RxIP: Monitoring the Health of Home Wireless Networks

Justin Manweiler, Peter Franklin, Romit Roy Choudhury
Duke University

***Abstract*—Deploying home access points (AP) is hard. Untrained users typically purchase, install, and configure a home AP with very little awareness of wireless signal coverage and complex interference conditions. We envision a future of autonomous wireless network management that uses the Internet as an enabling technology. By leveraging a P2P architecture over wired Internet connections, nearby APs can coordinate to manage their shared wireless spectrum, especially in the face of network-crippling faults. As a specific instance of this architecture, we build RxIP, a network diagnostic and recovery tool, initially targeted towards hidden terminal mitigation. Our stable, in-kernel implementation demonstrates that APs in real home settings can detect hidden interferers, and agree on a mutually beneficial channel access strategy. Consistent throughput and fairness gains with TCP traffic and in-home micro-mobility confirm the viability of the system. We believe that using RxIP to address other network deficiencies opens a rich area for further research, helping to ensure that smarter homes of the future embed smarter networks. In the near term, with the wireless and entertainment industries poised for home-centric wireless gadgets, RxIP-type home management systems will become increasingly relevant.**

I. Introduction

The Enterprise WLAN (EWLAN) network architecture has gained rapid popularity in single-administrator environments, such as universities, airports, and corporate campuses. In EWLANs, multiple wireless access points (APs) are connected to a central controller through a high-speed wired backbone. The controller assimilates a centralized view of the network, facilitating coordination that would be difficult over the wireless channel alone. The overhead of coordination is offloaded to the out-of-band wired infrastructure, freeing the wireless spectrum for productive data communication. Deployment experiences show reduced hidden/exposed terminals [7], [24], greater spatial reuse [16], [24], smarter association [19], and a host of other enhancements to the end-user experience [1], [4], [6]. These techniques have proven practical, with commercial systems available from Aruba, Cisco, and Meru [17].

Unlike EWLANs, residential wireless networks (RWLANs) do not share a common, centralized infrastructure. Each residential AP is typically purchased, installed, and configured by the resident, without any type of interconnection to its neighbors [2]. The advantages of EWLANs are apparently unavailable. This paper investigates the feasibility of using the Internet as a wired backbone to coordinate residential APs.By exchanging their globally-routable IP addresses through wireless beacons, APs can be made to communicate with neighboring APs over the wired Internet. This *out-of-band* communication channel can emulate some of the EWLAN advantages in residential settings, and yet, preclude the need for a central controller. While numerous possibilities emerge, our first step is to narrow our exploration of this architecture to a specific application. We develop *RxIP* (*Prescription: IP*), a network diagnostic and recovery tool, targeted at hidden terminals.

Motivation and Measurements

The increasing availability of fiber-to-the-home and 100 Mbps+ cable access (DOCSIS 3.0) are transforming wireless networks [3], [11]. The bottleneck is no longer the ISP, but instead, the wireless network itself. High-bandwidth multimedia applications within the home are further reducing the slack: Apple TV, HD streaming, Apple Time Capsules, SONOS music systems, etc., are demanding significantly more capacity. When neighboring apartments simultaneously run these applications, the interference floor will far exceed what we have experienced in the past. While physical layer technology may keep up with this demand, we argue that atypical, unanticipated scenarios will be unavoidable. A network-wide health-monitoring framework, similar to those in enterprise WLANs, would be valuable to provide sustained stability.

Hidden terminal impact on network stability. Hidden terminal problems have been reported to impact network stability, especially in view of TCP [13], [21], the dominant component of residential traffic [15]. The problem manifests into severe packet loss, to the extent that 802.11 link-layer retries do not ensure successful packet delivery. TCP experiences these losses, and assumes severe network congestion. In response, it overzealously reduces its congestion window, yielding intolerably-poor performance. Measurements confirm hidden terminal impact in enterprise networks [9], [24]. Typical residential settings exacerbate the problems; APs may be located far from the wireless devices, and may often be placed on or near the floor, increasing multipath, weakening links, and lowering bitrates [21]. As bitrates drop, channel occupancy inflates, and correspondingly, the probability of packet collisions increases.

Validating hidden terminal impact via measurement. To validate these observations from existing studies, we set up our own experiments in student apartments. In two nearby condos, we placed the APs near coax (cable) ports to mimic real-life deployments. While both APs transmitted TCP traffic to their respective clients, we systematically moved one of the clients to 100 different positions, with the aim of creating a throughput contour under hidden terminal interference.

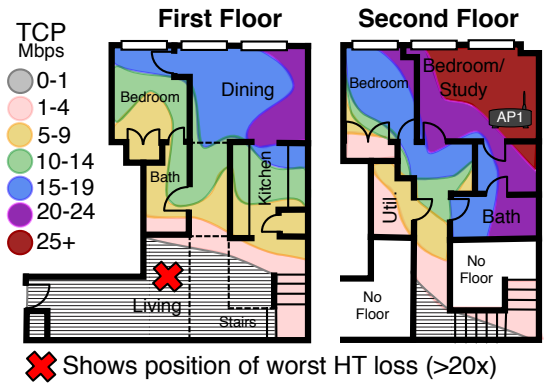

Fig. 1. TCP download throughput contour on two floors of the same apartment. A hidden terminal is placed in an adjacent apartment (not shown). Removal of the interference provides 8-12 Mbps in the living room (versus <1 Mbps shown).

Figure 1 illustrates a steep throughput degradation as the client moves away from its AP, becoming more susceptible to interference from a hidden terminal. Turning off the hidden interferer consistently improves the situation (more than 20x in the worst-interfered regions).

Figure 2 shows that hidden terminals are not pathological scenarios, and may arise with reasonably-high probability. As AP2 is moved further away from the AP1→C1 link, AP2 becomes a hidden terminal for AP1 and causes severe losses (from 5m to 35m on the x-axis, beyond which AP2 no longer creates significant interference). In our home networks today, we do not experience such impacts from existing hidden terminals [22]. The excessively-slow DSL/cable bottleneck trivializes wireless utilization, hiding the problem.

