JaldiMAC – Taking the Distance Further

Yahel Ben-David[*] , Matthias Vallentin, Seth Fowler, Eric Brewer[*]
Department of Electrical Engineering and Computer Science
University of California, Berkeley
{yahel,mavam,sfowler,brewer}@cs.berkeley.edu

ABSTRACT

WiFi has been promoted as an affordable technology that can provide broadband Internet connectivity to poor and sparsely populated regions. A growing number of deployments, some of substantial scale, are making use of WiFi to extend connectivity into rural areas. However, the vast majority of the 3.5 billion people living in rural villages [1] are still unserved.

To reach these people, new technology must be developed to make small rural wireless Internet service providers (WISPs) profitable. We have identified radio towers as the largest expense for WISPs; to reduce or eliminate this barrier to entry, we propose a novel point-to-multipoint deployment topology that takes advantage of "natural towers" such as hills and mountains to provide connectivity even over great distances. We make this design practical with a new TDMA MAC protocol called JaldiMAC[1] that *(i)* enables and is optimized for point-to-multipoint deployments, *(ii)* adapts to the asymmetry of Internet traffic, and *(iii)* provides loose quality of service guarantees for latency sensitive traffic without compromising fairness. To our knowledge, JaldiMAC is the first integrated solution that combines all of these elements.

Our evaluation of JaldiMAC suggests that it fulfills its design goals. Our scheduler is able to provide a 71% decrease in jitter and superior latency characteristics in exchange for a 5% increase in average RX/TX switches, as compared to a stride scheduler. Overall, we find that JaldiMAC performs surprisingly well at this early stage.

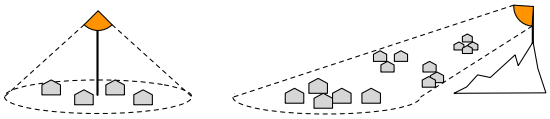
1. INTRODUCTION

We believe that WiFi is the most affordable and appropriate wireless technology for rural connectivity. Indeed, it is in use by a growing number of wireless Internet service providers (WISPs) in rural, developing countries [2, 3]. For example, the AirJaldi [4] network in rural India provides Internet access to 10,000 users in one of the world's most challenging environments, and spans an area of 100 km radius. WiFi offers substantial savings in both deployment and operational costs since it avoids spectrum licensing fees and the hardware is both low cost and available off-the-shelf. Its power consumption is low, making it ideally suited for locations without grid power.

Despite some successes, sparsely populated communities are commercially unattractive to service providers because the infrastructure investment required to service these poor markets is repaid so slowly (if at all). By examining the deployment and operational models used by rural WISPs (as noted by Raman and Chebrolu [5]), we have identified radio towers as the costliest expenditure required to enter rural markets.

In this paper we propose a novel deployment methodology and MAC protocol allowing the use of (potentially distant) mountains as "natural towers", greatly extending the coverage of wireless base stations and reducing the number of towers needed to service rural markets. The result will be dramatic savings, faster return on investment, and lower business risks when servicing communities with very low purchasing power.

Traditional WISP deployments use a design similar to that used in cellular networks: sector antennas with relatively wide beamwidths are placed on a tower in the center of the community (Figure 1a). To limit interference and reach customers close to the tower, these antennas must be angled toward the ground, which greatly reduces their coverage area. We propose to replace the sector antennas with a directional antenna placed on a mountain (Figure 1b). Because the antenna will be located further away from the community and at a much higher elevation, it can cover a much larger area despite its narrow beamwidth; its directionality will also reduce interference, allowing several such setups to coexist.



(a) Sector-based (b) Directional

Figure 1: Conventional cellular-like deployment versus our proposed topology leveraging distant elevations.

[*]Also AirJaldi.Org, Dharamsala, India.
[1] *Jaldi* is the Hindi word for fast.

NSDR '10 San Francisco, California USA

Goal	Solution	Section
Long-distance point-to-multipoint	TDMA with polling and bulk acknowledgements	§3.1
Asymmetric and dynamic traffic patterns	Dynamic TDMA layouts with decentralized reporting	§3.2
Per-station fairness	Assigning slots to stations using min-max fairness	§3.3
Best-effort QoS supporting traffic classes	Allocate transmission time based on requests	§3.4, §3.5, §3.6
Administrative preferential treatment	Weighted min-max	Future work
Adapt to changing physical RF conditions	Calibrate PHY on the fly (bitrate, channel width, etc.)	Future work
Link-layer recovery	Forward error correction (FEC) and retransmissions (ARQ)	Future work

Table 1: JaldiMAC's high-level goals.

