Distributed Scheduling and Delay-Aware Routing in Multi-Hop MR-MC Wireless Networks

Xianghui Cao, Member, IEEE, Lu Liu, Student Member, IEEE, Wenlong Shen, Student Member, IEEE, and Yu Cheng, Senior Member, IEEE

Abstract—In multi-radio multi-channel (MR-MC) networks with significantly expanded network resource space, many existing scheduling/routing algorithms rely on link based network model and apply different heuristics in algorithm design in order to achieve/approximate throughput optimality. In this paper, using a tuple-based multi-dimensional conflict graph model, we establish a cross-layer framework which facilitates systematically studying distributed scheduling and routing in multi-hop multi-path MR-MC networks. In this framework, each tuple-link is installed with a routing controller which feeds controlled amounts of data to the tuple-link output queues for scheduling and transmission. We rigorously prove that, under a set of certain conditions, the network is queue stable in mean sense under the distributed maximal scheduling policy. Based on Lyapunov optimization, we further propose a distributed delay-aware multi-path routing method which aims at minimizing the end-to-end delay of each commodity flow. Extensive simulation results demonstrate that the proposed joint scheduling/routing algorithm outperforms existing link based single-path and multi-path algorithms and tuple-based cross-layer control algorithm.

Index Terms—Multi-radio multi-channel networks, distributed scheduling, distributed delay-aware routing, queue stability

I. INTRODUCTION

Exploring the radio and channel diversities, multi-radio multi-channel (MR-MC) wireless networks can achieve significantly larger capacities than traditional single-radio single-channel (SR-SC) networks [1]–[4]. The MR-MC network model is a fundamental generalization of future wireless networks such as cognitive radio networks, IEEE 802.16 based mesh networks and the Long Term Evolution based cellular networks [5]–[7].

The problem of throughput-optimal scheduling of fundamental importance for achieving optimal performance while maintaining network stability has attracted much attention recently. In SR-SC networks, there are many works applying the back-pressure based algorithm to schedule links for data transmission [8], [9], where the scheduling decisions are usually made in centralized manners. The back-pressure scheduling is equivalent to solving a maximum weighted independent set (MWIS) problem, which is proven to be NP-hard. Suboptimal scheduling algorithms have been proposed to approximate throughput optimality with low computation complexity [10]–[12]. One popular example is the greedy maximal scheduling (also known as the longest queue first policy) which is based on the greedy MWIS algorithm [10]. Since centralized scheduling policies are not favored in large-scale networks, distributed scheduling policies have been studied. For example, in [13], distributed randomized scheduling policies are proposed where either each link randomly decides to participate in scheduling or the nodes randomly decide to setup connections between each other. As another approach, the distributed maximal scheduling allows at least one backlogged link in the interference set of a link be scheduled [14]. It has been shown that such an algorithm can achieve a capacity efficiency ratio (the largest achievable fraction of the optimal capacity region while guaranteeing network stability) of $\frac{1}{K}$ where $K$ is the interference degree, i.e., the maximum number of non-interfering links in the interference set of any link in the network [15].

In MR-MC networks, with significantly increased network dimension, link scheduling is often coupled with radio/channel assignment. There are only a few studies on designing and analyzing the performance of distributed scheduling policies in MR-MC networks. In [16], the distributed maximal scheduling policy has been applied in MR-MC networks based on link-channel pairs (LCPs), where a physical link is split into several LCPs with each LCP maintaining a queue to be considered in the maximal scheduling. A systematic approach that transforms an MR-MC network into an equivalent virtual SR-SC network through a tuple-based multi-dimensional conflict graph (MDCG) has been proposed in [17]. However, the analysis is constrained within single-hop networks.

