2014

Unleashing Exposed Terminals in Enterprise WLANs: A Rate Adaptation Approach

Jun Huang

Guoliang Xing

Gang Zhou
College of William and Mary

Recommended Citation

Huang, J., Xing, G., & Zhou, G. (2014, April). Unleashing exposed terminals in enterprise wlans: A rate adaptation approach. In INFOCOM, 2014 Proceedings IEEE (pp. 2481-2489). IEEE.

Unleashing Exposed Terminals in Enterprise WLANs: A Rate Adaptation Approach

Jun Huang†, Guoliang Xing† and Gang Zhou‡
†Department of Computer Science and Engineering, Michigan State University,
‡Department of Computer Science, College of William and Mary
†{huangjun, glxing}@cse.msu.edu, ‡gzhou@cs.wm.edu

Abstract—**The increasing availability of inexpensive off-the-shelf 802.11 hardware has made it possible to deploy access points (APs) densely to ensure the coverage of complex enterprise environments such as business and college campuses. However, dense AP deployment often leads to increased level of wireless contention, resulting in low system throughput. A promising approach to address this issue is to enable the transmission concurrency of *exposed terminals* in which two senders lie in the range of one another but do not interfere each other's receiver. However, existing solutions ignore the rate diversity of 802.11 and hence cannot fully exploit concurrent transmission opportunities in a WLAN. In this paper, we present *TRACK* – *T*ransmission *R*ate *A*daptation for *C*olliding lin*K*s, a novel protocol for harnessing exposed terminals with a rate adaptation approach in enterprise WLANs. Using measurement-based channel models, TRACK can optimize the bit rates of concurrent links to improve system throughput while maintaining link fairness. Our extensive experiments on a testbed of 17 nodes show that TRACK improves system throughput by up to 67% and 35% over 802.11 CSMA and conventional approaches of harnessing exposed terminals.**

I. Introduction

In the last decade, 802.11-based Wireless LANs (WLANs) have become an important pervasive communication infrastructure for mobile Internet access. WLAN service is now being provided in a majority of corporations and college campuses. The advances of 802.11 technologies have also led to a drastic increase of bit rate of WLANs. Many production WLANs today provide a maximum bit rate of 600 Mbps, while lower bit rates are still available for legacy 802.11 clients.

Despite the significant improvement on bit rates and spatial coverage, it remains challenging to deploy and operate high-performance enterprise WLANs that service hundreds of users. First, the uncertainties in environmental factors including wireless signal attenuation, multipath fading, interference, and inherently complex building structures, often lead to significant variability in network performance. Second, WLAN clients now comprise increasingly heterogeneous devices including smartphones, tablets, and laptops, which often differ in transmission power and supported data rates, making it difficult to ensure consistent user experience. To provide pervasive coverage and satisfactory Quality of Service (QoS), a common practice is to over-provision network capability by densely deploying access points (APs) [15], especially in complex environments such as large office buildings with many blind spots. This approach is also justified by the increasing availability of inexpensive off-the-shelf 802.11 hardware. Unfortunately, dense WLAN deployment often leads to increased level of contention among nearby APs, resulting in lower system throughput and poor user experience.

To improve the performance of densely deployed WLAN, a well-known solution is to maximize the number of successful concurrent transmissions of different links. However, this is challenging in WLANs due to the contention-based nature of *carrier sense multiple access* (CSMA). In particular, a sender may be prevented from transmitting due to the interference from a nearby sender, although the signal to interference-plus-noise ratio (SINR) of its intended receiver is high enough to ensure successful reception, which is referred to as the *exposed terminal* (ET) problem. Recent empirical studies show that ETs are common in production WLANs. In [20], 41% of the links in a WLAN suffer from the ET problem, whose throughput can be doubled when concurrent transmissions were allowed. Several solutions [20] [23] attempt to harness ETs to improve link concurrency of WLANs. However, a major drawback of existing solutions is that they either assume a uniform bit rate across the network [23] or only adapt the rates of ETs in interference free scenarios [20]. Without accounting for the rate diversity of 802.11, they cannot fully exploit concurrent transmission opportunities. On the other hand, existing rate adaptation schemes [4] [6] are designed to adapt the bit rate of a single link in response to signal attenuation and channel fading. They assume that interference is resolved by medium access control (MAC) protocols, which prevents them from exploiting ETs.

This paper proposes a novel approach to harness exposed terminals in enterprise WLANs by leveraging the rate diversity of 802.11. Today's WLANs provide bit rates typically ranging from 6 to 600 Mbps. Induced by different modulation and coding schemes, these rates provide different trade-offs between transmission efficiency and reliability. By optimizing the bit rates of ETs according to their interference conditions, we can allow more links to transmit concurrently, leading to higher aggregate system throughput. However, this approach calls for new techniques that are fundamentally different from conventional wisdoms of ET exploitation or rate adaptation. As bit rate impacts the reception performance of links, non-ET nodes may be converted to ETs when they operate at different rates. Moreover, bit rate directly determines the airtime of

978-14799-3360-0/14/$31.00 ©2014 IEEE

packets and the temporal duration of interference. Therefore, the bit rates of ETs must be *jointly* considered in order to achieve desirable system-level performance such as throughput and fairness, which is a paradigm shift from the existing per-link rate adaption algorithms. This paper makes the following contributions.

First, we define the *R*ate-adaptive *E*xposed *T*erminal (*RET*) problem, where the quality of a link is strong enough for successful packet delivery at certain bit rate while its sender is prevented from transmitting by CSMA due to the interference of other sender(s). We characterize the properties of RETs using micro-benchmarking experiments. Our results reveal the challenges of exploiting RETs that have not been addressed before. Different from ETs, simply allowing concurrent transmissions of RETs may lead to unfair channel usage and link starvation.

Second, we design *T*ransmission *R*ate *A*daptation for *C*olliding lin*K*s (*TRACK*), a novel protocol for harnessing RETs in enterprise WLANs. TRACK tunes the bit rates of concurrent transmissions based on online channel measurements that account for the effect of frequency selective fading, and jointly schedules the transmissions of downlink RETs through the backend wired LAN connecting multiple APs. The scheduling algorithm of TRACK can optimize different metrics of system performance such as fairness and aggregate throughput. Finally, TRACK only requires changes to WLAN APs and hence can work with off-the-shelf 802.11 clients.