2. RELATED WORK

Wireless MAC protocols have been studied extensively; we therefore spent considerable effort searching for existing solutions that would meet the requirements of rural WISPs and the deployment scenario described in §1. A suitable MAC would support deployments with many fixed (rather than mobile) stations communicating with the master over long distance links, with most stations only able to hear the master. It would have to deliver both broadband-class speeds and latency low enough for applications like VoIP. The entire package would also have to be affordable enough for developing regions. Many of these requirements have been addressed in isolation, yet to our knowledge they have not been integrated into a working solution.

Because we don't have space to survey all of the related work in this short paper, we focus on general approaches which have been explored in many studies and explain their shortcomings in the context of our deployment methodology.

2.1 WiFi / 802.11

Although WiFi was originally designed for LAN applications, its proliferation and low cost drove a remarkable level of research and development. In order to overcome the inherent limitations of the 802.11 protocol when used outdoors, many studies suggest replacing the CSMA/CA algorithm with Time Division Multiple Access (TDMA) [6, 7, 8, 9, 10, 11, 12]. This body of work shows that bandwidth is independent of distance when using TDMA [10]. However, this potential has not yet been fully exploited; none of these studies address the varying, asymmetric nature of Internet traffic, resulting in underutilization of the available bandwidth. Moreover, with the exception of Raman and Chebrolu [7] most of this work is limited to point-to-point links, which are infeasible for "last mile" distribution.

The 802.11e standard attempted to address some of 802.11's shortcomings by adding features to support Quality of Service (QoS) and polling; unfortunately, it may starve traffic in certain situations [13, 14], and to the best of our knowledge it has never been fully implemented.

2.2 Mesh Networks

Mesh networking is the foundation for many studies about last-mile distribution [7, 8, 12, 15, 16]. This approach implicitly requires that stations be in range of two or more other stations to allow relaying. This makes the use of directional antennas impractical, a serious flaw in the context of long-distance links where their ability to improve signal strength, reduce interference, and allow greater spatial density is critical. In our view, long-distance mesh networks are currently unrealistic, although recent work on electronically steerable antennas [17] may eliminate this restriction. We do borrow from this body of work the concept of dynamic time slot allocation, which was introduced in JazzyMAC [12], another study targeting long distance links.

2.3 Sensors Networks

Research into sensor networks has yield MAC protocols which are attractive for being light-weight enough to run on low-power embedded devices like those we aim to use. Nevertheless, the topologies are predominantly mesh-like, mostly for short distances and low bandwidth, and are unsuitable for the same reasons.

2.4 Cellular Data Networks

An impressive amount of work has gone into developing new protocols to support the exploding growth in the demand for mobile data. "WiMax" [18] (802.16) is considered state-of-the-art for this family of solutions, but it targets high density markets and mobile applications which have very different requirements from rural WISPs. Furthermore, WiMAX base stations are prohibitively expensive for rural communities.

3. JALDIMAC

In this section, we present the design of JaldiMAC, an adaptive TDMA-based point-to-multipoint MAC protocol with support for differentiated service classes. JaldiMAC is intended for use in combination with the novel deployment method described in §1. Our design also assumes the environment of a typical rural WISP where the stations are fixed, controlled by the service provider, and may each aggregate traffic for several machines. JaldiMAC takes advantage of these assumptions to provide better performance than a generic MAC protocol.

To make our deployment method practical and meet the needs of rural WISPs, JaldiMAC must achieve a variety of sometimes-conflicting goals. Based upon our experience with AirJaldi and that of others, we have distilled these requirements into the list presented in Table 1. We discuss each of these high-level goals in more detail in the subsequent sections.

3.1 TDMA

JaldiMAC uses TDMA because it outperforms the standard CSMA/CA scheme of 802.11 for long-distance wireless. Because TDMA terminology is not standardized, we briefly define our terms here before discussing our design choices. Figure 2 presents a TDMA *schedule* that partitions time into continuous intervals; we distinguish a *window* that represents the horizon of the scheduling algorithm. The schedule consists of *rounds* with an internal structure called a *layout*. Each round begins with a *contention slot* (§3.2) for unscheduled requests like station joins; the size of the round is then the length of time until the next contention slot. We define

a *slot* as the minimal allocatable time unit; a contiguous sequence of slots assigned to the same station is a *chunk*, and the set of slots a station receives in a given round is its *allocation*.

