Computer Networks 122 (2017) 16–28

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Variable-weight topology-transparent scheduling

Jonathan Lutz, Charles J. Colbourn, Violet R. Syrotiuk*

School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, Arizona 85287-8809, USA

ARTICLE INFO

Article history:
Received 4 June 2016
Revised 7 January 2017
Accepted 7 April 2017
Available online 13 April 2017

Keywords:
Wireless networks
Channel allocation
Combinatorial mathematics
Distributed algorithms
Adaptive algorithms

ABSTRACT

In this paper, topology-transparent scheduling for mobile multi-hop wireless networks is extended from constant-weight to variable-weight schedules. A construction of variable-weight topology-transparent schedules – the first of its kind – is provided based on transversal designs from the finite field. The schedules are integrated into the VWATT medium access control (MAC) protocol, enabling nodes to dynamically select their schedule weights to accommodate both their local traffic load and local topology while maintaining a guarantee on maximum delay. Simulations show VWATT to increase throughput compared to constant-weight topology-transparent schedules, to reduce both maximum and expected delay relative to schedules whose weights are not constrained to the set of weights, and to adapt rapidly to changes in topology and traffic load. The results suggest variable-weight topology-transparent scheduling as a viable approach to medium access control when a bound on delay is required.

© 2017 Published by Elsevier B.V.

1. Introduction

Topology-transparent scheduling [1] is a scheduled medium access control (MAC) mechanism designed specifically for applications in highly mobile multi-hop wireless networks requiring a delay guarantee, such as voice or video. Scheduled schemes usually assume that time is divided into discrete units called *slots*. A *schedule* (or *frame*) F is described by a binary vector $f_0 f_1, \ldots, f_{n-1}$ of length n with one element for each slot. Typically, each node in the network uses its schedule in a cyclically repeated way, with nodes synchronized on frame boundaries. If $f_j = 1$ then the node is permitted to transmit in slot k, when $k \equiv j \pmod{n}$; otherwise the node is silent and could receive.

Different from topology-dependent scheduling schemes where a node recomputes its schedule in response to changes in the network topology (see, as examples, [2,3]), in topology-transparent scheduling each node's schedule is fixed. For each node i with at most a threshold number of neighbours, its schedule guarantees a collision-free transmission to each neighbour. What makes the schedule 'transparent' to topology changes is that this property holds no matter which nodes are i's neighbours; however, to obtain the delay guarantee their number must be bounded by the threshold. (In [4], the delay guarantee is shown to degrade gracefully when the bound is exceeded.)

More formally, let N be the number of nodes in the network, and $D_{\max}$ the maximum neighbourhood size. Treat node i's schedule $F_i, 0 \le i \le N-1$, as a subset S_i of $\{0, 1, \ldots, n-1\}$ giving the slots in which i is permitted to transmit; i.e., if $j \in S_i$ then $f_j = 1$ in F_i, otherwise $f_j = 0$. In essence, F_i is the characteristic vector of the set S_i. Now, the combinatorial problem asks for each node i to be given a subset S_i with the property that the union of $D_{\max}$ or fewer other subsets cannot contain S_i. This may be expressed mathematically by requiring that

$$\left(\bigcup_{j \in S} S_j\right) \not\supset S_i, \quad \text{(the } \textit{cover-free condition}\text{)}$$

where $S \subseteq \{0, \ldots, N-1\} \setminus \{i\}$ with $|S| \le D_{\max}$. In the language of sets this is precisely a *cover-free family*. In other vernacular, these are equivalent to disjunct matrices [5] and to certain superimposed codes [6]; see [7]. Existing constructions for topology-transparent schedules [1,8] are shown to correspond to an orthogonal array in [9], and are generalized as cover-free families in [10]. Steiner systems, the densest cover-free families, provide the shortest possible frame length, and hence the shortest worst-case maximum delay, for a given $D_{\max}$ and N [11].

Fig. 1 demonstrates both the potential and the limitation of topology-transparent scheduling. In it we plot the MAC throughput, defined as the total number of packets delivered per second to one-hop neighbours, achieved by four different strategies; we discuss each strategy in turn. TDMA schedules are trivially topology transparent. In TDMA, the schedule for node i has only one transmission slot $S_i = \{i\}$ in a frame of length N. The *weight* of each schedule corresponds to the Hamming weight of its correspond-

* Corresponding author.
E-mail addresses: jlutz@asu.edu (J. Lutz), colbourn@asu.edu (C.J. Colbourn), syrotiuk@asu.edu (V.R. Syrotiuk).

http://dx.doi.org/10.1016/j.comnet.2017.04.023
1389-1286/© 2017 Published by Elsevier B.V.

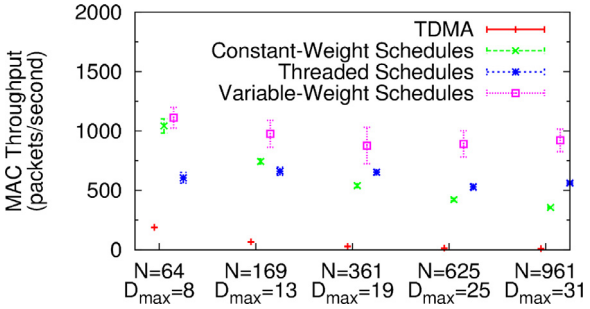

Fig. 1. Constant- vs. variable-weight topology-transparent schedules.

Table 1
Summary of notation used in Section 2.

