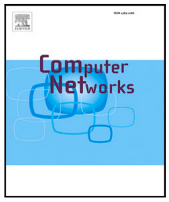
Computer Networks 241 (2024) 110213

Contents lists available at ScienceDirect

Computer Networks

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



Digital twin approach to estimating and utilizing the capacity region of wireless ad hoc networks☆

Yannis Thomas [a], Stavros Toumpis [a,*], Nikolaos Smyrnioudis [b]

[a] MMlab, Department of Informatics, AUEB, Greece
[b] CERN, Switzerland

ARTICLE INFO

Keywords:
Capacity region
Experiments
IEEE 802.11
Interference
Measurement
OLSR
Online ML
Performance evaluation
Wireless network

ABSTRACT

The capacity region (CR) of a wireless ad hoc network describes in detail the network's ability to carry traffic when operating optimally. However, the volume of measurements required to determine exactly the CR of a deployed network typically increases very fast with its size. We introduce a digital twins approach for estimating (i.e., measuring approximately) the CR of deployed networks using a modest number of measurements as well as estimates of measurements derived using online machine learning. The CR estimate may be thought of as the state of a digital twin of the network. We apply the methodology to an experimental network of 16 Raspberry Pi nodes using IEEE 802.11.

Furthermore, to showcase the utility of having a digital twin of the network, we compare the experimental network performance of the well-known Optimized Link State Routing (OLSR) protocol with both the theoretically optimal performance of the digital twin derived using the CR estimate and the experimental network performance of a novel, CR-inspired centralized protocol that creates network-wide time divisions based on the CR estimate; the performance of the latter protocol is found to be superior to that of OLSR and, more importantly, the three-way comparison provides important insights.

1. Introduction

Wireless ad hoc networks have been investigated for several decades now due to their ability to provide connectivity through multihop routing in cases where installing wired infrastructure is impossible or impracticable; they are currently being investigated for use in, e.g., satellite [1], sensor [2], Internet of Things [3], and vehicular [4] applications as well as distributed learning settings [5].

The fundamental capacity of a wireless ad hoc network to carry traffic when operating optimally can be described in detail in terms of its capacity region (CR), a properly defined multidimensional set consisting of achievable combinations of data rates. Unfortunately, the complexity of describing the CR typically increases very fast with the size of the network [6–8].

Motivated by this complexity, in this work firstly we develop the Sequential Expansion Algorithm (SEA), with which the CR can be estimated (i.e., measured approximately) using a manageable volume of measurements that must be scheduled and collected at a central network controller. The basic idea of the algorithm is to discover, by measurements, progressively larger sets of links that can be simultaneously active without excessively interfering with each other (as determined by checking for a set of interference-centric link strength conditions), by adding single, sufficiently strong links to such sets already discovered; we start the first iteration by combining singleton sets comprising single, sufficiently strong links. We implement this algorithm to a wireless network testbed comprised of 16 Raspberry Pis to arrive at an estimate of its CR.

Secondly, we show that the CR estimate can be gainfully utilized by a novel traffic optimization protocol, called the Network-Wide Time Division (NWTD) algorithm. NWTD is executed by the central network controller that specifies efficient time division schemes for the network based on solutions of a traffic optimization problem that models the network through the CR estimate. With this approach the network can significantly improve its performance (by around 34% in our reported experiment) with respect to the conventional distributed approach of using IEEE 802.11 together with a routing protocol (here, OLSR). The theoretical optimal performance using the CR estimate also provides valuable intuition. Therefore, we follow the Digital Twins approach of

☆ This work has appeared in preliminary, conference form in Thomas et al. (2021). A detailed exposition of the testbed used in this work appears in Thomas and Toumpis (2023) and is not included here. Research conducted by N. Smyrnioudis while he was a student at AUEB.

* Corresponding author.
E-mail address: toumpis@aueb.gr (S. Toumpis).

https://doi.org/10.1016/j.comnet.2024.110213
Received 21 October 2023; Received in revised form 8 January 2024; Accepted 22 January 2024
Available online 26 January 2024

optimizing the operation of a *physical* twin (that comprises the wireless nodes, the wireless channel, and the network controller) based on mathematical optimization taking place in a *digital* twin (that comprises SEA, the CR estimate, and an optimization module).

Regarding the novelty of our contributions, to the best of our knowledge, this work is the first to introduce a practicable method for estimating the CR of wireless ad hoc networks and then a method for utilizing the CR estimate (in particular using a Digital Twins approach) without resorting to a binary interference assumption (cf. [9] and Section 2.2). It is also the first to use ML in order to estimate the data rates of sets of interfering links.

Regarding our prior related work, we note that a preliminary version of this work appeared in [10]. That work included measurements using a preliminary 5-node network that provided inspiration for the development of this work. In the work at hand, which reports results from a 16-node network, the interference-centric link strength conditions and SEA (i.e., the key theoretical contributions of this work) have been significantly expanded, the assumptions on the network that delineate the scope of our work have been modified in their essential aspects, we introduce the digital twins approach (in particular the NWTD algorithm), and we compare the performance of NWTD with OLSR. Also, we note that in [11] we present in depth our developed testbed; here, that work is mentioned where appropriate but is neither repeated nor extended.

The rest of this paper is organized as follows. In Section 2 we present background and related work. In Section 3 we develop our mathematical framework. In Section 4 we present SEA and also introduce the NWTD protocol. In Section 5 we present our experimental results. In Section 6 we discuss the limitations of our work and future research directions, and in Section 7 we present our main conclusions.

2. Related work

2.1. Capacity region definitions

The problem of determining theoretical upper bounds on the capabilities of wireless networks has attracted significant sustained interest ever since the publication of the seminal work in [12]. There, it was shown that, under a uniform placement of nodes on a plane, if each of the n nodes selects at random a single destination for its packets, then each of the resulting n streams can have a data rate on the order of $n^{-\frac{1}{2}}$.

However, there are $n(n-1)$ potential unicast source–destination pairs in the network, each of them with a possibly different target data rate, which means that the capabilities of the network cannot simply be defined in terms of a single number, but rather a suitably defined CR.

There is, however, no single universally adopted definition of the CR of a network. In all related investigations (e.g., in [6,9,13–17]) the CR is indeed a multidimensional set that provides a complete description of the network's capabilities to carry traffic when operating optimally, but details differ and various related definitions have been investigated.

In particular, in some cases (e.g., in [6,16]), the CR is the set of all combinations of data rates achievable by the *source–destination* pairs in the network; in some other cases (e.g., in [18]), as in this work, the CR is the set of all combinations of data rates achievable by the *transmitter–receiver* links. (Note that a node acting as data *source* and a node acting as data *destination* might be at disparate locations in the network, thus necessitating the transmission of packets along numerous links placed in between them, each comprising a *transmitting* node and a *receiving* node.) Different choices have also been made on what is subject to optimization: in works with a stronger information-theoretic flavor (e.g., [14]), the optimization is over wide swaths of the protocol stack; in network optimization works, such as [6,16,17] as well as this work, many aspects of the network operation, such as the physical layer and the MAC protocols used, are fixed and not subject to any optimization.

However, in all except special cases, due to the many degrees of freedom that wireless networks possess and the fact that simultaneous transmissions typically interfere with each other in a non-trivial manner, the shape of the CR is very complex to describe. In fact, various flow optimization problems involving the CR are known to be NP-complete [8,19,20].

This complexity is regrettable, as there are two important advantages of knowing the actual CR of a deployed network, even approximately. Firstly, if the CR is known, we can use it to solve flow optimization problems optimally and compare the optimal solutions with the actual performance achieved by current protocols; this provides us with an absolute measure on how good these protocols are. Secondly, this process sheds light on the inefficiencies of current protocols and allows us to build better, CR-inspired ones, that operate closer to the optimal. Our work here exemplifies these advantages.

2.2. Wireless ad hoc network measurements

Due to the aforementioned descriptive complexity of the CR, published investigations focusing on CRs have typically been theoretical in nature, eschewing measurements, with the CR defined using mathematical models instead. This is regrettable, as mathematical modeling cannot always adequately capture the complexities of the wireless medium, the operation of the various elements of the protocol stack, and the interplay between them.

A notable exception is the work in [21]; there, however, the measurement-based CR model used is based on the commonly used assumption that interference is binary, i.e., two links that transmit simultaneously either do not interfere with each other at all or both fail to transmit anything [22,23]. This assumption simplifies the estimation of the CR significantly, because once the interfering link *pairs* are found, the CR can be given in terms of the maximal sets of non-interfering links, without the need of any more measurements. However, the assumption does not hold in general and, in particular, our testbed [11], and so is not employed here. Furthermore, even with this assumption, computing all maximal sets of non-interfering links is still computationally intensive (in [21] the problem is reduced to a maximum weight independent set (MWIS) computation, which is still NP-hard).

To the best of our knowledge, the research work presented here is the first in which a method for estimating the CR of a wireless multihop network is proposed, implemented, and evaluated at a testbed, without relying on the aforementioned binarity assumption or any other one that dispenses with measuring anything more than the interference between link pairs.

2.3. Machine learning

There is a significant body of work accumulated on the wider topic of Machine Learning (ML) techniques in wireless network settings [24]. Relatively close to our work, in [25–27] ML techniques are used to predict the downlink data rate of specific links in a cellular setting; in [28] a three-stage ML-based technique is used for passively identifying link faults; in [29] two different ML algorithms are used to detect a variety of anomalies in a wide area network; in [30] the authors propose the scheduling of links using a deep neural network that accepts as input the locations of nodes alone.

As part of this work, and with the goal of minimizing the volume of measurements needed for estimating the CR, we use a standard deep neural network (DNN) to perform regression on the data rate vectors of sets of simultaneously transmitting, and hence mutually interfering, links, based on the previously measured data rate vectors of *other* sets of simultaneously transmitting links. Therefore, we use ML to predict the effects of interference within sets of competing links of a wireless networks; to the best of our knowledge this aspect of our work is novel. We stress, however, that the ML module we use is a standard DNN, and not novel.

2.4. Centralized control: Software defined networking and digital twins

We note that our network controller is akin to Software Defined Networking (SDN) controllers used in WiFi [21,31], cellular [32], and ad hoc [33–35] networks. In particular, an SDN-type network controller was also used in the aforementioned work in [21] to measure the CR, using the binarity assumption, and then utilize it, in particular for performing backpressure routing on modified transmitters that do not employ the MAC layer of IEEE 802.11. Our controller, on the other hand, activates transmitters according to a time-division schedule that does not take into account buffer sizes, as backpressure does, and with which in each slot nodes employ the IEEE 802.11 MAC layer.

A Digital Twin (DT) is a virtual representation of a physical system, referred to as the Physical Twin (PT) [36]. The DT is aimed to be an accurate representation of the PT and its state must track as close as possible the state of the PT through a suitable communication medium. DTs are useful in a variety of contexts. Notably, optimal policies for the DT can be discovered, and then the PT can be instructed to follow these policies. However, there are other applications, such as anomaly detection (involving security breaches), prediction, and counterfactual analysis (where we investigate the effects of potential system changes).

The DT concept has been applied extensively, over the last decade, in the context of smart manufacturing and Industry 4.0 [36]. However, recently it has received significant attention within the wireless networking community [37]. Indeed, on one hand wireless networks can provide the communication link between the DT and the PT ("wireless for twins" research). On the other hand, constructing and maintaining a DT of a wireless network can help optimize its operation ("twins for wireless" research) [37].

