Appeared in the Proceedings of the Second IEEE International Workshop on Wireless Mesh Networks (IEEE Wimesh 2006), Reston, VA, Sept 25, 2006.

Interference Aware Routing in Multi-Radio Wireless Mesh Networks

Anand Prabhu Subramanian
State University of New York at Stony Brook
anandps@cs.sunysb.edu

Milind M. Buddhikot, Scott Miller
Lucent Bell Labs
{mbuddhikot,scm}@lucent.com

Abstract— **We address the problem of interference aware routing in multi-radio infrastructure mesh networks wherein each mesh node is equipped with multiple radio interfaces and a subset of nodes serve as Internet gateways. We present a new interference aware routing metric – iAWARE that aids in finding paths that are better in terms of reduced interflow and intra-flow interference. We incorporate this metric and new support for multi-radio networks in the well known AODV routing protocol to design an enhanced AODV-MR routing protocol. We study the performance of our new routing metric by implementing it in our wireless testbed consisting of 12 mesh nodes. We show that iAWARE tracks changes in interfering traffic far better than existing well known link metrics such as ETT and IRU. We also demonstrate that our AODV-MR protocol delivers increased throughput in single radio and two radio mesh networks compared to similar protocol with WCETT and MIC routing metrics. We also show that in the case of two radio mesh networks, our metric achieves good intra-path channel diversity.**

I. Introduction

The wireless mesh networks are emerging as a significant new technology that has experienced growing research and commercial interest [1–4, 8, 9, 31, 35]. Their promise of rapid deployability and reconfigurability makes them suitable for transient on-demand network deployment scenarios such as disaster recovery, homeland security, convention centers, hard-to-wire buildings and friendly terrains. They are also an attractive technology for long-lived infrastructure networks such as municipal broadband in dense metros and for providing low-cost backhaul to cellular base stations in remote rural areas.

One of the distinguishing aspect of mesh networks is the multi-hop forwarding or relaying of packets over the wireless links for communication between the component nodes. The form of mesh networks that are of most commercial interest are often called *Infrastructure-mesh* networks wherein the end-user devices such as PDAs, laptops do not participate in the packet relay and the relay nodes are part of the network infrastructure. Here, the network consists of two types of links: access links to the end-user devices and mesh-relay links between relay nodes to form the packet transport backbone. In order to guarantee minimal congestion in presence of dynamic traffic aggregation from access links, the mesh backbone must be designed to support high capacity and speed advantage over access links.

There are two main techniques to improve wireless capacity: (1) Improve data rate of the wireless channel that uses a fixed amount of spectrum by improving spectral efficiency in bits/sec/Hz. This can achieved by better modulations, multi-antenna techniques and better MAC protocols. For example, one can use 54 Mbps 802.11g links instead of 11 Mbps 802.11b links and MIMO antenna instead of single antenna systems. (2) Simultaneously use a large number of concurrent wireless channels and therefore, a large amount of spectrum.

This work is supported in part by NSF CRI Grant 454174 and Bell Labs Summer Internship Program.

In the second approach, each mesh node can use a single radio interface that is dynamically switched to a wireless channel in different frequency bands to communicate with different nodes. This however incurs frequent channel switching overhead of the order of 100s of microseconds which is comparable to packet transmission times. A more practical method for concurrent channel usage is to use multiple radio interfaces and dedicate a separate radio channel to each. We follow this model in our work.

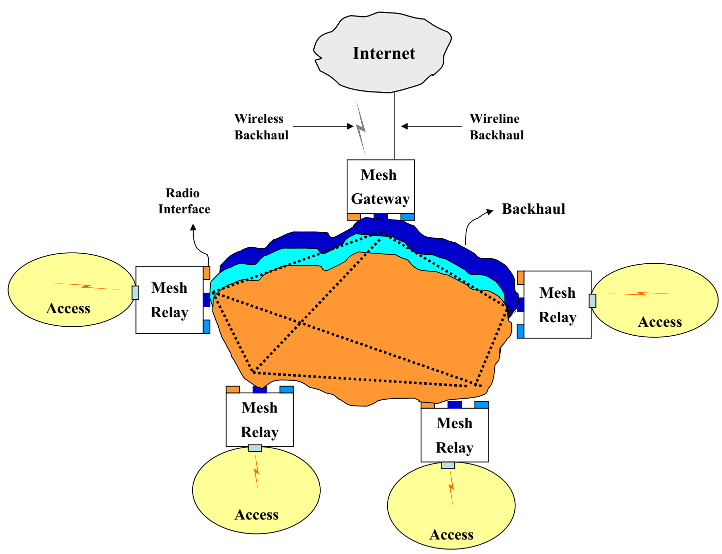

Fig. 1. Multi-radio mesh: reference architecture

Our reference architecture for multi-radio mesh network (Figure 1) based on this model consists of two new network elements: the relay and the gateway nodes. The relay elements are multi-radio systems that support two kinds of wireless network interfaces: access and relay, whereas gateway elements support: relay and Internet backhaul (uplink) interfaces. The enduser Mobile Nodes (MNs) access network using the access interfaces. The relay interfaces are used to construct a self-configuring, secure, managed, power-adaptive packet forwarding backbone between the relay and gateway nodes. The access links can be based on 3G (e.g:[11]) or 802.11[10] standards, whereas the relay links can be based on 802.16 or 802.11. The gateways are connected to the Internet via wired (Ethernet) or wireless (1xRTT, EV-DO, 802.16) uplinks. This model requires solutions to two important problems: (1) *mesh channel assignment* which assigns channels to radio interfaces at all nodes and *mesh routing* which requires efficient, high capacity routes to be computed between a source-destination pair of nodes.

Various approaches to solving the mesh channel assignment problem have been reported in literature [27, 29, 34]. In this paper, we focus on mesh routing problem.

