A Flexible Framework for Polynomial-Time Resource Allocation in Streaming Multiflow Wireless Networks

Gareth Middleton, Behnaam Aazhang, and Jorma Lilleberg

Abstract—We present a wide-area, multiflow ad-hoc network model leveraging information-theoretic rate control, emphasizing interfering rather than colliding transmissions. We seek to allocate resources in this network by optimizing scheduling, routing and power control to solve the max-min throughput problem for all flows involved. In general, our time-slotted, fully-interfering model leads to an NP-hard problem. Further, the rate-control element of the mixed-integer program results in a non-convex problem in the continuous domain.

The complexity of the joint problem makes an optimal solution prohibitively difficult to find, leading us to propose a two-pronged approach to determine a near-optimal resource allocation. First, we propose a novel decomposition technique, breaking the joint problem into a sequence of more tractable subproblems. Second, we present a data structure serving multiple purposes: it compactly represents network conditions as they evolve with time, and also serves as the basis for our cubic-time dynamic programming algorithms used to solve and catalog the subproblems. The result is a schedule, route, and power allocation for all data frames involved. We demonstrate the performance of our techniques on the max-min throughput problem, while also showing that they are sufficiently general to apply to a wide variety of optimality criteria in which decisions over transmission schedules and packet routing must be made.

Index Terms—Resource allocation, scheduling, routing, mixed-integer programming.

I. INTRODUCTION

RESOURCE allocation in large networks is fundamentally an interference-management problem, with degrees of freedom dictated by the network architecture and system configuration. In traditional wide-area network studies, the complexity of the problem has been limited by treating it as an interference-avoidance paradigm, employing techniques such as k-hop or graph-matching collision models. Even models inspired by the physical layer, such as those based on signal-to-interference-and-noise ratios (SINR) reduce to graph-based models because SINR thresholds enforce binary connection states in the network. This class of network model has produced a large body of resource allocation results (see below), but still treats the problem as interference-avoidance rather than interference-management. Resource allocation in such shared bandwidth scenarios is relevant in a variety of large-area, multihop network settings, including sensor-terminal deployments, emergency communications, and coverage-extension in decentralized cellular networks.

We seek to broaden the scope of the optimization, allowing the physical layer rate to be continuously variable, defined using an information-theoretic parameterization. Our model assumes that all transmissions interact; hence rates are governed by channel conditions on the link in question as well as those between the receiver and all other active transmitters. As a result, the rates of all links are determined jointly by the choice of active links and the power level of each transmitter. In this way, we think of simultaneous transmissions not as colliding data frames in which all information is lost, but as coexisting frames whose rates and power levels must be jointly selected. Thus the resource allocation problem becomes one of interference-management, where choice of power level and active transmitters determine the amount of data each frame can carry. The scheduling problem is known to be NP-hard in this environment [1], and since we upper-bound the rate using Shannon’s equation $\log_2(1 + \text{SINR})$, the rate control problem is non-convex in the power allocation. This leads to a tradeoff in complexity for performance: if frames are assumed simply to either collide or be received perfectly, the complexity of the allocation problem is reduced but network performance may be lower. In this paper, we study how the solution to a much more complex problem results in performance gains.

Our first contribution towards solving this problem is an extensible framework which allows for polynomial-time resource allocation with information-theoretic rate control. We will formulate the mixed-integer program, and show that the global solution is generally unavailable due to the complexity of the constraints. We then present our second contribution: an algorithm for our framework requiring only cubic time to yield a solution. While the optimality criterion in this paper is max-min throughput, we will demonstrate that our approach is sufficiently general as to apply to a wide variety of criteria, including route delay and flow utility. The key to our approach is a decomposition of the joint problem which allows for dynamic programming methods to be applied, while preserving as much of the interconnected nature of the problem as possible.
II. RELATED WORK

Resource allocation in wireless networks has been studied from two main viewpoints: large-scale information theoretic analyses and medium-access control studies. Both lend very different flavors to the same problem, giving very different results. The order analysis of large networks, pioneered by Gupta and Kumar [2], opened the door to a variety of information-theoretic network studies [3]–[6]. These works established the limits of transport capacity in networks of increasingly sophisticated design: multihop [7], hierarchical [5], even those inspired entirely by the physics of energy propagation [8], [9]. A common result is that there exists an asymptote at which interference management becomes impossible; i.e., when network load is so large that no technique can overcome it. This is known as the “interference limit.”

In contrast, network-layer studies of practical issues such as scheduling and routing packets on small timescales tend to emphasize interference avoidance, in order to guarantee packet delivery and flow throughput. In this domain, powerful and widely employed routing algorithms such as AODV [10] and DSDV [11] have been developed, building on the classical results of Dijkstra [12], and Bellman-Ford [13]. In the wireless context, DSR [14] was among the first protocols to exploit the wireless broadcast advantage in routing. Others have followed [15]–[17], but with the primary emphasis on route creation rather than schedule management. Scheduling for maximum throughput was addressed with the introduction of the backpressure algorithm [18], which can apply to the wireless environment if the protocol model of interference is assumed. In that case, a variety of algorithms have been proposed [19], [20], also for random-access networks [21]–[23], some with emphasis on congestion control.