To avoid collisions, TDMA requires some method of synchronization. Maintaining a synchronized clock with sufficient accuracy is challenging [10], so we avoid this approach and rely on *polling*. In JaldiMAC, the master transmits the downstream data for each station separately; the data is prefixed by the length of the receiving station's upstream slot. The station begins transmitting after its downstream slot is over, and continues until exhausting either its queues or its upstream slot. If a station's allocation is not contiguous, this process may occur several times over the course of a round.

The ARQ protocol used in 802.11 requires that each sent packet is acknowledged individually by the receiver; this is extremely inefficient for long distance links with high propagation delay. JaldiMAC instead employs a *bulk acknowledgement* scheme [10] using cumulative ACKs. Since our polling scheme alternates between downstream and upstream slots, we can implement this without unnecessary RX/TX switches by prefixing each slot with the ACK for the previous slot. This works because all stations can hear the master in our topology.

3.2 Dynamic Layout

Traditional TDMA schemes [6, 7, 10] use *static* layouts that only change when stations join or leave the network, and maintain a fixed ratio between upstream and downstream traffic. JaldiMAC uses the available bandwidth more efficiently by taking actual traffic needs into account to create *dynamic* layouts. Each station requests bandwidth in proportion to its anticipated traffic needs, and the master arbitrates these requests to produce a layout that is as "fair" as possible. Because each station may serve many users, requests are performed per session and not per station.

Stations normally make their requests during their upstream slots to minimize the overhead of sustaining long transfers. Stations that had no upstream slots in the previous round must request bandwidth in the contention slot. If there are many such stations, the potential for collision is high; to compensate, JaldiMAC proportionally expands the contention slot up to a maximum size. Collisions are further reduced by requiring each station to randomly choose a time within the contention slot for their requests.

3.3 Fairness

In the context of point-to-multipoint TDMA, *fairness* is the equitable sharing of time between clients. JaldiMAC's use of decentralized reporting (§3.2) makes it natural to define fairness relative to the stations' per-session requests, leading us to use *min-max* fairness. Each session is automatically granted requested time up to its even share of the maximum round size; the excess is then split evenly among sessions that require more.

3.4 Service Classes

Network applications have widely varying requirements in terms of bandwidth, latency, and jitter. For example, file transfers are primarily sensitive to available bandwidth, while VoIP needs very little bandwidth but is highly sensitive to jitter. JaldiMAC addresses these requirements by allowing stations to subdivide their requests into different *service classes*; these classes affect the station's allocation and slot scheduling.

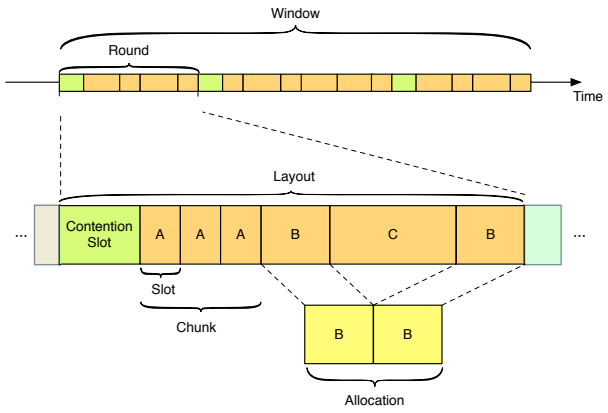


Figure 2: Our TDMA terminology.

Our experience suggests that two service models capture the requirements of the vast majority of types of traffic. *Bulk* traffic (the B class) seeks to maximize throughput with no regard for latency or jitter. *Latency sensitive* traffic (the L class) attempts to minimize latency and jitter at the expense of throughput. Fragmentation of latency sensitive packets can greatly increase latency; to avoid this, the maximum packet size required by the application determines a *minimum chunk size* (S) for its traffic. Similarly, a latency sensitive application often generates packets at a predictable rate. If a chunk is scheduled too early for the packet it must service, the chunk is wasted and latency is increased; however, a late chunk increases latency as well. To capture this, we assign latency sensitive traffic a *period* (P). Thus there are many latency sensitive service classes, depending on our choice of minimum chunk size and period; we can describe a specific class parametrically as $L(S, P)$. JaldiMAC is able to enforce the properties of these service classes effectively but imperfectly, as some combinations of requests are impossible to simultaneously satisfy; we describe JaldiMAC as offering "loose guarantees" for its service classes.