In our work, which falls on the latter research thrust, the PT comprises the wireless nodes, the wireless channel, and the network controller, whereas the DT comprises the SEA, the CR estimate (which acts as the DT state), and an optimization module.

3. Capacity region model

3.1. Wireless network utility maximization problem

We model our network as a directed graph whose vertices correspond to the nodes and edges correspond to unidirectional communication links between the nodes. Let $\mathcal{N}$ be the set of nodes, $\mathcal{L}$ be the set of links, and N and L be their respective sizes. The precise connectivity in the network can be succinctly given in terms of the $N \times L$ **adjacency matrix** A defined as follows:

$$A_{n\ell} = \begin{cases} 1, & \text{if } n \text{ is the transmitter of } \ell, \\ -1, & \text{if } n \text{ is the receiver of } \ell, \\ 0, & \text{otherwise.} \end{cases}$$

We model the transport of data in the network in terms of two vectors: the L-dimensional **flow vector**, x, whose component $x_\ell \geq 0$, $\ell = 1, \ldots, L$, is the average (over time) rate with which data flows across link ℓ, and the N-dimensional **divergence vector**, s, whose component s_n, $n = 1, \ldots, N$, is the average (over time) data rate with which node n inserts data into the network (therefore, if $s_n < 0$, node n is removing data with rate $-s_n$). The vectors x and s describe *average* (over time) rates in order to take into account networks in which the instantaneous data rates vary with time, due to the effects of medium access control, transmission adaptation, etc. As data must be conserved:

$$s_n = \sum_{m:(n,m)\in\mathcal{L}} x_m - \sum_{m:(m,n)\in\mathcal{L}} x_m, \quad n = 1, \ldots, N,$$

which, in matrix notation, and treating, from now on, all vectors also as column matrices, simply becomes $s = Ax$.

We model the traffic needs of each node n in terms of a **utility function** $U_n(\cdot)$, such that a divergence s_n offers node n a utility $U_n(s_n)$. Each utility function can be used to implicitly constrain the respective divergence to lie in a specific set, by setting it to be $-\infty$ outside that set.

Finally, we define the **capacity region** C to be the subset of $[0, \infty)^L$ that contains all flow vectors x that are achievable by the network, given the constraints imposed by the hardware, the protocols, and the wireless medium.

In this context we arrive at the following flow optimization problem:

Wireless Network Utility Maximization Problem (WNUM) — Abstract Form

$$\begin{aligned} &\text{maximize:} && \textstyle\sum_{n=1}^{N} U_n(s_n), \\ &\text{subject to:} && s = Ax, \quad x \in C. \end{aligned} \tag{1}$$

Observe that the parameters A and C describe the properties of the network, the parameters $U_n(\cdot)$ describe the traffic needs of the nodes, and the parameters s and x are subject to optimization.

3.2. Assumptions on the network

In Section 3.1 the CR was defined but its shape was left abstract. In Section 3.3 we will develop a specific model for it, based on three assumptions which we now present.

Assumption 1. *At each time instant the network controller (or the nodes collectively, if there is no network controller), can select which of the links are active or not, but all other properties of the link transmissions on the physical layer, such as the bandwidth of a potential subchannel that the link will use, its transmission power, its specific modulation and coding scheme (MCS), etc., are not subject to optimization*

Therefore, since there are L links, it follows that as far as transmissions are concerned, the network can be set to be, at any given point in time, in one among 2^L states, each state being a distinct subset of links $\mathcal{T} \subset \mathcal{L}$. We refer to these states as **transmission modes (TMs)**. Also, let the **size** $|\mathcal{T}|$ of the TM $\mathcal{T}$ be the number of its links. Finally, we will also refer to the TM that the network is in at a given time instant, as well as its links, as **active**.

Assumption 2. *There is a data rate $R_\ell(\mathcal{T}) \geq 0$ and a finite amount of time $T_\ell(\mathcal{T}) > 0$, such that when TM $\mathcal{T}$ is continuously active, the average link data rate of ℓ can be made (either by the network controller, or the nodes collectively, if there is no network controller) to converge, within time $T_\ell(\mathcal{T})$, to any value at most equal to $R_\ell(\mathcal{T})$, irrespective of the average link data rates achieved by other links in $\mathcal{T}$*

Intuitively, each active link causes a fixed amount of interference to the rest of the active links, irrespective of its data rate, so that each active link ℓ can be used by the network controller for transporting up to a maximum data rate $R_\ell(\mathcal{T})$. We associate with each TM $\mathcal{T}$ its corresponding **transmission mode rate vector (TMRV)** $R(\mathcal{T})$ which is an L-dimensional vector whose ℓ-th coordinate is $R_\ell(\mathcal{T})$, if link ℓ belongs to the TM, and equal to 0 otherwise.

Assumption 3. *The average link data rate with which a link ℓ can transport data when a TM $\mathcal{T}$ is active, even if $\mathcal{T}$ is not* continuously *active, is always at most equal to the rate $R_\ell(\mathcal{T})$.*

In more detail, suppose that in the time interval $[t_A, t_B]$ $\mathcal{T}$ was active for a set of non-consecutive intervals of total duration $D \leq t_B - t_A$, while in the rest of the time interval $[t_A, t_B]$ other TMs were active. Let B be the total amount of data transported across link ℓ while $\mathcal{T}$ was active. The assumption states that we must have $\frac{B}{D} \leq R_\ell(\mathcal{T})$. Intuitively, using the TM in a sequence of *non-consecutive* activations of arbitrary duration in between other TMs cannot help the network controller improve the performance of the TM with respect to the steady state described in Assumption 2.

To illuminate the TM concept and the quantities defined so far, in Fig. 1 we present three TMs in the case of a simple network model comprising four nodes and seven links. We also plot the average data rates if these TMs were combined in a time division; details appear in the figure.

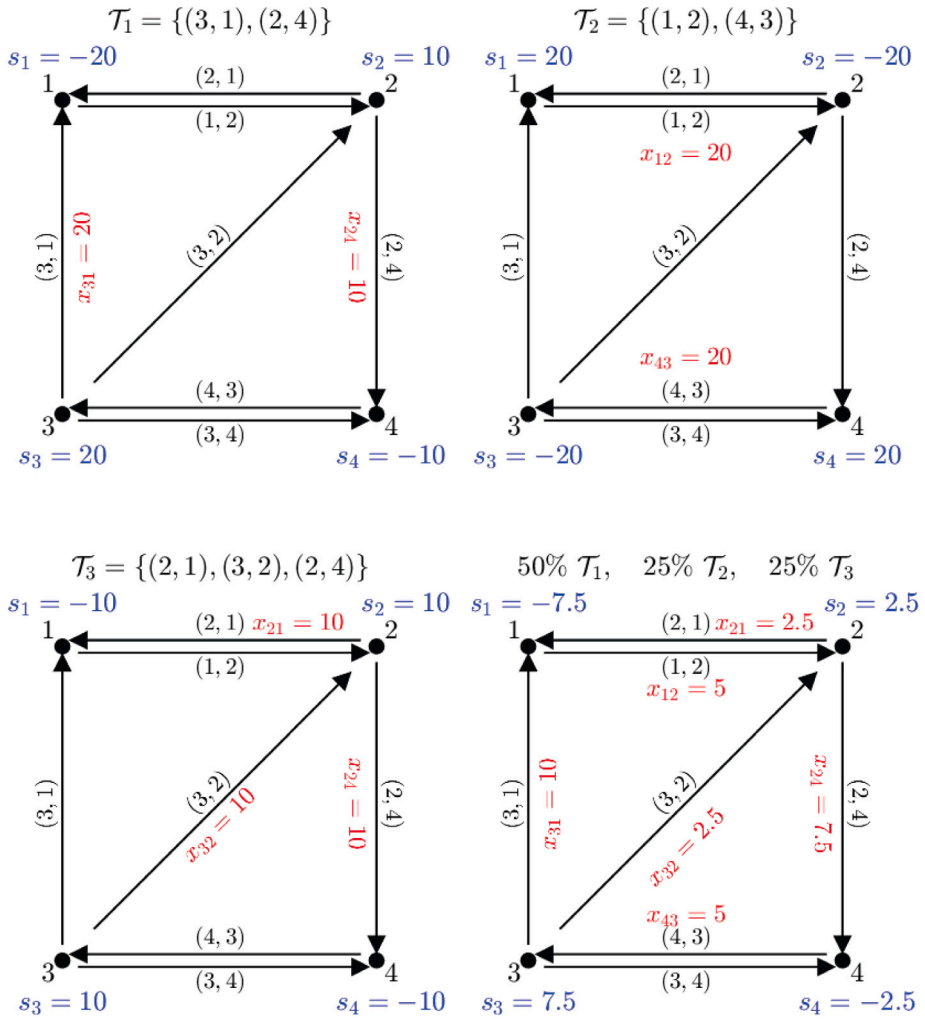


Fig. 1. An example network comprising four nodes, indexed $1, 2, 3, 4$ and seven links, indexed

$(1,2),\ (2,1),\ (3,4),\ (4,3),\ (3,1),\ (3,2),\ (2,4)$.

We plot 3 TMs,

$\mathcal{T}_1 = \{(3,1),(2,4)\},\ \mathcal{T}_2 = \{(1,2),(4,3)\},\ \mathcal{T}_3 = \{(2,1),(3,2),(2,4)\}$

with respective transmission mode rate vectors

$R_1 = (0,0,0,0,20,0,10),\quad R_2 = (20,0,0,20,0,0,0),\quad R_3 = (0,10,0,0,0,10,10).$

Link data rates x_{ij} not denoted in the figure are equal to 0. A time division comprising 50% by $\mathcal{T}_1$, 25% by $\mathcal{T}_2$, and 25% by $\mathcal{T}_3$ corresponds to average data flows and divergences shown on the bottom right and leads to a time division rate vector

$0.5R_1 + 0.25R_2 + 0.25R_3 = (5, 2.5, 0, 5, 10, 2.5, 7.5).$

.

3.3. Capacity region model

Given a set $\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ of K distinct TMs, with corresponding TMRVs $R_1, \ldots, R_K$, we define their **time division (TD) set** as the L-dimensional set

$$\mathcal{D}(\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}) \triangleq \Big\{ x \in \mathbb{R}^L : 0 \le x \le \sum_{k=1}^{K} a_k R_k, \quad \sum_{k=1}^{K} a_k = 1, \quad a_k \ge 0,\ k = 1, \ldots, K \Big\}.$$

We refer to the elements of $\mathcal{D}(\{\mathcal{T}_1, \ldots, \mathcal{T}_K\})$ as **time division rate vectors (TDRVs)**, and collectively to both TMRVs and TDRVs simply as **rate vectors (RVs)**.

The TD set of the set of TMs $\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$ is an important concept of our methodology because it contains exactly those RVs x that are achievable (in the sense that their components are average link data rates that are simultaneously achievable) if the network is restricted to operate using these TMs.

