A Framework for Frameless TDMA Using Slot Chains

Gentian Jakllari*, Mike Neufeld†, Ram Ramanathan‡
*IRIT-ENSEEIHT, University of Toulouse, France, †Akamai Technologies, ‡Raytheon BBN Technologies, USA
jakllari@enseeiht.fr, mneufeld@akamai.com, ramanath@bbn.com

***Abstract*—TDMA MAC protocols suffer from two drawbacks that have reduced its practical appeal, especially for infrastructure-less wireless networks: the need for *synchronization*, and the use of *frames*. While there has been extensive research towards achieving the required synchronization, the frames are assumed to be inherent in TDMA.**

In this paper we present a framework for frameless TDMA based on the new concept of a slot chain. A *slot chain* is a series of slots with a starting slot and inter-slot period that is chosen to match a given capacity request. We present an algorithm for interleaving slot chains in a conflict-free manner. The algorithm is optimal for a class of requests called *geometric* and is z-approximate for general requests, where z can be made arbitrarily small.

We show that any TDMA MAC protocol can readily replace using frames with using slot chains. Our simulation analysis shows that using slot chains results in much better capacity utilization when traffic is heterogeneous – as much as 100% in some cases.

I. Introduction

Real-time multimedia applications (packetized voice, video, gaming etc.) are an increasingly dominant presence in wireless networks. Unlike in wired networks where bandwidth is plentiful and predictable, media access for real-time multimedia in bandwidth-scarce wireless networks, especially in infrastructure-less networks, must be reservation-oriented, that is, provide capacity guarantees and admission control. The prevalent 802.11 DCF protocol is contention-based and inherently unsuitable for this purpose. While the community has invented several ingenious variations [1], [19], [15] to make it real-time friendly, it is clear that these are not long term solutions.

Time Division Multiple Access (TDMA) is contention free and, in principle, allows reservation guarantees in a natural manner. It is also known to offer better performance at higher loads. However, TDMA currently suffers from two drawbacks that have reduced its practical appeal, especially for infrastructure-less networks with a heterogeneous traffic mix: the need for *synchronization* and the use of *frames*. While the problem of achieving the required synchronization has been subject of numerous studies [4], [17], the frames have so far been considered inherent in any TDMA MAC protocol. Frames in TDMA divide time into groups of repeatable time slots and allocations of slots to users are made for a single frame. That same frame is repeated until a new allocation (new frame definition) is created. A node (or flow) that is allocated s slots in a frame of length F slots gets s/F fraction of the link capacity. A frame is useful in describing slot allocations succintly - otherwise one would have to describe every slot into the future individually and explicitly. However, frames define a rigid allocation granularity that results in poor support for heterogenous traffic. Specifically, the granularity of allocation in a F-slot frame is 1/F and capacity allocations are discrete-valued at 1/F, 2/F, 3/F etc. A request for (2 + ϵ)/F capacity ($\epsilon << 1$), for instance, will waste nearly 1/F in capacity.

We present a re-thinking of TDMA which preserves the core essence of "time divided" access but does away with traditional framing. Instead of frames, we propose a new concept called *slot chains*. A (periodic) slot chain is described by $\mathsf{C} = (s, p)$ where s is a starting slot number and p is the number of slots between two consecutive slots in the chain. Note that there are no boundaries whatsoever – the period between the slots in the chain, which could be arbitrarily small or large, determines the capacity of the allocation. As an example, a request for 0.1 of the bandwidth is allocated a slot chain one in 10 slots (e.g. slots 5, 15, 25 and so on, that is, $\mathsf{C} = (5, 10)$).

Using slot chains instead of frames is not trivial. The slot chains need to be conflict free, and interleaving slot chains with arbitrary periods is an unsolved problem. We present a polynomial-time algorithm that solves this problem and prove its correctness. The algorithm is optimal for a class of reservations requests called *geometric* requests and can be made arbitrary close to optimal for general requests.

We show how the slot chains can readily be adopted by any current or future TDMA protocol into a frameless TDMA protocol. As a test case we modify a recently published TDMA protocol to use slot chains instead of frames. ns-2 simulations show that doing so leads to much better capacity utilization when traffic is heterogeneous – as much as 100% in some cases.

This project was supported under AFRL contract FA8750-08-C-0148. Approved for Public Release; distribution unlimited: 88ABW-2010-2644, 27JUL10

In summary, our main contributions are as follows:

1) A new approach and algorithm for time divided access using slot chains that is free of the framing constraint and accommodates diverse flows far more flexibly.
2) Algorithms and optimality proofs for interleaving slot chains in a conflict free manner.
3) A framework for turning any TDMA MAC protocol into a frameless TDMA protocol.

The rest of the paper is organized as follows. In section II we introduce the concept of slot chains and state the problem. In Section III we present an optimal algorithm for geometric requests and a z-approximate algorithm for general requests. In Section IV we present a framework for frameless TDMA MAC protocols while the simulations are discussed in sections V. In section VI we survey the background for our work and finally in Section VII we provided some concluding remarks.

II. Slot Chains

The concept of *frames* is deeply ingrained in current TDMA approaches. Frames divide time into groups of repeatable slots. In particular, allocations of slots to users are made for a single frame and that same frame is repeated until a new allocation (new frame definition) is created. A node (or flow) that is allocated s slots in a frame of length F slots gets s/F fraction of the link capacity.