Though it is designed for normal Internet traffic, this model offers a combination of simplicity and flexibility that allows JaldiMAC to handle the needs of specialized applications that have unusual requirements. For example, a client may have an important application that recieves requests over the network and performs some time-consuming processing on them before sending urgent replies later in the round. JaldiMAC does not offer a way to explicitly request a gap between a station's downstream and upstream slots; however, an appropriately chosen L class will accommodate this need by placing an upper bound on how long the application must wait before getting a chance to transmit, regardless of the round size.

3.5 Ply Scheduling

Once fair allocations for the stations are determined, JaldiMAC creates a corresponding layout that maintains the service classes' requirements as closely as possible. To this end, we have developed *ply scheduling*, a new algorithm inspired by stride scheduling [19] but with better latency and

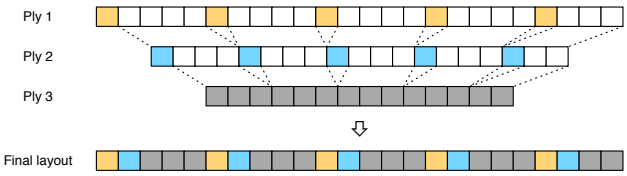


Figure 3: Each ply is made up of the empty slots in the previous ply.

jitter characteristics when minimum chunk sizes vary.[2]

Rather than filling in the layout sequentially, as stride scheduling does, ply scheduling places each service class separately. Each service class is placed into the gaps left by the previous classes; to avoid fragmentation, classes are placed in descending order with respect to the size of the gaps they can fit into. The result is that all L classes are placed before B, and L classes with larger minimum chunk sizes are placed first.

A *ply* is a virtual layout with as many slots as there are unassigned slots in the preceding ply; the first ply corresponds to the entire round. The scheduler starts by assigning slots in the first ply to the first service class being placed, grouped according to that service class's minimum chunk size and spaced according to its period (or simply placed sequentially, in the case of B). The slots that remain unassigned, viewed as a contiguous layout, form the next ply. The next service class is then placed in this ply in the same way, and the process continues until all slots have been assigned (see Figure 3).

The ply scheduler prioritizes minimizing jitter over minimizing latency. Indeed, the effect of considering the gaps in a previous ply as contiguous when laying out the next ply may be to increase latency in some situations. If there are L classes with large minimum chunk sizes in the layout, this effect can be substantial. However, we expect almost all latency sensitive traffic to have a minimum chunk size of 1 or 2 in practice, and in this case we have observed the effect on latency to be positive; we discuss this case further in §4.1.

3.6 Station Mapping

The result of ply scheduling is a layout that assigns slots to service classes. Before this layout is ready for use, the *station mapper* must convert service class assignments into station assignments.

JaldiMAC handles the L classes by assigning chunks in a round-robin fashion to each station with a request in that class. Since the minimum chunk size is constant within an L class, a stride scheduler can be used to carry out this assignment.

B requests can be placed with more freedom; JaldiMAC takes advantage of this to minimize RX/TX overhead. The process has two phases: in the first phase, the station mapper iteratively assigns a B slot to each station in turn until every station's requests have been satisfied. Whenever possible, a station will be assigned a B slot next to one it has already been assigned; if this is impossible, the station will be assigned a B slot in the middle of the largest remaining chunk. In the second phase, each contiguous group of B slots is rearranged to minimize RX/TX switches. This is achieved by grouping the B slots by station and choosing the chunks at the beginning and end of the group so that they are assigned to the same station as any L chunks that bookend the group.

The station mapper finalizes the layout by converting it from units of slots to seconds. This process reveals gaps in the layout caused by the quantization of the station's original requests into slots. These gaps can be used to squeeze in extra transmission time for sessions that did not receive all of their requested time due to fairness; we leave this possibility for future work.

[2]Stride scheduling is very similar to weighted fair queueing in the networking community.