To prove this property, first, let some vector x belong to $\mathcal{D}(\{\mathcal{T}_1, \ldots, \mathcal{T}_K\})$, for some suitable choice of a vector a. Then, let the network operate under a time division scheme in which each TM $\mathcal{T}_k$ is active for a percentage of time equal to a_k, for $k = 1, \ldots, K$ and, furthermore, all TMs are left continuously active for amounts of time sufficient for their link rates to converge to their maximum values, according to Assumptions 2 and 3, before activating the next TM. Then, the average data rate of each link ℓ, across the complete time division, will be $\sum_{k=1}^{K} a_k R_{k,\ell}$. However, if this time division is maintained but

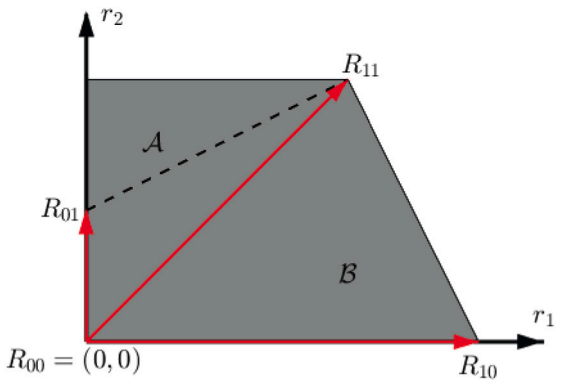


Fig. 2. A simple, two-dimensional CR of a network with only two links and four transmissions modes with rate vectors R_{00}, R_{01}, R_{10}, and R_{11} (subscripts denote which links are active). The CR, computed using (2), is denoted in gray. Note that the RVs R_{01} and R_{00} do not contribute to the CR. As an example of the importance of allowing the underutilization of links, note that the CR can be written as the union of two regions, $\mathcal{A}$ and $\mathcal{B}$, divided by the dashed line. $\mathcal{B}$ is the convex hull of the four RVs and so any point in $\mathcal{B}$ can be achieved by a time division of the respective TMs without an active link being underutilized. The points in $\mathcal{A}$, however, can only be achieved if the TM corresponding to R_{11} is used but its two links are underutilized.

now each link ℓ is underutilized, as Assumption 2 allows, by some appropriate data rates in some of the TMs, its average data rate can become equal to the given value $x_\ell \le \sum_{k=1}^{K} a_k R_{k,\ell}$. Therefore, x is achievable by the network.

Inversely, let some vector x that is achievable if the network is restricted to operate using the set of TMs $\{\mathcal{T}_1, \dots, \mathcal{T}_K\}$. Let $a_k \ge 0$, $k = 1, \dots, K$ be the percentages of time for which the respective TMs are employed to achieve x. As the variables a_k are time percentages, they must sum up to 1. Furthermore, x will be by definition nonnegative and at most $\sum_{k=1}^{K} a_k R_k$, componentwise, as each term of this sum expresses the vector of maximum possible (due to Assumptions 2 and 3) average data rates transferred when each TM $\mathcal{T}_k$ is active for the time percentage a_k. Therefore, x belongs to $\mathcal{D}(\{\mathcal{T}_1, \dots, \mathcal{T}_K\})$.

Finally, the CR of the network is simply defined to be the TD set of *all* TMs:

$$\mathcal{C} \triangleq \mathcal{D}(\{\mathcal{T}_1, \dots, \mathcal{T}_{2^L}\}). \tag{2}$$

To clarify the definition of the CR, in Fig. 2 we present the CR of a toy example network comprising only two links.

We note that this definition of the CR is related to the definitions of the CR used in [6,8], in the sense that in all cases CRs are created by combining the set of elements describing all the distinct ways with which the network can operate. In the case of [6], however, the elements are $n \times n$ *rate matrices*, where n is the number of nodes in the network, and the CR describes the combinations of *source–destination* data rates that are achievable in the network; a similar approach is adopted in [8], although the CR is not specified explicitly there. Here, the elements are *transmitter–receiver* data rates. In computing the CR, this approach is advantageous because the CR has a much smaller dimension (L instead of n^2) and its shape depends on optimal (link) scheduling and not on optimal scheduling *and* routing.

Embedding the CR model in the WNUM problem (1), we have:

Wireless Network Utility Maximization Problem (WNUM) — Explicit Form

$$\begin{aligned} &\text{maximize:} && \textstyle\sum_{n=1}^{N} U_n(s_n), \\ &\text{subject to:} && s = Ax, \quad 0 \le x \le [R_1 \ \dots \ R_{2^L}]a, \quad a \ge 0, \quad a^T \mathbf{1} = 1. \end{aligned} \tag{3}$$

In this form, $\mathbf{1}$ is a vector of size $K \times 1$ whose elements are all equal to 1, the parameters A and $R_1, \dots R_{2^L}$ describe the properties of the network, the functions $U_n(\cdot)$ describe the traffic needs of the nodes, and the parameters s, x, and the new **time division vector** a are subject to optimization.

Note that we have defined the CR in terms of 2^L TMs, so in order to, e.g., solve the WNUM problem in its explicit form, we will have to measure one by one the 2^L corresponding RVs; however, their number is prohibitively large in all but the smallest of networks. We stress, therefore, that this definition is not meant for *direct* application; rather, our methodology aims at finding a very small subset of TMs whose TD set approximates the CR well; this is the topic of the following section.

4. Estimation and utilization of the CR

4.1. Strong TMs

A TM $\mathcal{T}$ with RV $R(\mathcal{T})$ is defined to be **strong** when two checks, each with a corresponding set of (possibly multiple) conditions, are successful. Each check is successful only if all its corresponding conditions are satisfied.

The first check is the **topology check**; its conditions are the **topology conditions**, $\mathbb{C}_{\text{TOP}}$. These conditions only involve information on the network topology, denoted by $\mathbb{N}$, and do not involve measuring the RV $R(\mathcal{T})$, which is a time-consuming process. A variety of choices is possible for these conditions. For example, one or more of the following can be used:

- A node cannot be the transmitter of multiple links in $\mathcal{T}$.
- A node cannot be the receiver of multiple links in $\mathcal{T}$.
- A receiver R in $\mathcal{T}$ and each transmitter in $\mathcal{T}$ other than the one transmitting to it, T, must be separated by a distance at least equal to the distance between R and T.

The second check is the **measurement check**; its conditions are the **measurement conditions**, $\mathbb{C}_{\text{MEA}}$. Checking if these hold requires measuring $R(\mathcal{T})$ in the network, which is much more time-consuming. Again, various choices are possible, depending on the technology used. For example, one or more of the following can be used:

- All data rates of links in $\mathcal{T}$ must exceed a threshold b.
- For each link in $\mathcal{T}$, the product of its data rate and the physical distance between the transmitter and the receiver must exceed a threshold t.

The rational behind these conditions is that we must declare a TM to be strong if all its links can help substantially with the transport of information.

4.2. Identifying strong TMs

Next, we present the routine **strong**$(\cdot)$, which decides if a TM $\mathcal{T}$ is strong or not. Due to the need to make this routine as fast as possible, we make the compromise that it is allowed to make false negative errors, i.e., to declare that an actually strong TM is not strong. (The opposite, i.e., to declare that an actually not strong TM is strong, does not occur.)

In addition to the checks of Section 4.1, two more checks, with corresponding sets of conditions, are used. Each check is successful only if all its corresponding conditions are satisfied.

The first check is the **subset check**; its conditions are the **subset conditions**, $\mathbb{C}_{\text{SUB}}$, and involve the set of all measurements $\mathbb{M}$ that have taken place thus far. The idea is to glean information about the suitability of $\mathcal{T}$ based on the suitability of its subsets for which there are measurements in $\mathbb{M}$. As with the other sets of conditions, various choices exist. For example, let $\mathbb{T}$ be the set of all TMs measured thus far and found to be strong and the set $\mathbb{T}^c$ of all TMs measured thus far and found not to be strong. Then, one of the following two conditions may be used.

- $\mathcal{T}$ passes the subset check if all its subsets of size $|\mathcal{T}|-1$, i.e., those subsets created by removing a single link from $\mathcal{T}$, belong in $\mathbb{T}$.
- $\mathcal{T}$ passes the subset check unless any of its subsets of size $|\mathcal{T}|-1$ belongs in $\mathbb{T}^c$.

Observe that the second condition is a weaker version of the first. The rationale behind both these conditions is that we expect that the subsets of strong TMs must also be strong, for any reasonable choice of the other sets of conditions.

The second check is the **regression check**; its conditions are the **regression conditions**, $\mathbb{C}_{\text{REG}}$, and involve an online machine learning module that has been trained using the aforementioned set $\mathbb{M}$ in order to provide an estimate of the RV $R(\mathcal{T})$ of $\mathcal{T}$ by doing regression. Different choices exist, first on how to build the ML module and second on how to utilize the outcome of the regression. Regarding the first issue, the approach we adopted in the testbed is discussed in Section 5. Regarding the second issue, one of the following two conditions could be used, assuming there is only a single measurement condition (Section 4.1) that all rates of links in the TM must exceed a threshold b:

- $\mathcal{T}$ passes the check if all the rates of its links are estimated by the ML module to be greater than the above threshold value b.
- Water filling: $\mathcal{T}$ passes the check if the estimated rates of its links can all become greater than the above threshold value b if they are increased, each by some different amount, such that the sum of the increases does not exceed the product $K\mathcal{E}_{\text{RMS}}$ of an **error factor** K, which is a tunable parameter, and the root-mean-square (RMS) error $\mathcal{E}_{\text{RMS}}$ between past link data rate measurements and their respective estimations right before each measurement was taken; the RMS error is calculated using the last W_{RMS} RV measurements; however, if there have been fewer than W_{RMS} RV measurements, we lack confidence in the current estimate provided, so we set $\mathcal{E}_{\text{RMS}}$ to a very large value and the check is automatically successful.

Observe that the first choice is identical to the second choice for the special case when $K = 0$. The rationale behind these conditions is that a measurement should be avoided if, based on regression, we are relatively confident that the TM is not strong.

In the following, we will collectively refer to the topology, measurement, subset and regression checks as **strength checks**, and their conditions as the **strength conditions**.

We now specify the operation of the routine **strong**$(\cdot)$. Given an input TM $\mathcal{T}$, the routine checks if the TM passes the topology conditions; we denote the respective subroutine with **check**$(\mathcal{T}, \mathbb{C}_{\text{TOP}}, \mathbb{N})$. If the check is successful, the routine checks if the TM passes the subset conditions; we denote the respective subroutine with **check**$(\mathcal{T}, \mathbb{C}_{\text{SUB}}, \mathbb{M})$. If that check is also successful, then the routine checks if the TM passes the regression conditions; we denote the respective subroutine with **check**$(\mathcal{T}, \mathbb{C}_{\text{REG}}, \mathbb{M})$. If that check is successful as well, then we measure the RV $R(\mathcal{T})$ of the TM in the deployed network; we denote the respective subroutine with **measure**$(\mathcal{T})$. Finally, we check if the TM passes the measurement check; we denote the respective subroutine with **check**$(\mathbb{C}_{\text{MEA}}, R(\mathcal{T}))$. If that check is also successful, the TM is declared strong. The pseudocode of the routine is Pseudocode 1.

Pseudocode 1: routine **strong**$(\cdot)$

```
Input: T, C_TOP, C_MEA, C_SUB, C_REG, N, M
Output: boolean strong, M
1  begin
2      strong=false;
3      if check(T, C_TOP, N) AND check(T, C_SUB, M) AND
         check(T, C_REG, M) then
4          R(T) = measure(T);
5          M = M ∪ {R(T)};
6          if check(C_MEA, R(T)) then
7              strong=true;
8          end
9      end
10 end
```

Pseudocode 2: Sequential Expansion Algorithm

```
Input: T, C_TOP, C_MEA, C_SUB, C_REG, N, M
Output: S_ℓ, ℓ = 1, ..., L.
1  Set S_ℓ = ∅ for all ℓ = 2 ... L
2  Set S_1 the set of all single-link TMs {ℓ} for which strong({ℓ})
     is true
3  Set ℓ = 1
4  while ℓ < L and S_ℓ ≠ ∅ do
5      for T_test ∈ combine(S_1, S_ℓ) do
6          if strong(T_test) then
7              S_{ℓ+1} = S_{ℓ+1} ∪ {T_test};
8          end
9      end
10     ℓ = ℓ + 1;
11 end
```

4.3. Sequential expansion algorithm

The **strong**$(\cdot)$ routine declares a TM to be strong or not. As checking if all 2^L TMs are strong is impractical, in this section we present the **Sequential Expansion Algorithm (SEA)**, which judiciously decides which TMs to check.

