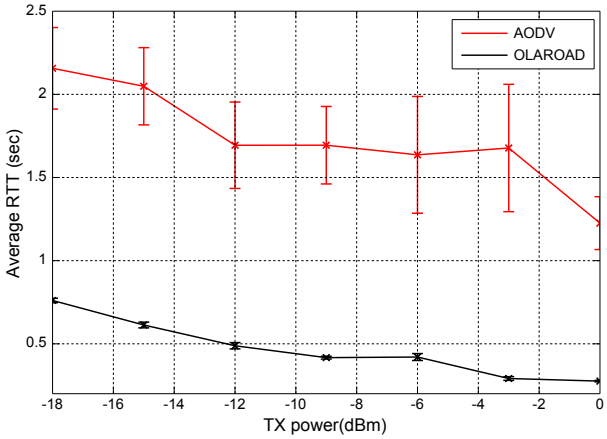


Fig. 7. End-to-end round trip time (RTT)

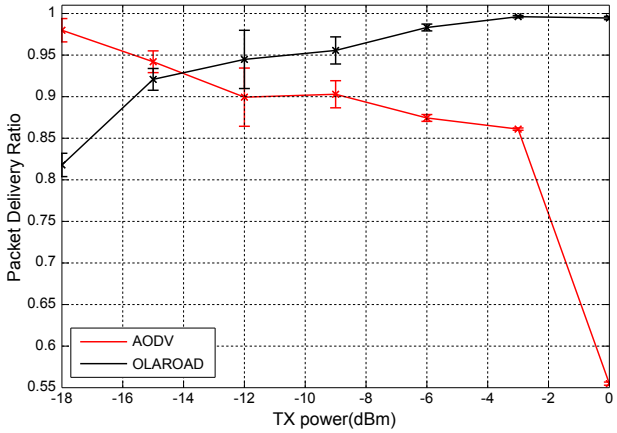


Fig. 8. Packet delivery ratio (PDR)

after that packet has finally been received successfully, the destination generates another packet and sends it back to the source. We measure the time duration between the transmit time stamp of the first packet and the reception time stamp of the second packet, both of which are logged at the source node. After repeating this measurement process 100 times, we obtained the mean RTT values shown in Fig. 7, where we observe that OLAROAD RRT is about 1/3 that of AODV, and also the confidence interval for OLAROAD is much smaller, indicating a much lower variance in RTTs. The first reason for the large difference in mean RTT values is the retransmission of AODV to repair the packet loss in the link layer. In fact, in wireless networks, there is a trade-off between the latency and the reliability, which is usually controlled by the error control schemes such as link layer retransmission. In our implementation of AODV, we set the maximum number of retries is seven, however, this number might be reduced, if the application is more latency-sensitive but error-tolerant such as voice or video transfer. Also, although an ACK protocol for OLAROAD has been proposed, we did not implement one in this experiment, because of the high reliability of CCT in this network. Another reason for the large difference in RTT is the interference caused by the HELLO messages in AODV, which make the data transmissions back off by CSMA/CA. However, we minimize its impact by modifying the local route repair to be initiated only by the nodes in the data route. In other words, the nodes that are not in the data route, do not send a RREQ even when they lose one of its neighbors by expiration.

C. Packet Delivery Ratio (PDR)

In this subsection, we focus on the end-to-end outage performance measured by PDR over 100 data packets sent from the source. We repeated 100 runs of this PDR measurement, and obtained the result given in Fig. 8. In the figure, OLAROAD performs better than AODV with the transmit power from -12dBm to 0dBm. However, in the poor wireless link condition with the lower transmit power (-18dBm and -15dBm), the PDR of AODV is higher than or similar to OLAROAD. Considering cooperative diversity, this might seem surprising, however, it is reasonable based on the low node degree and the link layer error control of AODV. With the transmit power of -18dBm, the route built by OLAROAD was actually similar to the one made by AODV, because the node degree is too low to gather many cooperators. As the average hop count of AODV and OLAROAD routes at -18dBm transmit power are about 8.5 and 8, respectively in Fig. 6, the maximum number of cooperator on each hop on average in the cooperative route of OLAROAD is just about two. However, in fact, the number can be less than two if some nodes cannot decode. Also, as previously stated, AODV has the data link layer error control, while OLAROAD does not. Therefore, this result suggests the need of error detection/repair in the link layer independent from the end-to-end error control, even though it is challenging to design because of the asymmetry in CT-based link connections. As the transmit power increases, OLAROAD shows better PDR, while the PDR of AODV decreases. One reason for this performance degradation in AODV is the HELLO messages cause increasing interference on the data packets. Also, AODV builds the shortest route regardless of the link quality, which in result causes frequent re-routing and retransmissions.