A. Research Contributions

We address the problem of interference aware routing in multi-radio infrastructure mesh networks wherein each mesh node is equipped with multiple radio interfaces and a subset of nodes serve as Internet gateways. We present a new interference aware routing metric – iAWARE that aids in finding paths that are better in terms of reduced inter-flow and intra-flow interference. We incorporate this metric and new support for multi-radio networks in the well known AODV routing protocol to design an enhanced AODV-MR routing protocol. We study the performance of our new routing metric by implementing it in our wireless testbed consisting of 12 mesh nodes. We show that iAWARE tracks changes in interfering traffic far better than existing well known link metrics such as ETT and IRU. We also demonstrate that our AODV-MR protocol delivers increased throughput in single radio and two radio mesh networks compared to similar protocol with WCETT and MIC routing metrics. We also show that in the case of two radio mesh networks, our metric achieves good intra-path channel diversity.

B. Outline of the Paper

This paper is organized as follows: Section II reviews the various routing metrics reported in literature and articulates the need for new routing metrics. It then presents the design of our new interference aware metric called *iAWARE* based on the physical interference model. In Section III, we present the design of our multi-radio routing protocol AODV-MR based on AODV-ST [28]. Section IV describes the implementation of the routing metric computation and the routing protocol in Linux 2.4. kernel using NIST AODV implementation [5]. Section V describes our testbed on which the measurement were taken and section VI presents the performance results. Finally, Section VII provides conclusions and outlines our on-going work.

II. Design of Interference Aware Routing Metric

In this section, we describe the motivation and design of our new interference aware routing metric for multi-radio wireless mesh networks.

A. Need for a New Routing Metric

Given a source and destination node, a routing protocol provides one or more network paths over which packets can be routed to the destination. The routing protocol computes such paths to meet criteria such as minimum delay, maximum data rate, minimum path length etc. A routing metric that accurately captures quality of network links and thus aids in meeting such criteria is central to computation of good quality paths. The design of routing metrics for wireless multi-hop networks is challenging due to following three unique characteristics of wireless links:

- *Time varying channels and resulting variable packet loss:* The wireless links suffer from short term and long term fading and result in varying packet loss over different time scales. When the distance between the communicating nodes is large or if environment is obstacle rich and causes fading, the loss ratio of the link can be high. A routing metric should accurately capture this time varying packet loss.
- *Packet transmission rate:* The packet transmission rate (or data rate) may vary depending upon the underlying physical layer technology. For example, 802.11a links have high data rate compared to 802.11b links. The data rate may also vary depending on the link loss characteristics when auto-rate control algorithms are used.
- *Interference:* Wireless links operating in unlicensed spectrum suffer from two kinds of interference: (1) *Uncontrolled interference* results from non-cooperating entities external to the network that use the same frequency band but do not participate in the MAC protocol used by network nodes. For example, microwave ovens, Bluetooth devices operating in 2.4GHz ISM bands interfere with 802.11b/g networks in the same band. (2) *Controlled interference:* This kind of interference results from broadcast nature of wireless links where a transmission in one link in the network interferes with the transmissions in neighboring links. The interference of this kind depends on factors such as the topology of the network, traffic on neighboring links etc. It is well known [16] that interference seriously affects the capacity of wireless networks in a multi-hop setting. It is important for a routing metric to capture the potential interference experienced by the links to find paths that suffer less interference and improve the overall network capacity. Interference can be either intra-path, wherein transmissions on different links in a path interfere or inter-path interference or inter-path wherein, transmissions on links in separate paths interfere. A more channel diverse multi-hop path has less intra-flow interference which increases the throughput along the path as more links can operate simultaneously if they operate on different orthogonal channels.

A good routing metric should find paths with component links that have low loss ratio, high data rate and experience low levels of interference. In order to motivate the need for a new routing metric, in the following, we give an overview of the various routing metrics proposed for multi-hop wireless mesh networks in the literature and discuss their limitations. We describe them in an order where each subsequent metric improves on the previous one

A.1 Hop Count

Hop count is the traditional routing metric used in most of the common routing protocols (AODV [26], DSR [20], DSDV [25]) designed for multi-hop wireless networks. This metric treats all links in the network to be alike and finds paths with the shortest number of hops. It also does not account for data rate and interference experienced by the links. This can often result in paths which have high loss ratio and therefore, poor performance.

A.2 Expected Transmission Count (ETX)

ETX proposed in [14], characterizes the link loss ratio using the expected number of MAC retransmissions needed to successfully deliver a packet from the sender to the receiver. The lesser the ETX metric for a link, the better is the link. The path metric is the summation of ETX of each link in the path. ETX does not consider the data rate at which the packets are transmitted over each link. Since ETX is measured using periodic broadcast packets which are sent at a very slow interval (usually 1 sec) [14], they do not reflect how busy a link is. ETX might vary when there is very high load due to 802.11 MAC unfairness [13] or when there is loss of the broadcast packets due to collision with packets from hidden terminals. However, when the sender of the ETX broadcast packet can hear (or sense) the neighboring transmissions, collision does not happen and ETX is not affected. Thus, ETX does not capture the interference experienced by the links completely. ETX was designed for networks using a single channel, so it cannot exploit the presence of multiple channels and find paths that have better channel diversity.

A.3 Weighted Cumulative Expected Transmission Time (WCETT)

Draves et al. proposed WCETT [15] for multi-radio multi-channel mesh networks. First, they propose ETT which improves upon ETX by capturing the data rate used by each link. ETT_i of a link i is defined as follows:

$$ETT_i = ETX_i \times \frac{S}{B_i} \qquad (1)$$

where S is the packet size used and B_i is the raw data rate (bandwidth) of the link i. ETT_i characterizes the expected MAC transmission time of a packet of size S over the link i. Both ETX and ETT do not consider the presence of multiple channels and therefore, find paths with less channel diversity. Also ETT characterizes the expected transmission time in the absence of interference in the network. To find paths with less intra-flow interference, the authors propose WCETT which is a weighted cumulative path metric using ETT. The WCETT metric of a path p is defined as follows:

$$WCETT_p = (1-\alpha) \times \sum_{i \in p} ETT_i + \alpha \times \max_{1 \le j \le \kappa} X_j \qquad (2)$$

where X_j is the summation of ETT of the links in path p operating on channel j , κ is the number of orthogonal channels available and $0 \le \alpha \le 1$ is a tunable parameter. The first component in the WCETT metric helps in finding path with links having less ETT. The second component improves the channel diversity and helps in finding paths with less intra-flow interference.