The algorithm discovers in sequence progressively larger (i.e., with more links) strong TMs by adding links to already discovered smaller strong TMs and checking if the resulting TM is strong. The justification of this approach is that we expect that subsets of strong TMs should also be strong, therefore supersets of TMs that are not strong need not be considered.

The pseudocode of the algorithm is Pseudocode 2. The input of the algorithm is the same as the **strong**$(\cdot)$ routine. The output of the algorithm are the sets $\mathcal{S}_\ell$, $\ell = 1, \ldots, L$, of the discovered strong TMs of size ℓ. The algorithm makes use of the **strong**$(\cdot)$ routine; in the interest of succinctness, we do not list that routine's input variables except the TM.

The algorithm starts with an initialization (line 1), followed by a preliminary part (line 2), where all strong TMs comprising a single link are discovered.

The main loop (lines 4–11) uses, in line 5, the routine **combine**$(\mathcal{S}_1, \mathcal{S}_\ell)$, which is input the set $\mathcal{S}_1$ of strong TMs of length 1 and the set $\mathcal{S}_\ell$ of strong TMs of length ℓ and returns the set of all distinct TMs of size $\ell+1$ (including those that are not strong) that are comprised of one link from $\mathcal{S}_1$ and the links of one TM from $\mathcal{S}_\ell$. Note that no link can appear multiple times in a TM.

Regarding the main loop's operation, in its first iteration, when $\ell = 1$, all single-link TMs of $\mathcal{S}_1$ are combined pairwise and the resulting 2-link TMs are added to $\mathcal{S}_2$ if found to be strong. More generally, in the ℓ-th iteration the TMs in $\mathcal{S}_\ell$, which were found to be strong in the $(\ell-1)$-th iteration, are combined with the TMs in $\mathcal{S}_1$ to produce $\mathcal{S}_{\ell+1}$. The process continues until, in the check of line 4, no strong TMs were discovered in the previous iteration ($\mathcal{S}_\ell = \emptyset$) or the maximum size of TMs has been reached ($\ell = L$).

Having executed SEA in a deployed network, we arrive at a set of K strong TMs, $\cup_{l=1}^{L} \mathcal{S}_l = \{\mathcal{T}_1^{\text{SEA}}, \ldots, \mathcal{T}_K^{\text{SEA}}\}$. Let $R_1^{\text{SEA}}, \ldots, R_K^{\text{SEA}}$ their respective RVs. Their time division set is a subset of the CR:

$$\mathcal{C}_{\text{SEA}} \triangleq \mathcal{D}(\{\mathcal{T}_1^{\text{SEA}}, \ldots, \mathcal{T}_K^{\text{SEA}}\}) \subseteq \mathcal{C}.$$

However, due to the manner the set $\mathcal{T}_1^{\text{SEA}}, \ldots, \mathcal{T}_K^{\text{SEA}}$ was compiled by SEA, we expect that C_{SEA} captures enough of the CR so that it can be used in its place, when solving the WNUM problem, even if $K \ll 2^L$, resulting in the following, constrained form of the WNUM problem:

Wireless Network Utility Maximization Problem (WNUM) - Constrained Form

$$\begin{aligned} &\text{maximize:} && \textstyle\sum_{n=1}^{N} U_n(s_n), \\ &\text{subject to:} && s = Ax, \quad 0 \le x \le [R_1^{\text{SEA}} \ \ldots \ R_K^{\text{SEA}}]a, \quad a \ge 0, \quad a^T\mathbf{1} = 1. \end{aligned} \tag{4}$$

In this optimization problem, the functions $U_n(\cdot)$ capture the traffic requirements of the network's users, the adjacency matrix A and the rate vectors $R_1, R_2, \ldots, R_K$ describe the network, and the vectors x, a and a are subject to optimization. The optimal values x^{OPT}, s^{OPT}, and a^{OPT}, which may not be unique, are arrived at jointly, by centrally solving the problem by the optimization module placed in the digital twin. If the utility functions are convex, which is a common assumption, the problem is convex. Furthermore, the optimization problem is separable (i.e., the objective function is a sum of functions of individual optimization variables) and therefore especially amenable to algorithms employing duality [38]. In our experiments, the shapes of the utility functions are such that the problem becomes linear.

4.4. Network-wide time division protocol

Solving the WNUM problem, in either its explicit or constrained form, we find the optimal value of the objective as well as the associated optimal traffic flow vector x^{OPT}, divergence vector s^{OPT}, and time division vector a^{OPT}. These provide benchmarks that we can use to evaluate the performance of practical protocols when they are tasked with achieving the same objectives.

Furthermore, the optimal vectors can be utilized directly, following a digital twin approach, under which the PT comprises the nodes, the channel, and the network controller, and the DT comprises SEA, the CR estimate, and an optimization module.

Under this approach, firstly, the DT executes the SEA, issuing requests to the network controller to initiate measurements for specific TMs and processing the responses, so as to obtain a CR estimate. Secondly, the PT and DT together execute the following novel, CR-inspired **Network-Wide Time Division (NWTD)** protocol: The network controller collects the traffic requirements of the network and sends them to the Digital Twin, which, using the CR estimate and the optimization module, solves the constrained form of the WNUM problem to arrive at an optimal time division vector a^{OPT} which it passes back to the network controller. The network controller then instructs the nodes to implement the optimal vectors, using a sequence of identical frames, each frame comprising a set of slots, each slot allocated to a single TM, such that each TM is active for a percentage of time prescribed by the respective component of a^{OPT}; if the vector $[R_1 \ \ldots \ R_K]a^{\text{OPT}} - x^{\text{OPT}}$ has positive components, their respective links should be underutilized, as Assumption 2 allows, to the extent prescribed by these components.

One complication of using the prescribed time division schedule is that it is possible that during some of the initial frames there will be slots when nodes will be required to transmit data that they have not already received. The solution is for the affected nodes to transmit dummy data whenever this occurs, until they buffer a sufficient volume of actual data; the resulting effects on the performance will be transient, and the long-term data rates will not affected.

NWTD is conceptually straightforward to describe and achieves the optimal performance when the CR is used or, when a CR estimate is used instead, the best possible performance based on that estimate. In most cases, however, it will not be practicable because *(i)* of the overhead required to determine the CR (estimate), *(ii)* it is centralized, *(iii)* when the network topology changes, due to mobility, fading, or other sources of variations, the CR (estimate) will be obsolete or must be computed anew, and *(iv)* when the traffic requirements change, the time division must also be calculated anew. However we put it forward as a modest starting point for the development of practicable CR-aware, protocols following the Digital Twin approach.

5. Experiments

5.1. The testbed

We use Raspberry Pi 4 Model B single-board computers [39] equipped with 4 GB of RAM, the standard internal IEEE 802.11ac Network Interface Card (NIC) and an external IEEE 802.11a/n/ac NIC. Nodes run the Raspberry Pi OS (which is based on Debian) with the 5.4.83-v7l+ kernel version, which comes with the brcmfmac wireless driver [40] for supporting the internal NIC; however the external NIC is supported by a proprietary wireless driver. All the used hardware is off-the-shelf, making the testbed and experiments easy to replicate.

The nodes are placed inside three different floors of a building of the Athens University of Economics and Business (AUEB) containing faculty offices and labs (other floors also contain classrooms). Placing the nodes on multiple floors creates a three-dimensional, hence richer, topology. Each node is placed in a permanent position in one of the offices, and is continuously connected to a power outlet. Experiments run at times when the building is known to be empty or almost empty of life (i.e., during the night), in order to minimize variations in the channel and interference by other devices.

Regarding connectivity, each node is controlled by communicating with an on-site network controller over the **control plane** and forms the experimental wireless network with the rest of the nodes over the **data plane**. In our testbed the two planes are implemented using different NICs.

Regarding the control plane, the nodes use their external NICs and the preexisting WiFi network of the building, which comprises multiple access points and uses both the 2.4 GHz and the 5 GHz band, to receive instructions by the network controller at times when there are no transmissions in the data plane. (In any case, an effort was also made for the two planes to use non-overlapping channels, subject to the constraints imposed by the operation of the preexisting WiFi network in the building, which was not under our control.) The controller enables the testbed operators to maintain and monitor the correct operation of all, main and auxiliary, software elements of the nodes, specifies the operation of the nodes in the data plane according to the instructions of SEA or the testbed operators, and collects measurements which it feeds back to them.

Regarding the data plane, it comprises the nodes' internal NICs running in ad hoc mode at a 2.4 GHz channel, with all settings at their default values (in particular, nodes do not exchange RTS/CTS packets), unless otherwise mentioned. During the execution of the experiments the wireless configuration utilities of Raspbian (iwconfig, iwlist, and wl) report that "IEEE 802.11" is enabled with "unknown bit-rate information" in "Ad-hoc" mode, and the nodes select automatically the MCS, according to their perceived radio quality and firmware.

Regarding the data plane channel, we selected a channel in the 2.4 GHz band as opposed to the 5 GHz band because, firstly, the resulting topology in the 5 GHz band was sparser and hence easier to optimize on and, secondly, because, as reported in [11], data rates in the 5 GHz band were less volatile. Therefore, in this sense the 2.4 GHz band represents more interesting conditions.

Here we report results for two experimental network topologies, both shown in Fig. 3. The first network topology comprises 6 nodes, and the second one 16 nodes. In the figure, nodes with different shades are in different floors (the lighter the shade, the higher the floor).

In the 6-node network topology we only use the 14 links between 7 node pairs; these pairs are connected, in the figure, with lines. There are

other links, but these are significantly weaker and nodes are hard-wired to not use them.

In contrast, in the 16-node network all existing links are available for use; node pairs for which at least one of the two links between them is able to achieve data rates greater than 1 Mbps are connected, in the figure, with lines. To give a sense of how well-connected the 16 nodes are, we note that each of the 16 nodes has 15 potential direct destinations, therefore there are $16 \times 15 = 240$ potential transmitter–receiver pairs; of these, when multiple links do not compete for the medium, around 50 are able to communicate directly with a rate exceeding 10 Mbps, around another 50 communicate directly with a rate between 1 Mbps and 10 Mbps, around another 20 are able to exchange some packets directly with a rate not exceeding 1 Mbps, and the rest do not communicate directly at all. These numbers are approximate due to the variability of the measurements, especially of weak links; this variability is discussed later on.

