A Case for Enhancing Dual Radio Repeater Performance Through Striping, Aggregation, and Channel Sharing[*]

Sayandeep Sen
IBM Research, India
sayandes@in.ibm.com

Michael Griepentrog
Cisco Meraki, USA
mgriepen@cisco.com

Jongwon Yoon
Univ. of Wisconsin-Madison, USA
yoonj@cs.wisc.edu

Suman Banerjee
Univ. of Wisconsin-Madison, USA
suman@cs.wisc.edu

ABSTRACT

The work in this paper is a systematic research and engineering effort in exploring the design space of multi-interface wireless repeater systems. We present the design, implementation and evaluation of a wireless repeater system, *Multifacet*, which opportunistically utilizes multiple interfaces to enhance capacity. The system is designed to be transparent to application endpoints and does not need any end application modifications for adoption. *Multifacet* incorporates several techniques to achieve efficient bandwidth utilization across multiple interfaces – (a) coordinated channel sharing, (b) a simplified backpressure based striping technique, (c) a single link abstraction, and (d) the ability to seamlessly migrate a client to the optimal channel. *Multifacet* is implemented on off-the-shelf dual band wireless repeater and demonstrates high speeds (337 Mbps) operation. On average, *Multifacet* performs 50% better than traditional AP/repeater setups, and in the best case, more than 2× better.

Categories and Subject Descriptors

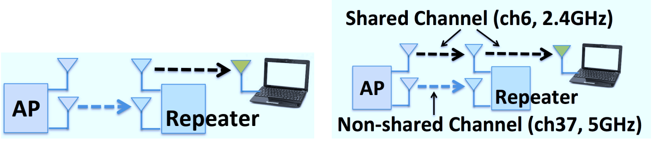
C.2.1 [**Computer Communication Networks**]: Network Architecture and Design – Wireless Communication

Keywords

Wireless repeater; Dual radio AP; Implementation

1. INTRODUCTION

There is a growing demand for high bandwidth wireless delivery mechanisms in varied home settings. For example, cable and satellite providers offer products which stream content from their *single* set-top box to multiple TVs customer's home over wireless. The gaming vendors are selling hardware with the ability to stream games from gamer's desktops to their living room (Steam Big Picture, NVIDIA Project SHIELD). To meet the increased demands for higher bandwidth, wireless equipment vendors and research community are continually striving to come up with new standards (802.11n, 802.11ac/ad etc.), which promise throughput in ranging from hundreds of Mbps to a few Gbps. Unfortunately, the predominantly harsh operating environment with high interference and sometimes non line-of-sight to the client implies that newer solutions that increase the raw data rates do not always deliver commensurate increase in client throughput.



(a) Traditional repeater (b) *Multifacet* repeater

Figure 1: Logical architecture of repeater.

Wireless range extenders, also referred to as repeaters [1,2,4,6], are the de facto solution to improve the performance of clients which experience bad connectivity from the Access Point (AP) either due to large distance of separation or excessive interference on the wireless channels and when bringing the AP close to the client or vice versa is not possible. A repeater enhances performance by dividing a single poor quality path into two improved segments by rebroadcasting signals from a given AP to devices that would normally be out of the AP's range or is not able to achieve a given user's performance constraints. Figure 1(a) shows the most flexible and sophisticated repeater system which features two wireless interfaces. Evidently this mode of operation achieves higher throughputs than a far away client directly attempting to connect to the AP. The dual-interface repeater serves the client through one of the radios (say, operating in the 2.4 GHz band) and the repeater, in turn, connects to the AP through its other radio (operating in the 5 GHz band). In fact dual-interface repeaters, together with dual-interface APs provide the maximum performance in terms of range, throughput, and overall experience, according to the vendors.

In this paper, we show that in certain settings dual-interface AP and repeater systems can deliver even higher throughput to their clients by making software-only modifications. The insight is to opportunistically utilize both the radios of the AP to send data to the repeater as shown in Figure 1(b). We design and implement *Multifacet* (or Multi interface Transport) based on the above insight to enhance client performance. As AP-repeater link shares the channel with repeater-client link, *Multifacet* can provide gains

[*] S. Sen and M. Griepentrog were graduate students at the University of Wisconsin-Madison during the course of this work.

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.
Copyright 2014 ACM 978-1-4503-2783-1/14/09 ...$15.00.
http://dx.doi.org/10.1145/2639108.2639133.

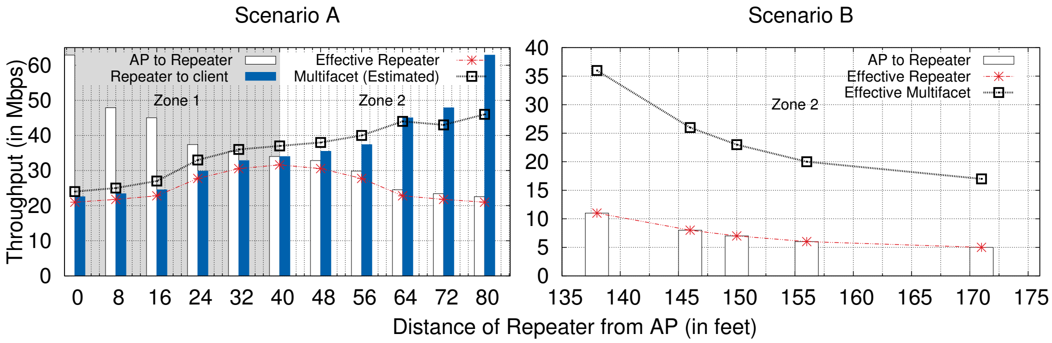


Figure 2: The performance benchmark of dual-interface repeater system and our formulation-based estimates of *Multifacet*. In the experiments the AP–repeater link was operating over a 20 MHz channel in 5GHz band and the repeater–client link was sharing a 20 MHz channel in 2.4 GHz band with the AP–repeater link (Figure 1(b)). We omit the repeater–client throughputs for Scenario B for clarity, which were always significantly high in this scenario.

only for certain channel configurations. Further, even in favorable conditions it is not straightforward to realize the throughput gains at client as one of the AP-repeater link shares the channel with repeater-client link.

How *Multifacet* differs from other general solutions of multi-hop multi-radio systems: Our repeater setup can be considered as a simpler version of a multi-hop multi-radio wireless mesh network. In particular, *Multifacet* just has two hops – one hop has two parallel links, while the other has a single link. While significant research exists in efficient wireless mesh network design, e.g., in channel assignment, routing strategies, load balancing of flows, and more [16–19, 21], such solutions are not pertinent to our problem. This is because of the fact that prior solutions *work on flows and do not address the problem of splitting individual flows across multiple interfaces*. A key uniqueness of *Multifacet* from prior mesh networking literature, thus, comes from its focus on efficiently splitting a flow over multiple interfaces at high data rates under varying channel conditions. Similarly, prior work has addressed the issue of using multiple parallel WiFi links simultaneously at full capacity (Glia [31]), it does not support partitioning of a single flow across these multiple interfaces. Hence, approaches such as Glia alone, are not suited to support high bandwidth media flows in high interference scenarios. The unique aspects of *Multifacet* is the set of algorithmic optimizations (a simplified back pressure technique, synchronous channel migration on demand, and soft reliability), and implementation-based refinements that present a single link abstraction to unchanged flow endpoints.

Key contributions: *Multifacet* is a systematic research and engineering effort to significantly enhance the performance and achievable throughputs when using repeaters in the most challenging wireless home environments. We make the following contributions:

- Show how the performance of dual-radio repeaters can be enhanced by leveraging multiple interfaces (§ 2).
- Design and implement *Multifacet* – a dual interface repeater system to opportunistically use multiple interfaces on a desktop platform and Netgear WNDR3800 (§3 and § 4).
- Show that *Multifacet* is within 15% of the theoretical optimal 90% of the time (§ 5.1). Demonstrate that our system works at high speeds (achieving 337 Mbps for TCP, § 5.4).

2. *MULTIFACET* IN A NUTSHELL

Multifacet is stylized for repeaters with the central goal of opportunistically leveraging the excess capacity of downstream channel. We explain the benefits that *Multifacet* can provide next.

When does *Multifacet* help? To answer the question, we first study how repeater performance varies (Figure 1(a)) in realistic settings while varying conditions. Specifically, we measure TCP throughput delivered by repeater systems using off-the-shelf dual-interface AP (WNDR4500), repeater (WN2500RP), and a client laptop, all operating over 802.11n channels in Figure 2. In Scenario (A), the separation between the AP and the client in a home environment (across multiple rooms and walls) is set to 80 feet, and the repeater is placed at different locations in between. It can be seen that the AP–repeater and the repeater–client links, each operating with different radios and different channels, individually get between 20 Mbps and 60 Mbps in different deployments of the repeater. The effective end-to-end throughput marked as effective repeater in Figure 2 is the minimum of the two hops and is limited to 20–30 Mbps. The problem of reduced throughput exacerbates with higher separation between the AP and the client. Scenario (B) shows an even greater separation between the two (up to 170 feet) leading to poorer performance, and the maximum end-to-end throughput using a dual-interface repeater is down to about 5 Mbps.