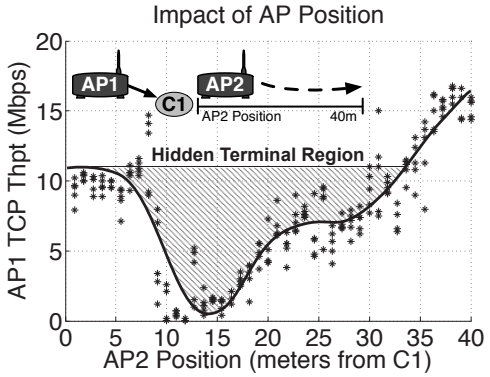

Fig. 2. As AP2 is moved away from the AP1→C1 link, graph shows decreasing and then increasing performance for AP1. AP2 becomes a hidden terminal at 5m, causing significant losses up to 35m away.

Figure 3 shows how mobility complicates hidden terminal outcomes, destabilizing interference relationships. We see how a client's movements (C2) dramatically affects throughput for the *other* hidden terminal link (AP1→C1). As a client moves throughout the home, its link may become stronger or weaker, depending on the distance to its AP. When its link strengthens, a reduction of packets losses makes its TCP sessions more aggressive (i.e., increased congestion window), increasing the channel occupancy. Correspondingly, clients associated to a neighboring AP may suffer, due to an increased probability of hidden terminal collisions. Performance reductions can be drastic (Figure 3). Effective recovery mechanisms must be responsive to client mobility.

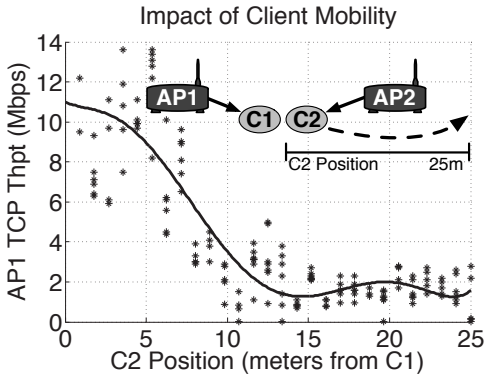

Fig. 3. As C2 moves towards its AP, it becomes less susceptible to hidden terminal interference from AP1. TCP more-fully utilizes the channel, and correspondingly, C1 is severely impacted by AP2.

Existing recovery mechanisms are insufficient. Enabling 802.11 RTS/CTS hidden terminal protection (for long packets) substantially reduces overall throughput [12]. ZigZag decoding [10] has been shown to be an effective hidden terminal mitigation system in software radio testing. However, without a practical implementation, it is not readily deployable in WLANs with commodity hardware and legacy clients. Centaur [24] successfully isolates hidden terminal traffic, but is only relevant to EWLANs with a low-latency backbone and central controller. Residential networks do not have such backbones, and suffer from the lack of coordination among chaotically placed APs [21]. Interference-aware, planned deployment is not an option, as lay users must be able to install these devices with plug-and-play simplicity. The onus is on network solutions to provide stable performance under strict constraints.

Out-of-band opportunities are available. We observe that the wired Internet already connects the majority of home APs, and the opportunity can be utilized to coordinate their operations, out-of-band. Of course, challenges naturally arise, including coping with Internet latencies, time synchronization, accurate fault diagnosis, and mechanisms for quick recovery. We systematically address these complications, moving towards a more stable RWLAN. While the efficiency of an ideal (enterprise-like) deployment may not be possible in unplanned networks, coordination-enabled, automatic network refinements may bring a network with unacceptable performance to adequacy. Besides, these solutions are practical without any client-side modifications. By remaining invisible to devices and users, the system can retain the necessary plug-and-play simplicity.

Our contributions in this paper are three-fold:

1) *We identify the Internet as a viable control plane for coordinating wireless APs in home networks.* While the core idea is not entirely novel [2], we believe that

our application and implementation in the context of residential networks enables new opportunities.

2) *We develop RxIP, a distributed hidden terminal diagnostic and recovery service.* Internet-based coordination enables cooperative mitigation among neighbor APs under TCP traffic and in-home mobility.
3) *We implement RxIP as a Click Router kernel module and experimentally characterize its performance with testbeds up to 12 nodes.* Results show a median throughput improvement of 57% against 802.11 (with RTS/CTS turned off) in symmetric hidden terminals, while also improving fairness.

II. RxIP Architecture

This section presents a high level overview of the system, followed by an outline of the underlying components. The design details are presented thereafter.

RxIP APs periodically announce their Internet IP addresses, through wireless beacons. Neighboring APs overhear these beacons and relay them one additional hop to ensure that they are received by potential hidden terminals. When APs learn about the presence of a new neighbor, they send a wireline probe to the specified IP address, establishing a control channel over the Internet.

To mitigate hidden terminal problems, RxIP relies on this direct, AP-to-AP coordination. The main idea is that APs monitor their wireless performance and periodically cross-check with nearby APs over the Internet. Bloom filters efficiently maintain the history of transmission timestamps at each AP, facilitating timing analysis for hidden terminal diagnosis. An observed correlation between two APs' transmission times (matched over the Internet) and collision rates (observed over wireless) raises suspicion of a hidden terminal scenario. Confirming the suspicion, hidden APs establish pair-wise partnerships to relieve the effects. Mitigation happens through a hybrid TDMA/CSMA schedule, implemented via token exchanges. The token exchange mechanism is designed to scale for complex interference relationships, ensuring that hidden APs never transmit during the same timeslot. The latency in Internet-based coordination is addressed by scheduling transmissions slightly in advance. APs that are not affected by hidden terminals continue their operations unaltered. Relative to unassisted CSMA, performance improves due to reduced collisions, fewer TCP disruptions, and higher bitrates. Moreover, Internet-based coordination frees up wireless bandwidth for productive data communication. We prove that coordination is correct and efficient in Section IV and present experimental results in Section VI.

Incentives. Since residential APs typically do not share a common administrative domain, they should be incentivized into protocol compliance only by service improvements for their own clients (we assume that APs may be selfish, but are non-malicious). Our distributed scheduling mechanism allows each AP to individually select precisely those peers with which it wishes to serialize its transmissions. Serialization is only required when both parties agree. By consensus-only serialization and peer monitoring to disincentivize cheating, we attempt to maintain incentive-compatibility for all APs.