An important point must now be made on what we define to be an active link and an active TM in our testbed. Taking into account the off-the-shelf components that we use, we do not define a link to be active only when it is actively transmitting; rather we define the link to be active when it is transmitting *or* it is actively contending for immediate access to the medium, following the rules of the IEEE 802.11 protocol. Suppose, for example, that there is a total of three links, ℓ_1, ℓ_2, ℓ_3 that have packets awaiting transmission and wish to transmit them as soon as possible, following instructions by the network controller or the testbed operators. The links will share the channel as prescribed by the IEEE 802.11 protocol, taking turns in transmitting, or transmitting simultaneously, if the topology permits, until one of them empties its queue or is instructed to stay off the contention for the channel; during the *whole* duration of this contention period, where the nodes are actively competing for access to the medium, according to the IEEE 802.11 protocol, the network remains in *one* TM, $\mathcal{T} = \{\ell_1, \ell_2, \ell_3\}$.

Regarding the validity of the assumptions of Section 3.2, note that we cannot formally prove mathematically that they hold, as our setting is experimental as opposed to mathematical; it will have to suffice to show that they are reasonable. To that effect:

1. Assumption 1 is reasonable, as the network controller, or the nodes individually (if there is no network controller) can specify which links are active at any time, in the sense described above, but the hardware does not give them other degrees of freedom.
2. Assumption 2 is reasonable, as we expect that if a TM is used continuously the data rates achieved by the links will converge to some fixed values, and nodes can pad transmissions with garbage bits to transmit anything from 0 to these maximum values.
3. Assumption 3 is reasonable, as we expect the network to be operating at least as efficiently during a continuous use of a specific TM $\mathcal{T}$ as when there is a sequence of transient activations of $\mathcal{T}$, interspersed with the activations of other TMs.

A central aspect of our experiments is the measurement of the RVs of specific TMs (i.e., line 4 of the **strong**$(\cdot)$ routine). To perform this measurement, the on-site server establishes independent single-hop TCP connections in the corresponding links using the Iperf tool [41]. When the connections are over, we record the volume of traffic transported through each link and, dividing by the duration of the measurement, we arrive at the RV of that TM.

In the following, when comparing the results of different experiments, all results were derived using the same protocol settings (e.g., the RTS/CTS mechanism was always on or always off), unless explicitly mentioned otherwise.

As a preliminary result, we note that we have consistently registered significant amounts of volatility of the measured RV data rates. As an indicative example, in Fig. 4 we plot the average and the standard deviation, over 10 experiments, of the data rates achieved by the links

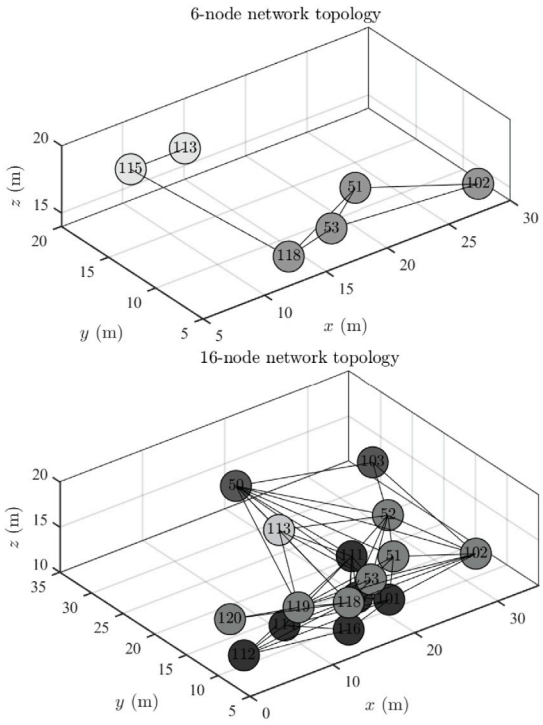


Fig. 3. The two experimental networks.

of two TMs of the 16-node network topology versus the duration of the measurements. One TM comprises three links (each link interfering with the other two) and the other TM comprises two, also interfering, links. In the legends, $x \to y$ signifies the link for which the node with id x is transmitting to node with id y. For the second TM, the sum of the two data rates is also plotted. Observe that there is significant variability of the measured data rates, which does not diminish perceptibly with the duration of the measurement.

5.2. Evaluating the accuracy/complexity tradeoff

In the first experiment, we evaluate how good a tradeoff the Sequential Expansion Algorithm achieves between estimating the CR accurately and avoiding an excessive amount of measurements. To this effect, we use the 6-node, 14-link network. The number of TMs in this network is 2^{14} which is relatively manageable in the context of this investigation. We execute the algorithm, for various choices of the topology and measurement conditions, and in each case record how much of the CR is discovered and how many measurements were needed.

In order to expedite the experiment, we adopt the following method: first, we measure the RVs of *all* TMs beforehand and store them. Then, we execute the stepwise expansion algorithm for various choices of parameters, and whenever a measurement is needed, we use the respective stored measurement. Using this approach, and a duration of measurements equal to 10 s, all measurements were completed in about 70 h. (We stress that the application of our method in deployed networks does not require this volume of measurements.)

As the CR of the network and its estimates are multidimensional, we need to use a conceptually simpler and easy to calculate figure of merit that captures its size. To this effect, we define the **aggregate data rate**

$$T_{\text{agg}}(\mathcal{T}_1, \ldots, \mathcal{T}_K) = \sum_{i=1}^{N} \sum_{j=1,\ j\neq i}^{N} T_{ij}(\mathcal{T}_1, \ldots, \mathcal{T}_K), \tag{5}$$

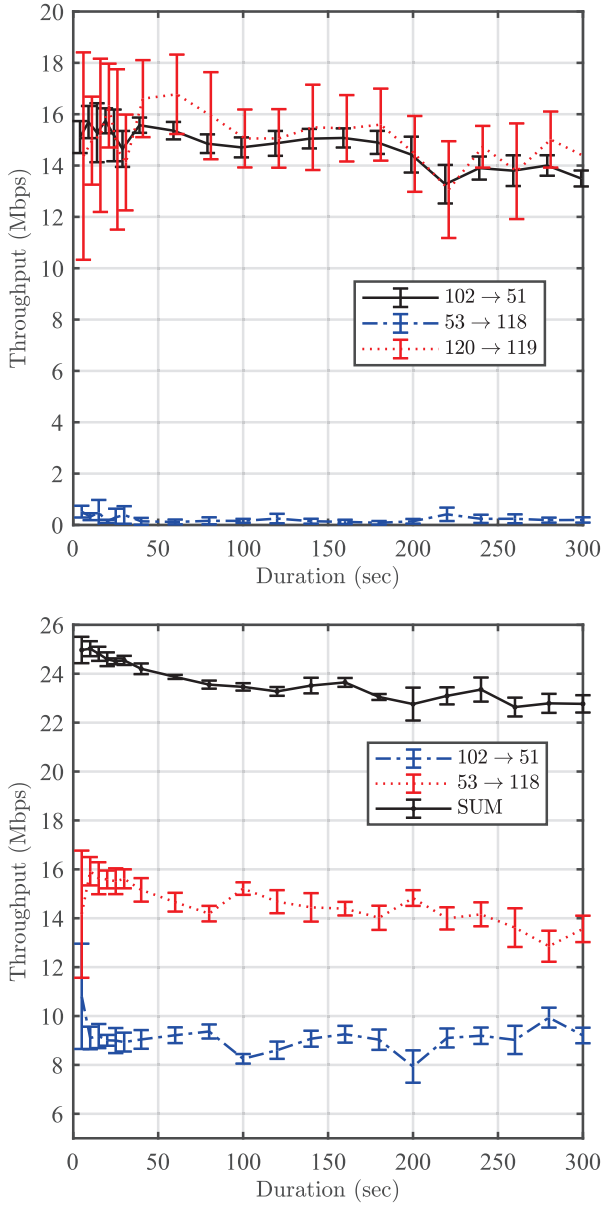


Fig. 4. Throughput versus duration for two indicative TMs.

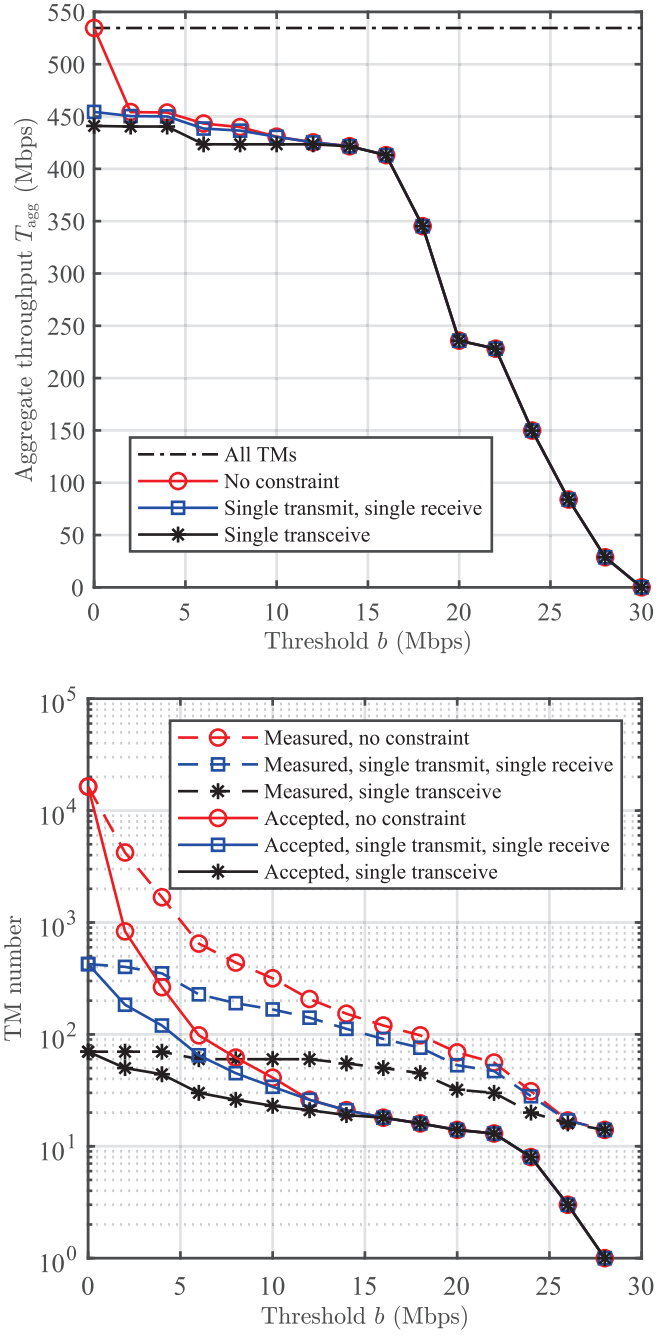


Fig. 5. Aggregate data rate T_{agg} (top) and Numbers of measured/accepted TMs versus b (bottom) versus b and various choices of the topology conditions.

where the **end-to-end** data rate $T_{ij}(\mathcal{T}_1, \ldots, \mathcal{T}_K)$ is the maximum data rate achieved when node i sends data traffic to node j, there is no other traffic need in the network, and the network operates using an optimal time division comprised of TMs in the set $\{\mathcal{T}_1, \ldots, \mathcal{T}_K\}$, as prescribed by solving the constrained WNUM Problem with

$$U_j(s_j) = -s_j, \quad U_i(s_i) = 0, \qquad U_k(s_k) = \begin{cases} 0, & s_k = 0, \\ -\infty, & s_k \neq 0, \end{cases} \qquad k \neq i, j.$$

The shape of the utility functions specifies that node j should remove from the network as much traffic as possible, any node k with $k \neq i, j$ is not allowed to either insert or remove traffic from the network, and the amount of traffic either being inserted in or removed from the network through node i is irrelevant. Jointly, these utility functions specify that node j removes the maximum amount of traffic that can be inserted, necessarily, at node i. Observe that even if we set $U_i(s_i) = s_i$, or any other finite, increasing function of s_i for that matter, the optimal flow will not change, but only the optimal utility.