Notation	Meaning of notation
N	The number of nodes in the network
n	The frame (schedule) length
m	The number of schedule weights
W	The set of schedule weights, $W = \{w_0, \ldots, w_{m-1}\}$
$S_{i,j}$	The jth schedule of node i, $0 \le j < m$
$\mathscr{S}_i$	The collection of schedules given to node i, $\mathscr{S}_i = \{S_{i,0}, S_{i,1}, \ldots, S_{i,m-1}\}$
$\mathrm{wt}(S_{i,j})$	The weight of the jth schedule of node i, $\mathrm{wt}(S_{i,j}) = w_j$
$\mathrm{W}_{\max}$	The weight condition bound, a fixed fraction of n
D	A subset of the set of nodes $D \subseteq \{0, \ldots, N-1\}$

ing characteristic vector. Hence all TDMA schedules have weight of one. TDMA supports the largest neighbourhood size but requires the longest frame length, $\mathrm{D}_{\max} = n = N$. As Fig. 1 shows, the throughput of TDMA is poor in general, but is optimal when the network is fully connected and saturated with traffic.

In [1,9,10], topology-transparent schedules are designed to minimize frame length while providing *at least one* successful transmission to each neighbour per frame. Alternatively, schedules have been designed to ensure *more than one* successful transmission per frame [8,12,13], improving minimum throughput at the expense of maximum delay. A number of other constructions have been proposed to generate schedules, some with frame length $n \ll N$ [14–17].

Fig. 1 shows that the throughput for constant-weight schedules using constructions from [1,9,10] is significantly better than TDMA, but degrades for large $D_{\max}$. For example, constructions based on orthogonal arrays produce schedules with k subframes of length v concatenated together, with one transmission slot per subframe; such schedules all have weight k. The use of constant-weight schedules places an unnecessary constraint on throughput in neighbourhoods smaller than $D_{\max}$, especially for nodes with large traffic demands. It also results in the loss of the delay guarantee in neighbourhoods larger than $D_{\max}$.

Protocol threading [18] has been proposed as a method to handle violations of $D_{\max}$. Several schedules, each supporting a different degree bound, are interleaved on a slot-by-slot basis. As Fig. 1 shows, this has a stabilizing effect on the throughput but results in a maximum delay significantly longer than that of TDMA. (The Appendix provides details on the schedules threaded to produce the results in Fig. 1 for each N and $D_{\max}$.) A different method for handling violations of $D_{\max}$ is used in [19], where each node selects a new constant-weight schedule at random; this voids the delay guarantee initially offered by the schedule.

Optical orthogonal codes (OOCs) were initially designed to support quality-of-service requirements in CDMA over optical fiber [20]. They were applied to topology-transparent scheduling in [21] to yield schedules with two weights. Each node is assigned a schedule whose weight is dependent on the node's desired quality-of-service. Both weights are small compared to the frame length and, once assigned to a node, the assignment is fixed. An empirical study in [22] suggests that multiple weights are beneficial when the weights are chosen well spread from one another.

To the best of our knowledge, we propose the first topology-transparent protocol in which nodes are able to vary their schedule weight dynamically according network conditions, while still achieving a guarantee on maximum delay *without* paying a high cost in throughput. Each node is provisioned with a set of schedules from a *transversal design* constructed from the finite field. Each schedule in the set has a different weight. Each node dynamically selects a schedule from the set with a weight that best accommodates its current network conditions. The guarantee on maximum delay is obtained by requiring that the weight of the schedules in each neighbourhood does not exceed a weight bound. Surprisingly, the variable-weight schedules are constructed at no cost to frame length n, $D_{\max}$, or N compared to the constant-weight schemes. Fig. 1 illustrates the potential our variable-weight scheme has to recover throughput lost to constant-weight schedules.

This paper makes four original contributions:

1. We formulate the problem of variable-weight topology-transparent scheduling (Section 2) that constrains the weight of a neighbourhood instead of its size.
2. We develop variable-weight topology-transparent schedules using a transversal design constructed from the finite field (Section 3).
3. We integrate the variable-weight topology-transparent schedules into VWATT, a MAC protocol that allows a node to adapt the weight of its schedule on-the-fly to changing network conditions while achieving a delay bound (Section 4).
4. Simulation results of VWATT show improved throughput while maintaining the delay characteristics of constant-weight topology-transparent schedules (Section 5).

The rest of this paper is organized into sections that describe the details of our contributions. After a discussion of future work in Section 6, we summarize and conclude in Section 7.

2. The requirements of variable-weight topology-transparent schedules

We now formulate the problem of variable-weight topology-transparent scheduling, introducing a weight condition that when satisfied, also satisfies the cover-free condition. We assume the network is homogeneous, where each node is equipped with a set of schedules of the same weights. In the sequel, we use set notation for schedules, always using Roman typeset (e.g., S) to denote a set and calligraphic typeset (e.g., $\mathscr{S}$) to denote a set of sets; Table 1 summarizes our notation for this section of the paper.

Let $W = \{w_0, \ldots, w_{m-1}\}$ be a set of m weights. We are to give each node i a collection of schedules $\mathscr{S}_i = \{S_{i,0}, S_{i,1}, \ldots, S_{i,m-1}\}$ where $\mathrm{wt}(S_{i,j}) = w_j$, i.e., the jth schedule has the jth *weight*. While our discussion in Section 1 focussed on throughput, a primary objective of topology-transparent schemes is to establish a delay guarantee. Suppose then that node i_0 has neighbours $i_1, \ldots, i_D$. If each node i_ℓ independently chooses the j_ℓth weight for its schedule, a delay guarantee of a single frame length holds at node i_0 only if

$$\left(\bigcup_{\ell=1}^{D} S_{i_\ell, j_\ell} \right) \not\supset S_{i_0, j_0}.$$

Download English Version:

https://daneshyari.com/en/article/4954582

Download Persian Version:

https://daneshyari.com/article/4954582

Daneshyari.com