Studies of joint resource allocation have recently been undertaken (see [24] and references therein): computing schedules and routes concurrently [25]–[28], and in some cases under the SINR model [29], [30]. In this light, resource allocation can be viewed as a product-multicommodity flow problem [31], so that tools such as relaxation and randomized-rounding [32] from the mixed-integer programming community may be applied. While this is a step towards interference management, the common requirement of fixed frame sizes limits the degrees of freedom in the network: requiring a specific SINR results in interference avoidance near a link in order for it remain active.

In this work, we propose a continuous domain for the size of data frames, drawn from information-theoretic characterizations of the channels between terminals. Combining the information-theoretic notions of the large-scale studies with the scheduling and routing constraints found in network-layer analyses allows us to treat the resource allocation problem as one of true interference management, in which we have degrees of freedom at both the physical and network layers.

The remainder of this paper is laid out as follows. We present our model and problem description in Section III, followed in Section IV by our novel tool, the Network-Flow Interaction Chart (NFIC). Section V describes the algorithms used on the NFIC to find the solution, and Section VI presents simulations results. Concluding remarks appear in Section VII.

III. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a half-duplex, synchronous, time-slotted system in which all terminals may transmit to each other over a common bandwidth, at rates determined by channel conditions between terminals and interference at the destination. We assume a square region of area \( A^2 \), over which terminals are scattered according to a 2-dimensional Poisson distribution with intensity parameter \( \lambda \). We will use \( N \) to denote the entire set of terminals, which has cardinality \( |N| = N \). From these terminals, \( F \) pairs of terminals are randomly chosen as sources and destinations for a set of information flows \( \mathcal{F} \). Data is transferred for the flows in a stream of arbitrarily many frames, where a frame is composed of small packets, the number of which is dictated by channel conditions. As such, frames crossing good channels can carry a larger amount of data than those being sent over poorer channels, leading us to model frame sizes as being from the positive real line. We index the frames with \( k \), with each frame being transmitted between two terminals in a single timeslot as it makes its way through the network from source terminal \( s(k) \) to destination terminal \( d(k) \). Any terminal in the network may act in a forwarding manner, and may also store data for future transmission. We define a schedule \( S \) specifying which transmitter and receiver pairs are active in each timeslot. A schedule specifying \( T \) timeslots exists in \( \mathbb{N}^{N \times N \times T} \), since each timeslot has \( N^2 \) binary variables specifying the transmission state of each link. With \( T_k \) we denote the timeslot in which frame \( k \) arrives at its destination.

A schedule is termed feasible if it specifies utilization of network terminals such that frames travel from their sources to their destinations without violating the routing and duplexing constraints of the devices. Defining the space of feasible schedules requires the introduction of a binary variable \( I^k_{t}(x,y) \), which is one if frame \( k \) is being transmitted from terminal \( x \) to terminal \( y \) in timeslot \( t \) and zero otherwise. Note that \( I^k_{t}(x,y) \) is indexed in both time \( t \) and space \( (x,y) \).

The set of feasible schedules \( S \) is defined by the following constraints:

\[
\sum_{x,y \in N} I^k_t(x,y) = 1 \quad \forall \; k, \; t \geq 1 \quad (1)
\]

\[
\sum_k \left( \sum_{z \in N \setminus \{x,y\}} (I^k_t(x,z) + I^k_t(z,x)) + \sum_{z \in N \setminus \{x,y\}} (I^k_t(y,z) + I^k_t(z,y)) \right) \leq 1 \quad \forall \; (x,y) \in N \quad (2)
\]

\[
\sum_{y \in N} I^k_t(s(k),y) = 1 \quad \forall \; k, \quad (3)
\]

\[
\sum_{x \in N} I^k_{T_k}(x,d(k)) = 1 \quad \forall \; k, \; T_k > 1 \quad (4)
\]

\[
\sum_{x \in N} I^k_t(x,y) = \sum_{z \in N} I^{k+1}_t(y,z) \quad \forall \; k, \; y \in N, \; t \geq 1 \quad (5)
\]

We consider unidirectional flows here, but bidirectional data may be handled by our framework by defining another \( F \) flows operating in the reverse direction.
These constraints apply to all frames \( k \) and are specified in terms of transmitter-receiver pairs \((x, y)\) and a third terminal \( z \). We regard the first time period to be indexed as \( t = 1 \). The first constraint requires that each frame be accounted for in each time period, even if it is in memory (i.e. \( I^k_t(x, x) = 1 \)) or at either \( s(k) \) or \( d(k) \). The second constraint precludes multicast and multiple-access combining, by ensuring that neither terminal in the active pair \((x, y)\) is acting as a transmitter or receiver in another link. This enforces the half-duplex constraint discussed earlier. The remaining constraints enforce feasible routes: constraint (3) fixes the start point of a frame’s route to its source, and (4) requires that the frame arrive at its destination at some timeslot \( T_k \geq 1 \). Continuity of routing is enforced by (5), and circular routes (traveling through the source or destination more than once) are disallowed by (6) and (7). We also define a set of feasible power allocations \( \mathcal{P}_{Tx} \in \mathbb{R}^N \), which is simply an \( N \)-dimensional cube in which all terminals are allocated power levels between 0 and the maximum terminal transmission power \( P_{Tx} \).