One limitation of the WCETT metric is that it does not capture inter-flow interference and when there are multiple flows in the network, it might end up finding routes in more congested areas of the network which results in poor throughput. In [36], the authors identify that the WCETT metric is not isotonic [32] [33]. *Isotonicity* is a property needed by link-state routing protocols to find loop-free and minimum weighted paths. However, on-demand distance vector or source routing protocols based on Bellman-ford algorithm can use non-isotonic metrics to find efficient paths [36]. In [15] Draves et. al use a source routing protocol ($LQSR$) which is a modified version of DSR [20] using the WCETT metric.

A.4 Metric of Interference and Channel Switching (MIC)

In [36], the authors propose MIC which improves upon WCETT by considering inter-flow interference. MIC for a path p is defined as follows:

$$MIC(q) = \frac{1}{N \times min(ETT)} \sum_{\text{link } l \in p} IRU_l + \sum_{\text{node } i \in p} CSC_i \qquad (3)$$

where N is the total number of nodes in the network. The two components IRU and CSC are defined as follows:

$$IRU_l = ETT_l \times N_l \qquad (4)$$

$$CSC_i = \begin{cases} w_1 & \text{if } CH(prev(i)) \neq CH(i) \\ w_2 & \text{if } CH(prev(i)) = CH(i) \end{cases} \qquad (5)$$

$$0 \le w_1 < w_2 \qquad (6)$$

where N_l is the set of neighbors that interfere with the transmissions on link l. $CH(i)$ represents the channel assigned for node i's transmission and $prev(i)$ represents the previous hop of node i along the path p. MIC is also *n*on-isotonic because of the second component (CSC) and the authors in [36], demonstrate some what complex ways to form virtual nodes and make the metric isotonic.

Note that MIC incorporates inter-flow interference by scaling up the ETT of a link by the number of neighbors interfering with the transmission on that link. In practice, the degree of interference caused by each interfering node on a link is not the same. It depends on the signal strength of the interferer's packet at the sender or the receiver. This varies depending on the position of the interferer with respect to the actual sender or receiver and the path loss characteristics. Also, the degree of interference depends on the amount of traffic generated by the interfering node. Even when the interferer is close to the sender or the receiver and is not involved in any transmission simultaneously, it does not cause any interference. MIC fails to capture the above mentioned characteristics of interference.

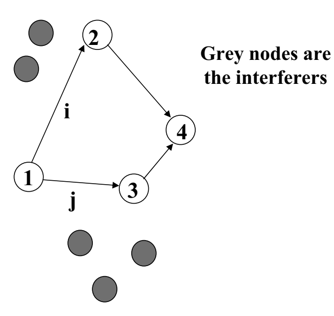

Fig. 2. Understanding interference

For example, consider Figure 2. Let us assume that each node generates uniform traffic. Consider the two links i and j with

$ETT_i > ETT_j$. Link i has two interfering neighbors which are close to the nodes 2 and cause high degree of interference. Link j have three interfering neighbors which cause less interference. MIC favors link i over link j, resulting in choosing the link with higher ETT and poor throughput.

The second component (CSC) captures intra-flow interference only in two consecutive links. The authors in [36] generalize it, but the decomposition of the nodes into virtual nodes to make the metric isotonic becomes more complicated.

MIC favors links incident on nodes with less number of interfering neighbors irrespective of whether the neighbor causes any interference or not. This results if finding paths along the boundary of the network where nodes have less number of neighbors and find longer paths. We observe this from our experimental results.

B. Our New Metric: Interference Aware Routing Metric (iAWARE)

In this section, we propose our new routing metric *i*AWARE, which addresses the aforementioned limitations of existing routing metrics. Our routing metric captures the effects of variation in link loss-ratio, differences in transmission rate as well as inter-flow and intra-flow interference.

When there is no interference in the network, ETT captures the quality of the link quite well as links with less expected transmission time give better throughput. But when there are more interfering flows in the network, this is not the case. We need to factor in the varying interference experienced by a link into the routing metric to find good quality paths. In order to do this, we need to model interference properly and factor it in the metric.

The protocol interference model [16] [19] and the physical interference model [16] are two models that have been proposed in the literature and studied well. We use the physical interference model to capture the interference experienced by links in the network. In this model, a communication between nodes u and v is successful if the $SINR$ (Signal to Interference and Noise Ratio) at the receiver v is above a certain threshold which depends on the desired transmission characteristics (e.g channel, data rate etc.). More formally, denoting the signal strength of a packet from node u at node v by $P_v(u)$, a packet on the link (u, v) from node u to node v is correctly received if and only if:

$$\frac{P_v(u)}{N + \sum_{w \in V'} P_v(w)} \geq \beta, \tag{7}$$

where N is the background noise, $V^{'}$ is the set of nodes simultaneously transmitting and β is a constant which depends on the data rate, channel characteristics, modulation scheme etc. If we consider a DATA/ACK like transmission on a link, then the $SINR$ values at both the nodes u and v should be greater than β.

This model is less restrictive compared to the protocol interference model [16] [19], since it does not use the concept of transmission range and interference range. It is also not restrictive to any medium access mechanism (802.11, CSMA/CA, TDMA). It only depends on the signal strength values which can be measured easily using commodity wireless cards as we discuss in section IV. It also has the advantage of measuring the parameters of the model using online data traffic in contrast to recent models [24] [21] proposed which use special kind of traffic to measure the degree of interference between links.