D. Route Discovery Time

The route discovery time is the latency to discover routes, which is a key performance metric for reactive (on-demand) routing protocols. This latency impacts the data rate, because the data packets are buffered until the route is built. In AODV, the discovery time is the time from the first RREQ transmission to the first reception RREP, which does not necessarily correspond to the first RREQ. In other words, if the source cannot receive the RREP corresponding to its first RREQ, but receives the RREP for the second RREQ trial, then the route discovery delay would be the time between the first RREQ transmission and the RREP (for the second RREQ) reception. Likewise, in OLAROAD, we measure the time between the first RREQ transmission and the first RCACK

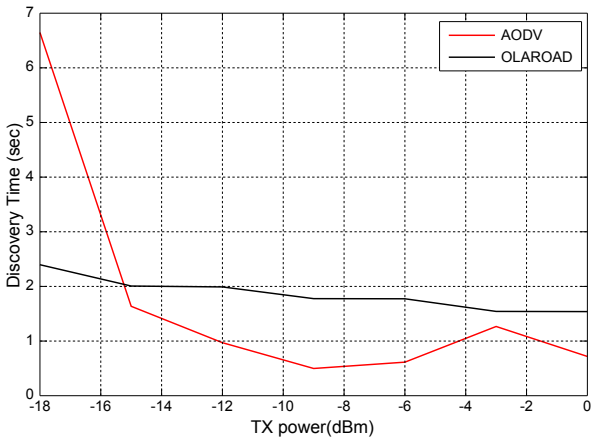


Fig. 9. Route discovery time

reception. In this way, our measurement of the route discovery time was conducted by 100 repeated experiments. Fig. 9 shows the *median* values obtained over the 100 runs, because route failure (infinite delay) happened sometimes. For the same reason, we could not calculate the confidence intervals. The result in Fig. 9 shows that AODV takes less time to finish the routing than OLAROAD except the transmit power of -18dBm, since OLAROAD has two stages of end-to-end handshake process. The reason why AODV gives the poor performance on -18dBm is the unstable link condition, so that the unicast of RREP requires many retransmissions.

Because of the two handshakes, OLAROAD has a longer route discovery time than AODV. If the nodes have small buffers, the large discovery time of OLAROAD might cause the buffer overflows. From an overhead perspective, the cost of OLAROAD's extra handshake will diminish if a large number of packets are sent along the route.

VII. Conclusion

In this paper, we demonstrated a CT-based routing scheme, OLAROAD, in comparison with the widely used on-demand routing protocol, AODV, on a linear network in a typical indoor environment. For a realistic evaluation, the protocols were implemented on a testbed using GNU Radio and USRP1 software defined radio. This paper provided the details of the challenge in CT-based routing, and the system structure of the physical, MAC, and network (routing) layers for the implementation of OLAROAD and AODV. The experimental results showed that OLAROAD improves the most of the performance metrics considered except the route discovery time. Moreover, we pointed out the need for the link layer error control in OLAROAD. In the future, we will consider other network topologies, demonstrate step-size control, and design a link layer error control scheme.

References

[1] Y. J. Chang and M. A. Ingram, "Convergence property of transmit time pre-synchronization for concurrent cooperative communication," *IEEE GlobeCom*, Dec 2010.