With these definitions and constraints in mind, we make precise the notions of routes and schedules. For frame \( k \), a route is the sequence of terminals it uses as it traverses the network. At time \( t \), the terminal transmitting frame \( k \) is a part of the route \( L^k_t = \{ x \mid I^k_t(x, y) = 1 \} \). Thus the route for frame \( k \) is \( L^k = \bigcup_t L^k_t \). The broader concept of a schedule specifies, at each time period, which links are active. A schedule \( S \) is therefore the synthesis of \( I \) across time and space: \( S = \bigcup_{t,k} I^k_t \). In this way, specifying the variable \( I^k_t(x, y) \) determines both the schedule at each timeslot and the routes for all frames simultaneously.

Resource allocation entails managing interference in two ways: first by specifying the transmission set \( I^k_t \) for all frames and timeslots, and second by selecting the power level each terminal uses while transmitting so as to optimize some network-wide parameter. Consider power \( p \in \mathbb{R} \) in use at the terminals at time \( t \) as given in the vector \( \mathbf{p} \). We refer to the set of vectors over all timeslots as \( \mathbf{p} \), where each element in a vector corresponds to one terminal. The space \( \mathcal{P} \) is the space of feasible power allocations, which may be smaller than \( \mathcal{P}_{Tx} \) if interference management constraints prevent nearby terminals from using maximum power.

Network utility is a function of scheduling decisions and available power, according to a mapping \( U : \mathbb{R}^{N \times T} \times \mathbb{R}^{N \times N \times N \times T} \rightarrow \mathbb{R} \). This function may be defined in terms of throughput, aggregate delay, or some other relevant performance metric. Computing the optimal utility \( U^* \) requires finding the optimal schedule \( S^* \) and power allocations \( \mathbf{p}^* \).

If \( \mathcal{P} \) resources are available, we write a vector-valued allocation function \( A : \mathbb{R}^{N \times T} \rightarrow \mathbb{R} \times \mathbb{R}^{N \times T} \times \mathbb{R}^{N \times N \times T} \) as \( A(\mathcal{P}) = (U^*, \mathbf{p}^*, S^*) \) where

\[
U^* = \max_{\mathbf{p} \in \mathcal{P}, S \in \mathcal{S}} U(\mathbf{p}, S),
\]

\[
(\mathbf{p}^*, S^*) = \arg \max_{\mathbf{p} \in \mathcal{P}, S \in \mathcal{S}} U(\mathbf{p}, S)
\]
link may be composed of several simultaneous, coordinated transmissions each of which is defined by a value in $p$. In this case $A$ is many-to-one and cannot be uniquely inverted. While we do not study this case here, our techniques may be extended to these network architectures.

We have presented these concepts in the abstract to illustrate their generality, but we will restrict our attention to a utility function based on information-theoretic rates. We will now describe the wireless environment we consider.

A. Physical Layer Model

We consider an extremely general physical layer model, one in which the size of a frame is determined by the information-theoretic capacity of the link between two terminals. This allows us to capture the notion of interfering—rather than colliding—transmissions, introducing rate control and allowing a study of interference management at both the physical and medium-access layers together. We consider frames of variable size, composed of a variable number of finite-sized packets. We assume a pathloss-dominated additive white Gaussian noise channel, and follow the worst-case assumption that transmissions interfere with each other purely as noise. All terminals are subject to a peak power constraint $P_{max}$, the thermal noise level is $N_0$, and the pathloss exponent is $\alpha$. With $D_{x,y}$ we denote the Euclidean distance between terminals $x$ and $y$, such that the rate $R$ on the link between the terminals at time $t$ for frame $k$ is upper bounded by [33]:

$$R_k^f(x, y) = \log_2 \left(1 + \frac{P_k(z) D_{x,y}^{-\alpha}}{N_0 + \gamma(y)} \right) \cdot I_k^f(x, y)$$

(9)

where the interference term $\gamma(y) = \sum_{z \in \mathcal{N}_{x,y}} P_k(z) D_{x,y}^{-\alpha}$. Here and after, we use $P_k(z)$ to denote the power used by terminal $z$ in timeslot $t$, a component of the vector $p$ discussed above. Without loss of generality, we will consider a normalized bandwidth of 1 Hz and timeslots of 1 second, such that $R_k^f$ is the size in bits of frame $k$ at time $t$. If the link between $x$ and $y$ is not active, the rate is necessarily zero, as is $P_k(z)$. Since a frame may traverse several links on its way to the destination, it may not be larger than what the bottleneck link can accommodate. As such we define the size of frame $k$ as $R_k = \min_f R_k^f$, with $R_k^f$ defined as in (9). Since our flows are infinitely backlogged, this limit on frame size does not result in fragmented data.

As the utility function $U$, we choose minimum throughput across the flows:

$$U(p, S) = \min_{f \in F} \eta_f$$

(10)

where the throughput $\eta_f$ is defined as the sum of the data transmitted scaled by the time in transit:

$$\eta_f = \frac{1}{T} \sum_{k=(f, f+1, f+2, \ldots, f+K F)} R_k$$

(11)

where we sum only the frames carrying data for flow $f$, and $K$ is the number of frames transmitted by flow $f$. As such, evaluating (8) with $U$ defined as in (10) results in computing the max-min throughput allocation in the network. This is NP-hard in general, and because the equation determining the throughput of $U$ contains a fractional form inside the logarithm of (9), $U$ is non-convex. Determining the fully optimal solution would require an exhaustive search through all $N^N$ possible schedules, each of which has a non-convex objective in the power allocation.