We define $interference\ ratio\ IR_i(u)$ for a node u in a link $i = (u, v)$ where ($0 < IR_i(u) \leq 1$) as follows:

$$IR_i(u) = \frac{SINR_i(u)}{SNR_i(u)} \tag{8}$$

where

$$SNR_i(u) = \frac{P_u(v)}{N} \tag{9}$$

$$SINR_i(u) = \frac{P_u(v)}{N + \sum_{w \in \eta(u) - v} \tau(w) P_u(w)} \tag{10}$$

Here $\eta(u)$ denotes the set of nodes from which node u can hear (or sense) a packet and $\tau(w)$ is the normalized rate at which node w generates traffic averaged over a period of time. $\tau(w)$ is 1 when node w sends out packets at the full data rate supported. We use $\tau(w)$ to weight the signal strength from an interfering node w as $\tau(w)$ gives the fraction of time node w occupies the channel. We discuss about the measurement of $\tau(w)$, $\eta(u)$, $P_u(v)$, and N in detail in section IV

Considering a bidirectional communication link $i = (u, v)$ for a DATA/ACK like communication, IR_i is defined as:

$$IR_i = min(IR_i(u), IR_i(v)) \tag{11}$$

Note that when there is no interference (no interfering neighbors or no traffic generated by interfering neighbors) $SINR$ of link i is equal to the SNR and thus IR_i is 1. In this case, the link i is independent of interference and the quality of the link is determined by the link loss-ratio and the data rate at which it operates captured by ETT_i.

We define our new link metric *i*AWARE of a link j as follows:

$$i\text{AWARE}_j = \frac{ETTj}{IR_j} \tag{12}$$

When IR_j for the link j is 1 (no interference), *i*AWARE$_j$ is simply ETT_j which captures the link loss ratio and packet transmission rate of the link j. ETT_j is weighted with IR_i to capture the interference experienced by the link from its neighbors. A link with low ETT and high IR will have a low *i*AWARE value. Lower the *i*AWARE of a link, better is the link.

Note that our model does not fully capture sender-side interference which results in backoffs and increases the expected transmission time. Ours is a first cut simple approach to factor in the varying interference with ETT. However, we show in section VI, that this simple approach helps to find paths with better quality when there are multiple interfering flows in the network. We are currently investigating more accurate ways to correlate ETT and IR.

In order to exploit the channel diversity and to find paths with less intra-flow interference, we define X_j as follows:

$$X_j = \sum_{\text{conflicting links } i \text{ on channel } j} i\text{AWARE}_i, 1 \leq j \leq \kappa \tag{13}$$

Here κ is the number of orthogonal channels available and we say that link $e_1 = (u, v)$ conflicts with link $e_2 = (x, y)$ if any one of the following inequalities is true.

$$\frac{P_u(v)}{N + \sum_{w\in\{x,y\}} P_u(w)} < \beta, \quad \frac{P_v(u)}{N + \sum_{w\in\{x,y\}} P_v(w)} < \beta$$

$$\frac{P_x(y)}{N + \sum_{w\in\{u,v\}} P_x(w)} < \beta, \quad \frac{P_y(x)}{N + \sum_{w\in\{u,v\}} P_y(w)} < \beta$$

Our new weighted cumulative path metric *i*AWARE(p) of a path p is defined as follows:

$$i\text{AWARE}(p) = (1-\alpha) \times \sum_{i=1}^{n} i\text{AWARE}_i + \alpha \times \max_{1\le j\le\kappa} X_j \quad (14)$$

Of course, there are practical limitations to accurate measurement of conflict and therefore, computation of the metric. We elaborate on these issues in the implementation Section IV.

The path metric *i*AWARE(p) is also non-isotonic like WCETT because of the second component. We design a new multi-radio on-demand distance vector routing protocol derived from AODV [26] and incorporate our new metric *i*AWARE to find efficient paths.

III. Design of AODV-Multi-Radio (MR) Routing Protocol

As mentioned in the Section II, our routing metric *i*AWARE is non-isotonic. Using on-demand distance vector routing protocols based on Bellman-Ford algorithm, we can find efficient paths without forwarding loops even when the metric is non-isotonic [36].

In this section we describe the design of our new multi-radio on-demand distance vector routing protocol derived from traditional AODV [26] protocol.

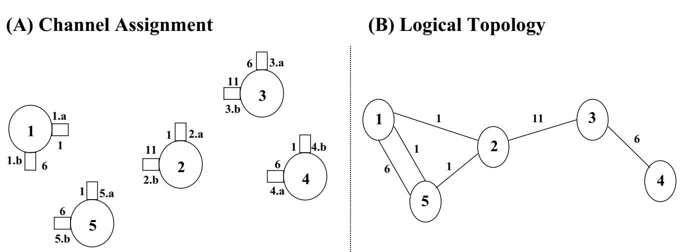

Fig. 3. Multi-radio nodes and logical connection topology

A. Channel Assignment and Logical Topology

As described in section I, each wireless router in the network has multiple radio interfaces. Each wireless interface is tuned to a different channel using efficient channel assignment algorithms [27, 30, 34]. The channel assignment to the interfaces gives a logical topology on which the routing protocol works. For example, see in figure 3. Node 1 and 5 have two interface tuned to common channels, so there are two links between the nodes 1 and 5. The routing protocol decides which link to use depending on the link metric (the link metric might be different depending on the degree of interference experienced in the channels the links are operating in).

B. Routing Protocol Design

Every node sends out periodic HELLO broadcast packets on each of its interfaces to discover their neighboring nodes. In our design, each interface in a node is assigned a unique IP address, so that when nodes send hello packets, they actually discover neighboring interfaces tuned to the same channel. This establishes multiple links between neighboring nodes if they have more than one interface tuned to a common channel. The HELLO broadcast packets are used to compute ETX and IR as discussed in section IV.