In Figure 4, we consider the two cases that warrant coordination. In a *symmetric* hidden terminal (AP1 and AP2), each AP appreciably interferers with its peer's client. Losses are roughly equitable; coordination provides immediate gains for both APs. In an *asymmetric* hidden terminal (AP3 and AP4), one AP has an advantaged position. The weaker link (AP4→C4) experiences excessive loss, leading to disproportionally-reduced congestion windows for TCP flows. The disparity between the strong and weak link becomes severely exaggerated. In such cases, the strong link may still consider coordination as incentive compatible, if there is an expectation that there might be a role reversal in the future. This may occur over long timescales with client mobility throughout the home and environmental changes (e.g., a closed door), or in short term due to stochastic fluctuations in the wireless channel.

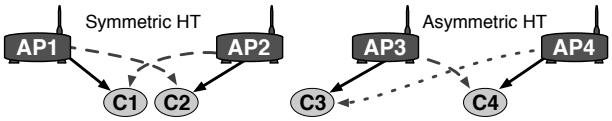

Fig. 4. Hidden terminal conditions.

Time Synchronization. In lieu of a global clock, RxIP APs maintain logical time synchronization with their coordination partners. Through periodic beacon reception and hardware timestamping, we maintain microsecond-granularity precision on a pairwise basis. When an 802.11 beacon is overheard, APs subtract the beacon TSF timestamp from the local TSF time of beacon reception, determining a clock offset. This passive technique does not interfere with existing AP-to-client 802.11 TSF synchronization, allowing the AP to maintain control over existing time-sensitive operations within its own BSS. Two-hop synchronization with hidden terminals is also feasible. For each of an AP's one-hop (directly-synchronized) peers, the AP uses the Internet to forward its synchronization offsets to all other one-hop peers. In Section VI-B, we evaluate synchronization precision.

III. Hidden Terminal Diagnosis

In this section, we divide the details of hidden terminal diagnosis into two subtasks: (1) ensuring that hidden terminals are the cause of performance degradation; and (2) isolating the particular hidden terminal at fault.

A. Ensuring Hidden Terminals are the Cause

Performance fluctuations are common in wireless networks, and may not be always attributable to hidden terminals. In light of this, we suggest checking for two conditions that indicate early evidence.

(i) Due to channel *reciprocity*, most AP-to-client links should exhibit a rough symmetry in their upload and download characteristics. The symmetry may be observable in the bitrates selected by 802.11 (e.g., for downlink DATA and

uplink TCP ACKs), and even in the delivery ratio of packets in each direction. However, hidden terminals are likely to induce stronger *asymmetry*. Downlink traffic to client C1 (Figure 4a) may suffer due to hidden terminal AP2, while the uplink transmissions from C1 to AP1 may retain a high delivery ratio/bitrate. Observing asymmetry could be a sign of nearby hidden terminals.

(ii) Received signal strength (RSSI) of client-transmitted packets, overheard at a neighboring AP, may be another indicator. In Figure 4a, AP2 may overhear C1's ACKs with reasonably high RSSI, but may not overhear AP1's DATA transmissions. Again, assuming rough channel symmetry, C1 is also likely to experience a strong RSSI from AP2, indicating the possibility of hidden terminals. Of course, it is important to ensure that AP1 is not within carrier-sensing range of AP2 (in which case they are not hidden). For this, AP2 can check whether it overheard AP1's wireless beacons in the past. If AP2 discovers that it has received AP1's IP address only through a *two-hop relayed beacon* (not from overhearing), then the evidence for a hidden terminal is stronger.

The above two conditions may not be conclusive; each test may incur false positives. Residential environments may exhibit inherent channel asymmetry [21]. APs formerly outside mutual carrier sense range (during beacon transmission) may no longer be hidden, due to channel variation. Even if the cause of performance degradation is indeed due to hidden terminals, an affected AP needs to accurately identify the culprit. Thus, triggered by the above symptoms, we propose a refined analysis, targeted to concretely isolate the specific hidden terminal.

B. Isolating the Hidden Terminal

A RxIP AP records timestamps for each packet it has transmitted in the recent past, allowing peers to determine when it has transmitted concurrently. A fixed-sized bloom filter can be used as an efficient data structure. When hidden terminal conditions are suspected, an AP initiates a challenge-response protocol with peer APs over the Internet. Each AP queries its peers with a suitably chosen timestamp (the choosing scheme will be discussed soon). Timestamps have millisecond-granularity, effectively slotting time into approximately packet-sized intervals. The peers convert the timestamp to local time (using up-to-date logical time synchronization), consult their bloom filters, and *report* back whether they performed a concurrent transmission at that time. APs maintain a *saturating counter* for each peer AP. For each received report, one of the following four cases results.

1) When an AP suffers a loss and a concurrent transmission is reported, the AP *increases* the counter for the peer by a *large* increment (collision).
2) When an AP transmits a packet successfully and a concurrent packet is reported, the AP *decreases* the counter by a *large* increment (no collision).
3) When the AP suffers a loss and no concurrent transmission is reported, the AP *decreases* the counter by a *small* increment (no concurrency).
4) When an AP transmits a packet successfully and no concurrent packet is reported, the AP *decreases* the counter by a *small* increment (no concurrency).

When a counter saturates high, the AP deems the peer to be a hidden terminal. If a counter desaturates for a peer with which no partnership is active, it is no longer considered a hidden terminal. Once a partnership is formed, counter desaturation reflects an expected alleviation of hidden terminal effects. To account for dynamic network conditions, especially those caused by client mobility, partnerships may be periodically disabled to check if the hidden terminal condition still exists.