A frame is useful in describing slot allocations succinctly - otherwise one would have to describe every slot into the future individually and explicitly. However, it has a number of disadvantages, especially when the flow rates are diverse:

1) Frames are restrictive in terms of the minimum rate (slots per unit time) that can be allocated. Given a frame length of F slots, the supportable rate is lower-bounded by 1/F.
2) The granularity of allocation in a F-slot frame is 1/F. That is, capacity allocations are discrete-valued at 1/F, 2/F, 3/F etc. A request for $(2 + \epsilon)$/F capacity ($\epsilon << 1$), for instance, will waste nearly 1/F in capacity.
3) Nodes participating in the TDMA will have to keep track of not only slot boundaries but also frame boundaries. If due to disruptions or attacks, the frame boundaries are misaligned, the whole protocol breaks down.

Note that since the flow rate mix may not be known in advance, it is cumbersome to size the frame to the lowest rate flow and allocate multiple slots per frame.

It might be argued that the frame be sized to accommodate the lowest rate flow and for higher flows, multiple slots scheduled per frame. However, there are several issues with this. First, the lowest rate flow may not be known in advance. Second, if slots are allocated back-to-back, as is typical, then real-time isochronous flows will not be able to utilize the additional slots without a lot of buffering. On the other hand, spreading out the slots within the frame is likely to result in uneven inter-slot distances across frames.

In summary, frames are not conducive for supporting highly heterogenous traffic mix efficiently. Future networks will have high traffic heterogeneity resulting from diverse applications such as voice, video, HD video, gaming, real-time imagery etc.

We investigate a radically new approach to TDMA allocations. While maintaining the basic slotting, we eliminate frames altogether. Instead, allocations are specified as *slot chains*. In general, a slot chain is any sequence of slot numbers. In this paper, though, we only consider *periodic* slot chains denoted as $\mathsf{C} = (s, p)$, where s is the starting number of the chain and p is the number of slots between every two slots, referred to as the *period* of the slot chain.

Figure 1 shows a simple example comparing the allocation in response to a request using frames and slot chains. For the requests shown, and sizing frames to the dominant flow (B), the frame length is 10 slots. A, B and C need 0.5, 1 and 2.1 slots per frame respectively, but get 1, 1 and 3 due to the rigidity of frames.

Slot chains, on the other hand, deal with this in a natural and maximally efficient manner. Each chain consumes exactly the capacity needed – in the case of C, fractional capacity is handled by multiplexing over two slot chains, one of which gets only one every 100 slots. Even in this simple example, slot chains consume 14% less capacity than frames. With more heterogeneity and more flows, the gap could obviously widen much more.

Slot chains remove many of the constraints of frames. However, an obvious problem is to allocate slot chains so that they don't conflict with each other. Note that it is not always clear whether two slot chains overlap at some slot in future. For example, in Figure 1, if A needed 125 kbps instead (that is, one in 8 slots), then starting at the same point would result in a collision in slot 33 with the first of C's flows. In the rest of this section, we state two related problems, and in the next section present an algorithm.

An *allocation request* R is a request for an allocation of capacity by a *user*, typically a traffic *flow*. The *allocation period* of a (slot chain corresponding to) a request R is W/R where W is the total node capacity (data rate). Thus, all allocation requests can be thought of as a request for a particular allocation period. Two slot chains are said to be in *conflict* if they have one or more slots in common. We use the term *interleaved* slot chains synonymously with conflict-freedom.

We consider a single node. Allocation requests R_i

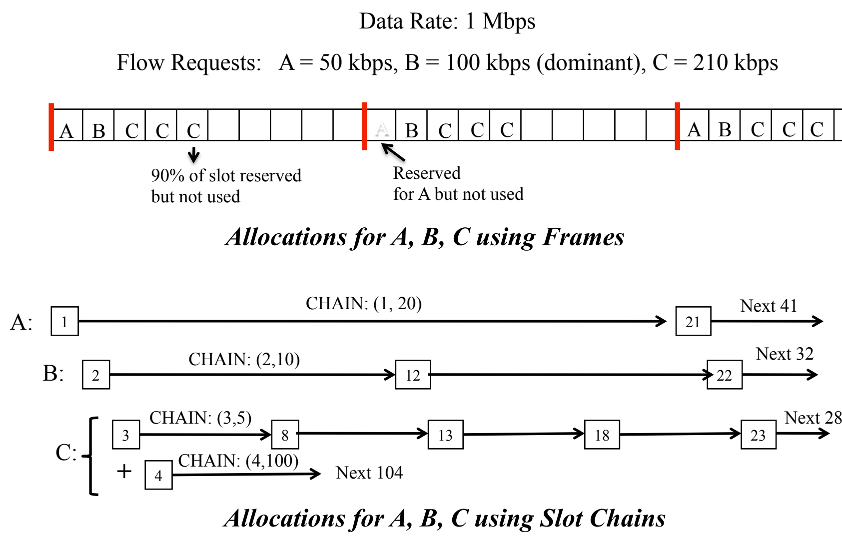

Fig. 1. Comparing example allocation requests via frames and slot chains. See text for explanation.

arrive sequentially from one or more users. Allocation requests are *not* known in advance. Each allocation request needs to be assigned slot chain(s) that is conflict-free with existing slot chains, unless and until there is no residual capacity. We term this the *Interleaved Slot Chains (ISC)* problem, defined as follows.