Regarding the topology conditions, we consider three alternative cases:

1. **No constraint**: There are no topology conditions.
2. **Single transmit, single receive**: Each node in the TM must receive from at most one other node and transmit to at most one other node.
3. **Single transceiver**: Each node in the TM must communicate, either as a transmitter or a receiver, with at most one other node.

Regarding the subset and regression conditions, we do not use any, for this experiment. Finally, regarding the measurement conditions, in all cases we require that a strong TM must have the rates of all its links be greater than a tunable threshold b.

In Fig. 5 we plot, on top, the aggregate data rate and, below, the number of measured TM and the TMs that were accepted as strong by the Sequential Expansion Algorithm as a function of the parameter b and for the three aforementioned cases of topology conditions. The measured TMs were those for which the **measure**$(\cdot)$ routine was invoked and a time-consuming measurement was executed in the network. It is crucial, therefore, to minimize their number.

The figures reveal a very promising tradeoff between that portion of the CR revealed (as quantified by T_{agg}) and the number of measurements required. For example, adopting the last topology condition and $b = 10$ Mbps, we arrive at $T_{\text{agg}} = 423$ Mbps which is 79% of the optimum value $T_{\text{agg}} = 535$ Mbps, by measuring only 60 TMs, which is 0.37% of all 2^{14} TMs.

The results of this experiment suggest that it is the strong links that form the bulk of the CR, and therefore if we had included all links in the network, and not the 14 strongest ones, the CR would not have changed materially. Also, as there is no reason for strong links to be less important in larger networks than in small ones, we anticipate that the tradeoff achieved in larger networks will be as favorable as the one exhibited here.

5.3. Evaluating the machine learning module

In our second experiment we evaluate how well our online ML module performs at providing estimates of RVs. To this effect, we perform a single execution of the stepwise expansion algorithm on the 6-node network with the strength conditions chosen so that all examined TMs pass all checks (therefore the ML module has the opportunity to train itself on all 2^{14} TMs) and report on how well the RV estimates and the respective RV measurements are aligned.

Regarding the implementation of the ML module, each link in the network is assigned a single feedforward, fully connected neural network (with 10 layers, the Rectified Linear Unit activation function, standardized input, with zero biases and no regularization term) which can be trained so that it can estimate the value of the data rate of that link if it is given an input TM that includes it. The training is based on the complete set of past RV measurements; however, in order to reduce the volume of computations needed, we retrain each such network whenever we receive 50 new measurements involving its respective link.

In order to have a basis for comparison, we also execute the stepwise expansion algorithm, with the exact same choice of conditions, and measure the same performance, but for two other networks as well.

The first is a **model network**, whose nodes are placed in the locations of the nodes in the experimental network, and for which the rate of a link is calculated using a simple Signal-to-Interference-and-Noise (SINR) model that allows for a node to be transmitting to multiple receivers and receiving from multiple transmitters at the same time (as is the case with the experimental network). We set the parameters of the model network so that the sum of all components of all distinct 2^{14} RVs is the same as the respective sum, denoted by $S_{\text{Exp}-5}$, of the experimental network, in the case of the Exp-5 measurement set defined below.

The second network is a **random network**, whose links are the same as those of the experimental network, but whose link data rates for each TM are chosen randomly and uniformly, and independently of other TMs, such that the expectation of the sum of all components of all distinct 2^{14} RVs is also equal to the aforementioned sum $S_{\text{Exp}-5}$.

The two networks represent opposite extremes: the model one should be very predictable, whereas the random one should be very unpredictable.

In Fig. 6 we plot the average (taken over a running window of size 300) RMS error between each measured data rate and its respective estimate, for each of the three networks. As in Section 5.2, in the case of the experimental network, we perform all measurements beforehand. Furthermore, three sets of measurements are used: a set of measurements with a duration of 5 seconds (Exp-5 set), a set of measurements with a duration of 10 seconds (Exp-10 set), and finally a set of composite measurements with a duration of 15 seconds created by taking the weighted averages of the rates of the first two measurements (Exp-15 set). In the plot, we normalize the average RMS errors so that at the first measurement it is equal to unity for all plots.

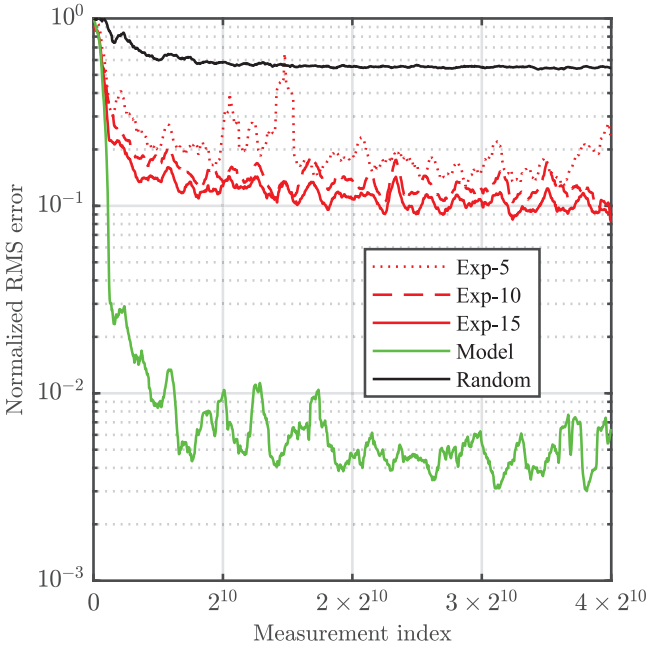


Fig. 6. RMS error versus measurement.

A number of comments can be made on the plot. Firstly, as expected, the RMS error is the smallest in the case of the model network, and in fact reduces by more than 2 orders of magnitude during the training. (This reduction incidentally provides validation that the neural networks used can indeed be trained.) Secondly, the RMS error is largest in the case of the random network. Indeed, the best strategy in this case is to select the estimate of each (randomly generated) link data rate to be its expected value; the neural network is quickly trained to follow this strategy, at which point the RMS error stops decreasing. Thirdly, we can see that the RMS error curves of the experimental network lie between the two other curves. This is also expected, as the underlying network on which the measurements are made indeed has a structure, albeit much more complex, and in addition the measurements exhibit a level of randomness (cf. Fig. 4). Finally, note that in the case of the experimental network the RMS error also decreases as the duration of the measurements increases and, consequently, their random fluctuations are reduced.

5.4. Capacity region estimations

We now present two experiments in which the CR of the 16-node network topology of Fig. 3 is estimated using the Sequential Expansion Algorithm; the two experiments differ in the set of strength conditions used. In contrast to the 6-node, 14-link network studied in Sections 5.2 and 5.3, we do not know the actual capacity of the network, due to the large number of links involved. Also, experiments run with the RTS/CTS handshake activated.

In order to expedite the execution of the experiments, the following scheme is adopted, which is somehow similar to the preemptive measurement of all TMs employed in Section 5.2: before issuing a command for a TM to be measured, if the measurement is already available from past experiments that were executed under identical

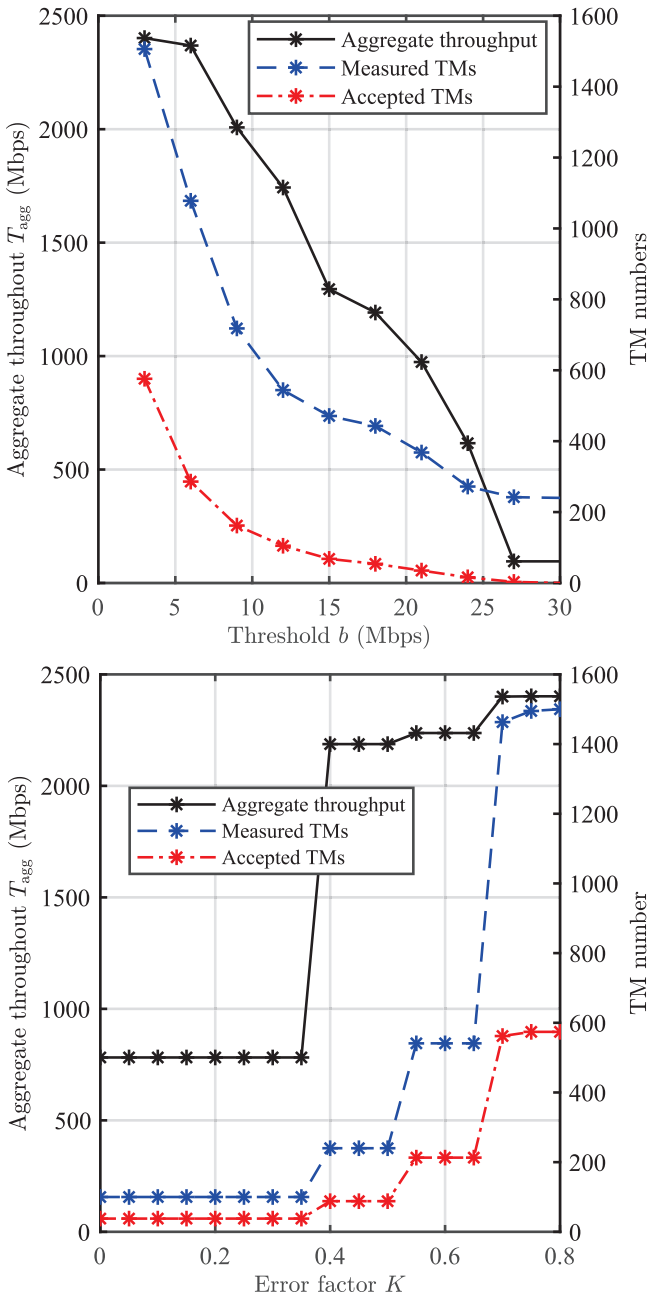


Fig. 7. Aggregate data rate and number of measured/accepted nodes versus the threshold b (left) and the error factor K (right).

conditions, then the past measurement is used instead. This scheme allows us, on one hand, to execute the algorithm with a much wider range of parameters in the same amount of time and, on the other hand, to continue experiments that were interrupted (due to power outages, software failures, etc.) right at the point where they were interrupted.

In our first experiment, we run the algorithm using the following set of strength conditions:

- Topology conditions: we require each node to participate in at most one link and that each receiver does not have an interfering transmitter closer to it that its own transmitter.
- Subset conditions: we require that the check succeeds only if all subsets of the TM under investigation that are created by removing a single link have been measured and were found to be strong.
- Regression conditions: the regression check is set to be always successful.
- Measurement conditions: a node is strong if all the rates of its active links exceed a tunable threshold b.

In Fig. 7 (left) we plot the aggregate data rate, T_{agg} (left y-axis) and the measured and accepted TM numbers (right y-axis) versus the threshold b. It is interesting to note that we can recover a large part of the CR discovered for $b = 3$ Mbps (for which $T_{agg} = 2402$ Mbps) for the relatively large value $b = 12$ Mbps (for which $T_{agg} = 1743$ Mbps, i.e., around 72% of the previous value), with 544 measurements, even in this case where no ML is used. Based on our study of the 6-node network we anticipate that the CR discovered for $b = 3$ Mbps is a large part of the whole CR.

