Glia: A Practical Solution for Effective High Datarate Wifi-Arrays

Sandeep Kakumanu and Raghupathy Sivakumar
Georgia Institute of Technology
Atlanta, GA 30313, USA
{ksandeep,siva}@ece.gatech.edu

ABSTRACT

Wi-fi standards have provisions for multiple orthogonal channels where the orthogonality allows them to be used simultaneously both in time and frequency without interference concerns. In this paper, we pose the following question: *Can devices use the multiple orthogonal channels in wi-fi networks simultaneously to realize a high data-rate wireless link and hence cater to applications requiring high bandwidths?* In other words, given that there are 3 orthogonal wi-fi channels in the 2.4GHz band and 12 orthogonal wi-fi channels in the 5GHz band, can a pair of devices each equipped with 15 wi-fi radios use all the available orthogonal channels to achieve a high data-rate link operating at 600Mbps? Surprisingly, we find through experimental evaluation that the actual observed performance when using all fifteen orthogonal channels between two devices is a mere *91Mbps*. We identify the reasons behind the low performance and present Glia, a software only solution that effectively exercises all available radios. We prototype Glia and show using experimental evaluations that Glia helps achieve close to 600Mbps data-rate when using all possible wi-fi channels.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Experimentation, Performance

1. INTRODUCTION

Even with the advancements made in wireless data technologies, there still remains a need to continue to bridge the bandwidth gap that exists between wire-line and wireless data networks. Wi-fi networks using IEEE 802.11 (a, g, or n) standards have considerably improved data-rates in wireless LANs, but also have provisions for multiple orthogonal channels that may be used by different networks operating in the same vicinity. The orthogonality of the channels allows them to be used simultaneously both in time and space without interference concerns. In this paper, we pose the following question: *Can devices use the multiple orthogonal channels in wi-fi networks simultaneously to realize a high data-rate wireless link and hence cater to applications requiring high bandwidths?* In other words, given that there are 3 orthogonal wi-fi channels in the 2.4GHz band and 12 orthogonal wi-fi channels in the 5.2GHz band, can a pair of devices each equipped with 15 wi-fi radios use all the available orthogonal channels to achieve a high data-rate wi-fi link?

We believe that such high data-rate wireless links will have use in greenfield environments where co-existence with pre-deployed networks is not a concern. Examples of such networks include enterprise network deployments and wireless backhauls for wireless mesh networks. Furthermore, even in environments that have prior wi-fi deployments, a solution that is fully backward compatible with normal wi-fi links and opportunistically provides high data-rate communication capabilities will indeed be desirable. We term such a set-up with multiple wi-fi radios mounted on a single device as a *wi-fi array*.

To the best of our knowledge, no efforts have been undertaken in related research to characterize achievable data-rates when using wi-fi arrays with all possible orthogonal channels in the 5.2GHz and 2.4GHz spectrum. Hence, we first experimentally determine the achievable data-rates using off-the-shelf (OTS) wi-fi radios. We use Microtik R52 miniPCI cards mounted on Routerboard IA/MP8 8-slot miniPCI-to-PCI adapters for our experimental set-up. Surprisingly, we find that while the expected application layer data-rate with a wi-fi array that uses 15 orthogonal channels (12 'a' and 3 'g') is approximately 600Mbps, *the observed performance is a mere 91Mbps*. We delve into this observation and identify two phenomena, both pertaining to the close physical proximity of the radios on the wi-fi array that together cause the performance degradation. Specifically, we find that *out-of-band (OOB) emission of energy* at a transmitting radio is strong enough at short distances (<1m) that it can trigger carrier sensing at a *nearby* radio operating on an orthogonal channel, and also corrupt the reception of packets at the other radio if it were receiving. Secondly, we find that *filter inefficiencies,* when two radios in close proximity are operating on orthogonal channels, also increases effective bit error rates further lowering performance.

We then present *Glia*[1] a practical software only solution that coarsely coordinates the different radios on a wi-fi array and in the process delivers the aggregate data-rate expected from the array. Glia uses a combination of medium access, scheduling, framing and channel management mechanisms that allow the radios on the wi-fi array to overcome the aforementioned problems. Perhaps, more importantly, we realize Glia as a software module that works

[1]Glia, Greek for 'glue', is a solution that effectively glues together wi-fi radios.

MobiCom'09, September 20–25, 2009, Beijing, China.

with any *off-the-shelf wi-fi radios, thus requiring no changes to the hardware or firmware* of the radios themselves. Using experimental evaluation, we demonstrate that Glia, with a 15 radio wi-fi array (12 'a' radios and 3 'g' radios) achieves approximately 600Mbps[2].

Note that there are several approaches to achieve high datarate wireless communication. Some of these techniques include *channel bonding* [1], *using higher frequency ranges of the spectrum* [2, 3], *wideband techniques* [4], *directional antennas*, *MIMO and adaptive array communication* [1], *radio bonding* [5, 6] and *advanced PHY layer techniques* [7, 8]. However, there are a few fundamental differences, and hence advantages, to the Glia approach to achieving high data rates: (i) Unlike all of the above solutions, Glia is a pure software based solution that works with *off the shelf radios*. We believe that this is a significant advantage when it comes to deployability and time to availability of the solution. (ii) wi-fi is by far the most ubiquitous wireless technology deployed in data networks today, and Glia is built a top wi-fi, and perhaps equally importantly is fully backward compatible with legacy wi-fi devices. (iii) Finally, to the best of our knowledge, despite the promise of high data rate wireless communication that other solutions offer, Glia is the first demonstrated experimental working solution that offers upwards of 600Mbps in data rate. We delve into other specific differences between Glia and the aforementioned alternatives later in the paper.

The contributions of this work are three-fold:

- We experimentally study the performance of a 15 radio wi-fi array and characterize the data-rate performance achievable using OTS radios as being a mere 91Mbps. We then identify the reasons behind the lower than expected performance.
- We present Glia, a software only solution effectively exercising a wi-fi array that coarsely coordinates the different radios on a wi-fi array to achieve the expected aggregate performance.
- We prototype Glia and demonstrate in a real experimental set-up that close to 600Mbps data-rate is achieved using only OTS wi-fi radios.

The rest of the paper is organized as follows: In § 2,we describe our setup of wifi-arrays and present the results of default testing of the setup. We also analyze the reasons behind poor performance in the default 802.11 operation. In § 3, we explore the core principles of our solution, Glia. In § 4, we present the software architecture of Glia and explain how each component of the solution works. In § 5, we present the performance evaluation of Glia using an implementation on the wifi-array testbed and also using ns2 based simulations. In § 6, we present the related work in this field, and finally we conclude the paper in § 7.

2. BASELINE PERFORMANCE AND MOTIVATION

2.1 Testbed Setup

First, we explain the setup used for experimentation. Two Intel core-2 based Dell Optiplex GX 520 desktops, running Ubuntu Linux (version 8.04, kernel 2.6.24), and equipped with 12 WLAN radios each, act as source and destination wifi-arrays. Since all the arguments we present in the paper are relevant only within a single band, we restrict the scope of the experimental set-up to only 12 radios belonging to the 802.11a 5.2GHz band. However, we revisit a complete set-up with 15 radios (12 a and 3 g) in the performance evaluation section. Atheros chipset (AR5413) based Microtik R52 802.11a/b/g miniPCI cards are used as WLAN radios. The cards are mounted on two Routerboard [9] IA/MP8 8-slot miniPCI-to-PCI adapters, each housing 6 cards. The open source Madwifi [10] driver is used for the WLAN cards. The 12 radios together occupy all the 12 available channels in the 5.2GHz spectrum. For the baseline experimentation, the Iperf application is used for generating UDP traffic. The source and destination wifi arrays are placed 10 meters apart. The RTS/CTS of the 802.11MAC protocol is turned off for all experiments. Figure 1 shows a photograph of one of the two wifi-arrays with 12 radios, while Figure 2 shows a schematic of the 12 radio wifi-array testbed.