B. Decomposition

We employ a frame-wise decomposition of this problem, solving a sequence of problems (8) for each frame $k$. This requires enforcing the data flow constraints in the variable $I_k^f(\cdot, \cdot)$ above, while optimizing power over a space $\mathcal{P}$ defined such that excess interference is not caused for existing traffic in the network. Each frame may be viewed as contributing an incremental element to the throughput of flow $f$; we seek to maximize this element. Denote with $a_k$ the arrival slot of the $k$th frame, such that $a_{k-1}$ is the arrival slot of the last frame carrying data for flow $f$. For each frame in the sequence we seek to solve

$$\max_{p \in \mathcal{P}, S \subseteq \mathcal{F}} \frac{1}{a_k - a_{k-1}} R_k$$

(12)

which is a result of the separability of the sum in (11). The throughput contribution is seen to be amount of new data divided by the number of timeslots elapsed since the previous frame arrived.

We may now compare our iterative, per-frame solution to the full problem expressed in (8) above. In Figure 1, we have exposed in abstract the three dimensions of the problem: power allocation $p$, link activation decisions $I_k^f(\cdot, \cdot)$, and frame $k$. The figure shows the joint problem, in which decisions made for each frame directly affect those made for the other frames. For the purpose of exposition, the shaded panes represent the schedules and power allocations feasible for each frame; in the joint allocation, all combinations are feasible subject to decisions made for other frames.

Our decomposition is shown in Figure 2. When problem (12) is solved for frame $k = 1$, all schedules and power allocations are feasible. After choosing the optimal sequence of terminals $I_k^f(\cdot, \cdot)$ for that frame, power allocations for terminals involved are reduced such that all links operate at the same rate. This schedule and corresponding power allocation then determine feasible schedules power allocations for the second frame, which is a smaller space than was available for the first. This process then repeats for subsequent frames.

While the decomposition makes our problem tractable in general, we must efficiently manage the data in the sequence of sub-problems. We now describe the data structure we have created for this purpose.

IV. N E T W O R K - F L O W I N T E R A C T I O N C H A R T

Until now, we have described the method for decomposition, but have not indicated how the subproblems are solved...
or how the complexity of the decomposition is managed. Our technique employs a data structure enabling both of these–decomposition management and subproblem solution–at the same time. Named the Network-Flow Interaction Chart (NFIC), it captures the spatial and temporal interactions of frames in a network with rate control. In particular, it allows us to easily populate the variable $I^t(x,y)$ to enforce the corresponding constraints, as well as to evaluate $A(P)$ for single frames in polynomial time, as we will show below.

The chart is illustrated for an example network in Figure 3. All terminals $x \in N$ in the network are represented as nodes $x_t$ in the chart, where the temporal dimension is exposed with the addition of the subscript $t$ to each node. Edges $e^{x,y}_t$ are drawn between each node to represent an action in the network at a particular time $t$; in the case of throughput optimization, the edge $e^{x,y}_t$ represents the transmission between terminal $x$ and terminal $y$ in timeslot $t$.

For a single frame, we will use the NFIC to establish a data-maximizing route between its source and destination. We will employ dynamic programming techniques in the vein of shortest-path routing [12], though on the temporally-aware

---

**Remark:** Although we consider throughput as the example
in this paper, a monotone semiring may also be defined to optimize for minimum delay: \( \mathbb{R}^+ \cup \{0\} \). Here, the values on each edge represent the delay associated with using that link, which sums up as we progress through the network. Other metrics, provided they form monotone semirings, may also be considered in our framework.

V. RESOURCE ALLOCATION ON THE NFIC

We now present the solution technique for problem (12) discussed above. Namely, we must establish how \( I_t^k(\cdot, \cdot) \) is populated, and how the space of available transmission powers \( \mathcal{P} \) evolves as frames are allocated.

A. Frame Sequencing

As discussed above, we will use shortest-path programming techniques to solve (12) optimally for one frame at a time. To determine the sequence in which frames should be allocated, we recall the max-min nature of the optimization formulation (10). Since the successive allocation of frames reduces available network resources, we allocate the frame we expect to require the most resources prior to those expected to require fewer.

We must determine this sequence without solving the joint problem. The pathloss-dominant nature of our environment provides such a method: since terminals are distributed according to a Poisson process, the average distance between terminals is \( \frac{\lambda}{P} \). A frame requiring more hops is more likely to experience a rate-limiting long hop, limiting its throughput. As such, we estimate that the flow with the longest end-to-end distance (Euclidean distance between the source and destination) should be considered for allocation first, followed by the one with the second longest end-to-end distance and so on. After a frame has been allocated for all \( F \) flows in this order, the sequence repeats for frame \( k = F + 1 \). This means that frame \( k \) is carrying information for flow \( k \mod F \), since we increment the frame counter \( k \) for each individual frame considered. Recall that we denote the terminal sourcing frame \( k \) as \( s(k) \) and its destination as \( d(k) \).