Third, we implement TRACK on commodity 802.11 nodes and evaluate its performance through extensive experiments on a WLAN testbed of 17 nodes. Our results show that, by effectively exploiting concurrent transmission opportunities, TRACK improves system throughput by 67% and 35% over 802.11 CSMA and conventional approaches of harnessing ETs while maintaining satisfactory link fairness.

The rest of this paper is organized as follows. Section II reviews related work. Section III presents experimental analysis of RETs. Section IV describes the design of TRACK. Section VI offers experimental results and Section VII concludes the paper.

II. Related Work

An effective way of boosting link concurrency in wireless networks is to exploit exposed terminals. CMAP [23] infers conflicting links using packet loss rate passively learned during concurrent transmissions, and opportunistically disables carrier sense when non-conflicting links are transmitting. CENTAUR [20] periodically measures conflict graph [5], and leverages centralized scheduling to mitigate downlink exposed terminals in WLANs. However, these systems either assume a uniform bit rate across the network [23], or rely on existing rate control algorithms designed for interference free scenarios [20]. In contrast, we demonstrate the significant impact of bit rate on exposed terminals, and present a practical protocol to improve link concurrency through rate adaptation for interfering links.

Several PHY layer designs have been proposed to improve link concurrency of wireless networks. In [8] [10], successive interference cancellation (SIC) is employed to recover collision packets. AutoMAC [9] exploits rateless coding to achieve link concurrency. However, these approaches are available only on software-defined radios and would require substantial modifications to commodity 802.11 receivers, making them difficult for practical deployment. Several recent works [19] study the approach of tunning transmission power to allow concurrent channel access in wireless networks. FLUID [17] exploits flexible channelization to improve system throughput of enterprise WLANs. Power control and channelization are orthogonal to our rate adaptation approach. The TRACK protocol proposed in this paper can be integrated with power control and flexible channelization to further improve the spatial reuse of WLANs.

Rate adaptation algorithms for 802.11 WLANs fall into two basic categories. *Link-layer algorithms* [4] [6] select bit rate based on frame delivery statistics such as packet delivery ratio (PRR). *PHY-layer algorithms* exploit PHY layer measurement to estimate channel quality. For example, CHARM [13] and SGRA [24] use signal-to-noise ratio (SNR) as an implicit indicator of channel quality. SoftRate [22] exploits per bit decoding confidence to direct rate adaptation. AccuRate [18] computes symbol dispersion to predict delivery performance of different rates. eSNR [11] measures channel state information to account for the effect of multipath fading. However, existing rate adaptation algorithms exclusively focus on interference free scenarios. For each packet transmission, they pick one rate that works best for an interference-free channel, assuming CSMA/CA has eliminated all potential collisions. In contrast, this work aims at boosting link concurrency by optimizing the bit rates of colliding transmissions. Therefore our approach is fundamentally different from existing rate adaptation paradigms.

III. A Measurement Study of Rate-adaptive Exposed Terminals

It is well-known that CSMA based wireless networks suffer from the exposed terminal (ET) problem. Previous work [20] [23] attempts to harness ETs to improve link concurrency. However, the impact of bit rate on ETs has not been studied. Modern WLANs provide rate diversity for trading off the efficiency and reliability of wireless communication. Generally, higher rate is efficient in modulation and coding, while lower rate is robust against noise and interference. In this paper, we revisit the classical exposed terminal problem in the context of rate-diverse WLANs. We define the *R*ate-adaptive *E*xposed *T*erminal (*RET*) problem as follows

Definition 1. Rate-adaptive exposed terminal (RET). *Given two links* $l_0 =< s_0, r_0 >$ *and* $l_1 =< s_1, r_1 >$, *where* s_i *and* r_i *are sender and receiver respectively, node* s_0 *is a rate-adaptive exposed terminal (RET) of* l_1, *if 1)* s_1 *can deliver its packets to* r_1 *at some bit rate R, but 2)* s_1 *is prevented from transmission due to the contention of* s_0.

A. Experimental Methodology

The goal of our measurement study is to characterize the properties of RETs. Since the existence of ET and RET is

TABLE I
THROUGHPUT ($10^6 \times bps$) MEASURED ON BENCHMARK TOPOLOGIES.

	Benchmark 1			Benchmark 2			Benchmark 3		
	$s_0 \to r_0$	$s_1 \to r_1$	Aggregate	$s_0 \to r_0$	$s_1 \to r_1$	Aggregate	$s_0 \to r_0$	$s_1 \to r_1$	Aggregate
802.11 CSMA	11.53	11.43	22.96	10.11	10.06	20.27	10.95	11.04	21.99
CT	2.76	3.12	5.88	0.91	1.20	2.11	11.47	1.69	13.17
CTRO	16.96	17.37	34.33	7.87	7.61	15.49	22.86	5.32	28.19

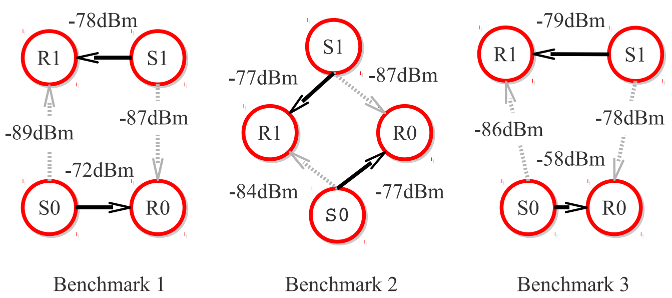


Fig. 1. Benchmark topologies. Data and interference links are marked with solid and dash lines. Average signal strength is labeled on links.

dependent on the channel access mechanism used on links, we employ three baseline methods in this study, including 802.11 CSMA, concurrent transmission (CT) and concurrent transmission with rate optimization (CTRO).

802.11 CSMA is the default channel access mechanism of WLAN's link layer. The bit rate is controlled by a state-of-the-art algorithm [11], which is designed for handling fading in interference-free scenarios.

Concurrent transmission (CT) simply disables CSMA on both links to enable simultaneous channel access. CT also employs the algorithm proposed in [11] for rate control, which selects the rate learned in an interference-free channel for concurrent transmissions. We note that simply allowing concurrent transmissions without rate adaptation is the common method of harnessing ETs. For a pair of *contending* links where the two senders lie in the carrier sense range of each other, loss-free delivery of CT indicates the existence of ETs.