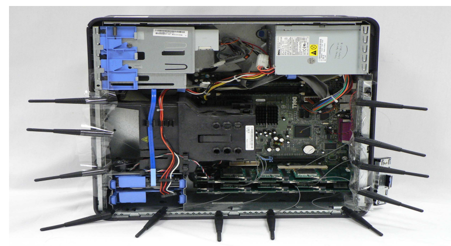


Figure 1: 12 radio wifi-array

2.2 Baseline Experimentation

In this Section, we present results of the baseline experimentation using the testbed.First, the individual per-channel data-rate is observed to be around 40Mbps [3], by running only one UDP iperf session across each channel at a time. The 12 channels used by the radios are supposed to be 'orthogonal', i.e, communication on one channel should not affect communication on any of the other channels. Thus the expected aggregate throughput, when all the 12 radios are operated simultaneously, should be around 480Mbps (40* 12). However, when simultaneous UDP iperf sessions are setup on each of the 12 channels, the observed aggregate throughput is only 70Mbps. Figure 4 shows the variation of aggregate throughput as a function of the number of simultaneous links active at the same time. Thus *only 15% of the ideal aggregate throughput capacity* is observed when off-the-shelf radios are used as-is for the wifi-arrays (OTS Wifi).

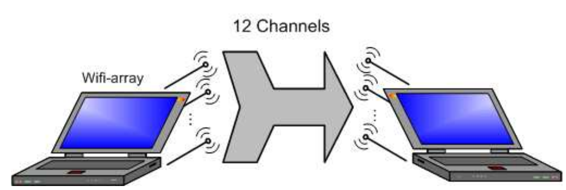


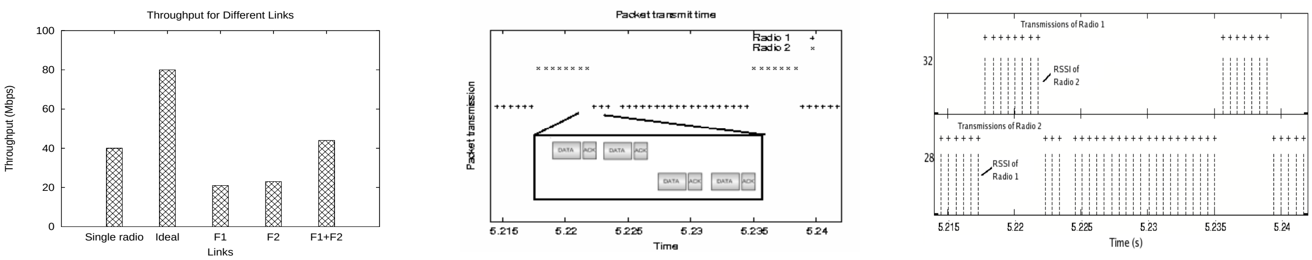
Figure 2: Schematic of 12-radio wifi-array Testbed

2.3 Analysis

In the previous section we observed that using all 12 channels at the same time using collocated radios gives a lower than expected

[2]While we don't perform extensive tests of Glia with 802.11n due to current bus speed limits in our experimental set-up, we do show a proof-of-concept that Glia works with 802.11n as well. Thus, a full set-up with Glia and 802.11n in the 2.4GHz and 5.2GHz bands could achieve over 1Gbps in data-rate.

[3]Note that the throughput we mention here is the actual application-level achievable throughput from the 802.11 links and not the raw datarates that the 802.11 standard specifies.



(a) Throughputs (b) Wireshark Analysis (c) Packet Error Rates

Figure 3: Experimentation with Collocated Tx/Tx

throughput performance. However, in practice WLAN network deployments do use orthogonal channels simultaneously. The key differentiating property of our experimental set-up when compared to such typical WLAN network deployments is the physical proximity between the radios using the orthogonal channels. To verify that this factor is indeed the reason for the poor performance we use a simple two channel experiment. Two adjacent channels in the 802.11a band (5.18GHz and 5.2GHz) are used for analysis. Figure 5 (a) shows the topology of the experiment, where two links operating on adjacent channels are setup. In this setup the two transmit radios are kept far apart (similarly the two receive radios are also kept far apart). However, the two transmit radios (similarly the two receive radios) are within transmit range of each other. The difference between this setup and a wifi-array setup with two radios is the absence of proximity between the radios. When the two links are active at the same time, the aggregate throughput is observed to be 78Mbps which is close to the ideal aggregate throughput of two channels. This points the reason for poor performance of the wifi-array setup to the proximity of the radios at the transmit and receive nodes.

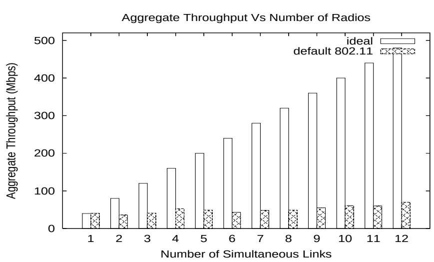


Figure 4: Throughput vs number of radios

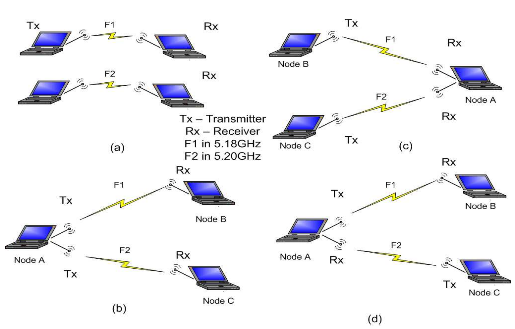


Figure 5: (a) Non proximal radios (b) Collocated Tx/Tx; (c) Collocated Rx/Rx; (d) Collocated Tx/Rx

To understand what exactly happens at each of the transmit and receive wifi-arrays, we experiment with a three node (A, B and C) setup, where node A has two radios while nodes B and C have only one radio each. Nodes B and C are placed far apart. The two radios at node A connect to either of nodes B or C on adjacent channels (5.18GHz and 5.2GHz). Depending on the direction of DATA flow in each of the two links, there are three possible scenarios, as studied below:

2.3.1 Collocated Tx/Tx

In this scenario, both the radios of node A are used for transmission(Tx) of DATA packets (refer Figure 5 (b), while nodes B and C act as receivers. Figure 3 (a) shows the ideal throughput of the two radio setup, and the observed individual and aggregate throughputs. We refer to the two links as F1 and F2. While the expected aggregate throughput is 80Mbps, the observed throughput is only 44Mbps. Thus, single link throughput is what is observed in-spite of the fact that two links on orthogonal channels are active at the same time. A deeper inspection, using the Wireshark packet analyzer shows that in fact only one link is active at any given time. Figure 3(b) is a visualization of the wireshark dump, which shows the times at which packets are sent across the two links. The figure also shows a zoomed version of a part of the visualization. It is clearly seen at any given time only packets belonging to one link are sent. This phenomenon occurs in-spite of the two radios of node A operating on orthogonal channels.

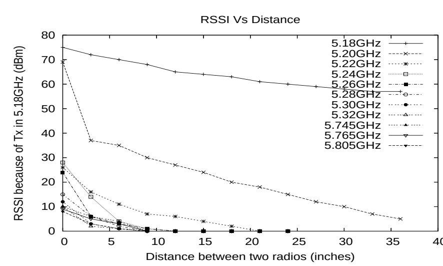
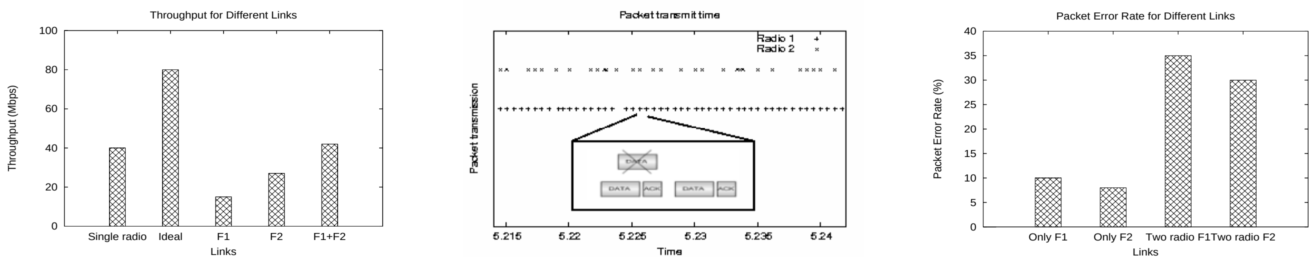


Figure 6: RSSI vs Distance

To verify the above phenomenon we investigate the RSSI (Received Signal Strength Indicator) values at both the radios of node A. The RSSI is used by 802.11 radios to perform physical carrier sense and is available readily as a hardware register on the physical device. Figure 3 (c) shows the RSSI at each radio of node A, when the other radio is transmitting DATA packets. It is observed that each radio records a finite RSSI when the other radio is transmitting. This RSSI triggers carrier sensing at either radio and prevents it from transmitting a packet when the other radio is transmitting. Thus even though the two channels are technically orthogonal to each other, there is some power leakage from a transmitting radio on one channel to the other. This leakage power is termed as Out-Of-Band (OOB) emission, and has been discussed in related literature [11].