Despite its advantage in achieving throughput-optimal scheduling in single-hop networks, the back-pressure based algorithm in multi-hop scenarios may incur excessively long delays and may not guarantee stability of the entire network [18]. On top of the scheduling policy, the network layer routing algorithm also plays a vital role for achieving optimal performance in multi-hop scenarios. There are several studies on joint scheduling and routing design [19], [20]. The back-pressure based scheme in [19] minimizes the lengths of the paths that can guarantee network stability. In [21], a joint congestion control, routing and scheduling algorithm is proposed which can guarantee both requirements of data rate and end-to-end delay. Several routing metrics for minimizing end-to-end delay are usually made in centralized manners. As another approach, the distributed maximal scheduling allows at least one backlogged link in the interference set of a link be scheduled [14]. It has been shown that such an algorithm can achieve a capacity efficiency ratio (the largest achievable fraction of the optimal capacity region while guaranteeing network stability) of $\frac{1}{K}$ where $K$ is the interference degree, i.e., the maximum number of non-interfering links in the interference set of any link in the network [15].
end delay in MR-MC networks are designed and implemented in [22]. However, many of them rely on centralized routing, which may be impractical for large-scale multi-hop networks. A distributed multi-path routing policy has been proposed in [16] in which the source node of a flow determines the fraction of the flow to be routed on each of the pre-determined paths. The fraction decisions are made by solving an optimization problem which is based on the heuristic of minimizing the congestion cost (which can be also viewed as total queueing delay). A distributed path selection algorithm is designed in [17], where the optimal path of a flow is selected such that the weighted sum of queueing delay and number of hops along the path is minimized. Since end-to-end delay consists of not only queueing delay but also transmission delay as well as the delay due to the scheduling policy used, neither of the above routing methods manifests the true delay in the routing decision process.

In MR-MC networks with a multi-dimensional resource space, many existing scheduling/routing algorithms rely on link based network model and apply different heuristics in algorithm design in order to achieve/approximate throughput optimality. Different from these studies, in this paper, we propose a cross-layer framework which facilitates systematic study of distributed scheduling and routing in multi-hop multi-path MR-MC networks. The framework is established based on the tuple-link based MDCG network model where a tuple is defined as a node-radio-channel resource vector in the multi-dimensional resource space. The transmitter tuple of each tuple-link implements a routing controller which feeds data traffic to the link’s output queue for further scheduling and transmission. An important role of the controller is to shape the incoming traffic from preceding tuple-links along the paths of a flow so as to maintain network stability. The function of the controller is similar to that of the regulator proposed in [15], [17]: a major difference in our case is that the parameters taken by the controllers are important routing variables that significantly vary the end-to-end delay performance.

We study the distributed joint scheduling and routing in the context of multi-commodity flow problem. We assume a set of pre-discovered potential paths for each flow and that each path delivers a fraction, which can be tuned dynamically, of the total amount of arrivals of the corresponding flow. The scheduling algorithm extends the maximal scheduling in traditional SR-SC networks into tuple-based MR-MC networks. By analyzing the drift of well-defined Lyapunov functions, we show that, under certain conditions, the network queue stability under distributed maximal scheduling is guaranteed in mean sense. Furthermore, based on the Lyapunov optimization method, we propose a multi-path routing optimization problem which aims at minimizing the end-to-end delay of each flow with both queueing delay and scheduling delay taken into account. Based on this, we develop a distributed delay-aware routing (DDAR) algorithm to decide the traffic fractions taken by the paths, where we apply the minimum-consensus algorithm [23] to ensure that the constraints of the routing optimization problem can be satisfied in a distributed manner. Finally, we conduct extensive simulations to evaluate the performance of the proposed method. The results show that our method outperforms existing algorithms including the link based single-path and multi-path schemes proposed in [16] and the cross-layer control based scheme proposed in [17].

The remainder of this paper is organized as follows. Section II presents system model. The scheduling policy and routing algorithm are described in Section III and IV, respectively. Section V presents simulation results, and Section VI concludes this paper.

II. System Model

In this section, we show that an MR-MC network can be represented as a tuple-based virtual SR-SC network, based on which we propose a cross-layer framework for designing scheduling and routing algorithms. We also describe the queue dynamics in such a framework. Notations used throughout this paper are summarized in Table I.

A. Tuple-based Generic Model of MR-MC Networks

Consider an MR-MC wireless network as an undirected graph \( G_p(N, L_p) \) with node set \( N \) and physical link set \( L_p \), where we assume each node uses the same transmit power. Each node \( i \) is equipped with a set of radios and each of the radios can operate on a set of orthogonal channels. A tuple \( p_i = (n_i, r_i, c) \) is a vector in the multi-dimensional network resource space, which means the resource allocation of node \( n_i \), one of its radios \( r_i \) and its operating channel \( c \). A tuple-link \( \ell = (p_i, p_j) = ((n_i, r_i, c), (n_j, r_j, c)) \) represents that the transmission from the radio \( r_i \) of node \( n_i \) to the radio \( r_j \) of node \( n_j \) is carried out over channel \( c \). For any tuple-link \( \ell \), its capacity \( w_\ell \) is also the capacity of \( \ell \)'s corresponding physical link. Based on the tuple-link model, a physical link connecting nodes \( n_i \) and \( n_j \) can be mapped into \(|M_i| \times |M_j| \times |C|\) tuple-links with each link indicating an assignment of radios and channel. With the fine-grained tuple-based modeling technique [5], the whole network can be mapped into a virtual SR-SC network as graph \( G(T, L) \) with vertex (tuple) set \( T \) and tuple-link set \( L \). In the following, if not specified, the terms link and tuple-link are used interchangeably.