B. Frame Allocation: Link Activation and Rate Control

We explicitly describe how to solve (12) for the first frame. Prior to allocating any frames, \( I_t^k(\cdot, \cdot) \) is zero for all terminals and \( \mathcal{P} = \mathcal{P}_{\text{Tx}} \), since there is no interference constraint in an empty network. We initialize the NFIC edge weights corresponding to the capacity that link represents, as

\[
\eta_t^k = \frac{d(k)}{t - a_{k-F} - 1} \quad \forall \quad t > a_{k-F}
\]

which is the amount of data at the destination at time \( t \) scaled by how much time has passed since the previous frame arrived. The arrival time of the previous frame for this flow is \( a_{k-F} \), zero for \( k \leq F \). Searching \( \eta_t^k \) across timeslots for the maximum value gives the frame contributing most to the throughput for this flow:

\[
t_{\text{opt}} = \arg \max_t \eta_t^k
\]

The route \( L^k \) is then constructed by tracking back through the NFIC starting at the destination node \( d(k)_{t_{\text{opt}}} \), using the history variable \( h_t^k \). The last terminal in the route is the destination: \( L^k_{t_{\text{opt}}} = d(k) \). The others are found using the recursive relation \( L^k_t = h_{t+1}^k \). The binary link-state variable \( I_t^k(\cdot, \cdot) \) may now be updated as well:

\[
I_t^k(L_t^k, h_t^{k_{t+1}}) = 1 \quad \forall \ t \leq t_{\text{opt}}
\]

An extremely simple execution of this algorithm is illustrated in Figure 4, where one frame is allocated in a network of four terminals.

To determine the power allocation for the terminals, we search \( L^k \) for the bottleneck link, since it sets the size for the frame along the entire route. We then reduce the power for all other terminals to match, such that all constituent links in \( L^k \) support frames of the same size, leading to the allocation \( \{P_1(s(k)), P_2(x), P_3(y), \ldots, P_{t_{\text{opt}}-1}(z)\} \). Performing
rate control in this manner serves the primary end of lowering the interference temperature in the broader network, so that frame \( k \) affects subsequent frames less, which also increases battery life in mobile devices.

This technique will solve (12) for an individual frame \( k \), populating \( I^f_k(\cdot, \cdot) \) and calculating the power allocation \( \mathbf{p} \). Note how the execution of the dynamic programming algorithm implicitly ensures that the schedule chosen will be feasible: in the solution for frame \( k \), only one link will be active per timeslot (1), there will be continuity (5), and the source and destination will be visited only once (6), (7). We now show how resources are removed from the NFIC to enforce the routing and duplexing constraints on \( I^f_k(\cdot, \cdot) \) for subsequent frames, Step 2 in the general algorithm outlined above.

**C. Resource Subtraction**

Constraints (1) - (7) must be satisfied at all terminals and all times in the network. By its design, the NFIC easily reflects those constraints in a graphical form. In particular, we will update the NFIC to satisfy the half-duplex constraint (2). To that end, we must make terminals unavailable while they are handling—either transmitting or receiving—any scheduled frame.

An edge \( e^{w,x}_{t} \) carrying frame \( k \) is represented as a 1 in \( I^f_k(x, y) \). Referring to (2), the sum over \( k \) means that no two edges from the same node may be active in the same timeslot. As such, after an allocation for frame \( k \) has been determined, we remove access to the terminals involved in the route of frame \( k \) from the NFIC. A consequence of the semiring in which we are working is that we can overwrite the weight of an edge with the annihilator \( \otimes_l \), such that any edge update \( \min \{ \} \) in subsequent frame allocations will always result in that edge being weighted with \( \otimes_l \). Thus we enforce the duplexing constraint as follows: for each active link in \( L^k \), edges entering and exiting both the source and destination nodes are assigned weight \( \otimes_l = 0 \). This removes the terminal from consideration in the shortest-path algorithm for any future frame. An illustration of this is shown in Figure 5, where the solid line is the allocation for a frame and the broken lines are those removed from the NFIC as described above.

All other edges with nonzero weights are available for subsequent frames. However, because we consider a fully interfering network, previously-allocated frames will be impacted by any newly-scheduled simultaneous transmission, meaning that interference must be taken into account. Given channel conditions, interference is purely a function of transmission power, which also exactly determines the size of a frame on a link. Managing feasible sizes of subsequent frames is therefore equivalent to managing interference imposed on frames already in the network.

We define a loss ratio \( \rho_{\text{lat}} \in [0, 1] \), the fraction of data loss permitted on existing frames as a result of the introduction of new ones. Consider the leg from \( x \) to \( y \) at time \( t \) in frames \( k \)'s path, which is the most recently allocated frame. This link operates at rate \( R^k_t \), which would drop to \( \tilde{R}^k_t \) in the presence of the new frame \( k + 1 \):

\[
\rho_{\text{lat}} \cdot R^k_t = \tilde{R}^k_t
\]

where \( \gamma_t(y) \) is the interference frame \( k \) is experiencing at terminal \( y \) from other existing frames \( \{1, \ldots, k-1\} \), and \( \tilde{\gamma}_t(y) \) is the interference it would experience if another frame is added to the network at time \( t \). This equation can be manipulated to find the maximum tolerated interference \( \tilde{\gamma}_t(y) \) in terms of frame \( k \)'s allocated rate and the tolerated loss fraction \( \rho_{\text{lat}} \):