Now striping traffic as advocated by *Multifacet* will be helpful when the repeater-client link has higher capacity than AP-repeater link as the extra capacity of repeater-client link can then be shared with AP-repeater link. As an example, consider the link capacities at 64 feet from AP in Figure 2. The throughput between the repeater-client is around 45 Mbps and for the AP-repeater links is 20 Mbps. Assuming similar data rates for both AP-repeater and repeater-client links would imply that $\sim$25 Mbps of capacity of shared channel is wasted. A part of this capacity can easily taken by AP-repeater link by operating on the channel. This wasted capacity is precisely what *Multifacet* intends to use for sending more traffic from AP to repeater by contending on the shared channel, in turn increasing throughput delivered to clients.

Note that the careful placement of repeater at a location does not completely obviate the problem of traffic scheduling. This is due to the time varying nature of wireless interference in dense urban home settings, which causes variability in channel conditions at different times of day. This apart, the client is expected to be mobile within the confines of his home, which might frequently change the optimal repeater placement.

How much gains can *Multifacet* provide? We present a numerical analysis to characterize the expected performance improvement due to *Multifacet*. We assume that perfect prediction and scheduling of traffic is possible to balance throughputs along upstream (AP–repeater) and downstream (repeater–client). Let D_S denote the single hop throughput of the downstream link in isolation between the repeater and the client. Similarly, U_S and U_{NS} denote the throughput of the two upstream links (each in isolation) between the AP and repeater. S corresponds to the link operating on the same channel as the downstream link, and NS corresponds to the link that is on the non-shared channel. The estimation of expected performance improvement is based on two observations: (i) For *Multifacet* to provide benefits, D_S has to be greater than U_{NS}. This is true for all experiments marked Zone 2 in Figure 2, and (ii) a queue build up in repeater–client radio would imply that the repeater is getting more packets from AP than it can send to the client. This is suboptimal[1] as it would simply lead to accumulation of packets at repeater without enough time to send them to client. Hence, *the fraction of the shared channel assigned to AP–repeater link should not cause queue build up in repeater–client radio*.

Now, let us focus on a single unit of time. For this time unit, assume that a flow's traffic is split at the AP to send U_{NS} data units over the non-shared channel and an additional K data units on the shared uplink channel to the repeater. This implies, the AP would occupy the shared channel for $\frac{K}{U_S}$ time units. To avoid buildup of packets in repeater, the repeater would need to transmit the entire $U_{NS} + K$ to the client within its time of operation on the shared channel. To do so, the repeater-client link should be active for $\frac{U_{NS}+K}{D_S}$ time units over the shared channel. Hence in steady state to avoid queue build up, the following invariant would be satisfied for a single time unit:

$$\frac{K}{U_S} + \frac{U_{NS} + K}{D_S} = 1 \tag{1}$$

The above equation states that the fraction of time occupied by AP-repeater link and repeater–client link on the shared channel should add up to 1. Rewriting,

$$K = \left(1 - \frac{U_{NS}}{D_S}\right) / \left(\frac{1}{U_S} + \frac{1}{D_S}\right) \tag{2}$$

and the total end-to-end throughput is $U_{NS} + K$.

In Figure 2 we also show the total end-to-end throughput of *Multifacet* that is estimated by the above formulation based on the measured values of U_{NS}, U_S, and D_S. We can see that under different scenarios (Zone 2), the total throughput due to *Multifacet* can be *more than double that of a standard and unchanged repeater implementation.* For example, in Scenario A at a separation of 80 feet, *Multifacet* in theory can provide up to 48 Mbps throughput 2.4× higher than standard repeater implementation (20 Mbps).

One observation is that *Multifacet* will not be useful in many scenarios where $U_{NS} > D_S$, as is true for all cases in Zone 1. However, there are sufficient other scenarios (Zone 2) where even the high-end and highly capable WiFi systems (our AP, repeater and client laptop use 2×2 MIMO and 802.11n), the end-to-end realized throughputs are as low as a few Mbps. In our experience these challenging scenarios, indeed, occur in large homes and spaces with thicker walls and greater degree of interference. In fact, this is why repeaters (including dual-interface versions) are being sold by all leading vendors in the marketplace. By allowing the overall system to be able to more than double the throughput using *Multifacet* can make a significant difference in the type of high-bandwidth applications that will be feasible under these circumstances.

What is hard about *Multifacet*? The basic concept of *Multifacet* is fairly simple – allow an individual flow to be striped across two WiFi paths between an AP and a repeater and aggregate the flow before forwarding it to the client. To ensure that wide deployment, *Multifacet* does not require any changes to applications and endpoints making the deployment possible by simply installing new software in the AP and the repeaters. This is achieved by providing a single link abstraction to ongoing flows even though internally, the packets are striped and aggregated across two interfaces.

The biggest challenge of the system is in its careful design of algorithms that (i) scales to very high 802.11n speeds (∼100 Mbps), (ii) reacts efficiently to changing channel conditions, and (iii) in its implementation in the specific AP and repeater form factors. With two parallel links operating at these speeds, and occasionally experiencing channel fluctuations, a small mistake in deciding how much traffic to send in each path can cause hundreds of packet reorders, and losses. The performance penalty is particularly high for TCP flows which react badly to bursty reorder and packet losses. Thus, our algorithms for striping traffic need to be *simple* and *nimble* so that they can efficiently react to even small changes to channel conditions across two parallel paths.

Further, given that one channel is shared between one of the upstream links and the downstream link, it is critical to design efficient control loop to coordinate between traffic forwarding decisions between these two links. The importance of efficient coordination becomes apparent by realizing that in its absence the upstream link can potentially dominate the downstream link on the shared channel. This will in turn lead to significant degradation in performance as repeater will keep getting more packets to deliver but would not get access to the channel to deliver it. To this end, we implement a light weight distributed control loop – a simplified back pressure technique – that monitors the available capacity of these links in making striping and traffic forwarding decisions.

In addition to the challenges of traffic striping, the availability of two parallel links in the upstream path presents us an interesting opportunity. If the non-shared link experiences poor performance, it might be more beneficial to switch the operating channel of this link to a better and non-congested one. We therefore designed mechanisms to opportunistically switch the non-shared link to different channels. We do not consider switching the shared channel as the only link between the client and repeater operates on it and switching channel can severely impact any active flows to the client.

3. *MULTIFACET* DESIGN

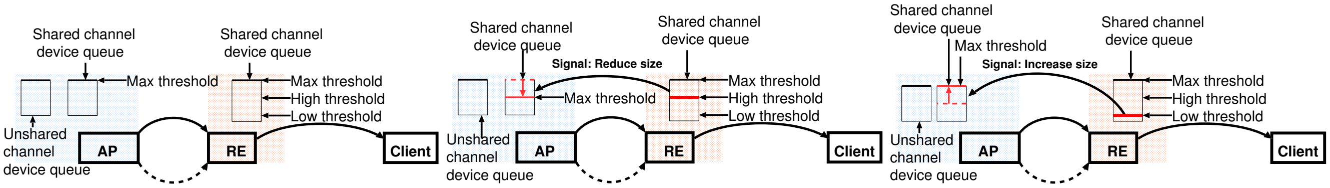
We describe the techniques used in *Multifacet* to address the challenges highlighted in previous section.

3.1 Simplified backpressure based striping

The goal of the striping algorithm is to ensure optimal utilization of the capacity of the constituent links under all channel conditions. This implies that the striping algorithm should be able to *quickly shift traffic load* from one interface to another if the channel conditions worsen.

The striping algorithm works in two phases: the bootstrap and the steady state. In the bootstrap state, the *Multifacet* AP assigns a fixed number (*Device_threshold*) of packets to all interfaces in a round-robin manner. The intention is to gauge the capacity of the constituent links by monitoring the performance of delivering the initial set of packets. Once, *Device_threshold* number of packets have been striped through all interfaces, the striping algorithm enters the steady state. In steady state, more packets would be scheduled only after feedback of success/failure is received from the underlying device. The above behavior ensures that *we only schedule packets the link can support*. Furthermore, we ensure that *each link always have some backlogged packets to transmit, or we would lose out on available transmission opportunities*. The faster link will send feedback more frequently and subsequently have more traf-

[1] In present context *optimal* throughput is defined as the maximum throughput that can be delivered to a client under the constraints of underlying link layer technology.


Shared channel device queue
Shared channel device queue
Max threshold
Max threshold
High threshold
Low threshold
Unshared channel device queue
AP
RE
Client
Shared channel device queue
Signal: Reduce size
Shared channel device queue
Max threshold
Max threshold
High threshold
Low threshold
Unshared channel device queue
AP
RE
Client
Shared channel device queue
Max threshold
Signal: Increase size
Shared channel device queue
Max threshold
High threshold
Low threshold
Unshared channel device queue
AP
RE
Client

(a) *Multifacet* logical design (b) Restrict traffic (c) Increase traffic

Figure 3: Co-ordinated transmission throttling for the shared channel shown for downlink transmission of packets. (a) Shows the logical architecture of the system wherein repeater periodically monitors the number of unsent packets at the repeater destined for the client. (b) The unsent packet queue grows beyond its maximum threshold, triggering a reduction in packets sent from AP over the shared channel. (c) The queue of unsent packets is below minimum threshold and AP can potentially send more packets to the repeater over the shared channel.

fic scheduled to it, while the link with lesser capacity would send feedback at slower pace, and thus would get less packets.

While in steady state, if the bandwidth demanded by the applications is well below the capacity of the underlying links, the *Multifacet* AP switches back to striping the packets in a round-robin fashion across the constituent links. The above state is indicated by a very small number of packets on *all* client interfaces for which no feedback has been received from the underlying device.