Concurrent transmission with rate optimization (CTRO) extends existing solutions designed for harnessing ETs by optimizing the rates of concurrent transmissions to maximize their aggregate throughput. In order to show the potential of rate adaptation, we exhaustively try all possible rate pairs, and measure the resulted optimal throughput. By Definition 1, for a pair of contending links, loss-free packet delivery of CTRO indicates the existence of RETs.

B. Characterizing RETs

We first quantify RETs based on the topology of a large-scale 802.11a/g WLAN deployed in the Engineering Building of Michigan State University. Our measurement involves 104 link pairs. The result shows that 64.4% link pairs are RETs. In comparison, only 27.8% link paris are ETs. Therefore, by optimizing bit rates, we can enable 131% more link paris to transmit concurrently. The detail of our measurement can be found in a technical report [12].

We then characterize the properties of RETs using three micro-benchmarking topologies, as shown in Fig. 1. We evaluate the throughput of the three baseline algorithms introduced in Section III-A. The results are reported in Tab. I. As shown in the following, these micro-benchmarks represent three typical cases of RETs with different levels of interference between concurrent links.

Benchmark 1. In benchmark 1, although concurrent transmission brings interference and results in significant packet loss under sub-optimal bit rate, the qualities of two links are good enough to support concurrent packet delivery at an alternative lower rate. Thus the two links are RETs of each other. The result shows that CTRO outperforms 802.11 CSMA by more than 50%, while CT performs the worst as it is oblivious to the concurrency opportunities provided by rate diversity. The result of benchmark 1 demonstrates the potential of harnessing RETs for boosting link concurrency.

Benchmark 2. Benchmark 2 gives another example of mutual RETs. Different with benchmark 1, we observe that concurrent transmissions cause strong interference on both links. Although both links can find a lower rate for reliable packet delivery, the aggregate throughput decreases compared with 802.11 CSMA. This is because the links have to compromise their modulation and coding efficiency to tolerate the increased interference caused by concurrency. This result demonstrates that, unlike the classical ET problem, exploiting RETs may not always improve system performance.

Benchmark 3. The two links in benchmark 3 are also mutual RETs. In particular, the channel quality of $s_0 \to r_0$ is much stronger than that of $s_1 \to r_1$. We observe that although CTRO outperforms 802.11 CSMA in terms of aggregate throughput, the throughput of $s_1 \to r_1$ is decreased. Moreover, in the case of CTRO, the throughput of $s_1 \to r_1$ is 17 Mbps lower than that of $s_0 \to r_0$. The result shows that the asymmetry in channel quality may lead to unfair exploitation of individual links when concurrent transmissions of RETs are enabled.

Summary. Our measurements show that harnessing RETs could significantly boost the link concurrency. However, compared with classical ETs, harnessing RETs is more challenging as concurrent transmissions of RETs may lead to lower system throughput or unfair channel usage among concurrent links.

IV. TRACK DESIGN

We present TRACK, a novel protocol of Transmission Rate Adaptation for Colliding lin$K$s. The goal of TRACK is to harness concurrent transmissions of RETs to improve the aggregate system throughput while maintaining satisfactory link

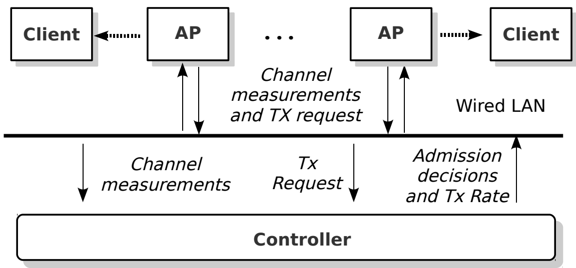


Fig. 2. The architecture of TRACK.

fairness. In this section, we first give an overview on TRACK, and then introduce its design in detail.

A. Overview

TRACK targets enterprise 802.11 a/g WLANs, where APs are densely deployed and connected with a high speed wired LAN. The architecture of TRACK is illustrated in Fig. 4. Exploiting the wired LAN as a messaging channel to APs, TRACK implements a centralized controller to perform admission control for downlinks (i.e., packets are transmitted from APs to clients). Downlinks are admitted to transmit concurrently, if this will improve the aggregate system throughput without compromising link fairness. Recent empirical studies showed that most network traffic in enterprise WLANs is downlink (e.g., 85% shown in [7]). Therefore boosting downlink concurrency will significantly improve overall system performance. In practice, the controller can be deployed on any server that is connected with APs through the wired LAN. The front end of TRACK is deployed on each AP, which collects and reports channel measurements to the centralized controller. When an AP has packets to send, it submits a transmission request to the controller. Based on the collected channel measurements and the set of currently active links, the controller makes admission decisions. If the link is admitted, the controller notifies the AP to start transmission, and configures the rates of concurrent downlink transmissions to tolerate interference. If it is rejected, the controller logs the requested link in a waiting queue, and recomputes admission decisions when one of the active links finishes transmission.

Although the basic design of TRACK is simple, deploying TRACK in practice is challenging due to the following reasons. First, in TRACK, APs and the centralized controller communicate through the wired LAN, which induces delay that affects the timing efficiency of link admission decisions. To amortize this overhead, TRACK uses a *packet batching* mode which groups downlink packets into blocks for transmission (see Section IV-C). Second, conventional SINR usually performs poorly in predicting link performance due to the effect of frequency selective fading [11]. To address this issue, TRACK adopts a novel metric called *effective SINR* to improve the accuracy of interference estimation (see Section IV-D). Third, although TRACK exploits the wired LAN as a messaging channel to coordinate the transmissions of APs, it is difficult to control clients and other non-enterprise WLAN devices. To

Algorithm 1: Downlink admission control of TRACK.