Problem II.1 (ISC)**.** *Given a set of existing slot chains* $\Psi = \{\mathsf{C}_1, \mathsf{C}_2, ...\}$ *and a new allocation request* R*, generate a set* Ψ' *of one or more slot chains such that* $\Psi \cup \Psi'$ *is conflict free, and* $R = W \cdot \sum_i (1/p_i)$*, where* p_i *is the allocation period in* $\mathsf{C}_i \in \Psi'$ *, and* W *is the node capacity (data rate).*

Thus, we need to generate one or more periodic slot chains[1] such that the allocations of the individual chains add up to the request R.

In our solution to the ISC problem, we shall first solve an "intermediate problem" which imposes the following constraint on the input: every request p_i is constrained to be of the form $B \cdot 2^n$ for some chosen B (termed *base*, where n is a non-negative integer. For example, if Base is 5, then the periods have to be one of 5, 10, 20, 40, 80, 160, 320.... We term this the *Geometric ISC* problem, defined below.

Problem II.2 (Geometric ISC)**.** *Given a positive integer* B*, a set of existing slot chains* $\Psi = \{\mathsf{C}_1, \mathsf{C}_2, ...\}$*, a new "geometric" request* R *such that* $p = W/R = B.2^n$ *for some non-negative* n*, determine the starting slot number* s *of a new periodic slot chain such that* (s, p) *is conflict free with* Ψ *.*

[1]Generalizing to allow multiple slot chains allows more diverse solutions. Since a flow can be multiplexed into multiple chains, this does not hurt in practice.

III. An algorithm for Interleaving Slot Chains

Before we describe our algorithm for IPSC, we show with a simple counter example that a greedy solution, namely one that assigns the first available starting point is not optimal.

Consider a base of 5, and the sequence of requests (20, 20, 20, 20, 20, 5). A greedy algorithm would produce the following slot chains:

First Request, period 20: 0, 20, 40, 60...
Next Request, period 20: 1, 21, 41, 61...
Next Request, period 20: 2, 22, 42, 62...
Next Request, period 20: 3, 23, 43, 63...
Next Request, period 20: 4, 24, 44, 64...
Next Request, period 5 : Can't be allocated (collision at 20)

However, the total requested capacity is only 1/5 + 5/20 = 0.45 of the available capacity. Thus we are denying the request when there is a lot of capacity. Thus a greedy solution is not optimal, and clearly suboptimal enough to not even be acceptable in practice.

Thus we require a different solution. Our solution, given in the remainder of this section, has two parts: 1) An optimal solution to the Geometric IPSC problem, which is a contribution in itself; 2) An simple algorithm for the (Generalized) ISC problem that decomposes the requested allocation into a set of Geometric allocations, and uses the solution from (1) for each.

A. Geometric ISC

For a given base B, we first partition the allocation space into B *primitive chains* S_0, S_1, .. S_{B-1}, starting at

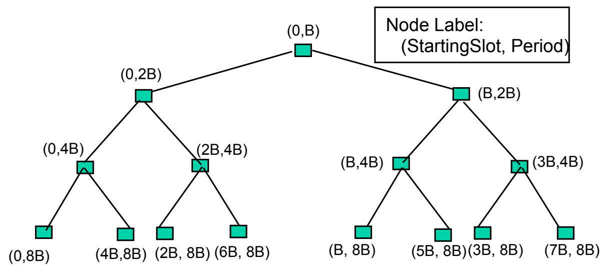

Fig. 2. Slot Allocation representation for the Geometric ISC. Each node labeled (x,y) represents capacity used by a chain with period x starting at y. One can (recursively) allocate a node OR both its children, but not both.

positions 0, 1, 2, ... $B-1$ respectively, each with period B. A primitive chain (*p-chain*) starting at $i \leq B-1$ represents the aggregate of all chains $(i, *)$, and holds a capacity of $1/B$. All allocations on p-chain S_i have to have allocations starting at $S_i + n \cdot B$. Thus, on p-chain 0, you can have sequences that go 0, 10, 20, 30... or 5, 15, 25, 35 ... or 40, 80, 120, 160... but not 1, 11, 21..... this sequence "belongs" to p-chain S_1.

The key idea behind the algorithm is as follows. Each p-chain can be thought of as a container for either one chain of period B (the p-chain itself), or two interleaved non-conflicting chains of period $2 \cdot B$ each of which takes alternate slots from the primitive chain, or four chains of period $4 \cdot B$ each, and so on. In other words, we can recursively partition a chain (j, P), holding capacity of 1/P into two chains (j, 2P) and (j+P, 2P), each holding a capacity of 1/2P.

The above can be represented quite naturally as a binary tree with nodes representing starting point, period combinations, as shown in Figure 2. For any node in the tree, you can either allocate the corresponding chain (shown alongside in the figure), or chains on both of its children.