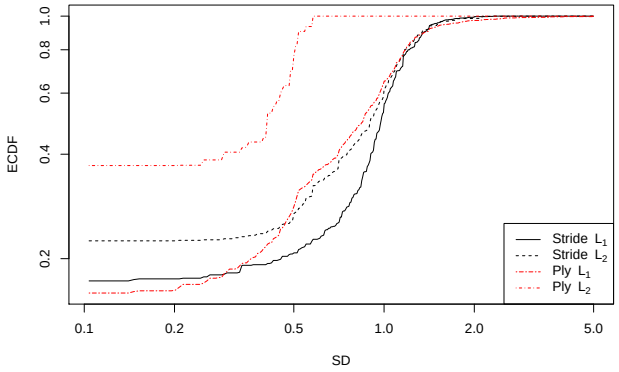


Figure 4: Comparing jitter for stride and ply. The x-axis (log-scale) shows the standard deviation (SD) of the period between two successive chunks for the given service classes. The y-axis (also log-scale) shows the empirical cumulative distribution function (ECDF).

4. EVALUATION

We present here a preliminary evaluation of JaldiMAC. Because we have not yet completed an implementation of JaldiMAC, we focus on comparing our ply scheduler to a standard stride scheduler. The same fairness and station mapping algorithms are used for both schedulers, and they are evaluated in the same custom simulator. We examine the effect of the schedulers themselves on jitter in §4.1 before looking at jitter and latency in the context of the end-to-end system in §4.2. Finally, we assess the context switch overhead induced by each scheduler in §4.3.

4.1 Scheduler-level Jitter

To compare ply and stride scheduling in isolation from the rest of the system, we fix a round size of 50 slots[3] and generate all possible distinct combinations of periods for a minimum chunk size of 1, resulting in a set of latency sensitive service classes L_1, and a minimum chunk size of 2, yielding L_2. Each choice of L_1 and L_2 is augmented by an allocation for the B class chosen to fill out the round. As shown in Figure 4, ply produces remarkably lower jitter values compared to stride for service class L_2 and still performs better for L_1.

It's unsurprising that the ply scheduler performs better for L_2, which is placed first and has an essentially optimal lay-

[3]It is worth noting that a realistic minimal time slot size for 802.11g hardware is 5ms [12], which would give this round a durations of 250ms. In practice, 250ms is our desired upper limit, as we would like to remain below the resulting 500ms round-trip latency.

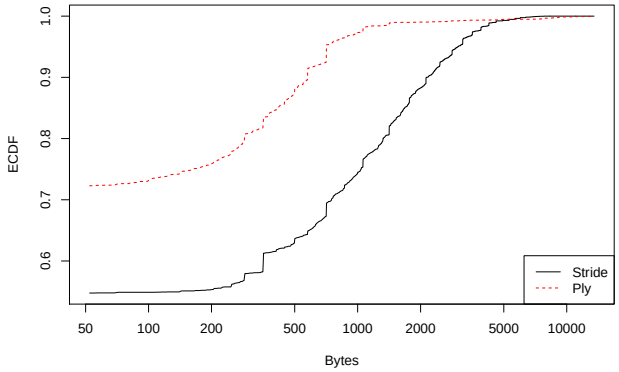


Figure 5: End-to-end jitter, analyzed in terms of the standard deviation of inter-chunk periods for each latency-sensitive request.

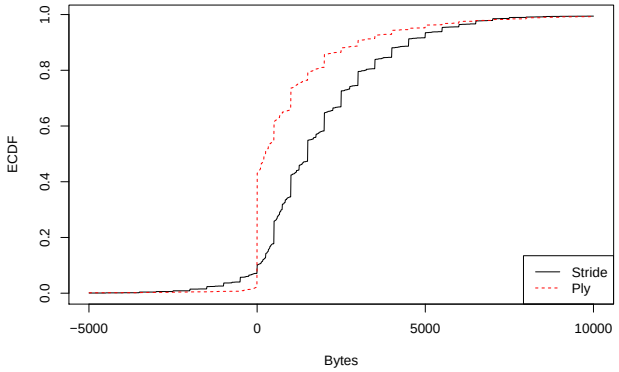


Figure 6: Difference between average period and requested period. A value closer to 0 means the scheduler accurately matched the station's request.

out. In the case of L_1, ply performs better because it uses the already-placed L_2 class to "resynchronize". When ply finds that it cannot assign a slot to L_1 because it is already occupied, it moves on to the next unoccupied slot, and tries to maintain the correct period from that point on; frequently no further correction is needed. This means that ply is often able to arrange different L classes so that they coexist harmoniously. In stride scheduling, by contrast, placement problems accumulate instead of disappearing; the linear, local nature of the algorithm provides no way to resynchronize.