A node begins a route discovery process, when it has a packet to send to some destination node and does not have a route to it. During route discovery, it sends a Route Request ($RREQ$) broadcast packet in each of its interfaces on different channels to which they are tuned. Each $RREQ$ packet sent, has a unique identifier which is a combination of the IP address of the interface on which it is sent and a $RREQ$ sequence number which is incremented for each $RREQ$ packet generated. This identifier is used to discard duplicate $RREQ$ packets received from other nodes and to prevent routing loops.

In order to compute our path metric when the RREQ traverses the network, we need the link metric (*i*AWARE) for each link traversed and the channel in which they are operating. So we overload the RREQ packet to carry the link metric and the channel of each link traversed. When an intermediate node receives an $RREQ$ packet, it first checks the unique identifier to see if it has already received the RREQ. In the event it has received this RREQ before, it creates a reverse route entry to the originator (source) of the RREQ packet. If it had already seen the RREQ packet but received the new one on a better path (based on the path metric computed using Equation 14), it updates the reverse route accordingly. It then appends the link information (*i*AWARE and channel) in the $RREQ$ message and forwards it by broadcasting in each of its interface.

For example in Figure 3, when node 1 begins a route discovery process to node 4, it sends out 2 RREQ broadcast packets with identifier $(1.a, n)$ and $(1.b, n)$ where n is the sequence number. Node 2 receives the RREQ with id $(1.a, n)$ on its interface $2.a$ and broadcasts it on both its interfaces. It does not receive the RREQ with id $(1.b, n)$ from node 1 as it does not have an interface tuned to channel 6[1]. Similarly node 3 receives the same RREQ in channel 11 through its interface $3.b$ and node 4 on channel 6 on its interface $4.a$.

When the destination-only flag is set in the $RREQ$ message, only the destination is allowed to generate an Route Reply ($RREP$). If the destination-only flag is not set in the $RREQ$, an intermediate node is allowed to send an $RREP$ if it has an active route toward the destination.

The $RREP$ message is unicast toward the source along the reverse route built during the $RREQ$ propagation. As the $RREP$ is propagated, the intermediate nodes build a forward route to the destination node. The $RREP$ packet is also over-

[1] Node 2 will get the RREQ with id $(1.b, n)$ on channel 1 from node 5.

loaded like the $RREQ$ packet to carry the link information and it is appended as it traverses toward the actual source node. When the source node receives the $RREP$ packet, it builds the route to the destination and sends out the queued data packets.

When an active route breaks, the node in the route that detects the break either sends an Route Error ($RERR$) message toward the source node or does a local repair by finding an alternative path to the destination.

IV. IMPLEMENTATION DETAILS

In the earlier sections, we omitted details about several implementation specific issues like ETT measurement, interference ratio (IR) estimation using signal strength and traffic measurements, and the interaction between the channel assignment and routing protocol. In this section we describe them in detail and also mention some practical limitations with current commodity hardware.

A. Implementation Architecture

The overall architecture of our routing protocol design is shown in Figure 4. We implemented our multi-radio AODV-MR routing protocol by extending the public domain implementation of traditional AODV available from NIST [23]. The routing protocol is implemented as a loadable kernel module and communicates with the channel assignment module and the device driver to compute the link metric *iAWARE* and use them to find routes reactively. The channel assignment protocol is implemented as a user module and assigns channels to each of the interfaces and communicates the assignment to the routing module through a *proc* interface. Currently, we have a static channel assignment algorithm like the MUP channel assignment algorithm [12] implemented. The device driver collects the signal and traffic information as described in the later section (Section IV-C) and exports them to the routing module. We used Prism 2 based cards and the hostap driver [18] to work in *Ad-Hoc* mode.

B. ETT Measurement

We use the HELLO packets sent by AODV to compute the expected transmission count. Each node broadcasts periodic HELLO packets (every 1 second) and computes the forward (d_f) and reverse delivery ratio (d_r) for a predetermined time interval. We used a time interval of 10 seconds in our implementation. The expected transmission count of a link is computed as $ETX = \frac{1}{d_f \times d_r}$. In order to compute ETT using the formula 1, we need the rata datarate at which the link operates. The hostap driver [18] adds a prism monitor header (also called as AVS header) to each received packet. The prism monitor header contains the data rate at which the packet was transmitted. But this header is added to the packets only when the card is set in a special operating mode called the *RFMon* mode. Radios based on prism 2/2.5 set to the *RFMon* mode cannot transmit packets. So we modified the hostap driver extensively to get the prism monitor header information in the normal *Ad-Hoc* mode and computed the link bandwidth averaging the data rate of the packets received every second and communicated it to the AODV module. The ETT of a link is then computed using the ETX, link bandwidth and the size of the packet (set to 1024 bytes in our implementation).

C. IR Measurement

The prism monitor header added by the hostap driver to the received packets also consist of the signal strength and the background noise when the packet was received. When a node receives a HELLO packet from a neighbor, the driver stores the IP address, signal strength and the background noise values to an in-kernel buffer which is exported to the AODV module. The AODV module uses these values for the calculation of *IR*. In practice, the driver knows the signal strength values only for the packets that are received properly. But for the calculation of IR of a link, we also need signal strength information from nodes that are not within the transmission range but can cause interference. We get this information by sampling the prism card register 51, which provides the interference noise in the environment [17]. We also modified the hostap driver to report the sending rate (packets sent per second) of each interface to the AODV module. Every node computes its sending rate on each of its interfaces and communicates to its neighbors by piggybacking it in the HELLO packets it sends. So each node knows the sending rate of its neighbors and use it in the computation of *IR*. Note that our measurement technique does not use any extra probing packets except the normal HELLO packets used by AODV to maintain connectivity information.

V. TESTBED

The measurement results presented in this paper were taken on a 12 node wireless mesh testbed deployed in the Computer Science department at Stony Brook University. We use attenuators to reduce the transmission range of the nodes and get various multi-hop topologies within a small area. The other advantage of using attenuators is that the effect of external interference on the network is reduced and it is helps to understand the effect of interference in the network. We are also able to create various interesting topologies unlike the static testbeds used in other works [15, 28].