With this background concept, the actual algorithm is easy to explain. The idea is to allocate the first available match to the request (recall that for Geometric ISC, the requests are discrete-valued at $B \cdot 2^i$ and hence will have to have a match in the tree, if available). We allocate using depth first search, allocating the first node (representing a chain) whose period matches the input period. We than mark the entire sub-tree rooted at that node as *closed* since allocating a slot chain means that that sub-chains aren't available anymore. The depth-first search does not go down *closed* sub-trees. Further, all of that node's ancestors are marked *occupied*, meaning that one can go down such ancestors (for the other sub-tree), but the ancestor itself cannot be allocated (since one of its sub-chains is allocated). Intuitively, we pack the space tightly (with the Geometric semantics) as we move right, and hence we only fail when there is no capacity. This is more formally proven in section III-B.

We discussed above the allocation within one p-chain, namely S_0. For a base B, we have B orthogonal p-chains, each with an aggregate capacity of $1/B$. Thus we have B trees, each rooted at and representing starting points 0, 1, 2.. B-1. An allocation on p-chain S_i is conflict free with an allocation on S_j. Although strictly not necessary, our algorithm only picks tree S_i if all trees S_0 through $S_{(i-1)}$ are full.

We now show, using a series of results, that Algorithm Geometric ISC produces conflict free slot chains. Although not strictly necessary, the proofs are easier to understand when developed in the context of an allocation tree as in Figure 2. We first state the concept formally.

Definition III.1. *A* Geometric Allocation Tree *with base B is a binary tree where each node v is associated with a slot chain $(s(v), p(v))$, and the following holds: $s(lc(u)) = s(u)$, $p(lc(u)) = 2 \cdot p(u)$, $s(rc(u)) = s(u) + p(u)$, and $p(rc(u)) = 2 \cdot p(u)$, where $lc(u)$ and $rc(u)$ denote the left and right child respectively. Further, if w is the root of the tree, then $s(w) = 0$, $p(w) = B$*

B. Correctness

We first show that two allocations within the same p-chain are conflict free.

Lemma III.1. *The children of any node in a Geometric Allocation Tree are conflict free*

Proof: Consider any node w and its children $u = lc(w)$ and $v = rc(w)$. By definition III.1, the slot chains of u and v, are, respectively, $(s(w), 2 \cdot p(w))$, and $(s(w) + p(w), 2 \cdot p(w))$. Letting $s = s(w)$ and $p = 2 \cdot p(w)$, the slot chains of u and v can be re-written as (s, p) and $(s + \frac{p}{2}, p)$ respectively. Suppose they are in conflict. Then there must exist positive integers n_1 and n_2 such that $s + n_1 \cdot p = s + \frac{p}{2} + n_2 \cdot p$, which implies $n_1 = (n_2 + \frac{1}{2})$.

However, this can never hold for positive integers n_1 and n_2 and thus the children of any node are conflict free. ■

Lemma III.2. *Suppose a node u is conflict free with a node v. Then, every descendant of u is conflict-free with every descendant of v.*

Proof: By definition III.1 the slot chain of both left and right children of a node v is a subset of the slot chain of the node. Thus, the descendants do not occupy any slot other than the those occupied by v. It follows that if u is conflict-free with v, it is also conflict-free with its descendants. Not that per terminology a node is a descendant of itself. ■

Lemma III.3. *A node u is in conflict with a node v if and only if v is either an ancestor or a descendant of u.*

Proof: The "if" part follows from a recursive application of definition III.1 since a child occupies the alternate slots of a parent's slot chain.

Now consider the "only if" part. We need to show that if a node v is not an ancestor or descendant of node u, then it is not in conflict with u. Let w be the common parent of u and v. Without loss of generality, u is a descendant of $lc(w)$ and v is a descendant of $rc(w)$ (note that per terminology a node is a descendant of itself). By lemma III.1, $lc(w)$ and $rc(w)$ are conflict free. Since u and v are descendants of $lc(w)$ and $rc(w)$ respectively, by lemma III.2, u and v are conflict free. ∎

The last step is to show conflict freedom between different p-chains.

Lemma III.4. *Consider two primitive chains (s_1, B) and (s_2, B) where $|s_1 - s_2| < B$. Then, the chains are conflict free*

Proof: Suppose they are in conflict. Then there must exist positive integers n_1 and n_2 such that

$$s_1 + n_1 \cdot B = s_2 + n_2 \cdot B$$

But this implies that $(s_1 - s_2) = (n_2 - n_1) \cdot B = n \cdot B$ for some positive integer n. But we are given that $|s_1 - s_2| < B$, and thus the p-chains are conflict free. ∎

Theorem III.1. *Algorithm Geometric ISC produces a conflict-free allocation of slot chains.*

Proof: Clearly, two given slot chains produced by (repeated invocation of) Algorithm Gemoetric ISC are either in the same p-chain or not. If they are not, then by lemma III.4 they are conflict free. Suppose then that they are in the same p-chain. Algorithm Geometric ISC stops exploring a node when it is marked full. Thus, it can never allocate a two slot chains where one is a descendant of another. Any other allocation within this tree is conflict free by lemma III.3. ∎

C. Optimality

We prove below that the algorithm is optimal. The optimality is a direct result of packing the allocation space tightly as we move from "left to right" in the tree (DFS).