(a) Throughputs (b) Wireshark Analysis (c) Packet Error Rates

Figure 7: Experimentation with Collocated Rx/Rx

Delving further, we characterize this OOB by studying RSSI values using different channels and different distances between the two collocated radios of node A. Figure 6 shows the variation of RSSI observed on one radio as a function of distance from the second radio, when the second radio is transmitting packets. In the figure we note that when the two radios are placed very close to each other, even channels that are as far as 5.18GHz and 5.805GHz (channels at extreme ends of the 802.11a spectrum) can be affected because of OOB emissions. *This power leakage can however be anticipated and the physical carrier sense mechanism can be suitably modified to account for the OOB.*

2.3.2 Collocated Rx/Rx

In this scenario, both the radios of node A are used for receiving(Rx) DATA packets (refer Figure 5 (c)). As in the previous scenario, Figure 7 (a) shows the ideal and observed throughputs of the setup. The observed aggregate throughput of the two links is 45Mbps. Again single link performance is what is observed. To investigate further, we perform Wireshark analysis of the two links. Figure 7 (b) shows a visualization of the times at which packets are sent on each link. While in the previous scenario, it was observed that only one link was active at any given time, in this scenario, packets are sometimes sent on either link at the same time. However, the aggregate throughput is low. To dig deeper, we zoom into the visualization and observe that some packets on either links do not start exactly at the same time, but overlap each other. In this case the reverse direction ACK from one of the radios overlaps with the DATA reception on the other. The ACK for the other DATA packet is not sent back, indicating a packet error. This reverse direction ACK will cause errors on the other DATA packet reception because of OOB emission. Further, we analyze the packet error rates of the received DATA packets [4]. Figure 7 (c) shows the packet error rates on each of the two radios, of node A, while under individual and simultaneous operation. The packet error rates are significantly higher under simultaneous operation confirming the earlier hypothesis that reverse direction ACKs can corrupt DATA reception. This phenomenon of ACKs corrupting DATA occurs irrespective of the channels used by the two radios (as long as the two channels are within the same band), albeit to varying degrees. *Thus it can be concluded that ACKs corrupt DATA.*

From the above observation, turning OFF 802.11 ACKs should result in ideal aggregation of the two links (assuming no background interference). However, a second phenomenon is observed when adjacent channels are used for the two Rx radios. Packet errors are observed in the reception of DATA packets in either radios. The packet error rate, and hence the aggregate throughput is different on the two radios, and varies depending on the location of node A. Even small differences in location can lead to a widely varying observed throughputs. The aggregate throughput is also affected by the adjacent channels being used by the two radios for the same location. However, the throughput remains fairly constant for a considerable amount of time (in the order of a few hours). Table 1 shows variation of aggregate throughput of the two radios and packet error rates with location and adjacent channels used. 802.11 ACKs are disabled for these experiments. The two different locations studied (1 and 2) are only 3 inches apart. As explained in [12] this phenomenon occurs because of the imperfect filter operations at the receive radios. The power from a transmission on a neighboring channel can be filtered along with the legitimate power on the current channel at a receiving radio. This extra power acts as interference and causes CRC errors resulting in packets being dropped at the receive radio. The effect of the extra power is observed only when the channel gains for the adjacent channel is high enough. The channel gains for the receive power can vary depending on location, time and channels being used.

Table 1: Packet Error Rates and Aggregate Throughput for Different Locations and Different sets of Adjacent channels used

Channels/Location	PER	Thrpt(Mbps)
5.18, 5.20/ A	0.01, 0.1	75.6
5.18, 5.20/ B	0.32, 0.24	56.7
5.24, 5.26/ A	0.5, 0.1	58.0
5.24, 5.26/ B	0.2, 0.21	62.8

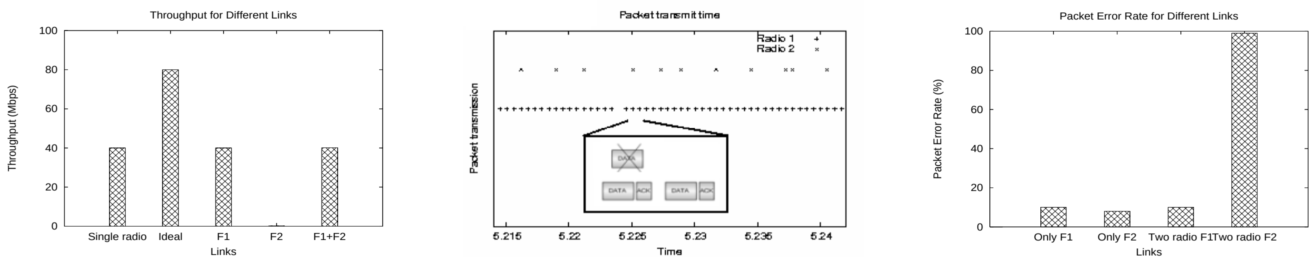
2.3.3 Collocated Tx/Rx

In this scenario (refer Figure 5 (d)), one radio of node A transmits packets (link F1) while the other radio receives packet (link F2). The throughput results in Figure 8 (a) indicate that while F1 gets a high throughput of 38Mbps, F2 gets only 1 Mbps. Wireshark analysis shows that while DATA packets are present in both the links, very few packets of F2 are ACKed. The reason for this phenomenon can also be attributed to the OOB emissions from the transmit radio of node A, which make it almost impossible for the other radio to decode its received DATA packets. Figure 8 (c) shows the unusually high packet error rates for F2, when both the radios are operating simultaneously. Thus, it can be concluded that it is not possible to simultaneously transmit and receive using collocated radios. *Since DATA transmission on one radio corrupts DATA reception on a collocated radio, simultaneous DATA transmission and reception through collocated radios is never possible.*

3. DESIGN ELEMENTS IN GLIA

In this section we present two broad design elements that allow aggregation of throughput capacities of multiple orthogonal channels. These design elements are based on the insights derived from the previous section. In § 4, we propose a software-only solution,

[4]The packet error rates can be figured from a hardware register on the physical WLAN device



(a) Throughputs (b) Wireshark Analysis (c) Packet Error Rates

Figure 8: Experimentation with Collocated Tx/Rx

known as Glia, using these two principles. The first principle, act-as-one, coarsely bonds individual radios and creates a single logical radio, that can use all the available channels at the same time. The second principle, exploit-the-many, allows the right radio-channel association for both the transmitting and receiving wifi-array, such that maximum aggregate throughput can be achieved. The two principles are explained in detail below.

3.1 Act-as-One

This design element facilitates multiple radios in a wifi-array to act as one single radio occupying all the channels at the same time. The key concept is to use coarse synchronization of the radios and allow near simultaneous transmission of data packets on the collocated radios.

3.1.1 Mutually exclusive Rx/Tx

In §s 2.3.2 and 2.3.3, we identify that transmission of a packet on a radio will render reception of any packet on a collocated radio useless. Hence it is essential to ensure that simultaneous transmissions and receptions of packets never occur in a wifi-array. However, it is possible to either simultaneously transmit from all collocated radios or simultaneously receive on all collocated radios. We now present a scheduler that achieves this behavior. If a single wifi-array interacts with multiple other wifi-arrays at the same time, it becomes difficult to schedule packets to/from those wifi-arrays (on different channels) such that unnecessary triggering of carrier sense and packet corruption is prevented. Hence, the scheduler allows a wifi-array to interact with only one other wifi-array at any given time.

3.1.2 Adaptive Carrier Sensing

In § 2.3.1, we identify that OOB emissions from one radio can trigger unnecessary carrier sensing at a collocated radio, thus preventing packet transmission on a radio if a collocated radio is already transmitting another packet. It is possible to estimate the effect of OOB power from a collocated radio. This estimate can be used to prevent an unnecessary carrier sensing, if the OOB from a collocated radio is anticipated. The default carrier sensing mechanism, of identifying if the received power is less than a threshold, can be thus replaced with a more intelligent adaptive carrier sense (ACS) mechanism. The new adaptive carrier sense mechanism will remove the estimated effect of a transmission from the received power before determining if the received power is greater than some threshold, to identify a legitimate carrier. If there are multiple collocated radios transmitting at the same time, the aggregate effect of all the radios by summing the estimated powers of each transmission should be used for the adaptive carrier sense.