The nodes used in the testbed are the Soekris net4826 [7] boxes that comes with a 233/266 MHz AMD Geode SC1100 processor and 128MB memory. Each nodes is populated with two Prism 2 based mini-PCI cards which do not have an internal antenna. All the cards operate in the 802.11b mode. The nodes run *Pebble Linux* [6] with the Linux 2.4.26 Kernel. We connected a 10dBm attenuator between the mini-PCI card and an external antenna. This helps to reduce the transmission range significantly (approx. 4 feet) and create multiple collision domains in a small area. It also reduces the cross interference between the two cards in the same node. Without the attenuators, we observed significant interference between the two cards in the same node when the antennas are placed close to each other, even when they operate in orthogonal channels. Similar experiences have been reported in [22].

We use the hostap driver [18] to run both the cards in Ad-Hoc mode. All the cards are set the default configuration. Specifically, they all operate using the same transmission power, perform auto-rate control and have RTS/CTS disabled. We use channel 1 and 11 so as to get maximum separation between the

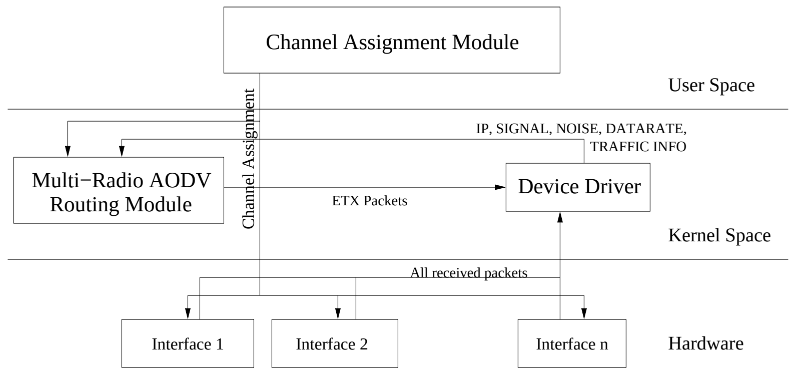

Fig. 4. Implementation architecture

frequency bands. All the measurements were taken when the activity of other 802.11b networks in the building were low to get more accurate results.

VI. Performance Evaluation

In this section, we present our early results in characterizing the performance of our new iAWARE routing metric and AODV-MR routing protocol.

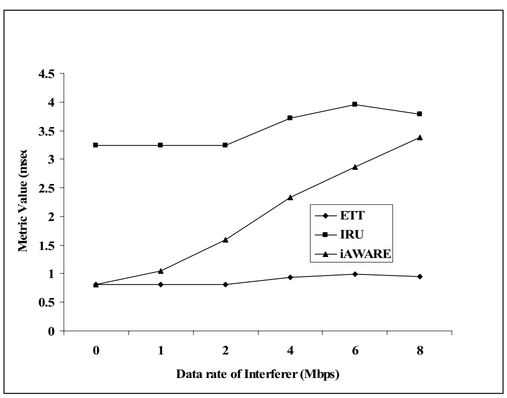

Fig. 5. Sensitivity of link quality metrics to interfering traffic

A. Sensitivity of Routing Metrics to Interfering Traffic

In Section II-A, we noted that existing link metrics such as ETX, ETT are insensitive to link traffic and presence of interfering traffic among neighboring nodes. Similarly, we observed that path metrics such as WCETT capture only intra-path interference and fail to capture inter-path (inter-flow) interference. In the following, our experimental results show how iAWARE models the quality of a link in the presence of varying interference.

In this experiment, we consider a controlled scenario wherein we study a single radio link between a pair of nodes in presence of interfering traffic among nearby neighbors. We have 6 nodes deployed in a single collision domain. We run a UDP flow at 8Mbps on the link we study. Initially there is no interfering traffic. We gradually increase the interfering traffic from 0 to 8Mbps and measure the throughput of the link we study and the metric values (ETT, IRU, and *iAWARE*) reported. Figure 5 shows how metrics ETT, IRU, and *iAWARE* behave in this case. The corresponding throughput of the link with varying interfering traffic is shown in the following table.

Interfering Traffic (Mbps)	Throughput (Mbps)
0	4.23
1	3.82
2	2.65
4	2.41
6	2.45
8	2.40

We can see that ETT remains almost constant except for a slight increase at very high load. This clearly shows that ETT does not capture the interference experienced by the link completely.The IRU metric (which is defined in Eqn 4 as part of MIC) tracks ETT as the number of neighbors is constant (=4). It over estimates the link metric even when there are no interfering traffic. Our new metric does quite well tracking the increased interfering traffic; it increases steadily as the interference increases. This can be observed from the measured throughput values and the *iAWARE* metric value. We observed similar trends in the metric values in the other links in the network.

B. Single Channel Single Radio Experiment

First, we show the performance of the different routing metrics when only one radio on each node is used and all of them are tuned to the same channel. For this experiment, we turned on one of the mini-PCI cards in each node and set it to channel 1 with auto-rate control.

As mentioned earlier, we have 12 nodes in our testbed. We have a total of 66 pairs of nodes in our testbed out of which we choose 25 pairs at random to study. We setup a 1 minute UDP transfer between them and measured the throughput and the path length of the routes chosen using different metrics. The sending rate of the UDP flow was fixed at 4Mbps. This flow was started in the presence of an interfering UDP flow (1 hop flow) between another random pair of nodes at 1Mbps.