repeat
  $update\_needed \leftarrow False$;
  if *transmission request received from downlink* q **then**
    if $\mathsf{Q} = \phi$ **then**
      $update\_needed \leftarrow True$;
    end
    $\mathsf{Q} \leftarrow \mathsf{Q} + \{q\}$;
  end
  if *downlink* q' *finishes transmission* **then**
    if $\mathsf{Q} \neq \phi$ **then**
      $update\_needed \leftarrow True$;
    end
    $\mathsf{L} \leftarrow \mathsf{L} - \{q'\}$;
  end
  for $i \leftarrow 1$ **to** $|\mathsf{Q}|$ **do**
    if $\mathcal{T}(\{q_i\} + \mathsf{L}) > \mathcal{T}(\mathsf{L})$ *and* $\mathcal{J}(\{q_i\} + \mathsf{L}) \geq \alpha$ **then**
      $\mathsf{L} \leftarrow \mathsf{L} + \{q_i\}$;
      Remove q_i from Q;
    end
    else
      break;
    end
  end
  Send updated bit rates to links in L;
until *stop* ;

address the conflicts between scheduled downlinks and non-scheduled uplinks, TRACK employs a novel approach called *selective CCA*, which allows a TRACK AP to detect and ignore the signal of scheduled downlinks in clear channel assessment, while preserving the CSMA-based contention between scheduled and other non-scheduled transmissions (see Section IV-E). In the following, we will introduce the design of TRACK in detail.

B. Link Admission Control

To improve link concurrrency, TRACK opportunistically harnesses RETs by admitting downlinks if they will improve the aggregate system throughput without compromising link fairness. In the following, we first formulate the problem of admission control of RETs, and then introduce the protocol design.

We assume that a *packet batching* mode [2] is employed by the WLAN, where packets are combined into blocks for transmission. We discuss the motivation and design of packet batching in Section IV-C. Link admission decision is made before the transmission of each block. Rate is controlled on a per-packet basis. Let L be the set of active links in the network, and l the new link that has a batch of packets to send. The concurrent RET problem can be formulated as follows. For a set of of active links L, a new link is allowed to transmit concurrently with the links in L, if and only if (1) $\mathcal{T}(\{l\} \cup \mathsf{L}) > \mathcal{T}(\mathsf{L})$, and (2) $\mathcal{J}(\{l\} \cup \mathsf{L}) \geq \alpha$, where $\mathcal{T}(\cdot)$ and $\mathcal{J}(\cdot)$ are the functions of aggregate effective throughput and link fairness under $optimal$ rate assignment. α is a pre-defined threshold on link fairness. The two constraints assure that concurrent transmissions of RETs will not lead to system performance degradation or link unfairness, as demonstrated in benchmark 2 and 3 in Section III-B.

We define the optimal bit rate for link i, denoted by γ_i^{opt}, as the maximum rate that assures reliable packet delivery. In our implementation, we consider a link to be reliable if its packet reception ratio (PRR) is higher than 95%. Formally, given a set of active links L, γ_i^{opt} can be derived as,

$$\gamma_i^{\text{opt}} = \max_{\gamma \in \mathsf{R}} \{g(\text{sinr}_i, \gamma) \geq 95\%\} \tag{1}$$

where R is the set of legitimate bit rates defined in 802.11 standard; $g(\cdot)$ is the model of PRR, which depends on the employed bit rate γ and the SINR measured at the receiver of link. The SINR can be computed as $\text{sinr}_i = \frac{rss_i^i}{n_i + \sum_{j \neq i} rss_i^j}$, where n_i is the noise floor measured at the receiver of link i; rss_i^j is the average signal strength of packets transmitted from the sender of link j to the receiver of link i. In practice, $g(\cdot)$ can be profiled offline, while rss_i^j and n_i should be measured at runtime to estimate SINR. We discuss accurate SINR estimation in Section IV-D.

Assuming that all links in L use the optimal rates computed by Eq. (1) for concurrent transmissions, the aggregate effective throughput can be computed as follows,

$$\mathcal{T}(\mathsf{L}) = \sum_i \frac{d}{h_{PHY}/\gamma_0 + (h_{MAC} + d)/\gamma_i^{\text{opt}}} \tag{2}$$

where d is the size of payload; h_{PHY} and h_{MAC} are the sizes of PHY and MAC headers, respectively. In 802.11, the PHY header is always transmitted with the lowest rate, denoted by γ_0. Since the optimal rate γ_i^{opt} is selected such that the resulted PRR is higher than 95%, we neglect the impact of packet loss on effective throughput.

We model the link fairness as follows. For each active link, we compute its channel utilization as $u_i = \gamma_i^{\text{opt}}/c_i$, where c_i is the channel capacity when there is no interference. In practice, c_i can be estimated by the optimal bit rate used in an interference-free channel. In this paper, we use Jain's fairness index to quantify the fairness of channel usages, although other fair measures can also be adopted. Formally, the fairness is computed by,

$$\mathcal{J}(\mathsf{L}) = \frac{(\sum_{i=1}^{k} u_i)^2}{k \times \sum_{i=1}^{k} u_i^2} \tag{3}$$

where k is the number of active links in L.

When receiving a request of packet batch transmission from AP, the controller logs the information of requested downlink in a *waiting queue*, denoted by $\mathsf{Q} = \{q_1, ..., q_n\}$, where links are sorted in decending order based on their waiting time. Let L be the set of active downlinks. The pseudocode of downlink admission control is described in Algorithm 1. Specifically, the controller makes admission decisions when (1) a request is received atop the waiting queue, or (2) a downlink finishes its transmission of packet batch. The controller attempts to accept transmission requests in a FIFO manner. It checks the throughput and fairness constraints from the top of the waiting queue. The process stops when the check fails at one of waiting downlinks. Then the controller notifies the APs of active downlinks to update their bit rates. To avoid corrupting ongoing packets of other concurrent downlinks before their rates are updated, the AP of newly admitted downlink should stagger its transmission by a packet air time.

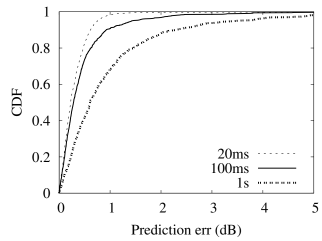


Fig. 3. Prediction error of different measurement period.

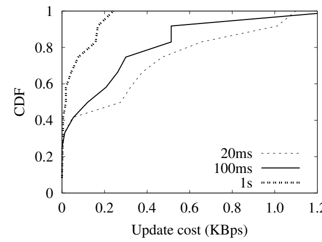


Fig. 4. Update overhead of reporting CSI to the controller.