\[
\tilde{\gamma}_t(y) = \left[ P_t(x) \left( 1 + \frac{P_t(x)}{N_0 + \gamma_t(y) D_{x,y}} \right)^{-1} \right]^{\frac{1}{\alpha}}
\]

\[
\times D_{x,y}^\alpha - N_0
\]
By scaling $\rho_{\text{Int}},$ we can select between full time-orthogonalization ($\rho_{\text{Int}} = 0$) and full interference ($\rho_{\text{Int}} = 1$). Calculation of an optimal value for $\rho_{\text{Int}}$ is topologically specific, though such a value can be determined experimentally. We carry out such a simulation in both low and high pathloss environments, for three information flows of six frames each. The results are reported in Figure 6. Lower values of $\rho_{\text{Int}}$ do not permit enough interference to allow the algorithm to exploit all degrees of freedom, while allowing too much interference results in a “decoupling” of frame allocation such that previously allocated frames are not considered in subsequent allocations. Network isolation in the high-pathloss case means we require a lower $\rho_{\text{Int}}$ to see network gains.

Since we have employed frame-wise decomposition, the next frame to be allocated in the NFIC will be considered in isolation; hence we know that at most one new link will be introducing interference to frame $k$. As such, we update NFIC edge weights in time $t$ as if any edge is used in isolation; to do so, we calculate the maximum power any node $z \neq x$ could use without causing more than $\hat{\gamma}_t(y)$ interference at $y$.

$$\hat{P}_t(z) \leq \hat{\gamma}_t(y) \cdot D_{z,y}^\alpha \quad \forall \ z \in \mathbb{N} \quad (23)$$

Evaluating (23) specifies the maximum power any terminal may use at time $t$ so as to cause too much interference on frame $k$. However, there are $k-1$ other frames which must be considered; some of which may be closer to terminal $z$ than frame $k$ in this timeslot, further limiting $z$’s power availability. We therefore update the power as the minimum of the previous maximum power or the maximum power as calculated above: $P_t(z) \leftarrow \min\{P_t(z), \hat{P}_t(z)\}$. We can now update all edges from $z$ at time $t$ according to

$$e_{t}^{x,w} \leftarrow \log_2 \left(1 + \frac{P_t(z)}{N^\alpha + \gamma_t(w)} D_{z,w}^\alpha \right) \quad \forall \ z, w \in \mathbb{N} \quad (24)$$

which is repeated for all $t$ in $L^k$. This redefines the space of feasible powers $\mathcal{P}$. With the NFIC so updated, we are ready to repeat the procedure and allocate resources for frame $k+1$. In this way—allocating resources for one frame optimally, then removing them from availability before allocating the next frame—we construct the master resource allocation for all frames in the network. In particular, the schedule is constructed from the sequence of routes: $S = \bigcup_k L^k = \bigcup_{t,k} I_t^k$.

The power allocation is the union of allocations at each timeslot, for each terminal: $p = \bigcup_{t,x} P_t(x)$. The resulting allocation is the approximate solution to (8) above, in which we have calculated $p$ and $S$ to maximize the minimum throughput among our flows.

D. Computational Complexity

A strength of our approach is its low complexity. Here we analyze the computational and storage requirements of the NFIC.

1) Calculations: Assume the overall schedule uses $T$ time slots. At each time slot, we must do node weight updates for all $N$ nodes. In the worst case (fully interconnected or “full-interference” network), this requires $N \oplus \text{operations},$ which here are minimizations. We must then do $N \cdot (N-1)$ edge updates for all outgoing edges. If we require $T$ timeslots for the schedule, this becomes $2TN^2$ node- and edge-weight updates. After computing the schedule, we must solve for the optimal power allocation, requiring exactly $T$ computations, and then update all other the edges in the graph accordingly. Since there are $N^2$ edges in each time slot, this update takes $TN^2$ steps. We repeat this for all $K$ frames scheduled, and arrive at the total

$$K(2T(N^2 - N) + TN^2) = KT^2(N^3 - 2) \quad (25)$$

In general, the length of the schedule (total number of timeslots used) is on the order of the number of terminals in the network, such that $T \geq N$, so that the complexity is $F(3N^3 - 2N^2)$, which is $O(N^3)$, considerably less than the $O(N^2)$ required for the exhaustive search, and also on the order of the best-known single-flow routing algorithms.

2) Storage Requirements: The $N$ nodes of the NFIC at any time period are associated with a weight and a history variable, and each of the $N^2$ edges are also weighted. As such the memory requirements for $T$ time periods are $T(2N^2)$, linear in the number of timeslots used and like the square of the number of terminals in the network.

E. Optimality of the NFIC Approach

The low complexity of our technique is one of its strengths; it results from decomposing the original problem in (10) into a series of problems (12). Since the NFIC and the operations we execute over it are drawn from a monotone semiring, we can show that the solution to (12) is strictly optimal for a given resource set $\mathcal{P}$; hence the first frame is allocated optimally, the second frame is allocated optimally given the first, etc. However, since our approach does not reallocate the first frame based upon the resources of the second, we cannot be said to solve (10) exactly. This is shown in Fig. 2 as a successive narrowing of solution spaces as frames are allocated.