Theorem III.2. *Algorithm Geometric ISC is optimal.*

Proof: First, since each p-chain allocates a disjoint subset of the allocation space, it is clear that if each p-chain is optimally allocated, then Geometric ISC is optimal.

Suppose allocation of each p-chain is not optimal. Then, there exists a sequence of requests such that Geometric ISC cannot allocate the last request, say $p = 2^k \cdot B$, but an optimal algorithm A can. Since Geometric ISC cannot allocate, because of the way the algorithm marks nodes, all nodes labeled (*,p) must have been marked *occupied*.

Now consider the nodes marked *occupied*. This means at least one of its descendants is allocated, and hence one of the node's children is also *occupied*. Thus the only sequences available for algorithm A would be those represented by a combination of descendants of *different* (*, p) nodes, say v_1 and v_2. This would imply that exactly one of the children of v_1 is *occupied* and exactly one of the children of v_2 is *occupied*. However, algorithm Geometric ISC proceeds in a depth first manner. Without loss of generality if v_1 is before $v2$ in the depth-first search, then, it is impossible that if all of the descendants of v_1 were not allocated (thereby allowing it be not *occupied*), a descendant of v_2 is *occupied*. Therefore, by contradiction, Geometric ISC is optimal. ∎

D. Generalized ISC

We now consider arbitrary requests, that is, the period is not constrained to be $B \cdot 2^k$ as in the geometric version of the problem. As mentioned earlier, the key observation is that such a request can be decomposed approximately into multiple geometric chains. For example, consider a base of 5 (which constrains chain periods to be 5, 10, 20, 40 etc.) and consider an input request for 1/12 (0.083) of capacity (ie, a period of 12). Allocating two geometric chains of periods 20 each yields an allocation of 0.1, allocating three chains of periods 20, 40 and 80, yields a better approximation of 0.087 and so on – the more chains we can decompose it into, the closer we come to the input request.

We note that in general it may not be possible to make these "component" requests add up exactly to the original request, but one can come arbitrarily close. We define a *z-approximation* of a given allocation period p as an allocation of capacity A such that $1/p \leq A \leq (1+z) \cdot (1/p)$. Note that by this definition the user never gets less than the amount ($1/p$) she asked for.

The approximation factor z represents an interesting tradeoff between capacity wastage and overhead of setting up the request. The smaller z is, the more components one needs to fulfill the request (if this is not obvious now, it should be after the algorithm description), and therefore the control overhead is higher. On the other hand, there is less wasted capacity.

We now describe a z-approximate solution to the (generalized) ISC problem by decomposing a given request into geometric requests. Suppose y_i, $1 \leq i \leq k$ ... are the geometric periods that comprise a z-approximate solution to the general period request. By definition, $y_i = B \cdot 2^{n_i}$ for non-negative integer n_i. Thus, we need to find n_1, n_2 etc. such that

$$\frac{1}{p} \leq \sum_{i=1}^{k} \frac{1}{B \cdot 2^{n_i}} \leq (1+z) \cdot \frac{1}{p} \qquad (1)$$

We can rewrite this as

$$\frac{B}{p} \le \sum_{i=1}^{k} \frac{1}{2^{n_i}} \le (1+z) \cdot \frac{B}{p} \quad (2)$$

Notice that the middle summation essentially picks some inverse powers of 2 to be part of the sum. If we denote by $w_j = 0, 1$ whether or not 2^j is picked, one can rewrite the above as

$$\frac{B}{p} \le \sum_{j} \frac{w_j}{2^j} \le (1+z) \cdot \frac{B}{p} \quad (3)$$

But the sequence of w_j is nothing but the binary representation of a fractional number[2]! Thus, one can reuse elegant algorithms for binary fractions for our purpose. Then, the set of Geometric ISCs corresponds to the 1 positions in the binary representation (in particular, if the j^{th} position has a 1, we invoke a Geometric ISC calculation with a period of $(B * 2^j)$. We go from left to right until we are within the z-approximation bound.

IV. Slot Chains in TDMA MAC Protocols

In this section we show how the slot chains can readily be used in lieu of frames in any TDMA MAC protocol. Note that every TDMA scheme has four parts:

1) A part that decides how much capacity a flow needs and therefore how many slots.
2) A data structure for organizing the slots – so far the frame.
3) An algorithm for allocating the slots across interfering nodes.
4) A distributed scheme/protocol for implementing the slot allocation.

In short, a given TDMA MAC protocol only needs to modify the second item on this list, namely the data structure, to start using slot chains instead of frames. In the following we describe this process in detail.