Bloom Filter Operations. To answer challenge-response probes, our timestamp data structure needs two operations, `ADD` (to insert a new timestamp) and `CHECK` (to test if a queried timestamp has been inserted previously). Because of its constant-time efficiency, a bloom filter is particularly well suited to this purpose. The bloom filter is maintained as a pair of bit arrays, initialized to 0. In a rotating fashion, one array is designated as `CHECK/ADD` and the other as `CHECK-ONLY`. During an `ADD` or `CHECK`, a timestamp is run through an MD5 hash, producing a 128-bit expansion. This digest is split into 8 values, simulating 8 independent hash functions. Each of the values serve as indices into the bit array. In an `ADD`, the corresponding bits in the `CHECK/ADD` array are set. In a `CHECK`, "yes" is returned if all 8 bits are set in *either* array, "no" otherwise. Once the `CHECK/ADD` array becomes saturated after many `ADD` operations, the `CHECK-ONLY` array is reset and is swapped with the `CHECK/ADD` array. In our implementation, a pair of 4096-bit (512-byte) arrays provide a false positive probability bounded by ≈ 0.05.

Preventing Misbehavior. In scenarios as in Figure 4b, AP4 may have an incentive to trick AP3 into believing that AP4 is a hidden terminal for AP3's client, C3. In reality, only AP3 is a hidden terminal to AP4's client – we call this an *asymmetric hidden terminal* condition. AP3 can prevent deception by AP4 through careful selection of challenge-response probes. Importantly, this is possible even while AP3 simultaneously experiences hidden terminal losses from other APs. Specifically, AP3 should choose its probing timestamps from both successful as well as failed transmissions (in a roughly-equal mix). Unless AP4 can guess which of the probes are from failed packets, it will not know when to "lie" that it was also transmitting concurrently. Random guesses are likely to cancel out on average, leaving AP3's saturating counter for AP4 unaffected. Thus, it is consistent with AP4's best interest to respond truthfully to AP3's probes, ultimately allowing AP3 to make a correct determination.

IV. Recovery by Coordination

RxIP mitigates hidden terminals through Internet-based coordination. The idea is to rotate channel access rights between the hidden interferers such that no two interferers transmit concurrently. Importantly, APs may experience multiple hidden interferers, together resulting in an interference graph. This section describes how RxIP coordinates APs over

this interference graph, ensuring deadlock-free operation, high channel utilization, and robustness to Internet latencies and dropped packets.

Once a pair of APs diagnose a hidden terminal fault, they may respond by establishing a channel *token*, to be passed back and forth. As in many existing token-based schemes, such as JazzyMac [20] in the wireless domain, token passes serve as a scheduling mechanism. At any time, only one AP is the *token bearer*. The other AP, that does not have the token, is free to transmit indefinitely. Unlike traditional token-based access control, the token bearer does not have the channel access right. Instead, it has the right to "purchase" a transmission timeslot on demand. In that sense, tokens are like money: a token bearer can buy a timeslot by giving the token to its counterpart. The counterpart now becomes the token bearer, and is able to purchase a subsequent timeslot with the same token. In this manner, interfering APs may reserve alternate timeslots in the future. Under certain circumstances, the token bearer may choose not to purchase the next timeslot. Instead, the token bearer holds on to the token and sends an *abstain notification*. The counterpart can then transmit during the "abstained" slot. Figure 5 shows a simple two-AP exchange.

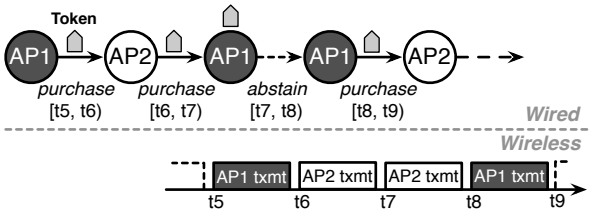

Fig. 5. Timeline of wired token exchange and wireless timeslots. AP1 purchases timeslot t5 to t6 by giving the token to AP2. AP1 may not be able to transmit at t7 (due to some other partnership, not shown). AP1 abstains from a token pass at t7, allowing AP2 to transmit. However, AP1 silences AP2 at t8 instead.

Coping with Internet Latencies. Every round of token-exchange on the Internet reserves the channel for some wireless transmissions in the near future. When the future time comes, the owners of the reserved timeslot transmit their data packets. One issue is that token-exchanges incur Internet-scale latencies, and if they are not fast enough, they may not be able to "stay ahead" of the actual wireless transmissions. To avoid this possibility, we choose our timeslot durations to slightly exceed the average token passing time (i.e., half of the RTT between APs). As we validate experimentally, longer timeslots do not impact any AP's long-term bandwidth share or aggregate throughput. However, delivery latency is adversely affected. TDMA schemes are known to incur higher latencies under light traffic conditions (unlike CSMA, a TDMA transmitter will need to wait for its turn) [23]. This increased latency correlates to the timeslot duration. Results in the next section show latency with varied, realistic slot durations.

Multiple Partnerships. Token passing becomes non-trivial when APs simultaneously need to partner with multiple hidden terminals. A channel access token is associated with each partnership. To transmit during a particular timeslot, an AP must satisfy the following requirement for *every* partnering AP: it either purchases that timeslot by expending the channel token to that partner, or receives an abstain notification for that slot from the partner. This ensures that all partners will remain silent for that slot. Importantly, this implies that an AP needs to gather all of its tokens, and spend them *simultaneously* to purchase the timeslot. Figure 6 illustrates the interactions between pairwise exchanges, and how each AP fairly receives its channel access rights. The movement of tokens between partnered APs schedules cyclical non-overlapping timeslots.

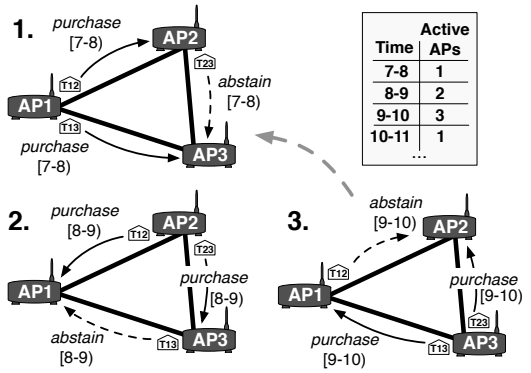

Fig. 6. Rotating channel access rights, established by token exchanges across multiple partnerships.