Received power is measured at a radio using RSSI [5]. For atheros cards, the RSSI is equal to $10log(SNR)$, where SNR is the Signal to Noise ratio, and is usually reported as an integral value in dBm. Thus, it is not possible to determine the accurate power received, given an RSSI reading. Further, if there are two components to some received power value, a higher power component can mask the lower value. For example, if two components of powers are 15dBm and 20dBm, the aggregate of the two is only 20dBm. It is possible that power from a collocated transmission mask the power of a legitimate background carrier and as a result the legitimate background carrier may not be detected by adaptive carrier sensing. Thus, there are two ranges of received power of a legitimate background carrier: 1) a region where a legitimate background carrier can definitely be identified, and 2) a region where the legitimate carrier can be masked by collocated transmissions and hence not be detected.

3.1.3 Coarse Synchronization

It is not always possible to identify a legitimate background carrier on a channel if collocated radios are transmitting some packet. It is possible to get complete information about a channel only if all collocated radios are idle. We propose a coarse synchronization across all radios in a wifi-array, where all radios in a transmit wifi-array start sending packets at the same time after physical carrier sense of their respective channels. Each radio sends one packet at a time and waits for an acknowledgment (ACK) from the corresponding radio at the receive wifi-array. An epoch is a time period during which a wifi-array sends packets on different radios and waits for ACKs. ACKs are sent by the receiver radios only after all packets in the epoch have been transmitted. This prevents ACKs corrupting receptions. If a particular radio of the transmit wifi-array senses its channel to be busy, it will not send any packet during that epoch. If some of the packets are not received during an epoch, they are retransmitted during the next epoch. The retransmission can happen through a different radio than the one in which the packet was sent around the first time. For providing fairness across all nodes occupying the channels, the transmit wifi-array performs a random backoff, similar to the random backoff in 802.11 MAC. There is however, only a single backoff for all radios. This ensures the coarse synchronization across all the radios.

This simple model of synchronization has three issues: a) It is not possible to perfectly synchronize all radios to send packets at the same time. There are several possible sources of delay along a packet path in the network stack. These delays are compensated as explained in § 4.1.2. b) Since there is a single backoff for all channels, it is possible to be unfair across users. If there are multiple users on a particular channel, packets belonging to the different users may collide with each other. An unsuccessful ACK will indicate such a loss of packet. Ideally in such a scenario, the transmitters should backoff for a larger time on that particular channel

[5] The reporting of RSSI is vendor specific. This fact poses a limitation, on our solution, of having to use cards from the same vendor. Further work is needed to combine cards from different vendors

during the next packet transmission. However, since all radios in a wifi-array have a single backoff, it is not possible to have a larger backoff for a particular radio. In this case, compensation is provided by not sending any packet in some epoch. c) A radio does not send any packet, during an epoch, if the corresponding channel is busy at the start of the epoch. However, it is not always possible to know if the channel becomes free before the epoch duration. This is because collocated radio transmission can mask the channel. This might be unfair to the wifi-array as other users in the channel might get access to the channel for a longer time than the wifi-array. However, this unfairness is allowed for the particular radio of the wifi-array.

3.1.4 *Framing*

While using a wifi-array, channel conditions may vary across different channels being used. Depending on the channel conditions, different rates of data transmission may be required for different radios,to ensure successful reception of the packets. However, different rates imply different transmit times for packets with same length. So when only one packet is sent across a radio in a single epoch, a slower radio will prevent faster radios from transmitting new packets. Thus a slow radio in a wifi-array can pull down the aggregate throughput achievable out of the wifi-array. However, if different radios, with different rates, use different packet sizes, such that the transmit time for any packet is the same, such wastage can be avoided. All packets from higher layer are joined to form a single byte stream. Suitable sized frames are created from this stream and given to individual radios. This variable size framing is also used to compensate the delays in packet transmission across radios. Given the link layer focus of Glia, we haven't focused on how different higher layer protocols will behave as a result of variable sized frames. A detailed analysis of how Glia interacts with higher layers will be part of our future work.

Table 2: RSSI and Aggregate Throughput for Different Combinations of Radio-channel Association for a 2 radio wifi-array

Combination #	Receive RSSI	Throughput(Mbps)
1	34, 36	70.1
2	31, 40	65.2
3	41, 38	78.2
4	33, 31	60.1

3.2 Exploit-the-Many

This design element exploits the presence of diversity of radios at source and destination wifi-arrays to maximize the achievable aggregate throughput. In § 2.3.2, we identified that imperfect filtering at the receiver radios leads to packet errors during reception of packets, when adjacent (yet orthogonal) channels are used simultaneously. This error rate depends on location of the radio antenna and even a small difference in location can lead to a huge improvement in aggregate throughput. However, the error rate does not change drastically during short intervals of time. The error rate observed has some correlation with the RSSI observed at a particular receiver radio, when both the channels are simultaneously being used. The higher the RSSI at the receiver, the higher the throughput of the radio. If there are n radios at each of the source and destination wifi-arrays, there are $n! * n!$ combinations to assign n channels to the different radios. It is possible to find a combination that gives the best aggregate throughput. For example, consider a 2 radio wifi-array. Table 2 shows the RSSI readings for the two receiver radios, and their corresponding aggregate throughputs (no carrier sense and no ACK), for the 4 different combinations of radio-channel associations, when both the transmit radios are simultaneously transmitting packets. Combination 3 performs the best, in terms of the aggregate throughput. Combination 3 also has the highest aggregate RSSI. Hence, the sum of RSSI of all receive radios, when all the transmit radios are active, is used as metric to determine the best combination.

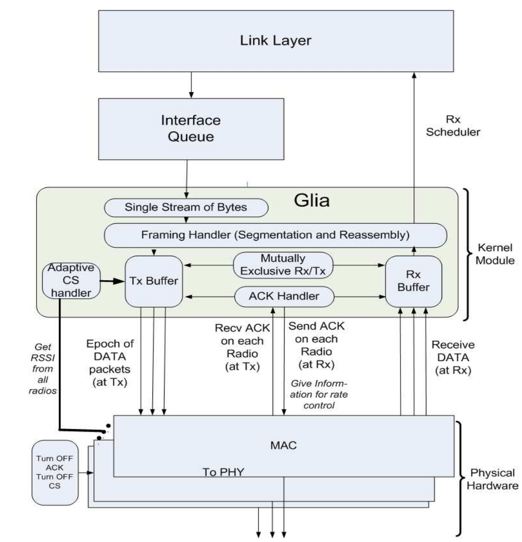


Figure 9: Software Architecture of Glia

4. SOFTWARE ARCHITECTURE

In this section, we present the details of how each principle, identified in § 3 can be implemented in a real system. We develop Glia as a software module that works with any off-the-shelf wi-fi radio, thus requiring no changes to the hardware or firmware of the radios themselves.

4.1 2.5 Layer Operation

We propose Glia as a 2.5 layer solution between the link layer and the medium access layer. Figure 9 shows the software architecture of Glia in the network stack. The correct operation of the solution requires the following from the 802.11 MAC: 1) The default carrier sense mechanism has to be turned OFF. Glia relies on adaptive carrier sense mechanism. Real-time RSSI values from the hardware are needed by the Glia layer. 2) The default 802.11 ACK mechanism has to be turned OFF. As discussed earlier, the default ACK is replaced with a delayed ACK, to compensate for the indeterministic delays in the network stack. We now present the various components of Glia in detail.

4.1.1 *Mutually Exclusive Rx/Tx*

The mutually exclusive Rx/Tx scheduler is required to prevent simultaneous transmissions and receptions. The pseudo code of this component is shown in Figure 10. There are two main functionalities of the scheduler: a) The first functionality prevents transmission of DATA packets on any radio if some of the radios of the wifi-array are receiving packets. b) The second functionality prevents transmission of packets to a wifi-array that is already in conversation with a third wifi-array. To achieve this, all wifi-arrays opportunistically snoop on packets that they hear. These packets need not be destined to a snooping wifi-array. However, the addresses on the snooped packets help the scheduler in determining if the intended destination is busy with some other communication,

INPUT:
$isIdle$ = Variable indicating if all radios are idle
$recvAddr$ = Address of the receiver
$snoopAddr[i]$ = Src and Dst addresses of opportunistic snooped packets, i = 1 to k, k = Total number of addresses
OUTPUT:
$isSend$ = Variable requesting to send packets to recvAddr
ALGORITHM
If ($isIdle$ == 1) {
  if ($recvAddr$!= $snoopAddr[i]$ $\forall$ i = 1 to k)
    $isSend$ = 1;
  else $isSend$ = 0; }

Figure 10: Pseudo Code for Mutually Exclusive Rx/Tx

in which case packets should not be sent to the receiver. If it is not possible to snoop packets of an intended receiver (this is possible if the local node is out of reception range of the transmission but within the carrier sense range of the transmission), the wifi-array will depend on adaptive carrier sense to determine if a particular channel is free. However, if the receiver wifi-array is busy with some other interaction, it will not send any ACKs.