3.2 Co-ordinated transmission throttling for congestion control

The goal of transmission throttling logic is to ensure that the uplink (AP–repeater) over shared channel occupies only the extra bandwidth after all downlink (repeater–client) traffic is sent. Figure 3(a) shows the logical setup for efficiently utilizing the spare capacity of the shared channel. The transmission throttling logic bootstraps from a state where *Multifacet* is operating like a traditional repeater, i.e., the AP is not sending any traffic over the shared channel. In steady state the system tries to incrementally probe the shared channel for extra bandwidth between the AP and repeater. The repeater continuously monitors the number of unsent packets destined to the client. The event of unsent packets count growing beyond a (user specified) threshold signifies that the AP traffic has exceeded its allowable bandwidth, thus taking away capacity from the repeater–client link. In this case the repeater module sends a congestion initiation signal to the AP module as shown in Figure 3(b). The transmission throttling module at AP reacts to the congestion message by reducing the *Device_threshold* parameter for the wireless interface operating on the shared channel. Reducing this parameter automatically reduces the number of packets scheduled by the backpressure packet striping algorithm. This in turn provides a larger share of the shared channel to the repeater.

Analogously, a lower than threshold number of unsent packets at the repeater signifies that the spare capacity is present on the shared channel. In this case, repeater signals the AP to increase the volume of traffic on the shared channel. The AP accordingly increases the percentage of traffic scheduled by increasing the *Device_threshold* for the wireless interface operating on the shared channel as shown in Figure 3(c). The repeater sends a status quo signal to the AP if the number of unsent packets stays between the lower and upper thresholds. We use a MIMD algorithm to adjust the *Device_threshold* at the AP.

Correctness of behavior: Note that the behavior of transmission throttling logic would be incorrect when the repeater-client link has enough traffic to saturate the shared channel on its own, however the AP-repeater link is still allowed to operate on the shared channel. From previous discussion (Figure 2), we know that repeater operates in two zones. We explain why our transmission throttling logic would behave correctly and choke AP–repeater link in both the zones. In zone 1, $D_S \leq U_{NS}$. Hence, $D_S \leq U_S + U_{NS}$ which implies that a queue of unsent packets would build up in repeater. The accumulated traffic will eventually cross the allowable threshold leading in repeater leading to choking of AP–repeater link (Figure 3(b)). In zone 2, assume that the AP–repeater link achieves K' throughput. Now, based on Equation 1, the repeater–client link can only support $K + U_{NS}$ data units per second. Whereas it is receiving $K' + U_{NS}$ data units per second and $K' > K$. Hence, $K' - K$ ($[K' + U_{NS}]$ - $[K + U_{NS}]$) amount of traffic will accumulate in repeater–client queue every second. Eventually, the queue size will cross the allowable threshold leading to reduction of AP–repeater traffic over shared channel.

3.3 A single link abstraction

We hide the underlying multiple parallel wireless links and present the abstraction of a single virtual link. This implies that packets of a single flow need to be delivered in order across the virtual link, analogous to sending packets across a single physical wireless link. To maintain single link abstraction we handle re-orders for the case when packets of a single flow is striped across two or more links. Note that any packets lost due to wireless error will be retransmitted by the source the resulting end-to-end re-orders, of course, do not need to be handled by our system.

Comparison to multi-path TCP: An alternative to our single link abstraction is to utilize a generalized multi-path transport protocol, e.g., MPTCP [29, 34, 39] depicted in Figure 4(a). It requires deployment at the two endpoints, and an ability in software to stripe packets across the different WiFi interfaces. The other alternative is the *Multifacet* (Figure 4(b)) that requires no endpoint changes but implements various algorithmic techniques that are deployed on two ends of the special wireless link. It is intuitive that the MPTCP approach (implemented at the transport layer) is efficient for general scenarios the multiple alternate Internet paths diverge and merge at arbitrary points in the Internet, but the *Multifacet* approach (implemented at the network layer) is more suitable for the specific scenario at hand. There are three advantages of *Multifacet* over the MPTCP. First, *Multifacet* works with unchanged endpoints and can be deployed in a transparent manner. Second, *Multifacet* exploits custom knowledge of WiFi properties to combine various data striping choices with selection of link parameters (PHY rates, re-transmission, etc.) to achieve higher throughputs than a WiFi-unaware MPTCP. Third, an end-to-end packet striping and adaptation mechanism of MPTCP operates at a control loop latency of 50-100 ms (typical Internet path latencies), while in *Multifacet*, the same mechanism operates at a control loop latency of 1-2 ms or less (typical single-hop WiFi latencies). All of these combined, leads to significant performance advantages as demonstrated in § 5.2.

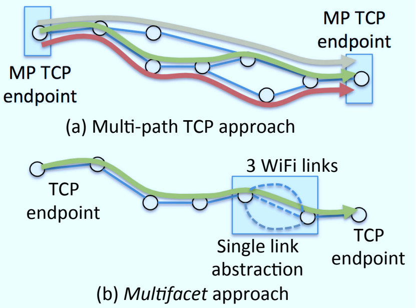


Figure 4: MPTCP is a generalized multi-path transport solution while *Multifacet* is customized to a single wireless hop with multiple parallel WiFi paths.

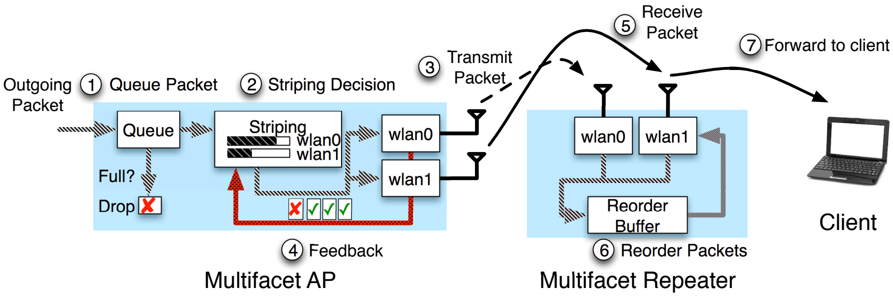


Figure 5: Flow of packet inside *Multifacet* system.

The performance inefficiency of MPTCP is due to the well known phenomenon of *self-interference* between two links operating over the shared channel. Self-interference causes the transmitter on upstream (downstream) link backing off after carrier sensing the transmissions of the downstream (upstream) on the same channel. In fact, it is to avoid this very problem the AP–repeater and repeater--client links are operated on different channels in state-of-art repeater systems. MPTCP suffers from the self-interference problem when operating on the two paths over the shared channel. As a result, MPTCP is incapable of saturating the links and resulting in low performance. In contrast, *Multifacet* actively prioritizes the operation of the repeater-client link over the AP-repeater link due to reasons explained in Section 1. This results in optimal avoidance of self-interference leading to high performance (evaluation in § 5.2).

3.4 Synchronous channel migration

To ensure higher capacity in undesirable circumstances, *Multifacet* triggers channel migration on a non-shared channel, especially when the quality of the non-shared channel between AP–repeater underperforms (below a threshold). During the channel migration process, the *Multifacet* AP first stops sending packets on the non-shared channel and sends a channel switch announcement (CSA) in its beacon frames. Once both the AP and repeater have migrated, the *Multifacet* AP carries out a bandwidth test similar to the one proposed in [28] for a duration of 20 milliseconds. If the achievable throughput is higher than what was previously observed, the channel migration process stops. Otherwise, we revert back to the original channel. We call this the *Adequate* scheme and contrast it with a *Best* scheme where the channel search explores all possible channels and picks the one that maximizes throughput (evaluation in §5.4).

Finally, note that we do not try to migrate the shared channel as a design choice. This is because of the fact that the only link between the client and repeater is operational on the shared channel, and hence the shared channel cannot be seamlessly changed without disrupting the flow to the client.

4. IMPLEMENTATION ASPECTS

Multifacet is implemented by augmenting various software components of both AP and repeater wireless network stack on a Linux 3.2.0 kernel. Specifically, we have modified the Linux *bridge*, *bonding*, ath9k, Linux mac80211 drivers and Hostapd AP daemon. It does not require any modifications at the client. For experimentation we used Atheros AR9220 and AR9380 chipset based 802.11a/b/g/n capable network interfaces for our system. The flow of packet (AP–repeater–client) is depicted in Figure 5. We describe the implementation details next.

Simplified backpressure based striping: Scheduling is performed within the bridge driver. Incoming packets are first enqueued in a small buffer, which we have added to handle fluctuations in link capacity (step 1 in Figure 5). We replace the forwarding logic of the bridge with our backpressure algorithm described in § 3.1 (step 2). If the scheduler can not find an available interface (where the number of unacknowledged packets is less than *Device_threshold*), no packets are sent. When an interface is available, we dequeue a packet from buffer, add an IP option (which contains a sequence number and the interface the packet was sent on) to the packet and send it on the appropriate interface (step 3). Once a packet succeeds or fails, the ath9k driver sends feedback to the bridge driver, which is then incorporated into striping algorithm. If the buffer is not empty, additional packets are sent to the interfaces until *Device_threshold* is reached or the buffer is emptied.