This experiment was repeated for the 25 pairs of nodes using the metrics hop count, ETT, IRU and iAWARE. Figure 6 shows the average throughput of the 25 flows on routes found by the different metrics in the presence of the interfering flow. As expected hop count performs the worst selecting long high-loss links and resulting in poor throughput. iAWARE performs

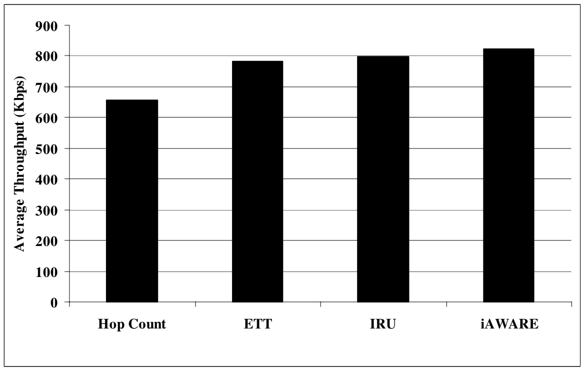

Fig. 6. Comparison of average UDP throughput in single channel case

the best among the tested metrics as it captures the interfering traffic in weighting the links so that it chooses links that are less affected by the interfering node. ETT does not capture the presence of the interferer and performs poorer than iAWARE by choosing links suffering more interference. IRU performed slightly better than ETT.

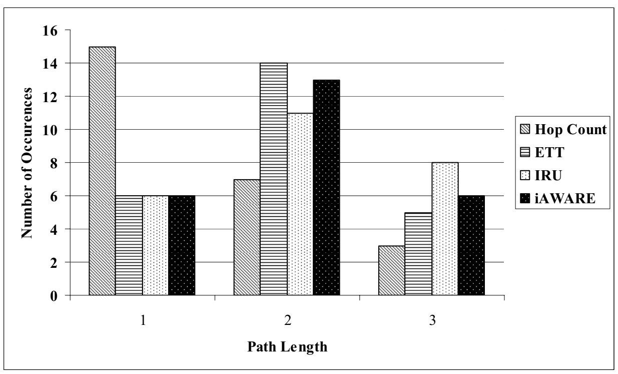

Fig. 7. Comparison of path length of routes found using different metrics in single channel case

To understand the performance in more detail, Figure 7 shows the path length chosen by each metric for the 25 paths computed. We can see that hopcount mostly chooses paths with single hops which have high loss ratios. In some case, we observed zero throughput in the paths chosen by hopcount metric. We can see that the number of 3 hop paths chosen by IRU is more than ETT and iAWARE. This is because, IRU assigns links with less interfering neighbors with a lower metric and so the boundary links are chosen compared to the links that have more neighbors.

The main observation from the above experiment is that, in the presence of interfering traffic, iAWARE performs better than ETT by capturing the inter-flow interference between the paths. In the absence of interference, iAWARE is no different that ETT as seen from the Equation 12. This baseline single channel experiment demonstrates the advantage of using iAWARE in comparison with the existing routing metrics.

C. Two Radio Experiments

In the previous section, we demonstrated the advantage of using iAWARE in a single channel single radio network. In this section, we present performance results of experiments when both radios in each node where turned on and tuned to orthogonal channels. We tuned the two radios to channel 1 and 11. We had proper antenna separation so that cross interference between the radios on the same node is less pronounced. The presence of attenuators also reduced the cross interference significantly.

We carried out the same set of 25 UDP flows but now in the presence of two interfering traffic one in each channel. We studied the performance of ETT, WCETT and iAWARE for this experiment. The value of α in Equation 2 and 14 was set to 0.5.

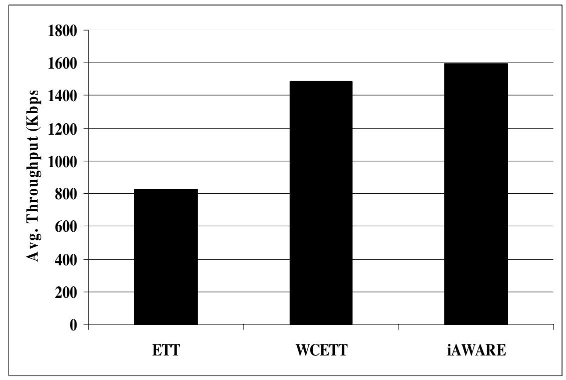

Fig. 8. Comparison of average UDP throughput in two channel case

The average throughput of the 25 flows in the presence of the two interfering flows is shown in Figure 8. WCETT and iAWARE clearly outperform ETT by properly exploiting the presence of multiple channels and finding paths with less intra-flow interference. iAWARE performs better than WCETT by choosing links not interfering with the background flows.

In order to show the channel diversity of the paths chosen by iAWARE we use a channel diversity index for each path as defined in [15] as follows.

$$CDI = \frac{min(N_1, N_{11})}{2 \times \lfloor N/2 \rfloor} \tag{15}$$

where N is the total number of hops in the path and N_1 and N_{11} are the number of hops in channel 1 and 11 respectively. For example, in a 2 hop path, if one hop is in channel 1 and the other in channel 11, the CDI is 0.5. In a four hop path, if one hop is in channel 1 and other three hops are in channel 11, then the CDI is 0.25. Note that the maximum value of CDI is 0.5.

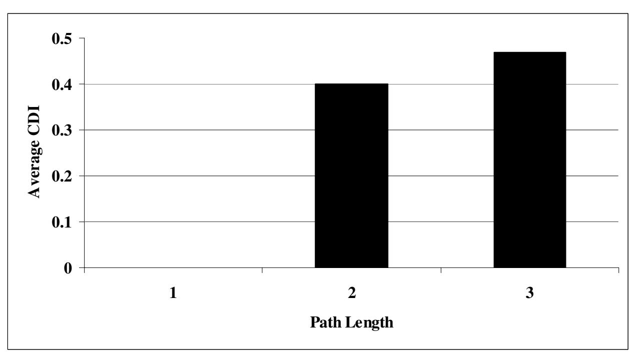

Fig. 9. Average CDI for paths found using iAWARE

Figure 9 shows the average CDI of the 25 paths found using the iAWARE metric. The CDI of a one hop path is zero. We would expect a CDI of 0.5 for the two hop paths. But we noticed that the CDI was about 0.41. The reason for this is the following. Though we used two orthogonal channels 1 and 11, we noticed that channel 11 had poorer loss characteristics compared to channel 1. This resulted in both iAWARE and WCETT to choose channel 1 for both hops for some of the two hop paths. The three hop paths had a CDI of 0.48. This shows how iAWARE chooses paths that have less intra-flow interference.