Unfortunately (10) is both exponentially complex and non-convex, so a direct comparison between our solution and the optimal solution is impossible. However, we can examine our performance in two limiting network conditions. For instance, in a small network with few terminals and few flows (e.g. $N = 5$ and $F = 2$), our technique will result in a
strictly TDMA solution: the first frame will consume nearly all network resources, such that the second frame will not be allocated until the first has cleared the network. In small networks which operate in the high interference regime (in which noise power is greater than signal power), TDMA is known to be optimal [35]. Similarly, in the case of a large network but also few flows, network isolation will allow our technique to simultaneously schedule frames for each flow, each one optimally as a result of our semiring construction; this is also the optimal overall solution.

While our technique is optimal in these boundary cases, it is in cases where interference is most problematic–large networks with many flows—that we cannot evaluate optimality directly. Cutset bounds may be calculated for our network, but their generality does not impose the scheduling and routing constraints we consider, resulting in a bound too loose to be of use. This routing constraint inspires the bounding result we present in Figure 7. Here, we bound the sum throughput for our flows by optimizing the first hop from their sources. If the first hop is optimized, no other allocation can possibly outperform it, hence this is an upper bound on performance. To establish this value, we search over all $N^F$ possible for hops and solve the power allocation problem (which is non-convex) using a random-restart technique. To illustrate the complexity of this problem, a network of $N = 40$ and $F = 5$, the parameters we use in our simulations, has approximately 102 million possible first hop combinations. We show an example of possible first-hops on the right side of Fig. 7: of these, we search over all combinations of active links.

Using this method, we are able to reduce network overhead requirements from learning $N^2$ to learning just $C \times (\frac{N}{C})^2$ channels, if we form $C$ neighborhoods. Performance degrades since interference is not managed across the network, but this trades off with the overhead required and the network coherence time—for rapidly changing topologies, the goodput of the distributed method is considerably larger than if the full-information solution is calculated. An extension of our approach could account for this overhead cost in the utility function of (10), subtracting from the rate of the link the amount of bits of overhead required to establish the connection.

VI. SIMULATION RESULTS

In the preceding sections, we have presented a low complexity algorithm to perform resource allocation in large networks. Here we show the performance of our techniques relative to other low-complexity approaches, and also present some network-layer implications of the NFIC allocation algorithm.

A. Scheduling and Interference Constraints

To illustrate how we handle the duplexing constraints on terminals and how interference is managed, we present an example allocation for one frame from each of three flows in Figure 8. The lines represent active links, enumerated by timeslot. For example, the green flow is active in timeslots $\{1, 2, 3, 4\}$. The longest flow (dotted blue) is allocated first, and follows a more or less straight-line path from its source to its destination. After that allocation, half-duplex constraints on the terminals used by the blue flow begin to bind, preventing the second (dashed red) flow from using any terminals occupied by the blue frame. For the red flow, the interference constraint binds more tightly; because as the blue frame is making its way ‘south,’ the red frame would have to lower its power if it took the direct path towards its destination, to avoid interfering with the blue frame. Instead, the NFIC algorithm determines that a better solution is a detour to the left, which allows the red frame to use higher power and achieve better throughput–spatially avoiding the blue frame.

The allocation for the third flow (solid green) demonstrates how the red frame took priority in that area. Because terminal
Each flow is given one dedicated timeslot to transmit data. To compare our scheme with another low-complexity scheme, we choose a round-robin (RR) allocation in which the two allocation techniques. The relative gain is calculated as:

\[ \text{gain} = \frac{T_{\text{NFIC}} - T_{\text{RR}}}{T_{\text{RR}}} \]  

(26)

Because the NFIC allocation removes resources only from those terminals already used by another frame and only scales resource availability at other terminals, simultaneous scheduling is possible subject to interference constraints. As such we simulate 1000 random topologies for a variety of network loads (number of flows), to study how simultaneous scheduling opportunities can improve network performance. We report our results in Figure 9, for a variety of network isolation levels (pathloss values). Note that for low isolation (\( \alpha = 2 \)), NFIC and RR perform roughly equally for all traffic loads. This suggests that simultaneous scheduling—while permitted by the scheduling constraints—is being prevented by the interference constraints. As the pathloss factor increases, opportunities for simultaneous network use increase, and NFIC scheduling begins to outperform RR allocation considerably, with up to five times higher throughput achieved under light traffic loads.

C. Simultaneous Allocation and Intra-Flow Interference

Simultaneous allocation can occur for frames in the same flow, provided the scheduling constraints remain satisfied. The fully connected nature of our network means that these frames will interfere with each other, an effect we term *intra-flow* interference. This is shown spatially in Figure 10, where frames from a single flow follow different routes to the same destination. Duplexing constraints permit source \( s \) to transmit in all timeslots, but multi-hop terminals \( x \) and \( y \) can only serve one frame for every two timeslots. However, because they are spatially separated, they may be active at the same time: significantly increasing throughput. In this example with moderate isolation (\( \alpha = 3 \)), the route multiplicity increased throughput by 20%.