Provable Properties of Coordination

Each RxIP partnership is an agreed contract between a pair of APs. We may express these terms as axioms, and use them to prove desirable properties about our system.

1) APs that receive the token from a token bearer may not transmit.
2) Token bearers that keep a token may not transmit.
3) An AP may transmit precisely when it receives no tokens and spends all held tokens.
4) The token bearer in a new partnership must be the bearer in all of its partnerships.

Based on these axioms, we have proven the following properties. In the interest of space, we state these without formal proof, only providing a sketch for our most important guarantee, *token passing will not deadlock*. We provide a technical report with the complete details.[1]

1) Protocol operation is *deadlock-free*.
2) An AP waits no more timeslots between transmissions than the number of coordination partnerships in which it is engaged.
3) A partnership between a pair of APs only induces silence if one is actually allowed to transmit.
4) Token passing implements optimal graph coloring for connected bipartite partnership graphs.

Proof Sketch for Deadlock-free Operation. Let $G(V, E)$ denote the directed coordination partnership graph where V is the set of APs and E is the set of coordination partnerships. Let $e \in E$ denote a directed edge from the token bearer to the non-bearer in a partnership. By the enforced partnership

[1] Available: http://bit.ly/b1507m

establishment procedure, the requested AP in a new partnership, $v \in V$, must have only outgoing edges. v cannot be in a cycle, thus the new partnership could not have created a cycle. Similarly, after a token pass, v has only incoming edges. Thus, v cannot be in a cycle, and so the corresponding partnerships cannot create a cycle. Neither partnership establishment nor token passing can create cycles in G, thus G is constructed and maintained acyclic. An acyclic graph must contain some vertex v with only outgoing edges. In G, this corresponds to an AP that is the token bearer in all partnerships. This AP may pass its tokens and transmit. Since the graph remains acyclic across token passes, some other AP must now have all tokens.

V. Additional Considerations

Coping with Token Loss. Tokens can be lost due to a number of pitfalls: APs may fail or become disconnected; packets may be lost or incur arbitrary delays and reordering; and non-compliant behavior can cause deadlock. APs continually monitor their partnerships for deadlock scenarios. All partnerships that could be at fault are temporarily severed and formed anew using the correct establishment procedure. Meanwhile, regular CSMA provides a natural fallback.

Address Translation. Network address translation (NAT) may apparently impose some difficulty in partnership establishment, since each AP must effectively act as an Internet-accessible server. In residential deployments, however, the AP itself typically serves as a NAT device and has a globally-routable IP on its gateway interface. In rare scenarios with an independent NAT device or multiple APs per home, UPnP (Universal Plug-and-Play) allows automatic configuration for NAT port forwarding.

Upload Traffic. In establishing TDMA schedules, we have not provided explicit scheduling for upload traffic. While this could be achieved with our architecture, complete scheduling would mandate client modification. Moreover, for download TCP traffic, there is greater benefit to protecting TCP data (received at the client) than ACKs (at the AP). TCP cumulative ACKS are highly redundant, as each ACK packet acknowledges every preceding received byte since the start of the session. TCP is only affected when multiple, consecutive ACKs are lost. Thus, hidden terminals among APs are more damaging than among clients for download flows. Given the predominance of download traffic in home networks (85% of residential broadband [15]), the potential gains from upload scheduling seem less compelling.

Incremental Deployability. In RWLANs, nearly all APs represent independent administrative domains. Thus, a practical system must be incrementally deployable. Our solution requires no changes to 802.11 clients. CSMA contention mechanisms still operate normally. Simply, no partnership are established with non-compliant APs. At worst, performance degrades to traditional 802.11.

VI. Evaluation

We take a systems-oriented approach in evaluating RxIP. Our prototype implementation provides the full functionality of our scheme, including (1) automated AP peer discovery; (2) precise two-hop time synchronization; (3) hidden terminal inference using link asymmetry, peer feedback, and bloom filter-based transmission timing analysis; and (4) maintenance of hybrid TDMA/CSMA schedules using token passing.

Our evaluation consists of three main analyses:

1) We characterize the ability of our system to *automatically detect, isolate, and recover* from hidden terminal scenarios.
2) We use a series of microbenchmarks to quantify important *performance attributes of our design and implementation*, including time synchronization and an ability to cope with Internet latencies.
3) We subject our system to larger (6-AP) topologies with an *inflated number of hidden terminals*. Performance gains over 802.11 reflect the robustness of our coordination-based TDMA and an ability to adapt to adverse network conditions.

Testbed Platform. We evaluated our system on a testbed of laptops, serving as APs and clients. Laptops were configured with Linux kernel 2.6.24.7, Intel Core 2 Duo CPUs, and Atheros chipset D-Link DWA-643 ExpressCard WLAN interfaces. For some UDP experiments, Soekris embedded PCs, configured with Metrix Pyramid Linux and Atheros 5213 chipset MiniPCI interfaces, served as supplementary clients. We implemented our system through in-kernel element extensions to the *Click Modular Router*. For precise TDMA schedule execution, we modified the MadWiFi 802.11 driver to provide Click interfaces to (1) access the TSF clock; (2) block the transmission queue and buffer waiting packets; and (3) transmit buffered packets and re-enable the transmission queue. We use 802.11b/g as there is not yet reliable 802.11n Linux driver support for our hardware. To consider the effectiveness of our approach under realistic bitrate conditions, all nodes use the popular SampleRate [5] loss-based bitrate selection heuristic.

Methodology. Our tests assume the wireless link to be the bottleneck. We compare our system against standard 802.11 DCF using `Iperf`, a widely-distributed network benchmarking tool. Only AP-to-client, download, traffic is considered. However, TCP results reflect the interaction of bidirectional traffic. Throughput, fairness, and jitter results are as directly measured by `Iperf`. Virtual carrier sense (RTS/CTS) is disabled for all tests.

A. Hidden Terminal Diagnosis and Recovery

We test system effectiveness in (i) symmetric hidden terminal conditions, (ii) asymmetric hidden terminal conditions, (iii) in varied interferer positions, and (iv) across client mobility for the interferer. RxIP provides stable performance across adverse hidden terminal conditions.