VII. Conclusion and On-going Work

In this paper, we addressed the problem of interference aware routing which is central to the design of high capacity multi-radio mesh networks. We presented a new interference aware routing metric – iAWARE that aids in finding paths that are better in terms of reduced inter-flow and intra-flow interference. We incorporated this metric and new support for multi-radio networks in the well known AODV routing protocol to design an enhanced AODV-MR routing protocol. We also described implementation of this protocol in Linux based mesh nodes and presented performance results from our wireless testbed consisting of 12 mesh nodes. Our experimental results showed superiority of our approach; specifically, we showed that in contrast to existing link metrics (e. g. ETT, IRU) and path metrics (e.g. WCETT, MIC), iAWARE tracks changes in interfering traffic, aids in delivering higher throughput and finds paths with good channel diversity.

We notice that our performance results are limited by the size of our test bed. Using only 12 nodes does not give us the opportunity to have large number of interfering flows and flows with longer hops. One of our ongoing work is setting up a larger realistic testbed and testing our routing protocol extensively. We are also currently investigating ways to incorporate the sender-side interference into the interference ratio and find better ways to correlate ETT and IR. In our experiment, the channel assignment of the radios was fixed through out the experiment. Understanding the interaction between the channel assignment and routing protocol is fundamental in designing high-capacity multi-radio mesh networks. One of our key research goal is to build a complete channel assignment and routing framework for high capacity multi-radio wireless mesh network.

References

[1] BelAir Networks. http://www.belairnetworks.com.
[2] Bell Labs MeshClusters. http://www.bell-labs.com/user/mbuddhikot/meshclusters/index.html.
[3] MeshDynamics. http://www.meshdynamics.com.
[4] Microsoft Research Mesh Networks Project. http://research.microsoft.com/mesh/.
[5] NIST AODV Implementation. http://w3.antd.nist.gov/wctg/aodv_kernel/.
[6] NYCwireless pebble linux. http://www.nycwireless.net/pebble.
[7] Soekris engineering. http://www.soekris.com/.
[8] StrixSystems. http://www.strixsystems.com.
[9] UCSB MeshNet Project. http://moment.cs.ucsb.edu/meshnet.
[10] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. ANSI/IEEE Std 802.11: 1999 (E) Part 11, ISO/IEC 8802-11, 1999.
[11] TIA/EIA/IS-835B - cdma2000 Wireless IP Network Standard. , Third Generation Partnership Program 2 (3GPP2), 2000.
[12] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks. In *Broadnets*, 2004.
[13] B. Bensaou, Y. Wang, and C. C. Ko. Fair medium accessin 802.11 based wireless ad-hoc networks. In *MOBIHOC*, August 2000.
[14] D.S.J. Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *MOBICOM*, 2003.
[15] R. Draves, J. Padhye, and B. Zill. Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks. In *MOBICOM*, 2004.
[16] P. Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 2000.
[17] HFA3863 Data Sheet:. Direct Sequence Spread Spectrum Baseband Processor with Rake Receiver and Equalizer. Intersil Corporation, 2000.
[18] HOSTAP driver. http://hostap.epitest.fi/.
[19] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of Interference on Multi-hop Wireless Network Performance. In *MOBICOM*, 2003.
[20] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing*, 353, 1996.
[21] A. Kashyap, S. Ganguly, and S.R. Das. Characterizing Interference in 802.11-based Wireless Mesh Networks . Technical report, Department of Computer Science, Stony Brook University, 2006.
[22] S. Liese, D. Wu, and P. Mohapatra. Experimental Characterization of an 802.11b Wireless Mesh Network. Technical report, Department of Computer Science, UC Davis, 2005.
[23] NIST Kernel AODV implementation. http://w3.antd.nist.gov.
[24] J. Padhye, S. Agarwal, V.N. Padmanaban, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *(IMC)*, 2005.
[25] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *SIGCOMM'94 Conference on Communication Architechture, Protocols and Application*, pages 234–244, 1994.
[26] C.E. Perkins and E.B. Royer. Ad hoc On-demand Distance Vector Routing. In *IEEE Workshop on Mobile Computing and Systems and Applications*, 1999.
[27] K. Ramachandran, E. Belding-Royer, K. Almeroth, and M. Buddhikot. Inteference Aware Channel Assignment in Mesh Networks. In *IEEE Infocom*, Barcelona, Spain, April 2007.
[28] K. Ramachandran, M. Buddhikot, G. Chandranmenon, S. Miller, E. Belding-Royer, and K. Almeroth. On the Design and Implementation of Infrastructure Mesh Networks. In *IEEE WiMesh2005*, San Jose, CA, Sept 2005.
[29] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks. *ACM SIGMOBILE MC2R*, 8(2):50–65, 2004.
[30] R. Raniwala and T. Chiueh. Architechture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM*, 2005.
[31] Roofnet. http://pdos.csail.mit.edu/roofnet/doku.php.
[32] J.L. Sobrinho. Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet. In *INFOCOM*, 2001.
[33] J.L. Sobrinho. Network Routing with Path Vector Protocols: Theory and Applications. In *ACM SIGCOMM*, pages 49–60, 2003.
[34] A. P. Subramanian, H. Gupta, and S.R. Das. Minimum-interference Channel Assignment in Multi-Radio Wireless Mesh Networks. Technical report, Department of Computer Science, Stony Brook University, March 2006.
[35] Tropos Networks. http://www.tropos.com.
[36] Y. Yang, J. Wang, and R. Kravets. Designing Routing Metrics for Mesh Networks. In *WiMesh*, 2005.