Although Algorithm 1 is not guaranteed to generate the maximum set of concurrent downlinks, it has two advantages. First, as transmission requests are accepted in a FIFO manner, it assures that no downlink will be starved when there is persistent contention from other APs. Second, compared with computing the optimal transmission schedule of all active links, Algorithm 1 yields lower computation cost. This is especially important for the enterprise WLANs with heavy traffic load. Lastly, although we adopt Algorithm 1 in this work, the implementation of TRACK can integrate other scheduling algorithms.

C. Packet Batching

Packet batching is a common practice to reduce engery consumption rushhour11mobisys [16] and channel access overhead [20] [23]. TRACK employs packet batching to amortize the communication overhead between APs and the centralized controller. Specifically, it aggregates multiple packets into a block for transmission. Channel access decision is made before the transmission of each packet block. However, bit rate is controlled on a per-packet basis. TRACK employs a block ACK scheme. Packet ACK is disabled during packet batch. When the transmission of a block finishes, the client replies the AP with a vector, where each bit indicates the reception of one packet. Similar with the per-packet ACK defined in 802.11, block ACK is transmitted using the lowest bit rate to mitigate the inefficiency caused by ACK loss. In our impelmentation, we set the duration of packet batch to 4 ms. Block ACK is transmitted with a rate of 6 Mbps. The packet bactching of TRACK can be integrated with sleep scheduling [14] [16] to improve engery efficiency.

D. Interference Estimation

TRACK employs SINR to quantify the interference caused by concurrent transmissions, and then maps SINR to PRR for rate selection. However, conventional SINR often performs poorly in predicting PRR due to the effect of frequency selective fading [11], where different sub-carriers of the channel suffer different degrees of fading due to multipath propagation, causing variation of delivery performance across sub-carriers.

TRACK mitigates the effect of frequency selective fading by extending the effective SNR model proposed in [11] to account

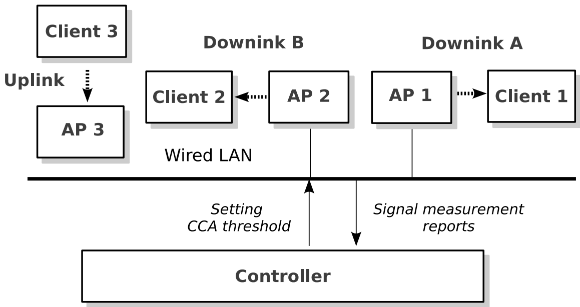


Fig. 5. A case study of coexistence of scheduled concurrent downlinks (A and B) with unscheduled uplink transmission. Downlink A and the uplink are out of carrier sense range of each other, while B and the uplink can hear each other. The objective of selective CCA is to disable carrier sense among scheduled downlinks, while preserving the channel contention between downlinks and the uplink. In this case, the transmitter of B (AP 2) is required to detect the signal of unscheduled transmitter (client 3) before accessing the channel, under the interference of scheduled on-going transmissions (from AP 1 to client 1). In our deployment, the measured signal strength from AP 1 to AP 2 is 15 dB stronger than the link from client 3 to AP 2.

for the effect of interference. The controller collects channel state information (CSI) measured on downlinks, and estimates sub-carrier SINRs for predicting uncoded bit error rate (BER) using the theoretical model. The uncoded BER is then averaged over sub-carriers and mapped back to obtain *effective SINR*. Different from the conventional SINR that simply averages SINR over sub-carriers, the effective SINR is calculated by averaging delivery performance across subcarriers to account for frequency selective fading.

The downlink CSI is measured as follows. First, AP periodically pings client, and uses the response packet to extract CSI. Then the channel reciprocity theory is applied to derive the downlink channel model. The measured CSI is reported to the controller if the signal variation on any subcarrier exceeds a threshold since last update. We observe that a threshold of 2 dB on subcarrier signal is enough to assure the accuracy of SINR estimation. We set the measurement period based on an empirical approach. We deploy 12 links to mimic the topology of a production WLAN, and then transmit back-to-back packets at a frequency of 500Hz to probe the CSI. Then we study the prediction error when different measurement periods are used. The result is shown in Fig 3. We observe that smaller period results in higher prediction accuracy. Specifically, a measurement period of 100ms is enough to limit the prediction error below 1 dB with a probability around 90%. Fig 4 shows the per-link update overhead incurred by reporting CSI to the controller. The result shows that the overhead is lower than 0.6 KBps per link in more than 90% cases, when a measurement period of 100ms is used. In our implementation, we will use a measurement period of 100ms. We study the performance of effective SINR driven rate selection in Section VI-B.

E. Coexistence with Non-Scheduled Traffic

Leveraging the architecture of enterprise WLAN to harness downlink RETs, TRACK uses a centralized controller to adapt the rates of concurrent downlinks. However, it is difficult for TRACK to control the packets sent by clients or non-TRACK WLANs deployed in vicinity, because the transmitters do not run TRACK. As a result, TRACK must be able to coexist with unscheduled traffic. A case of coexistence is given in Fig. 5. To avoid interfering unscheduled packet transmissions, we devise a novel clear channel assessment algorithm called *selective CCA* to handle non-TRACK traffic. The selective CCA allows a TRACK AP to detect and ignore the signal of scheduled concurrent downlinks in channel assessment, while preserving the CSMA-based contention between TRACK downlinks and unscheduled links. In the following we discuss the design and implementation of selective CCA in detail.

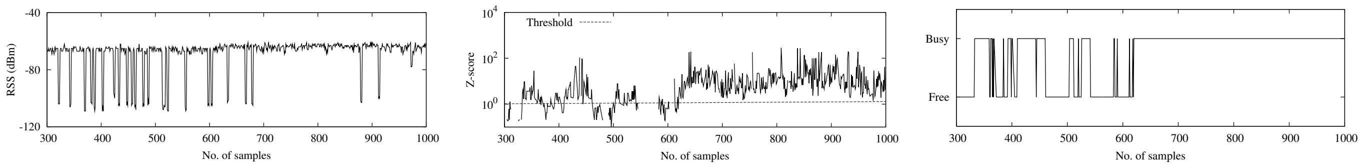
Suppose there is a scheduled concurrent downlink set of $\mathsf{L} = \{l_0, ..., l_m\}$. At the sender of downlink i, the measured signal strength of sender j is $rss_i^j\ dBm$. To allow concurrent transmission of L, the selective CCA sets the CCA threshold as $t_i = 10\log_{10}(\sum_{k\neq i}^{n} 10^{rss_i^k/10})$. The rationale behind is to set the CCA threshold by the total signal power of scheduled downlinks. When there is unscheduled traffic, the measured signal power at TRACK APs will exceed the calculated CCA threshold, causing selective CCA to detect a busy channel. To update the CCA threshold, TRACK APs periodically record the signal strength of overheard AP beacons. When TRACK configures bit rates through wired LAN, it also distributes the updated threshold.