Recall from the Section III that every p-chain can contain 2^n interleaving slot chains, each with period $2^n \times B$ and capacity $1/(2^n \times B)$, for $n = 0, 1, 2, ...$ For any given value of n and B, the tree representation of the Geometric Allocation Tree, Figure 2, can be transformed into a two dimensional table by "grabbing" only the leaf nodes from the binary tree. We show this transformation for $n = 3$ and $B = 5$ in Figure 3. The table representation is simpler conceptually and in implementation, yet it is equivalent to the binary tree representation. In particular, the allocation of a leaf node in the binary tree corresponds to allocating the box in the allocation table indicated by the line. Both the leaf node and the box in the allocation table have capacity of $1/(2^n \times B)$, or $1/40$ in our example. Allocation of a non-leaf node in the binary tree consists of (recursively) allocating the

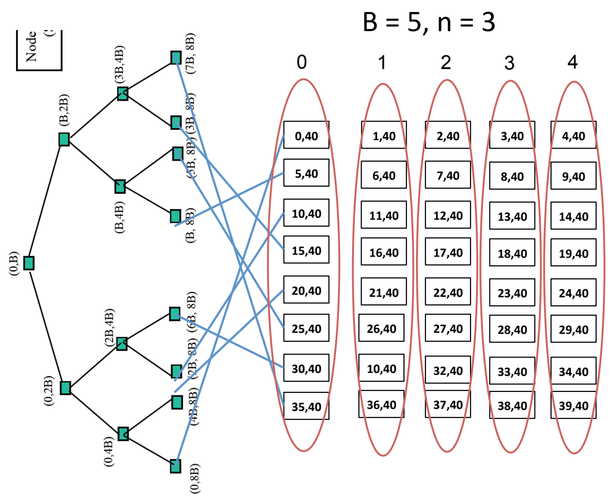

Fig. 3. The binary tree and the allocation table are equivalent. Allocation of a leaf node in the binary tree corresponds to allocating the box in the allocation table indicated by the line. Both the leaf node and the box have capacity of $1/(2^n \times B)$. Allocation of a non-leaf node consists of (recursively) allocating the boxes corresponding to its children. Thus, for example, allocating (2B,4B) for a given request would correspond to allocating boxes (30,40) and (10,40) – follow the lines from the children of (2B,4B)

boxes corresponding to its children. Thus, for example, if algorithm Geometric IPSC calls for allocating (2B,4B) for a given request, that would correspond to allocating boxes (30,40) and (10,40) – follow the lines, in Figure 3, from the children of (2B,4B). Note that, the table size is not static but it depends on the value of n, which in turn is selected by the Generalized ISC algorithm dynamically based on the capacity demands.

In a traditional TDMA MAC, part of the protocol translates a capacity demand into a number of slots in a frame. Switching to frameless would simply require replacing that part of the protocol with a two step process. Step 1, the Generalized ISC algorithm described in Section III-D is deployed to convert a capacity demand into a slot chain. Step 2, the binary tree node corresponding to the particular slot chain is converted into boxes in a two dimensional table, as shown in Figure 3. The rest is straightforward. A protocol that can work with slots on a one-dimensional array can just as easily work with boxes on a two-dimensional array!

V. Simulations

In this section we evaluate the performance of the slot chains as compared to traditional frames using the ns-2 simulator. As shown in Section IV, any TDMA based MAC can be modified to use the slot chains and therefore could be used for this evaluation. For this study, we have chosen to modify SITA[7]. The choice was purely for practical reasons as we had access to the SITA source code. To highlight this point, for the rest of the section

[2]In a fractional binary number (e.g. 0.101_2) the first digit is $(1/2)^1$, the second digit is $(1/2)^2$ and so on.

we refer to SITA with frames as Legacy TDMA while SITA with slot chains as Frameless TDMA.

A. Simulations Settings

In all the experiments, the frame size for the Legacy TDMA is set to 10 slots while the Frameless TDMA uses 10 p-chains. As in the previous examples the n for DISC is set to 3. With these numbers, the minimum capacity that SITA can allocate is $1/10$ of the available capacity, wxhile the minimum DISC can allocate is $1/(10 \times 2^3)$ of the available capacity. Note that for every frame size F the minimum capacity that SITA can allocate is 1/F, while for DISC, the minimum is $1/(F * 2^n)$ of the available capacity.

Unless otherwise specified, the packet size is set to 500 bytes, the physical channel transmission rate is set to 11 Mbps and the simulation time is 150 seconds. For all the experiments, we use CBR traffic over UDP, which is similar to the traffic a real-time multimedia application is expected to generate.

B. Experiment 1: Star Topology

We start with a simple example that nevertheless showcases the superiority of the slot chains in allocating the channel capacity to a variety of capacity demands. Consider a seven node star topology and assume that the node at the center receives six capacity demands, one for each of its one hop neighbors. Suppose that the capacity demands are 1/20, 1/20, 1/10, 1/5, 1/80, 1/2 of the available capacity and they each arrive separated by a 5 sec interval.

Figure 4 shows the observed throughput that each of the six flows achieves with Legacy and Frameless TDMA respectively. As we can see, the first five flows are served equally well by both approaches. However, the sixth flow is denied admission by the Legacy TDMA while it is accommodated by the Frameless TDMA. We purposely chose the sixth flow to be relatively heavy to show that the penalty in achieved throughput can be high. As shown in Figure 5, the Frameless TDMA is able to deliver almost twice as much data as the Legacy. Because the smallest allocation possible with the frame is 1/10 of the total channel capacity, 60% of the capacity ends up being allocated for accommodating the first five flows even though the actual demand is for 41.25%. Thus there is no capacity left for the sixth flow which needs 50% of the channel capacity. On the other hand, the Frameless TDMA uses precisely 41.25% of the channel capacity to accommodate the first five flows and therefore can admit and serve the sixth flow.