DEFINITION: $epoch$ = a period of time when radios in a wifi-array send out packets.
INPUT: $RSSI[i]$ = Current RSSI of radio i, i = 1 to n
$CSthresh$ = RSSI threshold for default Carrier Sense
$aCSthresh[i]$ = RSSI threshold, for radio i, for ACS, using estimated RSSI of collocated transmissions
OUTPUT: $oPkt[i]$ = Packet of suitable size to send on radio i
$isSendPkt[i]$ = 1 if $oPkt[i]$ should be sent in this epoch, 0 otherwise
VARIABLES: $isFree[i]$ = 1 if channel i is free
ALGORITHM:
for (i = 1 to n)
  if ($RSSI[i] < CSthresh$) $isFree[i]$ = 1
for (i = 1 to n) {
  if ($RSSI[i] < aCSthresh[i]$) {
    create $oPkt[i]$ of suitable size
    send $oPkt[i]$ on radio i } }

Figure 11: Pseudo Code for Coarse Synchronization

4.1.2 Coarse Synchronization

The coarse synchronization mechanism is used to send packets through all radios of a wifi-array, within an epoch. The pseudo code for this component is shown in Figure 11. A single backoff is used for all radios. The traditional carrier sense (CS) mechanism is replaced with the adaptive carrier sense (ACS) mechanism. RSSI values are estimated for all combinations of active collocated radios. These estimated RSSI values are used with the current RSSI to perform the adaptive carrier sense as explained in 3.1.2. Before sending out any packet in a epoch, ACS is performed on all radios to figure out, if their channels are free. All radios with free channels will send out packets in the current epoch. ACKs are sent by the receiving wifi-array on each radio to indicate the successful reception of the packet. The ACKs are sent using basic rate (defined in 802.11 PHY) to improve reliability. The ACKs are sent after the last packet in the epoch is received. Lost packets are retransmitted in the next epoch, possibly through a different radio (than the first time). ACKs are handled by an ACK handler as shown in Figure 9

Since perfect synchronization of all radios is not possible, the delays that occur as a result of various bottlenecks along the network stack have to be compensated. Since Glia is a 2.5 layer solution, only the delays that are caused below the link layer have to be addressed. The delays can be split into two parts:1) a constant deterministic delay (α) and 2) a variable delay (β), that is not fully deterministic. The deterministic delay occurs because of the system bus bottleneck. This delay can be as high as 11μs per packet if a PCI bus is used for the mounting the radios [6]. An X4 PCIexpress bus can reduce this delay to around 2μs. The α delay is compensated by variable packet size. The first packet is sent out with default packet size. Each successive radio is given a packet that is smaller than the previous one, such that the difference in packet size accounts for the deterministic delay. The goal of the compensation is to have the end times of all packets to be coarsely synchronized. This prevents the reverse direction ACK, on some radio, from corrupting DATA reception on a different radio. The β delay occurs because of system inefficiencies. A range of the β delay is precomputed and this delay is compensated by having an ACK timeout of maximum β after the last packet is sent out. Also before handing out the packet to the radio, a second ACS is performed to determine if no new packet has started transmission during the time between the two ACSs. If the second ACS indicates the presence of a some new background carrier on a particular channel, the channel is not used for this epoch. It is possible that the second ACS does not catch a legitimate carrier, because of masking. In such a scenario, collision will occur at the receiver, and an ACK will not be generated.

Individual radios are allowed to have their own rate control algorithm. However, since the default 802.11 ACK is turned OFF, the driver, which performs the rate control, does not have access to the successful packet delivery information. Instead, the ACK handler sends this information to the driver. After every packet transmission on a radio, the corresponding driver is given the information whether the transmission was successful or not. This information will be used by the driver to perform rate control.

INPUT: $rate[i]$ = Datarate for radio i, i = 1 to n
$iPkt[id]$ = Higher layer packets, id = packet number
OUTPUT: $oPkt[i]$ = Glia Packet for radio i
GLIA PACKET FORMAT:
|*Header*|*Segment*|*Header*|*Segment*|...|*Header*|*Segment*|
Segment = segment of bytes of some $iPkt[j]$
Where, *Header* = $(pnum, length, offset, more)$
$pnum$ = j, input packet number
$length$ = length of $iPkt[j]$ bytes used in this segment
$offset$ = Location of the first byte of Segment in the $iPkt[j]$
$more$ = 0 if this segment contains the last byte of $iPkt[j]$, 1 otherwise
VARIABLES USED: $pSize[i]$ = size of Glia packet on radio i
$tTime[i]$ = transmission time for Glia packet on radio i
ALGORITHM:
Convert all $iPkt[id]$ into one single byte stream
find k such that $rate[k]$ is maximum $\forall$ i
$pSize[k]$ = MTU
Choose $pSize[i]$ $\forall$ i != k such that $tTime[i]$ = $tTime[k]$
Take ($pSize[i]$ - Header size) bytes from byte stream, add Headers and create Glia packet

Figure 12: Pseudo Code for Framing

4.1.3 Framing

This component is used to send packets of different sizes through different radios within an epoch. Variable sizes may be required for accommodating multiple rates or for compensating the α delay.

[6]Assuming a packet size of 1500bytes and PCI bus speed of 133MBps

MTU is used for the fastest radio (to accommodate different rates) or the first packet sent out (to compensate the α delay). For accommodating variable rates, packet sizes are determined by ensuring constant packet transmit times. For compensating the α delay, successive packets are given increasingly smaller sizes. Figure 12 shows the pseudo code of the framing component for accommodating different rates. All packets from the higher layer are first combined to form a single byte stream. The packet size is determined for each radio and the corresponding number of bytes are given to the respective radio. The newly formed packets are termed as Glia packets. In order to aid in the re-assembly of the higher layer packets, from the Glia packet, a four tuple header is used for each segment of a unique higher layer packet. The packet format and the descriptions of the four tuple are shown in Figure 12. If a packet has to be retransmitted (because of packet loss), a new packet size may be required. In such a situation, the Glia packet may be further fragmented to make smaller Glia packets, or new segments may be added to make a larger Glia packet.

4.1.4 *Radio-Channel Association*

Radio-channel association involves the exploitation of diversity, possible because of the presence of a potentially large number of combinations ($n! * n!$ for an n radio wifi-array) channel assignments to the source and destination wifi-arrays. As discussed in § 3.2, the RSSI measurements at receive radio can be used to estimate the best possible combination. Even though the search space is very large, a significantly smaller number of experiments are sufficient to make the RSSI measurements. The fact that simultaneous transmission using only adjacent channels affect the achievable throughput on any channel, is used to reduce the number of experiments required to make the RSSI measurements. At the transmit wifi-array, three radios are simultaneously activated (we refer to simultaneous activation as sending DATA packets on all three radios at the same time after turning OFF CS and ACKs) using adjacent channels. The RSSI measurement is made for the middle channel on each of the n receive radios. This single experiment will give n RSSI readings for a particular combination of channel (the middle channel), transmit radio (radio at transmit wifi-array using the middle channel), and the receive radio. Changing the three channels of activation and the transmit radio for the central channel lead to a total of n^2 experiments. From these experiments it is possible to determine all the required RSSI values to compute the metric used to determine the best combination. The metric we use is the sum of RSSI readings, and this simple metric is found to provide a good combination that shows a high achievable aggregate throughput. We use a simple brute force search algorithm. A sophisticated algorithm will be part of our future work. The entire set of experiments can be automated. Once a suitable radio-channel association is selected, it can be used as long as the RSSI values at the receiver do not change significantly. The RSSI values can change if channel conditions have changed, because of mobility or time of operation.

4.2 Case Studies

While Glia is primarily designed for multi-radio wifi-arrays, it allows other background 802.11 traffic to co-exist. Further, Glia also allows wifi-arrays to communicate with legacy 802.11 devices. We consider four case studies, depending on the type of nodes present in the network and explain how Glia works in each scenario.

4.2.1 *Single wifi-array link:*

In this scenario a wifi-array A wants to talk to another wifi-array B. There are no other interfering sources. At node A, Glia gets packets from the higher layer, puts them all in a single byte stream, creates packets of variable sizes for different radios and hands over packets to the corresponding radios. Since there is no other transmission in the vicinity, every radio will sense the channel to be free. Each radio will send the Glia packets during the epoch. At the end of the epoch, The receiver node sends ACKs on each radio, if the corresponding packet was received successfully. If some packets are lost during the epoch, they are re-transmitted during the next epoch. Re-transmission might take place on a new radio.