A potential problem of selective CCA is that it may suffer lower carrier sensing sensitivity when the signal of unscheduled transmission is much weaker than scheduled links. Moreover, the signal power measurement often experiences temporal variations. The above two factors make it difficult to detect the signal of non-scheduled traffic among strong in-air downlink signals. We draw on statistical testing techniques to address this problem. When performing channel assessment, the AP first assumes a clear channel, and then performs the *z-test* on a window of collected signal samples to test the hypothesis. Suppose that the AP has a group of n measured signal samples which has mean μ and standard deviation σ. The AP computes the *z-score* by $z = \frac{\mu - thresh}{\sigma/n}$. The channel is deemed busy if the z-score rejects the hypothesis at a given confidence level. Fig. 6 shows the working trace of selective CCA on AP 2 of the link deployment given in Fig.5. We observe that AP 2 reliably detects the existence of uplink signal in presence of scheduled concurrent transmissions, as *z-test* is sensitive to the statistical variation of received signal strength.

V. TRACK IMPLEMENTATION

We have implemented TRACK in mac80211 [3] with ath9k [1]. To assure the practical deployability, we emphasize the principle of minimizing the modification of WLAN clients.

To adapt rate for concurrent transmissions, TRACK plugs effective SINR into an offline profiled SINR-PRR model to select the maximum rate that achieves reliable packet delivery. An important issue in profiling the SINR-PRR model is to limit the effect of multipath propagation in model profiling. We carefully place sender, receiver and jammer such that both of



(a) RSS trace measured at AP 2. (b) Z-score trace computed at AP 2. (c) Selective CCA output at AP 2.

Fig. 6. Trace of selective CCA at AP 2 of the link deployment in Fig. 5. Selective CCA samples the received signal strength at 250KHz. The size of CCA window is set to 5. A z-score threshold of 90% confidence is used. Uplink starts transmission around 600 samples. AP 2 senses a free channel with high probability when AP 1 is transmitting. After the uplink becomes active, AP 2 detects busy channel reliably.

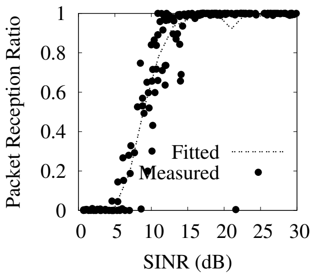


Fig. 7. The SINR-PRR model profiled using 12 Mbps rate.

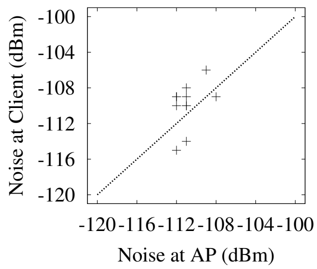


Fig. 8. Noise floor measured at clients and their associated APs.

the data and interference links have line-of-sight connections. We tune the transmission powers on sender and jammer to vary the SINR on receiver, and record PRR for each of 100 probes. The model profiled for Atheros-based cards of 12 Mbps is shown in Fig. 7.

For computing the effective SINR, TRACK requires channel measurements of downlinks on sub-carrier level to account for frequency selective fading. Ideally, this can be done by using channel sounding to extract channel state information (CSI) from received uplink packets, and then applying the channel reciprocal theory to derive downlink channel model [11]. However, Atheros cards used in our implementation currently do not expose CSI measurements to the user-space driver. To address this issue, TRACK employs the spectrum sampling tools of ath9k [1]. Specifically, clients transmits beacons periodically to probe CSI. Adjacent APs that receives probes switch to spectrum sampling mode to measure the signal strength on each subcarriers. Estimating effective SINR requires measuring noise floor at receiver. However, it is difficult to get clients' noise floor without deploying any code on them. We use the noise floor measured at the associated AP to approximate the noise floor of client. This approximation is reasonable as APs are usually densely deployed to ensure the coverage of enterprise WLANs, and hence the clients and their associated AP are often in proximity. We validate this assumption by measuring the noise floor in a 17-node testbed, including 6 APs and 11 clients. The noise floor measured for all 11 AP-client pairs are shown in Fig. 8. We observe that clients and their associated APs share similar noise floor.

VI. Evaluation

In this section, we evaluate the performance of TRACK. We evaluate TRACK against three baseline protocols to show the advantages of harnessing rate-adaptive exposed terminals (RETs). In the following, we will first introduce the experiment setting and then discuss evaluation results in detail.

A. Experiment Setting

We deployed a testbed consisting of 6 APs and 11 clients within the Engineering Building at Michigan State University spread over 1,800 square feet. Each node is a laptop equipped with Atheros AR928x radio, running mac80211 driver [3] with ath9k [1]. To ensure the realism of our testbed deployment, we place our APs close to the APs of a production WLAN deployed in the same area, and then randomly place clients around them. To further validate our deployment, we measure and compare the downlink quality of our testbed with the production WLAN in Fig. 9. We observe that signal strength of our testbed downlinks is slightly stronger, indicating that the part of deployment we emulated is denser than the production WLAN.

To demonstrate the efficiency of TRACK, we compare its performance with two baseline protocols, including 802.11 DCF (the default CSMA algorithm of 802.11) and a centralized scheduling algorithm for harnessing exposed terminals, which we refer to as *HET*. Different with TRACK, HET does not perform rate adaptation for concurrent transmissions. Instead, it uses the measured downlink channel model to estimate effective SINR and predicts resulted PRR for concurrent downlinks under default bit rate, i.e., the rate learned in interference-free channels. HET admits downlinks for concurrent transmission if all downlinks can achieve reliable packet delivery. Therefore HET only exploits ETs. We note that HET is similar to existing solutions designed to address the classical ET problem [20] [23]. For DCF and HET, we employ the algorithm proposed in [11] for rate control. For TRACK, the centralized controller estimates effective SINR, and plugs it into an offline profiled SINR-PRR model for rate selection.