Single link abstraction: The Linux bridge driver is used to implement the *Multifacet* AP interface. The WiFi NICs are in AP mode with distinct SSIDs and are ports of the bridge interface. Being ports of a bridge implies that all repeater/client traffic passes through the bridge code, where our code resides. To achieve the facade of a single interface at the repeater, we enslave the constituent wireless interfaces under a bond interface. The repeater interfaces associate with a specific AP interface based on the SSID. We add an IP option header at the AP to allow us to re-order packets at the repeater as well as to notify our striping algorithm of the (un)successful delivery of a packet.

Packet reordering: For re-order management, we have modified the bridge driver to assign a 32-bit sequence number to each packet which is stored as an IP option and implemented a circular re-order buffer. We also include an interface identifier to help lower layers provide feedback for our striping algorithm. As shown in Figure 5 (step 6), all packets arriving on different interfaces are placed in our re-order buffer. The AP assigned sequence number is used for re-ordering. In our implementation, the re-order buffer stores up to thousands of packets, but no longer than 10 milliseconds, for any missing prior packets. Typically most packets need to wait a few hundred microseconds to be correctly ordered when they arrive across different links.

Controlled sharing of channel: The spectrum is shared between the AP and repeater by monitoring queue build-up at the repeater for packets destined to a client. We modify the wireless stack on the repeater to periodically send control packets to the AP to indicate the presence of congestion, which is determined if the queue size passes a certain threshold. These control packets indicate to the AP

Evaluation scenarios	Section / Figure	Experiment variation / Results
vs. Traditional repeater	§5.1, Fig.6(b)	Uncontrolled: Avg. 35% TCP throughput gain, Max. 2.5×
	§5.1, Fig.7	Controlled: Avg. 70% TCP/UDP throughput gain, 8.7 dB PSNR gain
vs. MPTCP	§5.2, Fig.8	Varying path delay: 19% throughput gain (Max. 2×)
vs. Overhearing mesh	§5.3, Fig.9	110% throughput gain
Design choices of *Multifacet*	§5.4, Fig.12	Coordinated transmission throttling: *Multifacet* reacts within 200msec
	§5.4, Fig.13	Reorder buffer: 27% throughput gain with reorder buffer
	§5.4, Tab.5	Packet buffering: Avg. 10% TCP throughput gain, Max. 30%
vs. 802.11n	§5.5, Fig.16(b)	High interference: 6× throughput gain

Table 1: Summary of the key evaluation scenarios and results described in § 5.

to maintain, increase, or decrease the amount of traffic it should send. The transmission throttling is based on MIMD logic.

Synchronous channel migration: We modify Hostapd to insert channel switch announcements (CSAs) in beacon frames and mac80211 driver to send a specific (user-defined) number of beacons with CSA announcements to ensure successful reception of notification. A CSA notifies associated stations that the AP is switching to a different channel at a predefined time (in beacon interval). The modified mac80211 driver decrements the time until channel switch in each successive beacon. After sending the last beacon the Hostapd driver performs the channel switch.

Porting to WNDR 3800: Our initial prototype was built on a six core 2.6 GHz desktop with 8 GB RAM. When porting to our target platform, Netgear's WNDR 3800 (single core, 680 MHz, 128 MB RAM) experienced a 30% performance drop. To ensure optimal performance, we reduced the replaced mutex protected shared data structures with atomic variables in the packet striping and reordering logic resulting in a $\sim$10% improvement in performance. We also replaced workqueue based scheduling mechanism with the simpler and faster *tasklet* data structure leading to performance improvements in tens of Mbps. Finally, our initial port executed the entire *Multifacet* decision logic on a per packet basis leading to severe inefficiency on a single core system due to non-preemptible nature of most operations (e.g., mac80211, ath9k drivers stalling for updating information for every packet). Rewriting the code to operate on batch of packets led to amortizing of the stalling penalty leading to $\sim$15% performance improvement.

5. EVALUATION

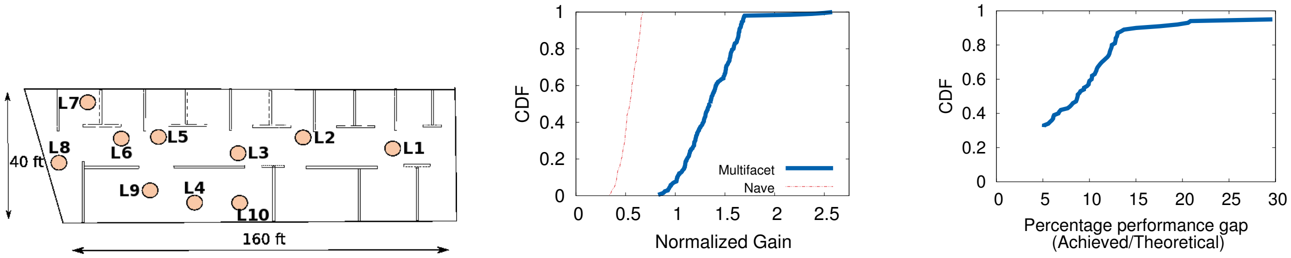
We benchmark the end-to-end performance of *Multifacet* by comparing to that of *Traditional repeater* – a baseline system where only a single link exists between AP–repeater and repeater-client as depicted in Figure 1(a); and *Multifacet*-naive – a simple strawman solution using round-robin discipline for striping across the two interfaces connecting repeater and AP without co-ordinated transmission throttling; under various controlled and uncontrolled settings (§ 5.1). All the repeater tests were conducted using our WNDR3800 based implementation for *Multifacet*. For *Traditional repeater* we used a vanilla OpenWrt image. In § 5.2 we compare the performance of *Multifacet* with *MPTCP*, an end-to-end multipath transport protocol using the modified kernel available at [5]. We have used the *coupled* congestion control algorithm for MPTCP experiments. We have used HighSpeed TCP congestion control algorithm for the results reported in the paper. We have observed similar results when operating TCP with BIC, CUBIC, TCP New Reno congestion control algorithms as well. Results omitted for the sake of brevity. We benchmark the advantages of *Multifacet* over overhearing based mesh routing in § 5.3. We micro-benchmark the efficacy of design choices made in *Multifacet* in § 5.4. Specifically, to understand the benefits of various design choices for link aggregation we conduct experiments with clients connected to the repeater via Ethernet and hence, data destined for the client traverses only a single hop (over two wireless interfaces) from AP to repeater. We use a desktop based setup to stress test the efficiency of various system components with high-bandwidth flows (we use $3X3$ MIMO in desktops comparing to $2X2$ MIMO in WNDR3800). Finally, we highlight connecting to repeater via Ethernet as an interesting design choice for implementing high bandwidth wireless solutions, when a single high speed (802.11n) link is unable to meet such demands (§ 5.5). We summarize the results in Table 1.

Traffic/media sources and metrics: Experiments were conducted in both the 2.4 GHz and the 5 GHz band with 40 MHz channel (unless stated otherwise), with the former being more interference-prone than the latter. We use *aggregate end-to-end* TCP throughput as our primary metric. In addition, we have experimented with end-to-end UDP throughput, HTTP-based dynamic streaming and RTP/UDP-based streaming to characterize performance enhancements offered by *Multifacet* to media applications. For the RTP/UDP-based video experiments, we used the Evalvid mp4trace streaming server [8] to send Ducks video clip [12] of 10 seconds duration encoded at an average bitrate of 20 Mbps (with maximum 40 Mbps). We report *Peak Signal to Noise Ratio (PSNR)*, a measure of perceptual quality of video as our primary metric. For TCP and UDP experiments, each result is the average of 20 iterations of 10 second runs. In different experiments, the traffic traversed some Internet path segment (varying between 0 ms and 100 ms) before it reached the wireless links. The additional delay of the wired segment was controlled using the NetEm tool [11]. The delay was introduced at the traffic source (Iperf or video server) machine and was kept constant for a given experiment (i.e., no variability was introduced).

Testbed: The experiments were conducted in a floor of a University building with around thirty five APs unregulated APs active across all the channels with five or more APs active on the channels on which we were conducting our experiments. Figure 6(a) shows the plan of the testbed and various locations used for different experiments. The presence of many unregulated access points gave us the confidence that our experimental results will be replicable over a large gamut of real world settings.

5.1 Overall Performance

In uncontrolled settings: We randomly select 30 location pairs for the AP, repeater, client and a pair of interfering nodes from the set of locations in Figure 6(a). For each location we compare TCP throughput for *Multifacet*, *Multifacet*-naive normalized to the performance of Traditional repeater and present results in Figure 6(b). *Multifacet* performance is 35% higher than Traditional repeater in 50% of the cases. For about 8% of the cases, the performance of *Multifacet* is same as Traditional repeater. These settings correspond to Zone 1 described in Figure 2. We also find that *Multifacet* performs about 2.5× better than Traditional repeater in the best case and the specific experiment setting corresponds to Zone 2, as it



(a) AP, repeater, client and interferer locations (b) TCP gain (c) Comparison to theoretical opt.