Using the online ML module described in Section 5.3 can also be advantageous. For our second experiment, we run the Stepwise Expansion Algorithm using the following choice of conditions:

- Topology and subset conditions: as with previous experiment.
- Regression conditions: The water-filling condition of Section 4.2 with $W_{RMS} = 100$. Also, we retrain the neural network whenever we receive 10 new measurements, and vary the value of the error factor K from 0 to 0.8.
- Measurement conditions: a node is strong if all the rates of its active links exceed a threshold $b = 3$ Mbps.

In Fig. 7 (right) we plot the aggregate data rate (left y-axis) and the measured and accepted TM numbers (right y-axis) versus the error factor K. Observe that if we select, say, $K = 0.45$, we compute an aggregate data rate $T_{agg} = 2187$ Mbps, which is equal to around 91% of the data rate $T_{agg} = 2402$ Mbps when the ML check is set to be always successful, as determined by the previous experiment. However, for $K = 0.45$, we only perform 240 measurements, which is 16% of the 1506 measurements executed if the ML module is always successful. In other words, by using ML we reduce the search space by 84% with only a 9% performance penalty with respect to the baseline when ML is not utilized.

5.5. Performance evaluation of OLSR and NWTD

In our final experiment, we show an example of how one can utilize CR estimates in order to evaluate in absolute terms the performance of known protocols, and also enable novel ones. The known protocol we investigate is OLSR [42], one of the most popular routing protocols for use in wireless networks. We employ the implementation of OLSR that is available through the official repository of the Raspberry Pi OS and is compatible with the ARM architecture of the Raspberry Pi Model 4 B. The novel protocol is NWTD, introduced in Section 4.4.

Regarding the implementation of NWTD, the identical frames comprise sets of slots of equal duration; as the number of slots in a frame must be finite, the optimal time division vector a^{OPT} is first rounded to multiples of the time fraction corresponding to one slot, and then the allocation of TMs to specific slots is made in a round-robin fashion (for the TMs), starting from the TMs that must be allocated to more slots (with ties broken randomly), and skipping in the round robin the TMs that have already been allocated their total number of slots in the frame.

Furthermore, NWTD uses the following three protocol optimizations, all motivated by the aforementioned variability of the measurements (cf. Fig. 4). Firstly, given an optimal time division a^{OPT}, if a proper subset $\mathcal{T}'$ of a TM $\mathcal{T}$ that appears in a^{OPT} is used in the time division instead of $\mathcal{T}$, but using for the $l(\mathcal{T}')$ link rates of its RV $R(\mathcal{T}')$ the corresponding $l(\mathcal{T}')$ link rates appearing in $R(\mathcal{T})$ (note that $R(\mathcal{T})$ has a few link rates more), and the objective is found not to decrease, then $\mathcal{T}'$ is used instead of $\mathcal{T}$ when creating the slot schedule. (This modification is useful because, due to measurement errors, when estimating the CR, there are occasionally TMs $\mathcal{T}$ whose performance in

a proper subset $\mathcal{T}'$ of their links is erroneously found to be superior to the performance of the TM $\mathcal{T}'$.) Secondly, at the start of a slot where a link is scheduled for activation, we only activate it if the buffer of its transmitter is not empty and, furthermore, its receiver is the final destination of the data packets or the buffer of the receiver does not already have a data volume exceeding a predetermined threshold $B^{\max}$. (The last condition ensures that data packets do not accumulate at intermediate nodes at the expense of other, starved nodes.) Thirdly, the slot stops either at its prescribed duration or sooner, whenever *any* of the transmitter buffers empties.

To simplify the design of the experimental testbed, in order to measure the performance of a specific time division schedule, we execute the TM of each slot and measure the data rates achieved, and then use this measurement to *emulate* the transport of data with a simple routine. Furthermore, if another measurement of the same TM, taken under identical conditions, has already been taken, and not used in any of the frames of the current time division schedule, that measurement is used instead. This approach significantly simplifies the design of the experiments and, since it allows using the same measurement in different time division schedules, significantly accelerates their execution as well.

In producing the following results, we used the strength conditions with which the CR estimate of Fig. 7 with $b = 3$ Mbps was computed, except that we used the more lax subset condition that the subset check only fails if a subset of the checked TM was found not to be strong. Also, the RTS/CTS handshake was activated. (We mention that we also performed the experiment without using RTS/CTS handshakes and the overall trends were similar, but details are omitted.) We set $B^{\max} = 112.5$ MB, in each frame there were 20 slots with a duration of 15 seconds, and 5 frames were emulated.

In Fig. 8 we compare the performance of OLSR and NWTD in terms of a scatter plot. Each point in the scatter plot corresponds to one among the $16 \times 15 = 240$ source–destination (SD) pairs. Its x-coordinate is the ratio of the average performance of OLSR (calculated using 50 runs for each SD pair) divided by the optimal data rate for that SD pair (which is calculated using the CR estimate). Its y-coordinate is the performance of NWTD, again divided by the optimal data rate. Therefore, we compare the two protocols between them and also with our estimate of the theoretical optimum. The points are circles when it is optimal, based on the CR estimate, for the SD pair to exclusively communicate directly, diamonds when it is optimal for them to communicate exclusively using a single other relaying node, and a square in all other cases. Observe that in a few cases the x and y coordinates of points exceed unity. This is due to measurement errors when estimating the CR.

A number of interesting facts emerge from these experiments. First, our NWTD scheme performs overall better, as exhibited by the fact that most scatter points are above the $y = x$ line. Indeed, the aggregate optimal data rate (as defined in Eq. (5)) is 2413 Mbps, however, the aggregate data rate over all SD pairs using OLSR is only 1349 Mbps, or 56% of the optimal, and the aggregate data rate over all SD pairs using NWTD is 1805 Mbps, which is 75% of the optimal and an improvement of 34% with respect to OLSR.

Second, although not reflected in the figure, the performance of OLSR is not consistent: the data rate of OLSR averaged over the 50 runs and the 240 SD pairs was 5.619 Mbps, whereas the average of the 240 standard deviations of the SD pair data rates was 2.236 Mbps. Our understanding is that the volatility of the physical layer eventually creeps in the network layer as well, producing oscillations on the routing tables and occasional network-wide failures.

Third, the performance of OLSR is significantly below the optimal. One reason is that OLSR is content to find *one* route as opposed to the *best one*. Another, related, reason is that when OLSR is used, suboptimal combinations of links, that exhibit significant interference, are used.

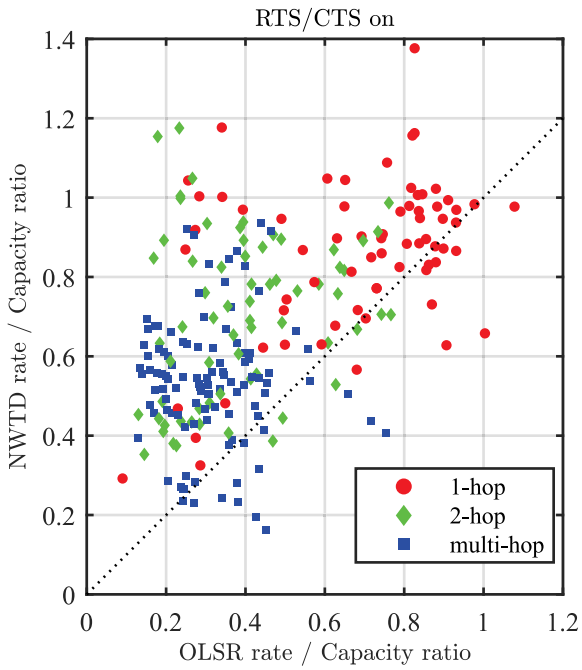


Fig. 8. Experimental network of 16 nodes.

Fourth, the performance of the NWTD protocol is also significantly below the capacity. This is caused, again, by the volatility of the channel. Indeed, in many cases the NWTD protocol aggressively uses in its time divisions TMs whose performance was overestimated in the CR estimation phase, resulting in poor performance. Nevertheless, the improvement of 33% with respect to OLSR is promising.

Finally, observe that both OLSR and NWTD suffer in the more challenging cases of the SD pairs corresponding to the blue squares. This is expected, as these SD pairs are separated by greater distances and involve more intermediate nodes.

6. Discussion: Limitations and extensions

Our work has a number of important constraints. Firstly, and crucially, estimating the CR is a time-consuming process. Therefore CRs cannot be estimated and utilized in cases where there is significant non-periodic mobility. The work is applicable, however, in the case where there is little mobility, as is the case in mesh networks, and also where there is periodicity, which allows measurements to remain pertinent. In addition, measuring the CR of a network in the low-mobility regime might provide information on, e.g., the properties of efficient routes, that is useful in the high-mobility regime.

Secondly, even when mobility is not an issue, we can only estimate, and not accurately measure, the CR, due to the following sources of error:

1. The assumptions of Section 3.2 might not hold perfectly.
2. We use some of the strong TMs, to construct the CR, instead of all TMs.
3. Any TM measurement error affects the shape of the estimated CR.

The seriousness of each of these sources of errors will depend on the precise platform on which our methodology will be applied. In the case of our platform, we conjecture that the effects of the first two sources were very modest, but the effects of the last one were important, and this source was the primary reason our NWTD protocol achieved only 75% of the theoretical maximum aggregate data rate and also the

normalized RMS error of the ML estimates were significantly larger that those derived for the model network (cf. Fig. 6).

Thirdly, as the input of our algorithms are actual testbed measurements, it is not possible to compute the complexity of our algorithms or evaluate the quality of the estimation they arrive at (as was done, e.g., in [8], and related works that are based on mathematical models.)

Finally, the execution of the SEA and NWTD protocols requires centralized operation, in line with the Digital Twins paradigm. Otherwise, SEA and NWTD cannot be executed, although lessons learned from networks where they are employed can be applied to other networks.

On the other hand, our work can be extended towards numerous directions. In terms of making our scheme faster, one important direction would be to improve the scalability of SEA by modifying it so that fewer candidates for testing are created in each iteration (line 5 of Pseudocode 2). In terms of reducing the overhead of the measurements, a very simple method would be to integrate the estimation of the CR while the network is operating, by serendipitously taking *passive* measurements; we anticipate this approach to be especially fruitful when the routing protocol used belong to the class of backpressure protocols [21]. In terms of the performance evaluation of protocols, it would be useful to study which percentage of the CR is achieved by routing protocols other than OLSR, and also to introduce CR-inspired protocols other than NWTD. In terms of RV predictions using ML, alternative neural network could be considered. In terms of complexity analysis, it would be useful to derive estimates of the use of resources by SEA using judicious assumptions bounding the data rate values. Also, it would be very useful to apply our methodology to other settings using alternative MAC/PHY layers, such as LoRa.

Finally, developing a distributed CR estimation method and an associated distributed time division (TD) schedule implementation method would be important extensions. One promising approach is to exploit the locality of the effects of interference in order to cluster nodes in different (possibly overlapping) interference zones. The establishment of zones fragments the search space of Transmission Modes (TMs), enables parallel measurements (thus accelerating the process) and limits the sharing of results and the computation of the zone-specific TDMA schedules within the zones (thus reducing complexity). The nodes would have to independently detect conflicts in the implementation of the produced global TD schedule, and respond locally and independently by introducing adjustments to their own transmissions schedule.