4.2.2 *Contending wifi-array links:*

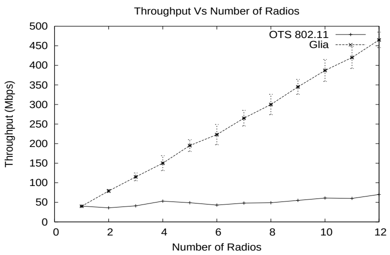
In this scenario several wifi-arrays contend with each other to transmit packets. Since Glia uses a single backoff for all radios in the wifi-array, and because all the radios are virtually glued together, there should ideally be a single virtual channel with multiple contending nodes (as in a single channel 802.11 network). However, since there is only a coarse synchronization across radios, and there is a finite delay between the start of packets on each radio, different wifi-arrays might take over control of different channels during an epoch. This will result in an epoch, where each of the transmit wifi-arrays use a subset of all the available channels. If the destination nodes of each of the transmit wifi-arrays is different, then Glia will essentially result in a situation with multiple links operating at the same time, with each link operating on a subset of the channels. However, consider a scenario where two wifi-arrays A and B want to talk to the same destination wifi-array C. Since the wifi-arrays A and B have different random backoffs they will likely start at a different time instants and hence only one of them takes over all the channels. On the other hand it is also possible that before all radios of the node that starts first start its transmission, the other node might start its own transmission. In this case, there are two possible situations. If the second node opportunistically snoops the packets of the first node, the mutually exclusive Rx/Tx scheduler will not allow the second node to talk to C. However, if opportunistic snooping is not possible, both nodes will go ahead and send packets on different channels. Node C will only ACK packets belonging to the first wifi-array and ignore all packets from the second wifi-array.

4.2.3 *Contending background legacy 802.11 links:*

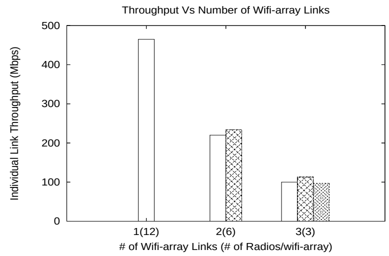
In this scenario, there are background 802.11 transmissions on some of the channels that are being used by a pair of wifi-arrays. Because of the random backoff, the channels with the background traffic will be shared between the corresponding radios of the wifi-arrays and the background 802.11 traffic. As discussed in the § 4.1.2, the radios of the transmit wifi-array will not use a channel if it is already being used by some other traffic. However, it is possible that OOB emissions mask the background carrier, and ACS fails. In such a situation, packets will collide, on the particular channel, at the receiver radios. This will result in a lost packet. The lost packets will be retransmitted at a later time.

4.2.4 *Contending foreground legacy 802.11 links:*

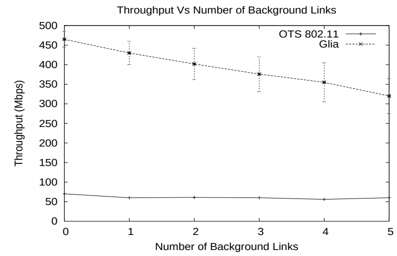
In this scenario, a wifi-array A has to interact with both another wifi-array B and a single-radio node C. The mutually exclusive Rx/Tx scheduler will ensure that only one of links (A with B or A with C) is active at any given time. There are four possible scenarios depending on the direction of communication between the A-B and A-C pairs. If the wifi-array A is transmitting to both B and C, then A will either transmit an epoch of packets to B or transmit a single packet to C. Now consider the scenario when A wants to send packets to B and A has to receive from C. In this case, when C is sending some packet to A, the scheduler will ensure that



(a) Effect of # of radios



(b) Multiple wifi-arrays



(c) Effect of background

Figure 13: Evaluation Results Part 1

no packet is transmitted from A. Similarly when A is transmitting packets, C will simply backoff because it sees a packet in its channel. The third scenario is when B sends to A and A sends to C. When B sends an epoch of packets to A, the scheduler will not let any packet from A to C. Similarly when A is sending a packet to C, B will opportunistically hear the packet and refrain from sending the epoch to A. The fourth scenario is when both B and C try to send packets to the wifi array A. In this case, the wifi-array B might not be able to opportunistically snoop C's packets, if they are out of transmit range of each other. C and B might be able to detect the other with carrier sense (physical or adaptive) else packets will collide on the channels and will be simply re-transmitted.

5. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of Glia on the 12 radio wifi-array testbed. We implement it as a software application, which hooks with the open source madwifi driver. The source-code of madwifi is modified to accept user-input values to any hardware register of the Atheros chipset (for each of the 12 radios), through the iwpriv command. The current RSSI of the chipset is mapped to a /proc file that can be accessed in real-time. The default CS of the chipset is turned OFF using the transmit-stomping feature. Traffic stomping works by telling the card to interrupt any reception of any data packet and shift to transmit mode when there are packets to send. The 802.11 ACK is turned OFF by setting the noACK parameter of the 802.11e QoS specification. All the other elements of Glia are implemented as user space C code. Traffic is generated using UDP datagrams. Unnecessary processes in the Linux OS (Example: X server) are turned OFF to reduce the indeterministic delays. ACS for a radio is performed using current RSSI value and pre-estimated RSSI values for OOB emissions. Although we don't implement the framing mechanism, we study the impact of variable frame sizes for different rates (§ 5.6). Radio-channel association is performed as an offline process by first performing the individual experiments, as described in § 4.1.4. The channel associations are computed offline and fed manually to the individual radios at source and destination wifi-arrays. All experiments are performed in an urban office environment. There are no background users on any of the 5.2GHz channels. However, there are users in the 2.4GHz spectrum. Unless otherwise specified, the results are provided for experiments using the 5.2GHz band. Unless otherwise mentioned, all results are obtained as a result of 10 experimental runs.

5.1 Single wifi-array Link

We first study the effect of number of radios on the throughput capacity of a single wifi-array link in an isolated environment (Figure 14). Each radio operates on a different 'a' channel. Figure 13(a) compares performance Glia with off-the-shelf (OTS) 802.11 operation. The OTS performance suffers for reasons identified in

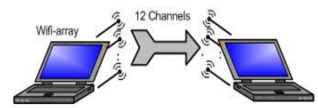


Figure 14: Single wifi-array Link

§ 2. However, Glia shows expected linear behavior of throughput, indicating the fact that all the channels are effectively being used. With all 12 radios, Glia is able to provide a throughput of about 465 Mbps very close to the ideal 480Mbps.

5.2 Multiple Contending wifi-array links

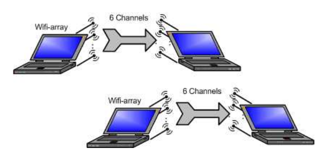


Figure 15: Multiple Wifi-Array Links

Here, we show how Glia operates in the presence of multiple wifi-array links (Figure 15). Due to the lack of enough equipment, we use lesser number of radios for each wifi-array, when experimenting with multiple wifi-array links. We use independent source destination pairs for each link. Figure 13(b) shows the individual link throughputs for different number of links. It can be observed that, in each scenario, all the wifi-array links get similar throughputs. In fact the links share the available bandwidth of all the channels they operate on. The single backoff across all the radios of a wifi-array ensures that the wifi-array acts as a single logical radio. It is however possible that different links use different sets of channels at the same time. However, on the average, each link gets approximately the same throughput.

5.3 Contending background 802.11 links

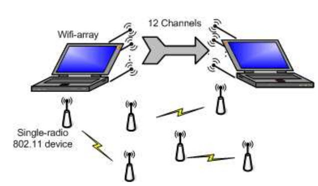


Figure 16: Glia Link with Background Traffic

Next, we study the fairness properties of Glia when there is legitimate background traffic on some of the channels (Figure 16. Multiple background single-radio 802.11 links are added to different channels used by a 12 radio wifi-array link. While Figure 13(c) shows how the number of background links affects the 12 radio throughput, Figure 17 shows the aggregate throughput of the background links. Results of Glia are compared with a OTS 802.11 wifi-array. The throughput of the wifi-array is much higher for Glia, as expected. While a Glia wifi-array tries to share any channel with a background link present on the channel, an OTS 802.11 wifi-array uses very little of any channel. Hence, in the case of OTS 802.11, background links get more time to transmit and as a result experience higher throughput.