Figure 6: (a) Deployment. (b) *Multifacet* improves on average 35% throughput gain over Traditional repeater (max. 2.5×). (c) *Multifacet* performance is within 15% of the theoretical performance 90% of time. The theoretical performance improvement is calculated using Equation 2

	Bottleneck (channel)	AP	Rpt	Client	Interferer	Placement mapping (Figure 2)
$\mathcal{C}1$	repeater–client (shared)	L1	L2	L5	L6–L8	Scenario A (zone 1)
$\mathcal{C}2$	AP–repeater (shared)	L9	L6	L5	L4–L10	Scenario A (zone 2, till 64 meter in Figure 2)
$\mathcal{C}3$	AP–repeater (non-shared)	L9	L6	L5	L4–L10	Scenario B

Table 2: Three bottleneck scenarios used for controlled experiments in § 5.1. The equivalent scenario in Figure 2 is also mentioned.

allows *Multifacet* to utilize shared channel capacity. These settings correspond to the cases when the capacity of the AP–repeater link is much higher on the shared channel than on the non-shared channel. This result makes sense as the Traditional repeater only utilizes the non-shared channel for uplink, whereas *Multifacet* is able to utilize an higher capacity shared channel in addition to the non-shared channel for uplink. However, *Multifacet*-naive performs worse than Traditional repeater as round-robin scheduling of packets over two interfaces leads to unconstrained transmission by AP–repeater on the shared link with repeater-client link. This in turn reduces the capacity of repeater-client below that of Traditional repeater.

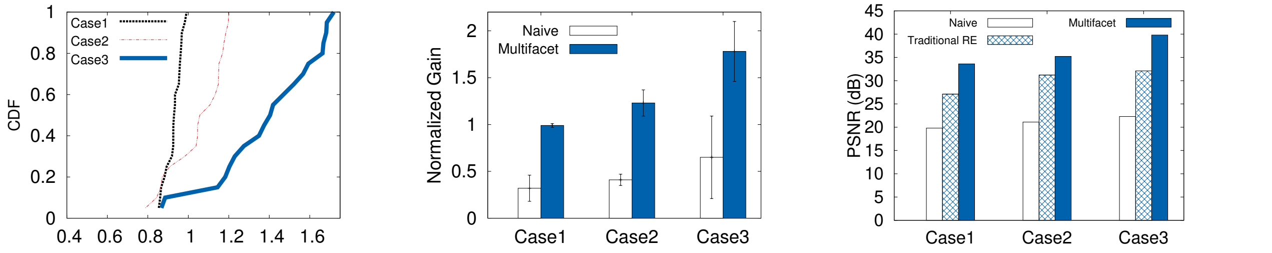
We benchmark the degree of inefficiency of *Multifacet* compared to the optimal bandwidth calculated using Equation (1,2). In Figure 6(c), we find that for 90% of the time *Multifacet* is within 15% of theoretically optimal performance. The inefficiency is due to the backpressure algorithm which operates in a conservative manner. We also notice that in the worst case *Multifacet* performance is 30% lower than theoretically optimal value. This was due to operation on a very weak repeater-client link and in the presence of an uncontrolled interferer which led to multiple triggering of congestion signals leading to TCP throughput drops.

In controlled settings: The performance improvements of *Multifacet* system over Traditional repeater depends on the relative capacity of the three links that it uses. To characterize this dependency of *Multifacet* performance, we create following three cases (Table 2) where the bottleneck link varies: ($\mathcal{C}1$) repeater-client link is bottleneck due to presence of a competing link close to client, ($\mathcal{C}2$) AP–repeater shared channel is the bottleneck due to presence of competing link near AP, and ($\mathcal{C}3$) AP–repeater non-shared channel is the bottleneck due to presence of a competing link on the same channel near repeater. We also mention the matching placement of AP, repeater and client scenarios (from Figure 2 in § 2) alongside each case. The capacity of the non-bottlenecked links were similar and we deterministically varied the capacity of bottlenecked link between 10–50% of the non-bottlenecked links by controlling the capacity of the competing link. The Traditional repeater system uses non-shared channel between the AP and repeater, and the shared channel between the repeater and client. For $\mathcal{C}1$ and $\mathcal{C}2$ we ensured the competing link had low power to not disrupt communication for the other link operating on the same channel.

We measure the end-to-end TCP/UDP throughput and calculate PSNR of RTP video stream under the three cases and present them in Figure 7. Normalized throughput gain is calculated as the throughput ratio of *Multifacet* to Traditional repeater. In $\mathcal{C}1$, repeater–client link is the bottleneck. This makes efficient utilization of spare capacity of shared channel futile. Hence, the performance of *Multifacet* is similar to that of Traditional repeater as the co-ordinated transmission throttling ensures that the AP does not stripe packets on shared link. In contrast, for $\mathcal{C}2$ and $\mathcal{C}3$, repeater–client link is not the bottleneck and hence *Multifacet* can start utilizing the spare shared channel capacity leading to client throughput improvement over Traditional repeater. $\mathcal{C}3$ is the most favorable condition for our system as Traditional repeater is bottlenecked by the AP–repeater link, whereas due to presence of spare capacity in the shared channel *Multifacet* delivers 70% higher throughput to the client (Figure 7(a)). In $\mathcal{C}2$, the expected gains are lower since the AP–repeater link over shared channel can only send minimal extra traffic, however *Multifacet* still provides 20% better throughput in this case.

We also present the normalized UDP throughput gain of *Multifacet* over Traditional repeater in Figure 7(b). Similarly, *Multifacet* improves the throughput significantly (at most 78%) when utilizing the additional link ($\mathcal{C}3$). Figure 7(c) shows that *Multifacet* provides an additional 8.7dB improvement in PSNR comparing to that of Traditional repeater in $\mathcal{C}2$. In all the experiments, *Multifacet*-naive performance is worse than Traditional repeater as the round-robin scheduling at AP stripes a high number of packets on the shared channel without any feedback about the congestion caused to the repeater–client link. This leads to packets being delivered on the non-shared AP–repeater link that cannot be delivered to the client. In contrast, *Multifacet*'s co-ordinated channel access logic desists the AP from sending traffic on the shared channel when enough capacity does not exist.

Multiple clients: We have experimented with four clients and presented the aggregate throughput received by the clients in Table 3. As shown, increasing the number of clients leads to a proportional drop in individual client throughput, while the aggregate throughput remains similar. We did not observe starvation at a specific client. This is expected as the co-ordinated transmission throttling algorithm ensures that the access to shared channel is given to the AP-client links and only when spare capacity is available, the AP-Repeater link is allowed to operate on the shared channel.



(a) Normalized TCP gain (b) Normalized UDP gain (c) PSNR

Figure 7: (a) and (b) *Multifacet* provides significant TCP/UDP throughput gains (up to 1.7×) in case of $\mathcal{C}2$ and $\mathcal{C}3$. (c) *Multifacet* provides good/excellent (MOS) video quality independent of channel conditions. (PSNR of $\geq$37 is excellent, 31-37 is good, 25-31 is fair, 20-25 is poor and $\leq$20 is bad Mean Opinion Score (MOS) [32].)

No. of Clients	*Multifacet* Aggr. (Avg.) TCP throughput	Trad. repeater Aggr. (Avg.) TCP throughput (Mbps)
1	31.2 (31.2)	19.1 (19.1)
2	30.5 (15.25)	19.8 (9.9)
3	31.1 (10.36)	18.2 (6.1)
4	30.8 (7.7)	19.2 (4.8)

Table 3: Aggregate and per client average (in brackets) TCP throughput performance of multiple clients when connected to *Multifacet* and Traditional repeater systems.

5.2 Comparison with MPTCP

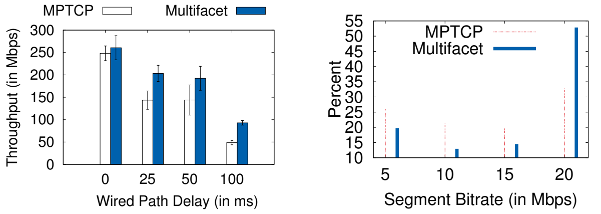
Varying path delay: We compare *Multifacet* and MPTCP's performance, with varying path latency between the source and destination, using the delay setting option of NetEm [11] tool (note that we did not introduce any variability in the delay). Figure 8(a) shows the performance of the two alternatives: (i) MPTCP which treats the two links as two separate paths and implements its congestion control algorithm by deciding how much traffic to send on each, and (ii) *Multifacet* which abstracts this into a single wireless link and uses the HighSpeed TCP congestion control algorithm. In the best case (no additional delay), MPTCP has 16% lower throughput.

MPTCP suffers from *self-interference* problem between two links operating over shared channel as we describe in § 3.3 and hence MPTCP is incapable of saturating the links leading to low performance. In contrast, *Multifacet* actively prioritizes the operation of the repeater-client link over the AP-repeater link resulting in optimal avoidance of self-interference leading to high performance. The performance gap increases with increasing wired path latency, with MPTCP's throughput at 100 ms path delay is about 50% of *Multifacet*. This is expected since *Multifacet*'s backpressure based flow management is closer to the wireless interfaces and hence, more responsive to small variations across two links. In contrast, MPTCP can only react in an end-to-end manner.

HTTP dynamic streaming: Next we compare the performance of *Multifacet* and MPTCP when a client is watching a 20 Mbps video. We encode 30 second segments of a 5 minute clip of Bluray quality (1080p) to create 5, 10, 15, and 20 Mbps video sources. The client fetches different video segments at the bitrate it believes the network has capacity for. Figure 8(b) shows the distribution of segments fetched by client for both *Multifacet* and MPTCP. We observed that 53% of segments fetched by *Multifacet* were 20 Mbps, versus 33% for MPTCP. Overall, the *Multifacet* clients enjoy higher video quality for longer durations than MPTCP clients.