[2] H. Jung, Y. J. Chang, and M. Ingram, "Experimental range extension of concurrent cooperative transmission in indoor environments at 2.4GHz," *IEEE MILCOM*, pp. 148 –153, Nov. 2010.

[3] L. Thanayankizil, A. Kailas, and M. A. Ingram, "Routing for wireless sensor networks with an opportunistic large array (OLA) physical layer," *Ad Hoc & Sensor Wireless Networks*, vol. 8, no. 1-2, pp. 79–117, 2009.

[4] L. Thanayankizil and M. A. Ingram, "Reactive robust routing with Opportunistic Large Arrays," *IEEE ICC Workshop*, June 2009.

[5] A. Scaglione and Y. Hong, "Opportunistic large arrays: cooperative transmission in wireless multihop ad hoc networks to reach far distances," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2082–2092, Aug. 2003.

[6] Z. Gao, Y. J. Chang, and M. Ingram, "Synchronization for cascaded distributed MIMO communications," *IEEE MILCOM*, pp. 387 –392, Nov. 2010.

[7] B. Sirkeci-Mergen, A. Scaglione, and G. Mergen, "Asymptotic analysis of multistage cooperative broadcast in wireless networks," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2531 – 2550, June 2006.

[8] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, and O. Ercetin, "A cross-layer framework for exploiting virtual miso links in mobile ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 6, pp. 579 –594, 2007.

[9] B. Gui, L. Dai, and L. Cimini, "Routing strategies in multihop cooperative networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 2, pp. 843 –855, 2009.

[10] J. Yackoski, L. Zhang, C.-C. Shen, L. Cimini, and B. Gui, "Networking with cooperative communications: Holistic design and realistic evaluation," *IEEE Comm. Magazine*, vol. 47, no. 8, pp. 113 –119, 2009.

[11] R. Ramanathan, "Challenges: a radically new architecture for next generation mobile ad hoc networks," *MobiCom*, pp. 132–139, 2005.

[12] A. Blair, T. Brown, K. Chugg, T. Halford, and M. Johnson, "Barrage relay networks for cooperative transport in tactical manets," *IEEE MILCOM*, pp. 1 –7, 2008.

[13] T. Halford and K. Chugg, "Barrage relay networks," *Information Theory and Applications Workshop (ITA), 2010*, 31 2010.

[14] T. Halford, K. Chugg, and A. Polydoros, "Barrage relay networks: System and protocol design," *IEEE PIMRC*, pp. 1133 –1138, 2010.

[15] A. Blair, T. Brown, K. M. Chugg, and M. Johnson, "Tactical mobile mesh network system design," *IEEE MILCOM*, 2007.

[16] B. Sirkeci-Mersen and A. Scaglione, "A continuum approach to dense wireless networks with cooperation," *IEEE INFOCOM*, vol. 4, pp. 2755 – 2763 vol. 4, 2005.

[17] S. Hassan and M. Ingram, "Modeling of a cooperative one-dimensional multi-hop network using quasi-stationary markov chains," *IEEE GLOBECOM*, 2010.

[18] Y. J. Chang, M. A. Ingram, and S. R. Frazier, "Cluster transmission time synchronization for cooperative transmission using software defined radio," *ICC Workshop on CoCoNet3*, June 2010.

[19] B. Crow, I. Widjaja, L. Kim, and P. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Comm. Magazine*, vol. 35, no. 9, pp. 116 –126, Sep. 1997.

[20] I. Chakeres and E. Belding-Royer, "AODV routing protocol implementation design," in *International Conference on Distributed Computing Systems*, 2004, pp. 698 – 703.

[21] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," Published Online, Internet Engineering Task Force, RFC Experimental 3561, July 2003.

[22] "GTNets," http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/.

[23] Ettus, "The universal software radio peripheral," Website, http://www.ettus.com/downloads/er_broch_trifold_v5b.pdf.

[24] "Gnu radio," http://gnuradio.org.

[25] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling mac protocol implementations on software-defined radios," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, 2009, pp. 91 –105.

[26] B. Sklar, *Digital communications: fundamentals and applications*. Prentice-Hall, Inc., 1988.