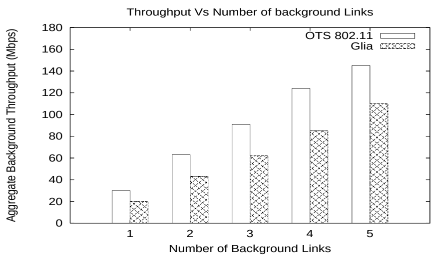


Figure 17: Aggregate of background Traffic

5.4 Contending foreground 802.11 links

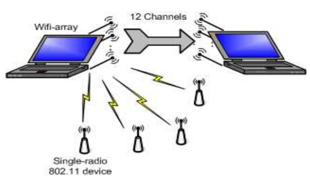


Figure 18: Glia Link with Foreground Traffic

Now, we study how Glia can coexist with other legitimate foreground 802.11 traffic. (Figure 18). Multiple single radio clients, each on a different channel, are added to the setup of a wifi-array link. A single wifi-array acts as the source for all the single radio clients as well as the other wifi-array. Due to lack of space, we do not study other situations of traffic directions. While Figure 19(b) shows how the number of foreground links affects the 12 radio throughput, Figure 19(a) shows the aggregate throughput of the foreground links. It can be seen that the throughput of the wifi-array link falls drastically with addition of new single-radio links. This is because the transmit wifi-array can send traffic to only one other destination at a given time. When the transmit wifi-array is talking to a single radio node, only one of the multiple radios is active.

5.5 Radio-Channel Association

As discussed in § 3.2, the radio-channel association plays an important role in achieving the best throughput out of a wifi-array. Further, the radio-channel association depends on the physical location of the source and destination wifi-arrays. In this experiment, using a single 12 radio wifi-array link, the location of the source wifi-array is fixed and the location of the destination is changed within the transmit range of the source. These different locations are all within the urban office environment. Glia's radio-channel association is compared with a dumb association in Figure 19(c). The dumb association just assigns channels to the radios in a sequential order. The results indicate that the throughput achieved with Glia is always higher than the dumb association. What is interesting to note is that a dumb association can lead to throughput that is only about 60% of the maximum achievable throughput.

5.6 Effect of Different Datarates

In § 3.1.4, a framing algorithm is proposed for using different rates at different radios of a Glia link. Instead of actually implementing the variable packet size algorithm, we study the effect of the variable frames size by manually setting the frame size for different rates. We study the effect of framing in a simpler 2 radio wifi-array setup. Table 3 shows aggregate throughput achieved by the two radios for constant frame size for both radios and variable frame sizes. When using a constant frame size, a slower radio will pull down the aggregate throughput as only one packet is sent on a channel during an epoch. However, with variable frame sizes the transmission time for packets in either channels is the same, thus increasing the aggregate throughput.

Table 3: Aggregate Throughput for Different Datarates (Mbps)

Rates	Const Pkt	Var Pkt
54, 6	10	43.1
54, 48	72	75.1
36, 12	18	37.8
12, 6	12	16

5.7 Glia in 2.4GHz band

Thus far we provided results of Glia operation in the 5.2GHz band. In this section we provide results of Glia operation in 2.4GHz band. The 2.4GHz band is a relatively congested band with lots of users. We show how Glia can aggregate the limited available bandwidth in this band. There are only three orthogonal channels that can be used in the 2.4GHz band (channels 1,6, and 11). Table 4 compares the aggregate throughput achieved by Glia with a default 802.11 implementation on a three radio setup using all the three channels. We run the experiments at two different times when the background traffic conditions are different. Glia can aggregate the available throughput at any given time.

Table 4: Glia in 2.4GHz Band

Scenario	Aggregate Throughput (Mbps)
802.11 (12:00pm)	20
Glia (12:00pm)	55
802.11 (12:00am)	34
Glia (12:00am)	95

5.8 Glia in dual band operation

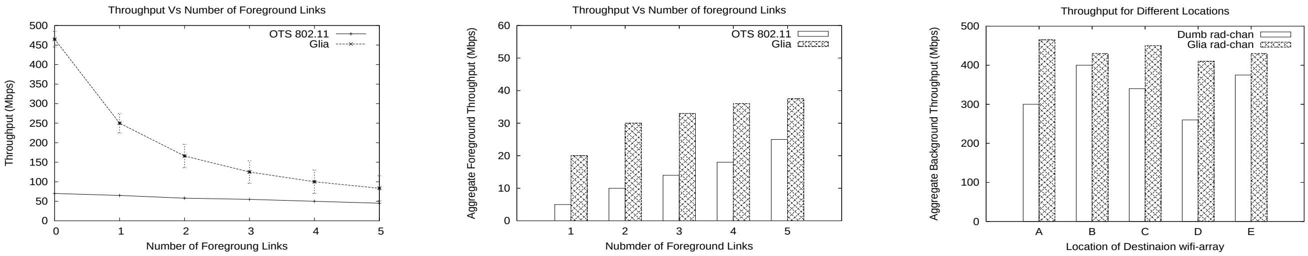
Since Glia uses independent radios for each channel, we can use it in all the 15 available orthogonal channels in the 2.4GHz and 5.2GHz unlicensed bands at the same time. Further, a transmission in the 2.4GHz band will not cause OOB emissions in the 5.2GHz band and vice versa. Hence we can run Glia on a 15 radio node independently, as two sets of Glia links, one in each band. Figure 20 shows the throughput vs number of channels used in such a 15 radio Glia link. These experiments were carried out at 12:00 am in the night when the 2.4GHz band is relatively free. Glia can show an aggregate throughput of 567Mbps with all the 15 radios.

Table 5: Aggregate Throughput for collocated 802.11n radios (in Mbps)

Scenario/Channels	Aggregate
Ideal Two-Radio	192
Default Two-Radio	81
Partial Glia	132

5.9 Glia in 802.11n context

802.11n is latest standard in the 802.11 suite of protocols. It offers higher throughputs among other benefits, by utilizing a variety of technologies like MIMO Multiple Input Multiple Output)



(a) Effect of foreground (b) Aggregate of foreground (c) Effect of Location

Figure 19: Evaluation Results Part 2

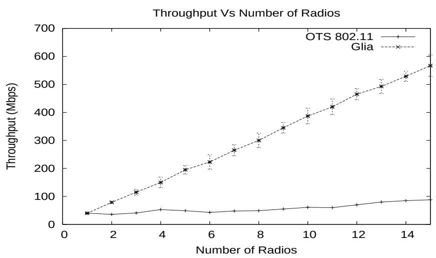


Figure 20: Glia in dual band operation

antennas, spatial multiplexing and a wider bandwidth (40MHz operation). Although, we present Glia in the context of 802.11 a/g, the design elements of Glia are valid even in the context of 802.11n. To study the impact of Glia in an 802.11n setting, we equip one of the wifi-arrays (Linux desktop) with two miniPCI 802.11n radios based on the Atheros 9160 chipset. The chipsets use the open source ath9k driver. Because the ath9k driver is still in a development stage, we were able to use the cards only in the client mode. Hence, we use two other 802.11n access points (a Linksys WRT600n and a Netgear WNR834) for the other ends of the wifi-array, connected via Gigabit Ethernet to two other Linux machines. The topology used is similar to the Tx/Tx topology of Figure 5(b). We use the 2.4GHz spectrum for experimentation. Since the 2.4GHz band has only about 60MHz of available spectrum, one of the two radios of the wifi-array can use a 40MHz channel and the other radio can use the remaining 20MHz channel. Table 5 shows the ideal and observed throughput when the two radios of the wifi-array transmit DATA packets simultaneously. It is observed that even 802.11n performs poorly in a wifi-array. We were able to disable CS of the 802.11n cards but not able to disable the 802.11 ACKs. Hence we were unable to fully implement Glia in the 802.11n context. However, disabling carrier sense, does show benefits in the aggregate throughput (confirming the OOB emission effect). The reason for not achieving the ideal throughput is that reverse direction ACKs collide with legitimate DATA packets. While this straw-man implementation shows the relevance of Glia in an 802.11n, we believe a full blown implementation can provide even higher aggregate throughputs. Ideally it should be possible to achieve about an aggregate throughput of around 1.2Gbps using all the 15 channels in the two bands.