5.3 Mesh routing with overhearing

Overhearing based mesh routing schemes [16, 22] improve performance over traditional mesh by allowing downstream nodes to overhear transmissions from upstream nodes. Overhearing also mandates that overhearing nodes operate on the same frequency as the upstream transmitter. In contrast, *Multifacet* operates the links over independent channels. To compare the performance of overhearing based schemes, we operate our repeater setup with both links on same channel. We calculate the throughput achieved when the client is directly associated with the AP. We assume 100% overhearing and consider the sum of throughput of repeater (over one channel) and the throughput between AP and client as the maximum potential improvements due to repeater with overhearing. Figure 9 shows the throughput attained by *Multifacet*, repeater with one channel and repeater with overhearing. As can be seen from the plot, overhearing improved performance by 21% over a simple repeater. The gains of overhearing are lower as AP is far apart from client which is typical setting for repeater. However, *Multifacet* outperforms repeater with overhearing by ~110% as the penalty of operating on a single channel over two hops (more than halving of bandwidth) is far higher than the gains of overhearing.



(a) Comparing MPTCP to *Multifacet* while varying delays (b) Comparison of the proportion of video segment bitrates

Figure 8: (a) *Multifacet* provides 2× throughput than MPTCP in case of 100 ms delay. (b) *Multifacet* clients fetch more segments at the highest bitrate.

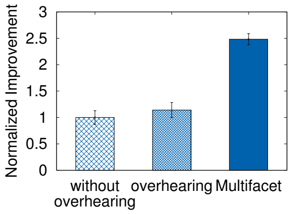


Figure 9: Overhearing provides 20% improvement, while *Multifacet* performs ~110% better than an overhearing repeater

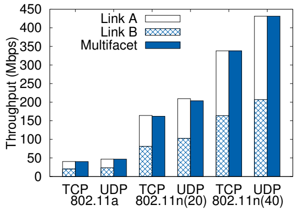


Figure 10: *Multifacet* is able to aggregate almost all available capacity of the two links across multiple technologies and at various speeds of operation.

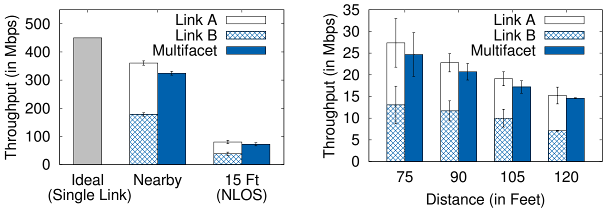
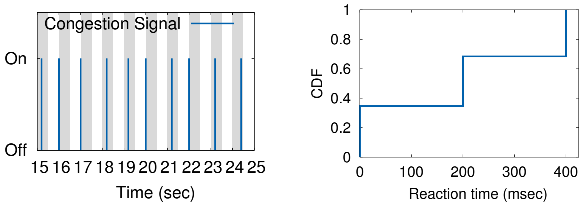


Figure 11: Comparing *Multifacet* with a single 802.11n link. The real applicability of *Multifacet* occurs at a NLOS separation of 70 feet or more.



(a) Congestion control pattern (b) CDF of reaction time

Figure 12: (a) *Multifacet* quickly reacts to interference. (b) Reaction time is less than 200 msec in 70% of cases.

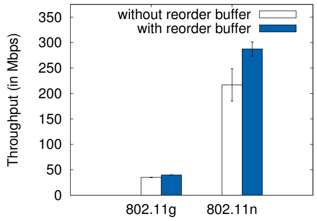


Figure 13: Throughput drops drastically when we do not have a reorder buffer to ensure TCP packets are not delivered out of sequence. The throughput drop is around 5 Mbps (10%) in 802.11a/g mode and 60 Mbps (25%) in 802.11n mode.

5.4 Micro Benchmarks

We study the contribution of specific design features of *Multifacet* to overall performance.

Ability to aggregate bandwidth over single wireless hop at highest speeds: We run the experiments in the desktop setup with WiFi NICs (3×3 MIMO) which offers higher speeds than WNDR3800 with (2×2 MIMO). In Figure 10, we present the throughput obtained using *Multifacet* with two WiFi links when compared to a single WiFi link of the same type. The plot shows both UDP and TCP throughputs when using (i) 802.11a links only, (ii) 802.11n links using 20 MHz channels, and (iii) 802.11n links with 40 MHz channels. We also show the throughput of each individual link when operating in isolation. These experiments were conducted in interference-free channels (achieved using RF cables). Hence, it captures the best case throughputs observed for each scenario. *Multifacet* using dual 3×3 MIMO interfaces is able to provide 337 Mbps TCP throughput and 431 Mbps for UDP. More importantly, in all cases, *Multifacet* is able to provide the sum of each link's bandwidth, and even for TCP throughputs with congestion control mechanisms. The ability to deliver capacity equal to the sum of the bandwidth of constituent links also implies that the additional queuing and buffering latencies of *Multifacet* system are negligible.

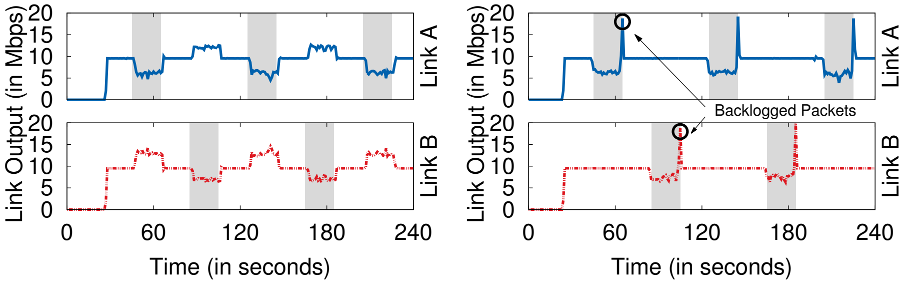
The above results calibrate *Multifacet* performance at the highest throughput cases. For more practical settings, we next evaluate *Multifacet* over-the-air in an uncontrolled (but relatively clean, 5 GHz) environment, where the (non line-of-sight) distance between the client–AP pair was varied. Figure 11 shows that the throughput of *Multifacet* was always within 10% of the aggregate bandwidth of individual links. This figure also brings out one of the key motivations for multi-interface systems. While a single 802.11n link achieves 180 Mbps throughput when minimally separated, it drops to 43 Mbps at a mere non line-of-sight separation of 15 feet, and to about 7-8 Mbps at a separation of 120 feet. In the presence of additional interference, these throughputs drop even further. *It is in these type of scenarios that Multifacet, with its ability to aggregate throughput, has practical use cases in indoor environments.*

Co-ordinated transmission throttling: To evaluate how quickly *Multifacet* adapts to the channel capacity, we project an interference every one sec during 200 sec experiment. The interference traffic at 30 Mbps was sent between two laptops placed near the client laptop with 500 msec ON and 500 msec OFF periods on the shared channel. The interference was sent at a low enough power to ensure no disruption to AP-repeater link over the shared channel. In the ON period, the queue of unsent packets destined for client starts to build up at repeater. We characterize the reaction time of transmission throttling algorithm after introducing the interference.

Figure 12(a) highlights one particular example of time series when the interference is injected and the congestion algorithm is triggered. We have observed similar pattern (quick start of transmission throttling) in the number of experiments and present some part of it for the sake of brevity. We also present the CDF of response time of *Multifacet* in Figure 12(b) as showing almost 70% of response time is less than 200 msec (especially, 35% cases react promptly as soon as the interference is injected) and transmission throttling algorithm is triggered within 400 msec in all cases. This shows the efficacy of our congestion detection and scheduling accordingly. The quick response ensures that *Multifacet* AP-repeater link utilizes the shared channel only when spare capacity is available and backs-off quickly when spare capacity decreases. The reaction time can be further reduced by increasing the frequency of congestion feedback however at the cost of added traffic overhead and more importantly due to increased possibility of reaction to transient channel variations. Based on empirical experimentation we find that 200 msec provides optimal transmission throttling.

Benefits of reorder buffer: We evaluate the utility of a reorder buffer by comparing the performance of *Multifacet* system with and without one. For our experiments we have used a reorder buffer size of 2048 packets, which we have empirically found to work well for links operating in both 802.11a/g and 802.11n mode. In Figure 13, we find that in absence of a reorder buffer the TCP throughput degrades by 5 Mbps for the *Multifacet* system with links operating in 802.11a/g mode while it degrades by 60 Mbps with 802.11n links. The performance penalty is due to the triggering of TCP's retransmission and transmission throttling mechanism in presence of re-orderings in the received data.

Backpressure based striping algorithm: We evaluate the reaction times of the striping algorithm as a function of capacity changes in one of its constituent links. To characterize how quickly the backpressure based data striping algorithm can shift traffic load when one of its links becomes bad, we carry out an experiment, in which we introduce interference in the two channels of operation in an alternating on/off fashion. This interference is in the form of a traffic flow of 20 Mbps during the on duration that last 20 sec at a time. We measure the output from each link through these periods of interference. In our backpressure based system (Figure 14(a)), the output of the interfered link goes down, and causing a shift of the corresponding traffic into the other link. For instance, in time durations 45-65 sec, Link A sheds its load to Link B *immediately* which



(a) Backpressure

(b) Round-Robin

Figure 14: (a) The proportion of packets shift over time in the presence of other stations. (b) The Round-Robin scheme does not compensate for the reduced capacity by scheduling the backlog on the other interface.