B. Performance of Rate Selection

We evaluate the rate selection scheme of TRACK against packet SINR driven rate selector, which averages online measured subcarrier SINR to get packet SINR, and maps it to a bit rate by looking up the profiled SINR-PRR models. We conduct experiments by randomly selecting sender and receiver in our testbed to form the data link, and then place a node around the link to serve as interferer. We first measure the channel state information by probing the signal of sender and interferer, and

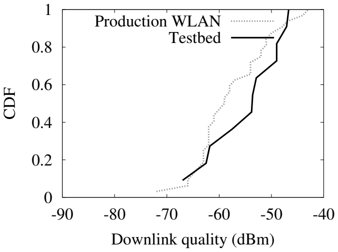


Fig. 9. Testbed link quality vs production WLAN.

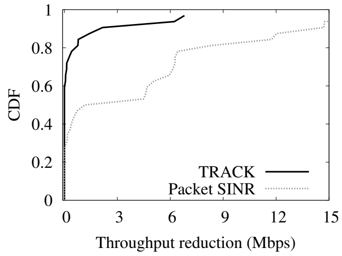


Fig. 10. Throughput performance of rate adaptation.

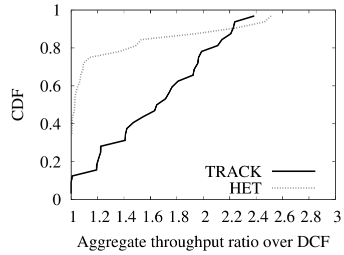


Fig. 11. Aggregate throughput on 2-AP topology.

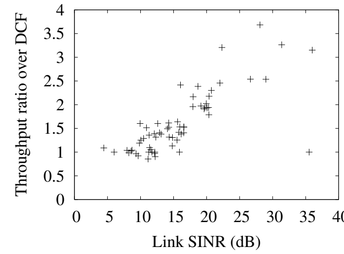


Fig. 12. Throughput improvement vs interference.

then log the rates selected based on packet SINR and effective SINR. Then we measure packet delivery performance for all bit rates, each using 10 probes. The rate which results in the best throughput is selected as the optimal rate. Then we repeat the measurement by 100 rounds. The experiment is conducted on 32 randomly selected settings of link pairs.

We compare the throughput achieved by packet SINR and effective SINR based rate selection in Fig. 10. The result shows the throughput reduction compared with the optimal bit rate. We observe that the effective SINR based approach does not decrease throughput in 60% cases, implying the accurate selection of optimal rate. Even when a sub-optimal rate is selected, the caused throughput reduction is always less than 6 Mbps. Overall, the rate selection scheme of TRACK outperforms the packet SINR driven rate selector in terms of the achievable throughput.

C. Performance on Two-AP topologies

In this section, we evaluate the performance of TRACK on two-AP topologies. We conduct the experiments by randomly picking two downlinks that are within the carrier sense range of each other, and then measure throughput for each link using broadcast traffic in a period of 1 minute. The experiment is repeated for 32 randomly chosen settings of link pairs.

Fig. 11 compares TRACK with HET in terms of the aggregate throughput improvement ratio over DCF. We observe that TRACK performs better than HET as it harnesses RETs by rate adaptation, while HET only exploits ETs. Specifically, compared to DCF, TRACK doubles the throughput with a probability higher than 65%, while HET only results in better performance than DCF in less than 60% cases. We further study the impact of interference on TRACK performance in Fig. 12. We find that TRACK performs better on strong links. In particular, the throughput improvement ratio over DCF can be higher than 2.5x when the SINR of link is larger than 25dB. The reasons are two-fold. First, strong links that support higher bit rates usually suffer more from the overhead of channel contentions and backoff procedure. Therefore, they benefit more by enabling concurrent transmissions, which mitigates the MAC layer overhead. Second, strong links usually perform poorly when coexisting with links of lower rates. This is because each time when the slow link wins in channel contention, it takes more time to transmit a packet than faster links, causing the rate anomaly problem [21]. Allowing concurrent transmissions by rate adaptation, TRACK avoids this problem, thereby enhancing the performance of faster links.

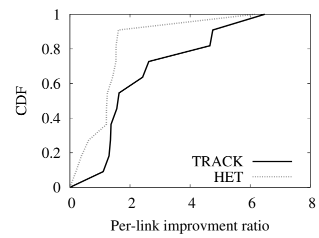


Fig. 13. Per-link UDP throughput

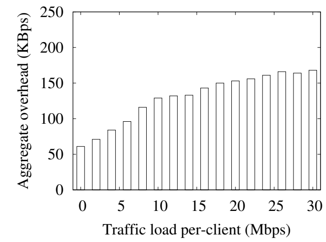


Fig. 14. Overhead.

D. Throughput, Fairness and Overhead

We conduct a large-scale evaluation on our testbed to study the performance of TRACK with both of UDP and TCP traffics. We first evaluate the downlink throughput of TRACK against DCF and HET. All links are configured to send saturated traffic. To mitigate the impact of short-term channel variation, we perform the experiment by switching between the three protocols in a round-robin fashion. Each protocol runs for one minute in each round. The experiment lasts for 2 hours.

We observe that the aggregate throughput achieved by DCF, HET and TRACK for UDP traffic are 63.3Mbps, 78.5Mbps and 105.7Mbps, respectively. For TCP traffic, the results are 50.6Mbps, 59.7Mbps, and 80.7Mbps for DCF, HET and TRACK. The results show that TRACK performs the best among all protocols. Specifically, TRACK achieves a 1.67x throughput over DCF, and improves the aggregate throughput by 35% over HET. We further compare the UDP throughput improvement ratio over DCF achieved by TRACK and HET for individual links. The result is given in Fig. 13. We observe that TRACK achieves better throughput on most of the links. In particular, when compared with HET, TRACK boosts throughput by up to 6x, and achieves an improvement ratio of more than 50% on 54.5% of the links.