We study this on a larger scale in Figure 11. Here, NFIC scheduling algorithms run for three flows with 50 frames each, in a network with a mean of 40 terminals. We count the mean number of routes frames use between source and destination, and plot that against the mean throughput increase over single-route allocation for all flows. Even when we artificially limit the routes available to only two, we see an increase over single-path routing on the order of 10%. When we allow four and five routes for the frames, we see throughput on average 40% better than single-route allocation, and for some topologies the increases are nearly double. This is because as route multiplicity increases, the duplexing penalty is mitigated and interference management becomes easier, hence all frames may be transmitted at higher rates. This illustrates an important consequence of NFIC allocation: that, in contrast to conventional routing schemes which establish a single route for the frames in a flow, the spatial-temporal network representation in the NFIC allows us exploit more available resources in the network, to achieve a higher throughput. This is further demonstrated in Figure 12, where we show how the number of routes a given flow uses increase with resource competition: as more flows require access to the network, NFIC allocation routes frames in different ways to maximize spatial and temporal reuse. In lightly loaded networks with few flows, route diversity is low, but it increases considerably with network load. Both of these results are independent of
among them can be successfully managed. This offers a new
slide with each other, but can exist together if the interference
of networks 

with 3 flows of 50 frames each. The number of routes available to each flow is
artificially constrained to be in \( \{2, 3, 4, 5\} \), and the mean across all three after
NFIC allocation is reported here. As frames are permitted to take more routes
through the network, mean throughput increases. Figures here are reported for
\( \alpha = 2 \).

pathloss, since route selections are made based only on relative
link capacities; hence these curves remain the same for values of
\( \alpha \neq 2 \).

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a general information
communication model which captures both physical and
network layer effects. This formulation permits the study of a
wide variety of physical layer technologies in the context of
medium-access control algorithms, the first steps of which we
have undertaken here. While an exhaustive search is required
to solve the original NP-Hard problem, we have developed a
polynomial-time decomposition technique which yields good
results under all network loads.

Our guiding notion is that frames should not necessarily col-
lude with each other, but can exist together if the interference
among them can be successfully managed. This offers a new
degree of freedom—the continuous power domain—to exploit
when computing resource allocations for large networks. We
have demonstrated that such a model can be created and studied, and we have shown results that an interference-
aware approach to large-scale resource allocation significantly
outperforms conventional techniques.

Our model and technique pave the way towards a more
comprehensive study of cross-layer design, where physical
layer technologies expressed as information theoretic qua-
tities may be evaluated at the wide-area level and on the
medium-access layer. This paper has shown the benefits of
efficient interference management and spatial re-use, demon-
strating performance gains beyond schemes of comparable
complexity. In the future, the generality of our framework and
algorithms will allow for the study of more advanced physical
layer technologies such as cooperation, where interference
among cooperating terminals increases throughput. In this
networks, medium-access schemes must be aware of physical-
layer interference at all times in order to realize gains.

The generality and efficiency of our approach permits its
application to a wide variety of networks and optimality cri-
cera; here we studied a fully ad-hoc network for max-
min throughput, but a semiring may be defined around a
delay-constrained network just as easily and may be applied
to networks with access points and disconnected terminals,
simply by modifying the NFIC. The cubic-time nature of our
scheme allows for quick computation of resource allocation,
making for rapid comparison between multi-flow allocation
and traditional techniques.

REFERENCES

the capacity of large extended ad hoc wireless networks,” IEEE Trans.
gap in the capacity of wireless networks via percolation theory,” IEEE
wireless networks: information theoretic and physical limits,” IEEE
routing,” in Proc. 1999 IEEE Workshop on Mobile Computing Systems
and Applications, pp. 90–100.
distance-vector routing (DSDV) for mobile computers,” ACM SIG-
vol. 16, pp. 87–90, 1959.

Fig. 11. Relative throughput gain over single-route allocation for networks
with 3 flows of 50 frames each. The number of routes available to each flow is
increases as the number of flows increases and competition for resources increases,
NFIC allocation is reported here. As frames are permitted to take more routes
through the network, mean throughput increases. Figures here are reported for
\( \alpha = 2 \).

Fig. 12. Number of routes frames take as a function of flows in the network,
where \( A = 20, \lambda = 1, \text{ and } \alpha = 2 \). 50 frames are allocated for each flow. As
the number of flows increases and competition for resources increases, NFIC
allocation results in frames traveling through the network in an increasing
variety of ways.
L. Chen, S. H. Low, M. Chien, and J. C. Doyle, “Cross-layer congestion

P. Raghavan and C. D. Thompson, “Randomized rounding: a technique

R. Madan, D. Shah, and O. Leveque, “Product multicommodity flow in

C. Scheideler, A. W. Richa, and P. Santi, “An o(log n) dominating set

T. ElBatt and A. Ephremides, “Joint scheduling and power control for


Gareth Middleton was born in Johannesburg, South Africa. He received the B.S.E.E. (with highest honors), M.S., and Ph.D. degrees from Rice University in 2005, 2007, and 2011 respectively. From 2005 to 2011, he was a Texas Instruments Distinguished Fellow at Rice, where he worked as a researcher in the Center for Multimedia Communication. In 2005 and 2007, he was a visiting researcher at Nokia Technology Platforms in Oulu, Finland. He is currently a post-doctoral researcher at Rice, where he pursues his interests in large-scale wireless networking, information theory, and mixed-integer programming. Dr. Middleton is a member of the IEEE,Eta Kappa Nu, Tau Beta Pi, and Phi Beta Kappa.