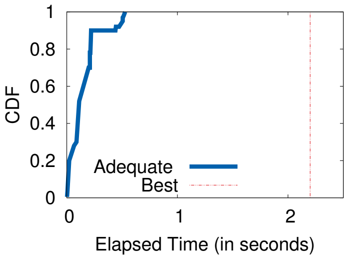


Figure 15: CDF of time duration required by the channel selection algorithms to converge on next channel.

Multi-radio mesh routing	*Multifacet*
1.62	2.15

Table 4: Table comparing performance of a multi-radio joint routing (adapted from ROMA [18]) and channel assignment algorithm with *Multifacet* packet striping. The values are normalized to that of a traditional repeater.

Buffering	TCP Throughput 802.11a/g (Mbps)	TCP Throughput 802.11n (Mbps)
Without	37.21 (1.82)	81.64 (3.34)
With	41.54 (0.29)	89.53 (0.26)

Table 5: Adding a packet buffer in *Multifacet* AP leads to several Mbps of throughput improvements.

proportionally increases. In contrast, in a round-robin based system (Figure 14(b)), no such load shifting would happen, and the interfered link would eventually release its backlogged packets in one burst (around 65 sec), leading to overall poor end-to-end behavior.

Comparison with multi-radio mesh routing: Traditional multi-hop, multi-radio mesh routing and channel assignment algorithms operate at flow level. In other words, they do not stripe packets from individual flows across multiple interfaces. Hence, such algorithms will just offer performance equal to that offered by a traditional repeater system. To highlight the relative benefits of our packet backpressure based packet striping and co-ordinated transmission throttling algorithm, we implement the routing logic of one such routing algorithm ROMA [18] as our packet striping decision logic[2]. Table 4, shows that *Multifacet* offers a 115% performance enhancement over traditional repeater system, compared to 62% gains offered by the routing logic of ROMA. The reason for ROMA's bad performance is that the routing logic explicitly tries to minimize the interference for the repeaterclient link on the shared channel[3]. Thus it is very conservative usage of the shared link. In contrast, *Multifacet* due its tight control loop can aggressively utilized the shared link while leading to a higher performance.

Packet buffering at *Multifacet* AP: To understand the benefits of the small packet buffer at the *Multifacet* AP, we determine the achievable TCP/UDP throughput with and without the packet buffering. We used a packet buffer of 200 packets for experiments and present the results in Table 5. As we can see, installing a packet buffer in the *Multifacet* AP increases TCP throughput by 10% when operating in 802.11a/g and 802.11n mode. The above results validate our decision of placing a small buffer in *Multifacet* AP. Note that above reported experiments were done in normal channel conditions without any external interference. We have found that the gains due to packet buffering goes up to 30% when the link conditions are more variable due to greater channel contention from other sources. The result is in agreement with our intuition that a packet buffer allows us to leverage bursty channel conditions by letting us send packets when the conditions are favorable.

Synchronous channel migration: We now illustrate the advantages of the *Adequate* channel selection algorithm over the more exhaustive but slower *Best* scheme, as described in § 3.4. In our system we limit the maximum number of channels that the AP needs to probe to 10. To trigger a channel switch, we degrade the quality of an operational channel quality by introducing interference. In both approaches, the AP uses a CSA with 2 beacon intervals, and hence the channel switch happens 200 msec after the CSA is triggered by the AP. Since the throughput measurement carried out by the sender takes about 20 msec for each channel, the total time to migrate to the next channel and measure its performance takes about 0.22 sec. Figure 15 shows that in 86% of the cases, *Adequate* needs to probe just 3 channels (i.e., 0.66 sec) to identify a channel that is sufficient to support the offered load. As we probe at most ten channels, the *Best* scheme needs around 2.2 sec to probe all channels and identify the best switch candidate.

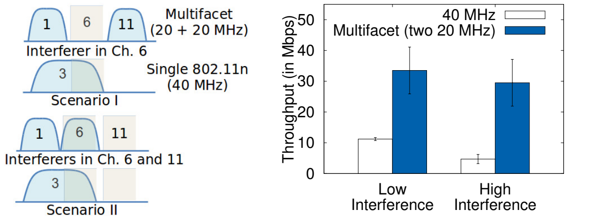
5.5 Does 802.11n and other high speed technologies obviate *Multifacet*?

One could, perhaps, argue that the existing 802.11n standard with a maximum PHY data rate of 450 Mbps (for a 3×3 MIMO system) is adequate for all media delivery needs, and hence eliminates the need for a multi-radio solution. Our experiments reveal that under good channel conditions and minimal signal attenuation, a single 802.11n link can achieve throughputs in the 60-210 Mbps range. In this performance range, the client connected via *Multifacet* would achieve an even better throughput. However, at these speeds, the single 802.11n solution is more than adequate for all practical HD media applications, including the scenario where we attempt to stream a Blu-ray player to a nearby television set.

While the use of these high speed technologies advance the throughputs at the high end, when the signal attenuates across one or more walls in an indoor environment, or if there is sufficient interference in the channel, the actual throughputs of 802.11n can degrade sharply *to below 20 Mbps, and sometimes even below 5 Mbps.* It is in these lower throughput ranges that a throughput aggregation

[2] To maintain fairness, we implement packet reordering logic for ROMA as well.

[3] Minimizing intra-path interference is the desired behavior of ROMA. When we let ROMA reduce interference over AP-repeater link its performance is worse



(a) Interference scenarios (b) Throughput gain of *Multifacet*

Figure 16: (a) A single 40 MHz 802.11n link (using 3×3 MIMO) performs far worse than a *Multifacet* system configured to use a total of 40 MHz of spectrum, using AP-repeater links operating on two 20 MHz channels on the 2.4 GHz band. (b) Experiment corresponding to interference Scenario II shown in (a).

system such as *Multifacet* becomes necessary and useful. A dual radio AP-repeater link can easily achieve double the throughput in these scenarios as it uses twice the spectrum (of a single WiFi link).

This gain is, however, not simply because of a wider spectrum occupancy of AP-repeater link. The flexibility provided by the multiplicity of wireless interfaces is a significant contributor to performance gain. If we limit the dual radio AP-repeater link to a total of 40 MHz (each WiFi interface uses a 20 MHz channel) and compare it with a single 40 MHz 802.11n link, in certain cases, we still observe performance advantages of our solution. For instance, in Scenario I shown in Figure 16(a), there is interference in channel 6 of the 2.4 GHz band (say, due to an independent AP). A single 40 MHz 802.11n link operating in this part of spectrum would experience significant interference from this source, while the proposed solution would be able to easily avoid this interferer. Even in Scenario II shown, where both solutions experience interference in at least 20 MHz of their operating spectrum, the overall impact of interference is far less for dual link AP-repeater link.

Figure 16(b) shows the significant advantages of the multi-interface approach to flexible channel bonding (operating at the MAC layer) compared to the PHY layer bonding approach. The figure shows results from experiments conducted in the 2.4 GHz band in corresponding to Scenario II of Figure 16(a). In this case, there are two other interferers occupying channels 6 and 11 and the AP-repeater picks channels 1 and 6 for operation. In contrast, the 802.11n 40 MHz system (operating with same amount of spectrum) picks channel 3. When the interferers emit a low volume of interfering traffic, *Multifacet* AP-repeater link aggregation achieves *more than double the throughput while occupying the same amount of spectrum.* When the interferers emit a higher volume of interference, *the gains are more than 6×.*

6. RELATED WORK

Multi-interface networking: Networking over multiple wired interfaces has been used in context of bandwidth aggregation, load-balancing and multi-homing. For example, researchers [26, 27, 38] presented mechanisms to bundle multiple, highly predictable, wired ATM links to construct a high bandwidth link. In contrast, in *Multifacet*, the WiFi links have significant variability and unpredictable changes in channel conditions and our solution is stylized to the various properties of two hop wireless networks.

Multipath TCP [29] and SCTP [37] are transport layer protocols designed to send a single flow over multiple paths. These schemes need modifications to end host systems and applications to handle packet re-ordering and related issues. In contrast, *Multifacet* is a custom dual interface repeater which does no need application of system modifications at end host modifications. Authors in [25, 33] study end-to-end flow striping techniques, however, our flow striping is not end-to-end but restricted to a single hop.

Akin to MPTCP, mHTTP [23] utilizes simultaneous multi paths over a network which only requires application layer supports and modifications on socket interface. In contrast, *Multifacet* is implemented on MAC and network layers for efficiently operating multiple links while maintaining transparency.

Multi-interface links: Glia [31] is a system, similar to *Multifacet* to an extent as they explore the use of parallel WiFi links simultaneously for high throughput. However, it differs in the core challenge being addressed by *Multifacet* – Glia does not stripe packets of a single (TCP or UDP) flow over different WiFi links and instead focuses on the aggregate throughputs of a large number of flows partitioned across different links. FatVAP [28] is a system to opportunistically aggregate bandwidth from multiple WiFi APs using a single client WiFi radio interface. Again, in contrast to *Multifacet*, the FatVAP system does not stripe packets from a single flow over these different APs. Finally, the MAR system [35] explores bandwidth aggregation over multiple low bandwidth 2.5G links for vehicular applications. In their experiments, the system limits each flow to a single 2.5G network only. In contrast, we explore the challenges of striping a single flow over two high-speed WiFi links.