Symmetric Hidden Terminals. *We show that RxIP substantially improves performance for both links in symmetric hidden terminals.* For these tests, two APs are placed outside of mutual carrier sense range, creating the hidden terminal.

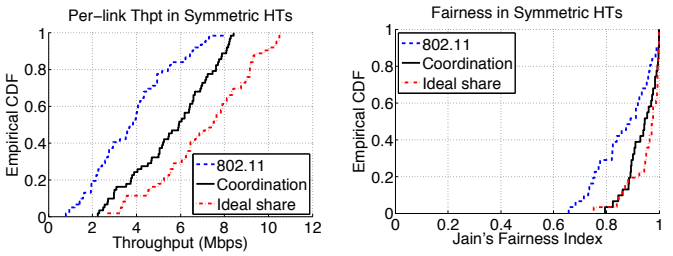

Fig. 7. (a) TCP provides a median 57% gain over 802.11 under symmetric hidden terminals. (b) RxIP extracts the majority of available gain. (c) Despite the already-symmetric conditions, RxIP further improves fairness.

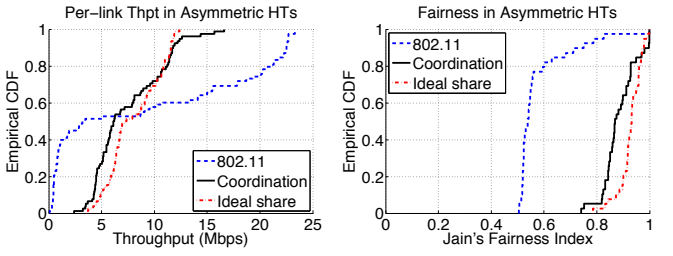

Fig. 8. TCP throughput and fairness under asymmetric hidden terminals. (a) Coordination balances the asymmetry, closely approximating an ideal 50-50 channel share. (b) Fairness improves dramatically.

Each AP has a single associated client, placed symmetrically in between, providing similar AP-to-client and interferer-to-client channel qualities for each link. A third AP serves as a relay for time synchronization. APs rely on automated hidden terminal inference mechanisms to request and accept partnerships. While the resulting topology exhibits typically-symmetric performance characteristics, channel fluctuations exacerbated by TCP congestion window throttling, occasionally break symmetry. When one link suffers a period of disproportionate loss, it cuts its TCP congestion window by an excessive margin. The other link, benefiting from the now-clearer channel, experiences a loss reduction and correspondingly increases its window.

Figure 7 (a,b) presents our results with TCP download traffic. In these symmetric conditions, we find a mean 53% (median 57%) throughput gain from coordination, with 91% of links experiencing an improvement. Despite the already-symmetric topological construction, fairness improves by a mean 8%. This is expected, as hidden terminals render 802.11 backoff ineffective.

Asymmetric Hidden Terminals. *In an asymmetric hidden terminal, when the strong link agrees, coordination can provide both links stable performance (Figure 8).* In asymmetric conditions, one AP's link suffers such severe losses that TCP fails to saturate the link, receiving only negligible throughput. The other AP gains a clear channel. Given this extreme condition, the advantaged AP may still be willing to enter a partnership if there is an expectation of future role reversal (e.g., from client mobility, discussed later). Coordination in asymmetric hidden terminals may be expected to decrease aggregate network throughput, at the gain of far-greater fairness and longer-term stability. Bandwidth formerly monopolized by a high-rate link is partially reallocated to the weaker link. However, this effect is lessened in practice, as coordination may reduce losses on both links.

We test asymmetric hidden terminals as in the symmetric case, except that APs are configured to participate in partnerships if there is a gain for *either* peer. We conduct these tests in an apartment complex. In one apartment, we position an AP at the cable point-of-presence in a study and a client in the common room (the weak link, Figure 1). We place a second AP and client in an adjacent apartment space (a large shared lobby area), serving as a strong link. In Figure 8, we see how coordination redistributes channel access to closely match an ideal 50-50 share. Compared to the symmetric case, we see greater efficiency as partnerships are entered freely, reducing the number of losses during fault detection.

Interfering AP Location. *RxIP coordination prevents hidden terminal losses, irrespective of the interfering AP's location.* Figure 9 shows stable performance, irrespective of interferer location, as opposed to extreme highs and lows with 802.11-based wireless coordination alone.

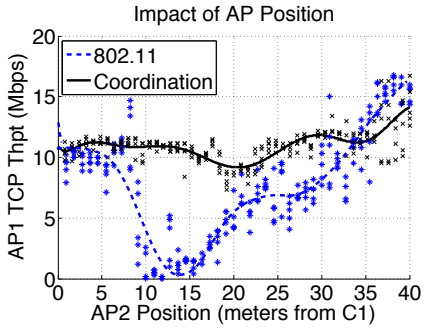

Fig. 9. RxIP protects the AP1-C1 link from performance degradation regardless of AP2 position.

Client Mobility. *RxIP alleviates TCP-imposed instability, caused by the neighbor's client mobility.* Figure 10 shows client C2's movement dramatically affecting throughput for the *other* hidden terminal link (AP1→C1). This may apparently seem counter-intuitive, but as losses impact TCP, channel occupancy inflates, and other links are correspondingly affected. With RxIP, coordination protects both links, ensuring stable performance at all client locations. In C2 positions 0-6m (X-axis), AP1 sacrifices some channel access time to AP2 (an asymmetric hidden terminal with AP1 as the stronger link). In exchange, AP1 is protected when the AP2 link strengthens (i.e., C2 moves to positions 6-20m).