5.10 TCP performance with Glia

Thus far in the paper we have not explicitly considered the use of TCP (Transmission Control Protocol) as the transport layer protocol for traffic that is sent over Glia. However, there are some important implications of using Glia with TCP based traffic. Glia, as we have presented it, does not explicitly provide in-order packet delivery. Since packets are served opportunistically on the different links on an array, it is possible that packets arrive out-of-order at the receiving end depending upon the bandwidths and delays along the different links in an array. However, TCP *interprets* sustained out-of-order packet delivery as a sign of network congestion (it infers a loss on the third DUPACK for a particular sequence number) and will cut down the rate at which a connection is operating. Fortunately, a simple extension to Glia that explicitly does re-sequencing at the receiving end will address this above problem. We defer further investigation of such techniques and an in-depth study of the impact of Glia on other higher layer protocols for future research.

6. RELATED WORK

There have been some works that identify practical issues with using multiple radios on a single node. In [13], the authors study a three node, two-hop testbed, with the common node having two 802.11 radios. They study only the two-hop behavior of the network and conclude that if a single node contains 2 wireless cards alone, these cards will not be able to receive or transmit traffic at the same time, unless their antennas are separated by more than 35db. In [14], the authors identify the interference across two wireless interfaces on the same node, each using a different channel. Similarly, in [15, 16, 17],the authors argue that it is not possible to simultaneously use two radios on the same node. In [11], the authors study the challenges and opportunities for multi-radio coexistence on a single node. Unlike in other works, the authors study coexistence of radios belonging to heterogeneous technologies like 802.11, WiMAX, and Zigbee.

Channel bonding techniques have been known for some time and have been proposed for the new 802.11n standard [1]. However, the standardized Channel bonding in 802.11n is only for 2 adjacent channels. Glia, on the other hand, can bond any number of channels, even if they are non adjacent. Further, new physical hardware conforming to the 802.11n standard is necessary for getting the benefits of such channel bonding. The 802.11n hardware is, however, compatible with existing 802.11 a/b/g devices. The maximum application bandwidths of commercial 802.11n equipment is in the order of 180Mbps [5]. Efforts are on for ratifying a new wifi standard known IEEE 802.11 Very High Throughput (VHT) [18]. Throughput in excess of one gigabit per second, using 100MHz of bandwidth in either the 5.2GHz or 60GHz spectrum, is the goal of this initiative. The new standard would likewise need new hardware. It is not yet clear if the new standard would be backward compatible with existing 802.11 a/b/g devices. Other wideband solutions have been shown to work in principle by works such as [19, 2, 4]. In [19], the authors present a wideband solution in the 5.2GHz spectrum, known as SWIFT, that can coexist with other narrow band devices in the same frequency by weaving together non-contiguous unused frequency bands. The maximum bandwidth shown by SWIFT is close to 500Mbps. All these wideband solutions need new physical hardware and are not compatible with other wifi devices [20, 21]. Advanced antenna technologies, like directional antennas, MIMO, and adaptive antenna arrays have been developed for existing standards. However, these technologies require additional hardware level modifications. While these products are backward compatible with other wifi devices, and conform to existing 802.11 standards, they require new physical hardware to provide higher bandwidths. The maximum per-client bandwidth advertised by such products is 300Mbps. Several wireless networking companies offer multi-radio wifi APs [5, 6]. However, these products bind the radios on different bands (2.4GHz and 5.2GHz). The multiple radios cannot be used to operate in the same frequency band. The maximum advertised throughput using such products is around 300Mbps. Advanced physical layer techniques like [7, 8], can also be used to provide a high bandwidth wireless link. However, these techniques require major changes to existing standards and also need new physical hardware. While these advanced physical layer techniques could be made to be standards compliant, they require new physical hardware to obtain the benefits. Such advanced techniques have only been demonstrated at bandwidths of around 11Mbps (802.11b).

In [22], the authors present 2P, a MAC protocol for long-distance 802.11 mesh networks. The proposed work uses two radios, with directional antennas operating on the same channel, at every node. Although the directional antennas face different directions, it is found that some amount of leakage power from one antenna, causes problems at the other antenna because of side-lobes. The solution proposed in this work is similar in the sense that transmission and reception of data packets at a node are synchronized with each other. There are two important differences between 2P and our work. Firstly 2P works only on one channel while Glia works on multiple orthogonal channels. Secondly 2P is not backwards compatible with other legitimate 802.11 traffic. WildNet [23] builds upon 2P to improve the loss resiliency of long distance mesh networks.

A commercial product called 802.11abg+n is manufactured by Xirrus, Inc [20]. The product is a 16 radio wifi AP with directional antennas. The AP uses 16 radios to divide 360 degrees into 16 sectors, each of which is served by a separate radio. However, the AP cannot use 16 different omni-directional radios as Glia does. More importantly, the notion of providing bandwidth aggregation is not supported on a single link to a single client. Hence, the throughput deliverable to a single client is restricted to that of a single radio

7. CONCLUSION

In this paper, we identify the practical issues of aggregating throughput of multiple orthogonal channels using multiple off-the-shelf radios in a wifi-array. We analyze the reasons for poor performance in such wifi-arrays using a combination of wireless packet trace analysis, and spectrum analysis. We present a practical software only solution, known as Glia, that can achieve close to theoretical aggregation. We evaluate our solution with an implementation on a 12 radio wifi-array testbed. A Mobility analysis will be part of our future work.

8. ACKNOWLEDGMENTS

We would like to thank the shepherd, Sung-Ju Lee, for guiding us through the revision of the paper and for his insightful comments. We would also like to thank the anonymous reviewers, for their valuable feedback. We would also like to thank Sriram Lakshmanan and Chen-Lin Tsao, for the numerous discussions throughout the course of this work.

9. REFERENCES

[1] "Ieee 802.11n standard, http://en.wikipedia.org/wiki/802.11n," Jun. 2009.
[2] M. Weiss, M. Huchard, A. Stohr, B. Charbonnier, S. Fedderwitz, and D. Jager, "60-ghz photonic millimeter-wave link for short- to medium-range wireless transmission up to 12.5 gb/s," *Lightwave Technology, Journal of*, August 2008.
[3] "Gigabit wireless, available online at http://nicta.com.au/research/projects/wireless_gige," Jun. 2009.
[4] "Ultra-wideband, http://en.wikipedia.org/wiki/ultra-wideband," Jun. 2009.
[5] "Linksys dual band ap, http://www.linksysbycisco.com/us/en/products/wrt400n," Jun. 2009.
[6] "Buffalo nfiniti wireless dual band ethernet converter, http://www.buffalotech.com/home/," Jun. 2009.
[7] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: interference cancellation for wireless lans," in *ACM MobiCom '08*, 2008.
[8] K. C.-J. Lin, N. Kushman, and D. Katabi, "Ziptx: Harnessing partial packets in 802.11 networks," in *ACM MobiCom '08*, 2008.
[9] "Routerboard, available online at www.routerboard.com/," Jun. 2009.
[10] "Madwifi, available online at http://madwifi-project.org/wiki," Jun. 2009.
[11] J. Zhu, A. Waltho, X. Yang, and X. Guo, "Multi-radio coexistence: Challenges and opportunities," in *ICCCN*, 2007.
[12] "The effects of adjacent channel rejection and adjacent channel interference on 802.11 wlan performance, available online at http://whitepapers.techrepublic.com.com/abstract.aspx?-docid=70499," Jun. 2009.
[13] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy, "Experimenting with a multi-radio mesh networking testbed," in *Winmee*, 2005.
[14] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A multi-radio unification protocol for IEEE 802.11 wireless networks," in *IEEE International Conference on Broadband Networks (BROADNETS)*, Sep. 2004.
[15] S. Liese, D. Wu, and P. Mohapatra, "Experimental characterization of an 802.11b wireless mesh network," in *IWCMC '06*.
[16] Y. Liu, Y. Xiong, Y. Yang, P. Xu, and Q. Zhang, "An experimental study on multi-channel multi-radio multi-hop wireless networks," in *IEEE Globecom*, 2005.
[17] C.-M. Cheng, P.-H. Hsiao, H. T. Kung, and D. Vlah, "Adjacent channel interference in dual-radio 802.11a nodes and its impact on multi-hop networking," in *IEEE Globecom*, 2006.
[18] "Ieee 802.11 vht study group, available online at http://www.ieee802.org/11/reports/vht_update.htm," Jun. 2009.
[19] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to share: narrowband-friendly wideband networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, 2008.
[20] "Xirrus 16 radio wifi-array, http://www.xirrus.com/products/arrays-80211abg+n.php," Jun. 2009.
[21] "Ruckus wireless, http://www.ruckuswireless.com/," Jun. 2009.
[22] B. Raman and K. Chebrolu, "Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks," in *MobiCom '05*.
[23] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," in *NSDI*, April 2007.