Multi-hop wireless communication: Authors in [13–21] have proposed channel assignment, routing, network coding etc. for improving the performance of mesh networks. Researcher in [24, 30, 36] have presented resource reuse and rate allocation algorithms to exploit spectral efficiency in mesh networks. We note that *Multifacet*'s gains are independent of rate adaptation algorithm used in constituent links. This is due to the fact that *Multifacet* deals with efficient scheduling of traffic over the links operating between AP-Repeater-Client. In contrast, a rate adaptation algorithm will be concerned with identifying the best modulation and coding scheme (MCS) for a given channel condition and not with when and how many packets to send on a given link. The key idea in such prior works is to optimally leverage multi-radio capability for improving throughputs across such networks. Our two-hop configuration of *Multifacet* is a special case of the multi-hop multi-radio mesh network, where there are two parallel links in the upstream hop and a single link in the downstream hop. Additionally, once a channel is assigned to the downstream hop (to the client), it cannot be easily changed or else this can be disruptive to the client when a high bandwidth flow is active. In our design of *Multifacet*, we focus primarily on techniques to opportunistically use the excess bandwidth of upstream links to enhance performance. Such prior works *operate at a flow level and do not address the problem of splitting individual flows across multiple interfaces*. The key uniqueness of *Multifacet* from prior mesh networking literature, thus, comes from its focus on efficiently splitting a flow over multiple interfaces at high data rates under varying channel conditions. Hence, this paper is quite complementary to the general body of work in the context of multi-radio multi-hop wireless mesh networks.

Wireless range extenders: There are several commercial dual band range extenders [2–4], in addition to traditional routers [1,6], which aim to provide extended wireless coverage both indoors and outdoors. In spite of having multiple radios, these products simply re-broadcast traffic from the AP based on a static configuration. *Multifacet* improves upon this by intelligently scheduling packets sent to the repeater while providing a single link abstraction. Range extenders [9, 10], which operate both links on a single channel are inferior in performance to dual band wireless range extenders and have been superseded by dual band devices.

Future MIMO based systems: MIMO based technologies such as directional MIMO antennas [7] have been used in outdoor settings to increase the range and capacity of wireless systems. Versions of such techniques customized to operate in indoor settings can provide an alternative solution for extending wireless coverage. Thus, obviating the need for repeater altogether. The main difference of *Multifacet* from such systems is that we make software only modifications on COTS repeater to enhance performance of state-of-art repeater systems, whereas, MIMO systems will necessitate hardware changes.

7. CONCLUSIONS

Wireless repeater devices are a special case of two hop wireless mesh setups and are extensively used in various settings to enhance the quality of wireless access for clients. Repeater products are available today from all leading vendors (Netgear, Linksys, Belkin, DLink, Apple, etc.), and especially dual-interface repeaters are top selling products. In this work, we present the design, implementation and evaluation of *Multifacet* – a software solution for dual-interface repeaters for providing increased bandwidth capacity to clients. *Multifacet* needs only software modifications to work on off-the-shelf dual band wireless APs and repeater. We demonstrate its superiority over conventional approaches through experimentation in varied wireless environments. Video demonstration of the performance of *Multifacet* can be found at `http://youtu.be/0W1MjlT6MkY`

Acknowledgements

We thank Michael Bower for his help in conducting repeater experiments. We are grateful to the anonymous reviewers whose comments helped bring the paper to its final form. All authors are supported in part by the US National Science Foundation through awards CNS-1040648, CNS-0916955, CNS-0855201, CNS-0747177, CNS-1064944, CNS-1059306, CNS-1345293, CNS-1343363, and CNS-1258290.

8. REFERENCES

[1] Apple AirPort Express, `http://www.apple.com/airportexpress`

[2] Cisco range extender, `http://www.pcmag.com/article2/0,2817,2399489,00.asp`

[3] Belkin range extender, `http://www.belkin.com/us/F9K1106-Belkin/p/P-F9K1106`

[4] Netgear range extender, `http://netgear.com/home/products/wireless-range-extenders/WN2500RP.aspx`

[5] MultiPath TCP - Linux Kernel implementation, `http://mptcp.info.ucl.ac.be`

[6] OpenWrt, `http://www.openwrt.org`

[7] Directional outdoor MIMO, `http://www.fortinet.com/products/fortiap/outdoor-MIMO-antennas.html`

[8] EvalVid - A Video Quality Evaluation Tool-set, `www.tkn.tu-berlin.de/research/evalvid/`

[9] Linksys RE1000, `http://support.linksys.com/en-us/support/rangeexpanders/RE1000`

[10] Netgear WN2000RPT, `http://www.netgear.com/home/products/wireless-range-extenders/wn2000rpt.aspx`

[11] Network Emulation, `http://www.linuxfoundation.org/collaborate/workgroups/networking/netem`

[12] Xiph.Org–Test Media page, `http://media.xiph.org/video/derf`

[13] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks", *ACM Sigcomm*, 2005.

[14] K. Chebrolu, B. Raman and S. Sen, "Long-Distance 802.11b Links: Performance Measurements and Experience", *ACM MobiCom*, 2006.

[15] J. Camp, J. Robinson, C. Steger and E. Knightly, "Measurement driven Deployment of a Two-tier Urban Mesh Access Network", *ACM MobiSys*. 2006.

[16] S. Chachulski, M. Jennings, S. Katti and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing", *ACM Sigcomm*, 2007.

[17] B. Ko, V. Misra, J. Padhye and D. Rubenstein, "Distributed Channel Assignment in Multi-Radio 802.11 Mesh networks", *WCNC*, 2007.

[18] A. Dhananjay, H. Zhang, J. Li and L. Subramanian, "Practical, Distributed Channel Assignment and Routing in Dual-radio Mesh Networks", *ACM Sigcomm*, 2009.

[19] A. Raniwala and T. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Radio Wireless Mesh Networks", *IEEE Infocom*, 2005.

[20] K. Ramachandran, I. Sheriff, E. Belding and K. Almeroth, "A Multi-Radio 802.11 Mesh Network Architecture", *ACM Mobile Networks and Applications*, 13(1):132-146, 2008.

[21] R. Draves, J. Padhye and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks", *ACM MobiCom*, 2004.

[22] E. Rozner, J. Seshadri, Y. Mehta and L. Qiu, "SOAR: Simple Opportunistic Adaptive Routing Protocol for Wireless Mesh Networks", *IEEE Transactions on Mobile Computing*, 2009.

[23] J. Kim, R. Khalili, A. Feldmann, Y-C. Chen and D. Towsley, "Multi-Source Multi Path HTTP (mHTTP): A Proposal", *arXiv*, 2013.

[24] L-W. Chen, Y-C. Tseng, Y-C. Wang, D-W. Wang and J-J. Wu, "Exploiting Spectral Reuse in Routing, Resource Allocation, and Scheduling for IEEE 802.16 Mesh Networks", *IEEE Transactions on Vehicular Technology*, 58(1), 2009.

[25] K. Chebrolu, B. Raman and R. Ramesh, "A Network Layer Approach to Enable TCP over Multiple Interfaces", *ACM Wireless Networking*, 11(5):637-650, 2005.

[26] J. Duncanson, "Inverse Multiplexing", *IEEE Communications Magazine*, 1994.

[27] P. Fredette, "The Past, Present, and Future of Inverse Multiplexing", *IEEE Communications Magazine*, 1994.

[28] S. Kandula, K. C-J. Lin, T. Badirkhanli and D. Katabi, "FatVAP: Aggregating AP Backhaul Capacity to Maximize Throughput", *ACM NSDI*, 2008.

[29] H. Han, S. Shakkottai, C. Hollot, R. Srikant and D. Towsley, "Multi-path TCP: A Joint Congestion Control and Routing Scheme to Exploit Path Diversity on the Internet", *IEEE/ACM Transactions on Networking*, 14(6):1260-1271, 2006.

[30] J. Tang, G. Xue and W. Zhang, "Cross-layer Optimization for End-to-End Rate Allocation in Multi-Radio Wireless Mesh Networks", *Wireless Networks*, 15(1), pp 53-64, 2009.

[31] S. Kakumanu and R. Sivakumar, "Glia: A Practical Solution for Effective High Datarate Wifi-Arrays", *ACM MobiCom*, 2009.

[32] Z. Orlov, "Network-Driven Adaptive Video Streaming in Wireless Environments", *IEEE PIMRC*, 2008.

[33] D. S. Phatak and T. Goff, "A Novel Mechanism for Data Streaming Across Multiple IP Links for Improving Throughput and Reliability in Mobile Environments", *IEEE Infocom*, 2002.

[34] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik and M. Handley, "Improving Datacenter Performance and Robustness with Multipath TCP", *ACM Sigcomm*, 2011.

[35] P. Rodriguez, I. Pratt, J. Chesterfield, R. Chakravorty and S. Banjeree, "MAR: A Commuter Router Infrastructure for the Mobile Internet", *ACM Mobisys*, 2004.

[36] S. Kompella, S. Mao, Y. T. Hou and H. D. Sherali, "On Path Selection and Rate Allocation for Video in Wireless Mesh Networks", *IEEE/ACM Transactions on Networking*, 17(1):212-224, 2009.

[37] R. Stewart, "Stream Control Transmission Protocol", *RFC 4960, IETF*, 2007.

[38] C. B. S. Traw and J. M. Smith, "Striping within the Network Subsystem", *IEEE Network*, 9(4):22-32, 1995.

[39] D. Wischik, C. Raiciu, A. Greenhalgh and M. Handley, "Design, Implementation and Evaluation of Congestion Control for Multipath TCP", *ACM NSDI*, 2011.