4.2 End-to-end Jitter and Latency

We now consider the performance of ply and stride scheduling in the context of the end-to-end system.

In our first experiment, we generate 10,000 random sets of requests. Each set contains random requests for five stations. To simplify the analysis, the requests for each station are constrained to contain no more than one L class; an arbitrary number of B requests are allowed. (This still means JaldiMAC must contend with up to five different L classes for each set of requests.) Figure 5 shows that both stride and ply perform quite well, with a median jitter of 0; however, the average jitter is 248% higher for stride than for ply.

Our second experiment compares the difference between the requested period and the average allocated period for each request; the results are shown in Figure 6. Periods higher than requested mean that the scheduling algorithm has introduced unnecessary latency; ply's better ability to deliver the requested period will therefore be observed as lower latency by applications. Additionally, stride scheduling sometimes delivers a period that is too short; this is shown in the figure by the portion of its ECDF that has a negative value. In this situation, the latency sensitive application may not yet have delivered the next packet, so the slot may go to waste. Ply always avoids this problem, which is highly desirable.

4.3 RX/TX Context Switches

It is desirable to minimize transitions between sending and receiving on long distance wireless links, since each switch takes time and reduces bandwidth. At the same time, JaldiMAC's loose latency and jitter guarantees inherently require more RX/TX switches.

To examine this tradeoff, we generate 10,000 random sets of requests, as in §4.2, and look at the number of RX/TX switches as a function of round size. We find that stride and ply do not differ significantly in the number of context switches (mean difference 0.74). Further, the number of context switches only increases slowly with round size, which makes ply well-suited for practical deployment in networks with a broad range of traffic levels.

5. FUTURE WORK

Our innovative method of deployment greatly reduces loss caused by interference, and rural long distance links typically experience little loss [20] in any case. We therefore do not expect significant loss rates in practice. If a problem does arise, we will consider various link-layer recovery schemes such as ARQ and FEC. Additionally, there are a wide range of ways that we can tune physical channel parameters: varying channel width, adapting the bitrate, adjusting transmit power, and switching frequencies. Optimizing these parameters dynamically is a challenging research problem.

WISP networks exhibit tree-like structures, with stations as leaves and intermediate nodes acting as aggregating relays that forward traffic to the root. The ability to maintain loose service guarantees across a multilevel hierarchy of links will be important for JaldiMAC in real-world deployments. In this paper we describe JaldiMAC as applied to a 2-level tree, but we plan to generalize the algorithm to operate over multiple hops in the same vein as Hohlt et al. [21].

Our target hardware platform for JaldiMAC is 802.11n, as it offers an order of magnitude more bandwidth then 802.11a/g and flexible control of important PHY parameters while maintaining the same cost levels. 802.11n's potential for use in long distance links has received little study, but our initial experiments suggest that it works well.

We plan to deploy two testbeds in parallel: one by AirJaldi in rural India, and one in the Bay Area of California, where we already have several 802.11n long distance links in place. By comparing the two testbeds, we expect to gain insight about the difference in interference patterns between urban and rural environments. As soon as the prototype reaches maturity, AirJaldi will begin deployment throughout rural India.

6. CONCLUSION

Although WiFi is a cost effective and promising technology for extending broadband Internet connectivity to poor and sparsely populated communities, its proliferation remains slow. Our first contribution is a novel deployment method designed to address the economic challenges faced by small rural WISPs. We identify towers as the biggest barrier to entry and suggest placing base stations on mountains or hills instead, even if the subscribers are far away. Our methodology is distinctive in its use of directional antennas to minimize interference while maximizing the service area.

Rather than retrofitting existing MAC protocols for this environment, we present JaldiMAC, a new MAC protocol that achieves the following goals: *(i)* enable point-to-multipoint setups while taking advantage of the broadcast capabilities of the topology, *(ii)* adapt to the asymmetric, varying nature of Internet traffic, and *(iii)* provide loose quality of service guarantees for latency sensitive traffic without compromising fairness.