7. Conclusions

Capacity regions have a central place in wireless network optimization research; in this work, first we presented a method for estimating them in deployed networks through measurement, using online machine learning to reduce the amount of measurements needed, and then showed that the derived estimates can be utilized both for evaluating existing protocols as well as creating new ones that operate according to a Digital Twins approach. We evaluated our method in a testbed that uses IEEE 802.11 but, due to its generality, the method can be applied with other PHY/MAC protocol combinations.

Our experiments lead to numerous promising results and insights. Firstly, the volume of measurements needed to estimate the CR sufficiently well is manageable. For example, as shown in Section 5.2 for the case of the 6-node network, using the Sequential Expansion Algorithm we arrived at an aggregate data rate which is 79% of the optimum value by measuring 0.37% of all TMs. Secondly, as shown in Section 5.5, online machine learning can be used effectively to estimate the performance of simultaneously active transmitter–receiver links, thus significantly reducing the volume of measurements needed. Finally, NWTD performs significantly better than OLSR (in particular, 33% in terms of an aggregate data rate metric), notably due to its ability to make judicious choices on which sets of transmitter–receiver links to activate at any given time.

CRediT authorship contribution statement

Yannis Thomas: Conceptualization, Investigation, Methodology, Software, Supervision, Validation, Writing – review & editing. **Stavros Toumpis:** Conceptualization, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing, Funding acquisition. **Nikolaos Smyrnioudis:** Conceptualization, Investigation, Methodology.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yannis Thomas and Stavros Toumpis report financial support for this paper was provided by the Hellenic Foundation for Research and Innovation. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.), Greece under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: HFRI-FM-17-352, Project Title: Wireless Mobile Delay-Tolerant Network Analysis and Experimentation).

References

[1] R. Wang, M.A. Kishk, M.-S. Alouini, Reliability analysis of multi-hop routing in multi-tier LEO satellite networks, 2023, arXiv preprint arXiv:2303.02286.
[2] K. Skiadopoulos, A. Tsipis, K. Giannakis, G. Koufoudakis, E. Christopoulou, K. Oikonomou, G. Kormentzas, I. Stavrakakis, Synchronization of data measurements in wireless sensor networks for IoT applications, Ad Hoc Netw. 89 (2019) 47–57.
[3] L. Atzori, A. Iera, G. Morabito, Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm, Ad Hoc Netw. 56 (2017) 122–140.
[4] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, Y. Zhou, Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2377–2396.
[5] Y. Liu, Y. Li, L. Su, E. Yeh, S. Ioannidis, Experimental design networks: A paradigm for serving heterogeneous learners under networking constraints, in: IEEE INFOCOM 2022, 2022, pp. 210–219.
[6] S. Toumpis, A.J. Goldsmith, Capacity regions for wireless ad hoc networks, IEEE Trans. Wirel. Commun. 2 (4) (2003) 736–748.
[7] M. Kodialam, T. Nandagopal, Characterizing the capacity region in multi-radio multi-channel wireless mesh networks, in: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, 2005, pp. 73–87.
[8] R. Gummadi, K. Jung, D. Shah, R. Sreenivas, Computing the capacity region of a wireless network, in: IEEE INFOCOM, 2002, pp. 1341–1349.
[9] R. Laufer, L. Kleinrock, The capacity of wireless CSMA/CA networks, IEEE/ACM Trans. Netw. 24 (3) (2016) 1518–1532.
[10] Y. Thomas, N. Smyrnioudis, S. Toumpis, Experimental measurement of the capacity region of wireless networks, in: 2021 19th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt, 2021.
[11] Y. Thomas, S. Toumpis, TWIST: Thin-waist wireless testbed for measuring interfering traffic stream throughputs, in: Proc. IEEE WoWMoM, 2023.
[12] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Trans. Inform. Theory 46 (2) (2000) 388–404.

[13] M. Noori, S. Rahimian, M. Ardakani, Capacity region of aloha protocol for heterogeneous IoT networks, IEEE Internet Things J. 6 (5) (2019) 8228–8236.
[14] C. Li, S. Weber, J.M. Walsh, On multi-source networks: Enumeration, rate region computation, and hierarchy, IEEE Trans. Inform. Theory 63 (11) (2017) 7283–7303.
[15] L. Liu, B. Yin, S. Zhang, X. Cao, Y. Cheng, Deep learning meets wireless network optimization: Identify critical links, IEEE Trans. Netw. Sci. Eng. 7 (1) (2020) 167–180.
[16] K. Choumas, T. Korakis, I. Koutsopoulos, L. Tassiulas, Implementation and end-to-end throughput evaluation of an IEEE 802.11 compliant version of the enhanced-backpressure algorithm, in: International Conference on Testbeds and Research Infrastructures, Springer, 2012, pp. 64–80.
[17] L. Deng, C.-C. Wang, M. Chen, S. Zhao, Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms, IEEE/ACM Trans. Netw. 25 (6) (2017) 3473–3486.
[18] M. Johansson, L. Xiao, Cross-layer optimization of wireless networks using nonlinear column generation, IEEE Trans. Wirel. Commun. 5 (2) (2006) 435–445.
[19] A. Köse, H. Gökcesu, N. Evirgen, K. Gökcesu, M. Médard, A novel method for scheduling of wireless ad hoc networks in polynomial time, IEEE Trans. Wirel. Commun. 20 (1) (2020) 468–480.
[20] P.-J. Wan, L. Wang, A. Huang, M. Li, F. Yao, Approximate capacity subregions of uniform multihop wireless networks, in: Proceedings IEEE INFOCOM, 2009, pp. 1–9.
[21] R. Laufer, T. Salonidis, H. Lundgren, P. Le Guyadec, A cross-layer backpressure architecture for wireless multihop networks, IEEE/ACM Trans. Netw. 22 (02) (2014) 363–376.
[22] Y. Li, L. Qiu, Y. Zhang, R. Mahajan, E. Rozner, Predictable performance optimization for wireless networks, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 413–426.
[23] T. Salonidis, G. Sotiropoulos, R. Guerin, R. Govindan, Online optimization of 802.11 mesh networks, in: Proc. 5th International Conference on Emerging Networking Experiments and Technologies, 2009, pp. 61–72.
[24] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah, Artificial neural networks-based machine learning for wireless networks: A tutorial, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3039–3071.
[25] A. Samba, Y. Busnel, A. Blanc, P. Dooze, G. Simon, Instantaneous throughput prediction in cellular networks: Which information is needed? in: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management, IM, IEEE, 2017, pp. 624–627.
[26] A. Al-Thaedan, Z. Shakir, A.Y. Mjhool, R. Alsabah, A. Al-Sabbagh, M. Salah, J. Zec, Downlink throughput prediction using machine learning models on 4G-LTE networks, Int. J. Inf. Technol. 15 (6) (2023) 2987–2993.
[27] D. Minovski, N. Ogren, C. Ahlund, K. Mitra, Throughput prediction using machine learning in LTE and 5G networks, IEEE Trans. Mob. Comput. (2021).
[28] S.M. Srinivasan, T. Truong-Huu, M. Gurusamy, Machine learning-based link fault identification and localization in complex networks, IEEE Internet Things J. 6 (4) (2019) 6556–6566.
[29] J. Zhang, R. Gardner, I. Vukotic, Anomaly detection in wide area network meshes using two machine learning algorithms, Future Gener. Comput. Syst. 93 (2019) 418–426.
[30] W. Cui, K. Shen, W. Yu, Spatial deep learning for wireless scheduling, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1248–1261.
[31] O. Aldhaibani, F. Bouhafs, M. Makay, A. Raschellà, An SDN-based architecture for smart handover to improve QoE in IEEE 802.11 WLANs, in: 2018 32nd International Conference on Advanced Information Networking and Applications Workshops, WAINA, IEEE, 2018, pp. 287–292.
[32] A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Comput. Netw. 167 (2020) 106984.
[33] K. Poularakis, Q. Qin, E.M. Nahum, M. Rio, L. Tassiulas, Flexible SDN control in tactical ad hoc networks, Ad Hoc Netw. 85 (2019) 71–80.
[34] K. Poularakis, L. Tassiulas, T. Lakshman, Modeling and optimization in software-defined networks, Synth. Lect. Learn. Netw. Algorithms 2 (2) (2021) 1–174.
[35] S. Çğay, T.T. Sarı, G. Seçinti, Sonar: Software-defined network and radio framework for FANETs, in: 2021 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN, IEEE, 2021, pp. 268–273.
[36] F. Tao, H. Zhang, A. Liu, A.Y. Nee, Digital twin in industry: State-of-the-art, IEEE Trans. Ind. Inf. 15 (4) (2018) 2405–2415.
[37] L.U. Khan, Z. Han, W. Saad, E. Hossain, M. Guizani, C.S. Hong, Digital twin of wireless systems: Overview, taxonomy, challenges, and opportunities, IEEE Commun. Surv. Tutor. (2022).
[38] S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[39] https://www.raspberrypi.org.
[40] https://wireless.wiki.kernel.org/en/users/Drivers/brcm80211.
[41] https://iperf.fr.
[42] T. Clausen, P. Jacquet, RFC3626: Optimized link state routing protocol (OLSR), 2003.

Yannis Thomas is a postdoctoral researcher at the Department of Informatics of AUEB and member of the Mobile Multimedia Laboratory (MMlab). He received his Diploma in Informatics (2009), his MS in Information Systems (2012) and his Ph.D. in Computer Science (2018) from the Department of Informatics of AUEB, Greece. He has extensive participation in a series of research projects funded by the European Commission (EC), the ESA, and Greece, such as the EC-funded projects "Publish Subscribe Internet Technology", "iP Over IcN-the betTer IP" and "Secure Open Federation For Internet Everywhere", and the ESAfunded project "The role of satellite in the future Internet". His research interests include multiflow transport, congestion and flow control in current Internet and Next Generation networks, including Mobile Ad hoc NETworks (MANETs) and heterogeneous Satellite-Terrestrial networks. He has coauthored more than 20 refereed articles in international journals, books, and conferences, including top-tier venues such as the IEEE INFOCOM and IEEE/ACM Transactions on Networking. He has also been reviewer for international journals, such as the ACM Transactions on Mobile Computing, the Elsevier Computer Networks and the Elsevier Computer Communications.

Stavros Toumpis was awarded the Diploma in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 1997, the M.S. degrees in Electrical Engineering and Mathematics from Stanford University, CA, in 1999 and 2003, respectively, and the Ph.D. degree in Electrical Engineering, also from Stanford University, in 2003. From 1998 to 1999 he worked as a Research Assistant for the Mars Global Surveyor Radio Science Team at Starlab, Stanford, providing operational support and conducting research on radio occultations, under the supervision of Prof. G. L. Tyler. From 2000 to 2003 he was a member of the Wireless Systems Laboratory at Stanford, conducting research on wireless networks, under the guidance of Prof. Andrea J. Goldsmith. From 2003 to 2005 he was a Senior Researcher with the Telecommunications Research Center Vienna (ftw.), in Vienna, Austria. From 2005 to 2009 he was a Lecturer at the Electrical and Computer Engineering Department of the University of Cyprus. From 2009 to 2020 he was an Assistant Professor and from 2020 he is an Associate Professor in the Informatics Department of the Athens University of Economics and Business in Athens, Greece.

Nikolaos Smyrnioudis is a Software Engineer working with Database systems in CERN. He has studied Computer Science at the Athens University of Economics and Business for his Bachelor's studies and Machine Learning at KTH for his Master's studies. His research interests are NLP, Machine Learning, Explainability in Machine Learning, Wireless Ad Hoc Networks, and Machine Learning applications in the context of networks.