B. Microbenchmarks

Internet Latency. *RxIP is compatible with Internet-scale latencies, shown through emulation of realistic RWLAN conditions.* By artificially delaying all coordination traffic, we match the link characteristics of 768 Kbps upload broadband connections for each AP with varied AP-to-AP RTTs. We select our timeslot conservatively, at 1.25X the one-way (half-RTT) imposed latency between partnered APs plus 5ms (for non-emulated delays). APs schedule token passes in advance by twice the timeslot duration. We deploy a 3-AP topology and enforce that all APs partner together. We

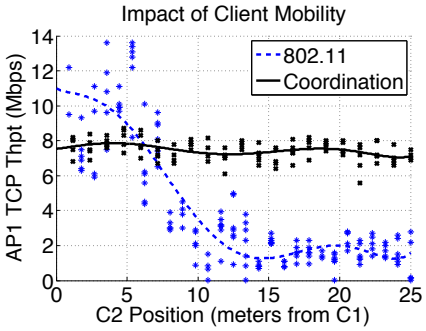

Fig. 10. As C2 moves from position 0 to 20m, its link strengthens, becoming less susceptible to hidden terminal interference from AP1. TCP more-fully utilizes the channel, and correspondingly, C1 is severely impacted by AP2. Coordination protects both links.

validate that throughput is stable across all artificially-varying Internet RTTs (drawn from apartment-complex measurements, Figure 11a). We report AP-to-client delivery latency as the metric of interest in Figure 11b. For reference, residential measurements in [8] show a median last-hop delay of ≈7ms/13ms for cable/DSL. We observe a mean RTT of 21.5ms.

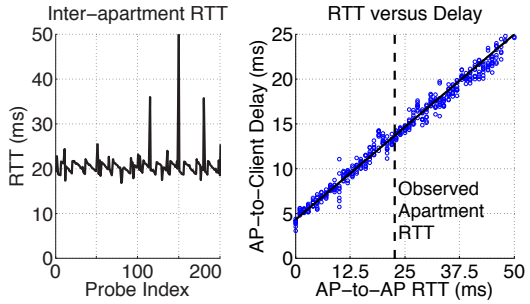

Fig. 11. (a) RTT between APs across an apartment complex using 1.5Mbps cable. (b) AP-to-client delivery latency exhibits a linear relationship to the Internet RTT between partnered APs (2x AP-to-AP delay).

Two-hop Time Synchronization. Beacon timestamps allow APs to maintain μs-granularity synchronization with one-hop neighbors. To maintain time synchronization to a hidden terminal with a tenuous or nonexistent wireless link, we utilize an intermediate AP, within one-hop range of both APs individually. By combining two direct synchronization clock offsets, an AP derives a logical synchronization across two hops. To evaluate the precision of two-hop time synchronization, we deploy a three AP topology with all APs in single-hop range. For this test, we use an exceedingly-long 500ms beacon interval, increasing staleness to strain our system. To determine the loss of accuracy imposed by the addition of a second hop, we compare the synchronization offsets determined by one-hop and two-hop synchronization mechanisms. We find a mean difference of 1.5μs with a standard deviation of 1.2μs and max of 5μs (Figure 12). We expect this to be representative of additive error across each hop of a multi-hop synchronization. Therefore, in a typical hidden terminal scenario, we anticipate mean total error to be bounded by $5 \cdot 2 = 10\mu$s. Thus, our synchronization facilities are more than sufficient for hidden terminal analysis, using timing to find concurrent packets.

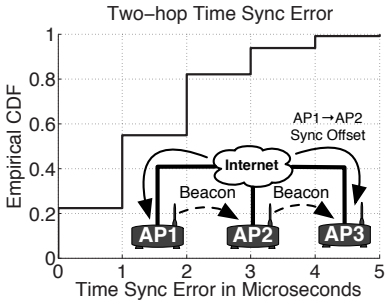

Fig. 12. (Inset) Intermediate APs relay clock offsets for time synchronization between hidden terminals. (Graph) Second-hop time synchronization error attributable to wired relay mechanism latency.

C. Scalability of Partnership-based TDMA

It may be unlikely that a real-world AP would encounter enough hidden terminals to necessitate many concurrent partnerships. Indeed, it is difficult to create such a scenario with the limited number of nodes available in our testbed. However, we wanted to evaluate the scalability and robustness of our system under such an adverse environment. To this end, we modified our APs to disable carrier sense and deployed them in dense topologies. By creating an extreme proportion of hidden terminals (artificially), these tests necessitated many partnerships, providing greater system strain. Under these conditions, reported performance results are *not in any way intended to be representative of a deployed system. Instead, performance enhancements are reflective of an ability of the system to adapt to more complex partnership formation.*

Methodology. With carrier sense disabled, bidirectional traffic, including both TCP and link-layer acknowledgments, induces many collisions irrespective of external interference. Therefore, we consider unidirectional flows without link-layer ACKs (broadcast UDP traffic with MTU-sized datagrams). Since effective rate control is difficult without per-packet feedback, we use a fixed 12 Mbps bitrate. Transmission timing analysis for hidden terminal detection is similarly not possible (the AP cannot isolate which packets may have collided). Instead, APs rely only on peer RSSI feedback regarding occasional client upload packets. For regular 802.11, we leave carrier sense *enabled* and topologies under test have few, if any, natural hidden terminals. Thus, we consider the extent to which coordination mechanisms can be as effective as 802.11 in scheduling channel access.

6-link Testbed Benchmarks. We deployed 6 APs and 6 clients into 30 distinct topological configurations within our university facility. APs and clients were randomly dispersed in varied dense configurations. In Figure 13a, we present a CDF of per-link throughput (1.8X mean aggregate throughput gain over 802.11). Figure 13b shows a 2.5ms improvement in mean jitter. Finally, Figure 13c shows that fairness is not negatively impacted by the coordination approach. We achieve a mean Jain's fairness index of 0.78, compared to 0.76 for 802.11.