C. Experiment 2: Grid topology

In this experiment we evaluate the performance of the Frameless TDMA over long flows that are subject to high interference. For this purpose we use a 25 node

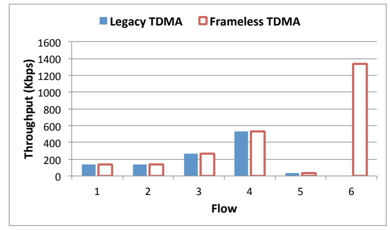

Fig. 4. The Frameless TDMA allocates precisely the capacity requested which allows it to admit one more flow than the frame based Legacy TDMA.

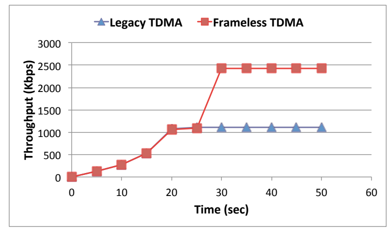

Fig. 5. The Frameless TDMA achieves significantly higher throughput due to is precise channel allocation made possible by the slot chains.

Manhattan grid. Five source destination pairs are selected such that the source is on one side of the grid and the destination on the other side.

We want to compare the Frameless TDMA to the Legacy TDMA using different degrees of heterogeneity for the capacity demands of the five considered flows. For this purpose we introduce a new metric, which we refer to as *heterogeneity*. The heterogeneity metric quantifies the difference in capacity demands among the traffic flows by using a single number. In this specific experiment we define 0 as the minimum value for heterogeneity and 3 as the maximum. A heterogeneity of zero is assigned to the case in which all the flows have the same capacity demand and the capacity demands are multiples of 1/10, which is the minimum capacity the Legacy TDMA can allocate in our simulations. When heterogeneity equals one, the capacity demands are selected uniformly at random from 1/20th to the whole capacity. Similarly, for heterogeneity of two and three, the capacity demands are selected from 1/40 to 1 and 1/80 to 1 respectively.

Figures 6 and 7 depict the performance of the Frameless and Legacy TDMA protocols as a function of heterogeneity. As expected when heterogeneity is zero, the two approaches perform exactly the same. However, as heterogeneity increases the Frameless TDMA outperforms the Legacy TDMA in terms of number of the admitted flows (Fig. 6) as well as the overall network throughput (Fig. 7).

VI. Related Work

Prior work related to this paper can be broadly classified along the following lines: access techniques,

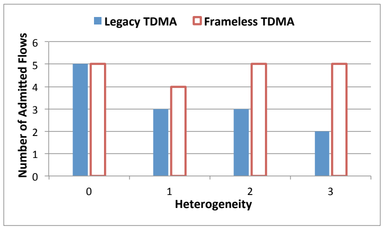

Fig. 6. As the heterogeneity increases, the Frameless TDMA admits more flows than the Legacy TDMA which is limited by the rigidity of the frames.

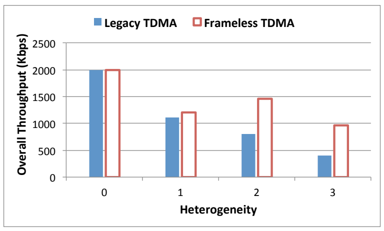

Fig. 7. As the heterogeneity increases, the Frameless TDMA admits more flows which translates into higher overall network throughput.

in particular CSMA/CA and TDMA; hybrid and other variants; and overlay implementations. We discuss each of these briefly below.

Random Access MAC Protocols: IEEE 802.11 DCF (based on CSMA/CA) has become the de-facto standard reference for the research and development of MAC protocols for MANETs. However, 802.11 was designed and engineered for WLANs. When used in MANETs, it presents numerous shortcomings, such as unfairness [5] and spatial bias [18]. Furthermore, the inherent weakness of CSMA/CA in handling real-time traffic [16] is exacerbated in the multihop setting [11], [12]. Finally, in emerging networks such as long-distance, underwater and airborne networks [10], [2], [9], the propagation delays are too high for the CSMA to be an overhead-sensible solution.

TDMA MAC Protocols: TDMA offers superior performance under high loads [3], and better capacity guarantees (crucial for real-time multimedia applications) when compared with CSMA/CA. However, it requires tight network wide synchronization [3], [6] and has poor channel utilization under bursty traffic. TDMA MAC is somewhat more attractive in the military context [20], and emerging networks [10], [9].

Hybrid variants of CSMA: In light of the inherent weaknesses of the CSMA in general and 802.11 in particular and its dominance in the market, there has been significant activity in improving the 802.11 MAC. This has resulted in protocols that can be broadly classified in two categories: protocols that provide service differentiation, and CSMA/TDMA hybrids.

Service differentiation. In the first category [19], [1], the modification still maintains the CSMA nature of the protocol but make changes to the backoff procedures so that the probability of accessing the channel depends on some assigned weights. However, assigning weights is non-trivial, and in the presence of multiple flows with the same priority the protocol with fall back to pure CSMA.