Our evaluation suggests that JaldiMAC's ply scheduler, when compared to a traditional stride scheduler, is able to provide a 71% decrease in jitter and superior latency characteristics in exchange for a 5% increase in average RX/TX switches.

In the near future, we plan to complete the design of JaldiMAC and begin experimenting with it on our testbeds in the Bay Area and rural India. In the long term, we aspire to have WISPs like AirJaldi take advantage of our deployment method, MAC design, and the low-cost 802.11n hardware platform to bring affordable Internet access to deserving rural communities all over the world.

7. ACKNOWLEDGMENTS

We would like to thank Rabin Patra, Sergiu Nedevschi, Anmol Sheth, Matt Podolsky, and Matt Tierney for the valuable feedback and discussions. Also, we thank the anonymous reviewers for the enlightening comments.

8. REFERENCES

[1] "The United Nations Population Database - World Urbanization Prospects: The 2007 Revision." http://esa.un.org/unup/.

[2] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, and K. Fall, "The case for technology in developing regions," *IEEE Computer*, vol. 38, no. 6, pp. 25–38, 2005.

[3] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer, "Beyond pilots: keeping rural wireless networks alive," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, (Berkeley, CA, USA), pp. 119–132, USENIX Association, 2008.

[4] "AirJaldi.Org - Wireless Network, Dharamsala, India." http://www.airjaldi.org.

[5] B. Raman and K. Chebrolu, "Experiences in using WiFi for Rural Internet in India," *IEEE Communications Magazine*, January 2007.

[6] K. Paul, A. Varghese, S. Iyer, and B. R. A. Kumar, "WiFiRe: Rural Area Broadband Access Using the WiFi PHY and a Multisector TDD MAC," *New Directions in Networking Technologies in Emerging Economics, IEEE Communications Magazine*, 2006.

[7] B. Raman and K. Chebrolu, "Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks," in *ACM MOBICOM*, August 2005.

[8] Ashutosh, Nirav, and Bhaskaran, "Implementation and evaluation of a tdma mac for wifi-based rural mesh networks," p. 6, 2009.

[9] A. Sharma, M. Tiwari, H. Zheng, and S. U. C. Barbara, "Madmac: Building a reconfigurable radio testbed using commodity 802.11 hardware," in *First IEEE Workshop on Networking Technologies for Software Defined Radio (SDR) Networks*, September 2006.

[10] 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*, 2007.

[11] A. Rao and I. Stoica, "An overlay mac layer for 802.11 networks," in *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, (New York, NY, USA), pp. 135–148, ACM Press, 2005.

[12] S. Nedevschi, R. Patra, S. Surana, S. Ratnasamy, L. Subramanian, and E. Brewer, "An adaptive, high performance mac for long-distance multihop wireless networks," in *ACM MOBICOM*, 2008.

[13] R. Lenagala and Q.-A. Zeng, "Study of dynamic mac layer parameters for starvation prevention in the ieee 802.11e mac layer protocol," in *Wireless and Mobile Computing, Networking and Communications, 2006. (WiMob'2006). IEEE International Conference on*, pp. 124–131, June 2006.

[14] S. Park and D. Sy, "Dynamic control slot scheduling algorithms for tdma based mobile ad hoc networks," in *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–7, Nov. 2008.

[15] "The AirJaldi Mesh Router, AirJaldi.Org, Dharamsala, India." http://drupal.airjaldi.com/node/9.

[16] "Freifunk Community Wireless Mesh Network, Berlin, Germany." http://start.freifunk.net/.

[17] "Tarana Wireless, Berkeley, CA." `http://www.taranawireless.com/tech`.

[18] S. A. Ahson and M. Ilyas, *WiMAX Handbook - 3 Volume Set*. Boca Raton, FL, USA: CRC Press, Inc., 2007.

[19] C. A. Waldspurger and W. E. Weihl, "Stride Scheduling: Deterministic Proportional-Share Resource Management," tech. rep., 1995.

[20] A. Sheth, S. Nedevschi, R. Patra, S. Surana, L. Subramanian, and E. Brewer, "Packet Loss Characterization in WiFi-based Long Distance Networks," in *IEEE INFOCOM*, 2007.

[21] B. A. Hohlt, *The Design and Evaluation of Network Power Scheduling for Sensor Networks*. PhD thesis, EECS Department, University of California, Berkeley, May 2005.