With fairness and jitter improvements simultaneous to appreciable throughput gains, these tests reflect an ability of coordination-based TDMA to efficiently partition channel ac-

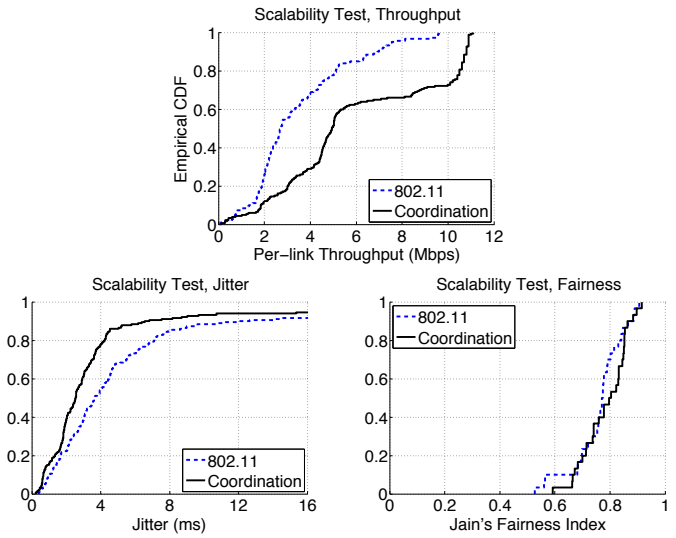

Fig. 13. Scalability test, 30 random 6-link topologies. CDF (a) throughput, (b) jitter and, (c) fairness.

cess. While we expect that throughput gains are primarily attributable to reduced exposed terminals, and are thus not representative of a deployed system, they reflect positively on the robustness of the design and implementation of our distributed TDMA approach.

VII. Related Work

Enterprise Network Management. Centralized EWLAN management has been considered in the context of fault diagnosis [7], protocol extensibility [18], security enhancements, such as detecting rogue APs [4], AP channel assignment and power control [1], client association [19], and client localization [6]. Centaur [24] and Shuffle [16] consider conflict-based per-packet link scheduling, allowing hidden terminal mitigation via scheduling. RxIP accomplishes similar timing-based isolation for hidden terminals, but our Internet-based architecture allows deployment within RWLANs without shared infrastructure or a low-latency interconnect.

Hidden Terminal Mitigation. Substantial prior work has considered hidden terminal detection and recovery [12], [14]. While ZigZag decoding [10] has been shown to be effective in USRP testing, it cannot support legacy hardware. RxIP may be readily deployable in WLANs with commodity hardware clients.

Network Measurement. [8], [11], [15], [22] characterize the performance of residential broadband. [21] presents an extensive measurement study of home wireless network performance. [2] suggests that these networks may be dense and prone to user misconfiguration. [9], [24] characterize hidden terminal losses. [13] recognizes the exacerbated impact of hidden terminals on TCP.

Related Techniques. Z-MAC [23] considers hybrid TDMA/CSMA for sensor networks, suggesting gains deriving from reduced contention irrespective of hidden terminal presence. JazzyMac [20] inspires our in-advance token-based establishment of TDMA schedules. SPIE [25] uses bloom filters for scalable per-packet state.

VIII. Conclusion

This paper considers the Internet as a medium for AP-to-AP coordination of the wireless channel. Although similar in principle to existing approaches, we believe our application to the residential domain expands opportunities previously reserved for the enterprise. As implemented in our Click Router prototype, RxIP APs may (1) detect the presence of a hidden terminal, (2) isolate the cause to a particular peer AP, and (3) mitigate hidden terminal performances losses by establishing an interference-aware hybrid TDMA/CSMA schedule. By peer-to-peer negotiation of the wireless channel, traditionally-centralized techniques for enterprise wireless networks may now be extended to the home as well. Immediately, residential deployments can benefit from fault diagnosis/recovery, improved coverage, and optimized frequency assignments. Extension of this platform leaves a rich area open for exploration.

References

[1] N. Ahmed and S. Keshav. Smarta: A self-managing architecture for thin access points. *CoNEXT*, 2006.
[2] A. Akella et al. Self-management in chaotic wireless deployments. In *MobiCom*, 2005.
[3] N. Anderson. Slow internet meets its waterloo as 105mbps comes to iowa. *Ars Technica*, Dec 2009.
[4] P. Bahl et al. Enhancing the security of corporate Wi-Fi networks using DAIR. In *Mobisys*, 2006.
[5] J. Bicket. Bit-rate selection in wireless networks. Master's thesis, MIT, 2005.
[6] R. Chandra et al. A location-based management system for enterprise wireless LANs. *NSDI*, 2007.
[7] Y. Cheng et al. Automating cross-layer diagnosis of enterprise wireless networks. In *SIGCOMM*, 2007.
[8] M. Dischinger et al. Characterizing residential broadband networks. In *IMC*, 2007.
[9] Y. C. et al. Jigsaw: solving the puzzle of enterprise 802.11 analysis. In *ACM SIGCOMM*, 2006.
[10] S. Gollakota and D. Katabi. ZigZag decoding: Combating hidden terminals in wireless lans. In *SIGCOMM*, 2008.
[11] D. Han et al. Mark-and-sweep: getting the "inside" scoop on neighborhood networks. In *IMC*, 2008.
[12] G. Judd and P. Steenkiste. Using emulation to understand and improve wireless networks and applications. In *NSDI*, 2005.
[13] S. Katti et al. XORs in the air: practical wireless network coding. *IEEE TON*, 16(3):497–510, 2008.
[14] F. Li et al. Passive and Active Hidden Terminal Detection in 802.11-based Ad Hoc Networks. In *Infocom*, 2006.
[15] G. Maier et al. On dominant characteristics of residential broadband internet traffic. *IMC*, 2009.
[16] J. Manweiler et al. Order matters: transmission reordering in wireless networks. *MobiCom*, 2009.
[17] Meru. The evolution of wireless LANs, 2009.
[18] R. Murty et al. An architecture for extensible wireless lans. In *HotNets VII*, 2008.
[19] R. Murty et al. Designing High Performance Enterprise Wi-Fi Networks. In *NSDI*, 2008.
[20] S. Nedevschi et al. An adaptive, high performance mac for long-distance multihop wireless networks. In *Mobicom*, 2008.
[21] K. Papagiannaki et al. Experimental characterization of home wireless networks and design implications. In *Infocom*, 2006.
[22] R. Raghavendra et al. Wi-Fi networks are underutilized. Technical report, MSR, 2009.
[23] I. Rhee et al. Z-MAC: a hybrid MAC for wireless sensor networks. *IEEE/ACM ToN*, 2008.
[24] V. Shrivastava et al. Centaur: Realizing the full potential of centralized wlans through a hybrid data path. In *Mobicom*, 2009.
[25] A. Snoeren et al. Hash-based IP traceback. In *SIGCOMM*, 2001.