CSMA/TDMA hybrids: The common goal in the design of CSMA/TDMA hybrids [8], [14], [13] is to combine the strengths of random access and TDMA approach into a single protocol. In [8], [14] it is proposed using CSMA for accessing the channel and then holding it, in TDMA fashion, for as long as it is necessary. Thus, no synchronization is required and the channel allocation is based on traffic demand, while at the same time there are capacity share guarantees. However, both works assume that the real-time traffic packets arrive at a specific and universal inter-arrival period. This assumption would not hold in scenarios where different kinds of real-time traffic, e.g video, VOIP, are present simultaneously in the network. Finally, both protocol have no efficient way of dealing with node mobility. ZMAC [13] starts by performing slot assignments as a pure TDMA . However, in contrast with the latter, the slots are accessed using CSMA, with the owner merely having a higher probability of winning the contention. However, ZMAC requires slot synchronization for its TDMA part, otherwise it will fall back to pure CSMA.

VII. Concluding Remarks

The typical mobile device today comes with a wide variety of real-time applications (VoIP, regular video, HD video, gaming, and other interactive applications). While the real-time and delay-sensitive nature of these applications fits poorly with CSMA/CA, the heteoregeneity of rates and bursty nature fits poorly with TDMA.

We have described a framework for frameless TDMA based on the concept of slot chains and capable of allocating the channel capacity in accordance with traffic demands. Our simulation analysis shows a distinct performance advantage in allocation efficiency especially in the presence of heterogeneous flows.

Our framework is applicable to infrastructure-based networks (Wireless LANs) as well as to mesh networks and MANETs. Our work shows that a new class of protocols that relax the rigidity of TDMA by using slot chains instead of frames holds great promise for next-generation MAC protocols. Much work, however, needs to be done to make slot chains more general and couple it to flows at the network layer.

Acknowledgements

We thank researchers at Scientific Systems Company Inc. Rajesh Krishnan, Carlos Gutierrez and Mike Perloff, and our sponsor David Krzysiak at AFRL for their support of this paper.

REFERENCES

[1] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 8: Medium Access control (MAC) Quality of Service Enhancements, IEEE Std. 802.11e-2005 (November 2005).

[2] Pravin Bhagwat, Bhaskaran Raman, and Dheeraj Sanghi. Turning 802.11 inside-out. *SIGCOMM Comput. Commun. Rev.*, 34(1):33–38, 2004.

[3] P. Djukic and P. Mohapatra. Soft-TDMAC: A Software TDMA-based MAC over commodity 802.11 hardware. In *INFOCOM 2009, IEEE*, pages 1836 –1844, 2009.

[4] P. Djukic and P. Mohapatra. Soft-tdmac: A software-based 802.11 overlay tdma mac with microsecond synchronization. *Mobile Computing, IEEE Transactions on*, 11(3):478 –491, march 2012.

[5] Violeta Gambiroza, Bahareh Sadeghi, and Edward W. Knightly. End-to-end performance and fairness in multihop wireless backhaul networks. In *ACM MobiCom '04*.

[6] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. IEEE 802.11s: The WLAN mesh standard. *Wireless Communications, IEEE*, 17(1):104 –111, 2010.

[7] G. Jakllari and R. Ramanathan. A sync-less time divided mac protocol for ad hoc networks. *IEEE MILCOM 2009.*

[8] C. R. Lin and M. Gerla. Asynchronous Multimedia Multihop Wireless Networks. In *IEEE INFOCOM '97*.

[9] Jim Partan, Jim Kurose, and Brian Neil Levine. A survey of practical issues in underwater networks. In *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*.

[10] Bhaskaran Raman and Kameswari Chebrolu. Design and evaluation of a new MAC protocol for long-distance 802.11 mesh networks. In *ACM MobiCom '05*.

[11] Ram Ramanathan. Challenges: a radically new architecture for next generation mobile ad hoc networks. In *ACM MobiCom '05.*

[12] Ananth Rao and Ion Stoica. An overlay MAC layer for 802.11 networks. In *ACM MobiSys '05.*

[13] Injong Rhee, Ajit Warrier, Mahesh Aia, and Jeongki Min. Z-MAC: a hybrid MAC for wireless sensor networks. In *ACM SenSys '05*.

[14] Sumit Singh, Prashanth Aravinda Kumar Acharya, Upamanyu Madhow, and Elizabeth M. Belding-Royer. Sticky CSMA/CA: Implicit synchronization and real-time QoS in mesh networks. *Ad Hoc Netw.*, 5(6):744–768, 2007.

[15] J.L. Sobrinho and A.S. Krishnakumar. Real-time traffic over the IEEE 802.11 medium access control layer. *Bell Labs Technical Journal*, pages 744–768, 1996.

[16] Yuan Sun, Irfan Sheriff, Elizabeth M. Belding-Royer, and Kevin C. Almeroth. An experimental study of multimedia traffic performance in mesh networks. In *WiTMeMo '05*.

[17] Bharath Sundararaman, Ugo Buy, and Ajay D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

[18] Karthikeyan Sundaresan, Hung-Yun Hsieh, and Raghupathy Sivakumar. IEEE 802.11 over multi-hop wireless networks: problems and new perspectives. *Ad Hoc Netw.*, 5(6):744–768, 2003.

[19] Nitin H. Vaidya, Paramvir Bahl, and Seema Gupta. Distributed fair scheduling in a wireless LAN. In *ACM MobiCom '00*.

[20] C.D. Young. USAP multiple access: dynamic resource allocation for mobile multihop multichannel wireless networking. *IEEE MILCOM 1999*.