As discussed in Section III-B, exploiting RET may result in link unfairness. To show that TRACK improves the overall throughput without unfairly exploiting individual links, we measure the Jain's fairness for the throughputs of individual links, and compare it with DCF in Table II. The result shows that for both TRACK and DCF, the throughput fairness drops as traffic load increases, since heavy traffic load will intensify

TABLE II
PER-CLIENT TRAFFIC LOAD VS THROUGHPUT FAIRNESS.

Traffic load	UDP		TCP	
	DCF	TRACK	DCF	TRACK
2 Mbps	0.970	0.987	0.972	0.987
6 Mbps	0.873	0.996	0.861	0.994
10 Mbps	0.781	0.937	0.768	0.916
16 Mbps	0.698	0.792	0.664	0.729

TABLE III
IMPACT OF UNSCHEDULED TRAFFIC ON THROUGHPUT RATIO.

Traffic load ratio (uplink:dnlink)	HET/DCF		TRACK/DCF	
	Dnlink	Uplink	Dnlink	Uplink
0.250	1.189x	1.036x	1.544x	1.153x
0.600	1.199x	1.039x	1.550x	1.009x
0.800	1.198x	1.030x	1.509x	1.048x
1.000	1.171x	1.003x	1.276x	1.121x

channel contentions, thus exacerbating the unfairness problem of CSMA. However, we observe that TRACK consistently performs better than DCF, as it exploits rate adaptation to allow more transmission opportunities on starved links.

We evaluate the overhead of TRACK by measuring the load of control messages over the wired LAN, including AP packets for reporting channel measurements, transmission requests, and TRACK controller commands for configuring bit rates and selective CCA threshold. We show the result of UDP traffic in Fig. 14. The TCP result is similar. We observe that overhead increases with per-link traffic load as higher traffic load incurs more frequent interactions between controller and APs. However, the aggregate traffic load is still lower than 200KBps even when the network becomes saturated, while the modern LANs often offer a bandwidth in the order of 100MBps.

E. Impact of Non-scheduled Traffic

We conduct experiments to study the impact of unscheduled traffic. We repeat the experiment introduced in Section VI-D with different uplink traffic loads, and compare TRACK with HET in terms of the UDP throughput improvement ratio over DCF. Tab. III shows that TRACK performs consistently better than HET as uplink traffic load increases. For instance, when the ratio between uplink and downlink traffic load is 25%, which is typical in enterprise WLANs [7], HET improves the downlink throughput by 18.9%, while TRACK achieves a gain of 1.54x, while maintaining similar uplink throughput. The results demonstrate the advantage of exploiting RETs, as well as the effectiveness of TRACK to avoid interference on unscheduled traffics.

VII. CONCLUSION

In this paper, we study a new problem – *R*ate-adaptive *E*xposed *T*erminal (*RET*) in WLANs, where the quality of a link is strong enough for successful packet delivery at certain data rate provided by 802.11 while its sender is prevented from transmitting by CSMA due to the interference from other sender(s). We present *TRACK*, a novel protocol for harnessing RETs. We implement TRACK on commodity 802.11 nodes and evaluate its performance through extensive experiments on a WLAN testbed of 17 nodes. Our results show that TRACK improves system throughput by up to 67% and 35% over 802.11 CSMA and a conventional approach of harnessing ETs, without compromising link fairness.

VIII. ACKNOWLEDGMENT

This work was supported in part by U.S. National Science Foundation under grants CNS-1218475, CNS-0954039 (CAREER), CNS-1253506 and CNS-1250180.

REFERENCES

[1] ath9k-linux wireless. http://linuxwireless.org/en/users/Drivers/ath9k.
[2] Ieee 802.11e-2005. http://standards.ieee.org/.
[3] Linux wireless. http://linuxwireless.org/,.
[4] Minstrel specification. http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel/minstrel.txt.
[5] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of rf interference. In *ACM CoNext*, 2008.
[6] J. Bicket. Bit-rate selection in wireless networks. Master's thesis, MIT.
[7] J. Eriksson, S. Agarwal, P. Bahl, and J. Padhy. Feasibility study of mesh networks for all-wireless offices. In *ACM MobiSys*, 2006.
[8] S. Gollakota, S. D. Perli, and D. Katabi. Interference alignment and cancellation. In *ACM SIGCOMM*, 2009.
[9] A. Gudipati, S. Pereira, and S. Katti. Automac: rateless wireless concurrent medium access. In *ACM Mobicom*, 2012.
[10] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: Interference cancellation for wireless lans. In *ACM MobiCom*, 2008.
[11] D. Halperin, W. Hu, A. Sheth, and D. Wetheral. Predictable 802.11 packet delivery from wireless channel measurements. In *ACM SIGCOMM*, 2010.
[12] J. Huang, G. Xing, and G. Zhou. Unleashing exposed terminals in enterprise wlans: A rate adaptation approach. Technical Report MSU-CSE-12-11, Computer Science and Engineering, Michigan State University, East Lansing, Michigan. http://www.cse.msu.edu/cgi-user/web/tech/reports?Year=2012.
[13] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation. In *ACM MobiSys*, 2008.
[14] J. Manweiler and R. Roy Choudhury. Avoiding the rush hours: Wifi energy management via traffic isolation. In *ACM MobiSys*, 2011.
[15] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing high performance enterprise wi-fi networks. In *NSDI*, 2008.
[16] A. J. Pyles, Z. Ren, G. Zhou, and X. Liu. Sifi: exploiting voip silence for wifi energy savings insmart phones. In *ACM UbiComp*, 2011.
[17] S. K. Rayanchu, V. Shrivastava, S. Banerjee, and R. Chandra. Fluid: improving throughputs in enterprise wireless lans through flexible channelization. In *ACM MobiCom*, 2011.
[18] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Accurate: Constellation based rate estimation in wireless networks. In *NSDI*, 2010.
[19] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-mac: Model-driven concurrent medium access control for wireless sensor networks. In *IEEE INFOCOM*, 2009.
[20] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra. Centaur: Realizing the full potential of centralized wlans through a hybrid data path. In *ACM MobiCom*, 2009.
[21] G. Tan and J. Guttag. Time-based fairness improves performance in multi-rate wlans. In *USENIX Annual Technical Conference*, 2004.
[22] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *ACM SIGCOMM*, 2009.
[23] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing exposed terminals in wireless networks. In *NSDI*, 2008.
[24] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang. A practical snr-guided rate adaptation. In *IEEE INFOCOM*, 2008.