Two tuple-links \( \ell_1 \) and \( \ell_2 \) are said conflicting with each other (i.e., they cannot be simultaneously active for data transmission) if either of the following conditions is met: (1) their beginning or ending tuples share a common radio; (2) they operate on the same channel and are physically within each other’s interference range. Based on the conflict relationships among all the tuple-links, a multi-dimensional conflict graph (MDCG) \( G_c(L, X) \) can be established with vertices as all the tuple-links and edges as the set of conflict relationships. For each \( \ell \in L \), denote \( I_\ell \) as the set of all tuple-links that have edges with \( \ell \) in \( X \). By convention, let \( \ell \in I_\ell \). Assuming the protocol interference model [17] with which a scheduling policy should make sure that, if \( \ell \) is scheduled, none of the links in \( I_\ell \) other than \( \ell \) is scheduled simultaneously. For \( \ell \), define its interference degree \( K_\ell \) as the maximum number of tuple-links in \( I_\ell \) that can be scheduled at the same time. Let \( K = \max_\ell K_\ell \) be the network interference degree.

We study the scheduling and routing issues in the context of multi-commodity flow problem in which there are a set of
flows $\mathcal{F}$ injected into the network. For each flow $f \in \mathcal{F}$, the average traffic arrival rate at its source node is $\lambda_f$. Take the simple network shown in Fig. 1(a) for example, where each node has one or two radios and there are one channel available (thus each node has one or two tuples as shown in this figure). There are two commodity flows between two pairs of source and destination nodes, respectively. Each flow may go through multiple multi-hop paths until reaching its destination. The problem is to design proper scheduling and routing policies to deliver as much flow to corresponding destinations as possible with low end-to-end delay. We consider a slotted system in which time is divided into slots of unit length. The scheduling with low end-to-end delay. We consider a slotted system in 

**B. Queue Dynamics**

We assume that each commodity flow $f$ is associated with a set of candidate paths $\mathcal{P}^f$. The incoming packets of flow $f$ in slot $t$ will be routed through the path $\mathcal{P}^{f,i}$ with probability $\eta^{f,i}(t)$. By applying the MDCG based off-line planning technique [17], a relatively small set of $\mathcal{P}^f$ and the corresponding long-term average of $\{\eta^{f,i}\}$ for each flow $f$ can be obtained in a centralized manner. Alternatively, as discussed in [16], the candidate set $\mathcal{P}^f$ can be discovered online by the source node of each flow. In the following, we assume that the paths $\mathcal{P}^f$ have been discovered and are fixed over time.

The transmitter node of link $\ell$, i.e., $b(\ell)$, maintains both an input and an output queues, i.e., $Q_{\text{in},\ell}$ and $Q_{\text{out},\ell}$, respectively, to buffer the packets of flow $f$ to be delivered over path $\mathcal{P}^{f,i}$, where $\ell$ is on path $\mathcal{P}^{f,i}$. For each link $\ell$, its transmitter node $b(\ell)$ is installed with a routing controller which maintains the set of input queues $\{Q_{\text{in},\ell}^f | f \in \mathcal{F}, i \in \mathcal{P}^f, \ell \in \mathcal{L}^{f,i}\}$ and feeds controlled amounts of data to the corresponding output queues for transmission. If $\ell$ is on none of the candidate paths of flow $f$, we simply have $Q_{\text{in},\ell}^f(t) \equiv 0$ and $Q_{\text{out},\ell}^f(t) \equiv 0$. In the following, for ease of exposition, we focus only on links such that each of them is on at least one path. That is, by using $Q_{\text{in},\ell}^f$ and $Q_{\text{out},\ell}^f$, we imply that $\ell \in \mathcal{L}^{f,i}$. The input and output queues are illustrated in Fig. 1(b).

The routing controller of link $\ell$ is characterized by a set of parameters $\{\mu^{f,i}_{\ell} | \ell \in \mathcal{L}^{f,i}\}$. As shown later in this paper, the routing controller plays an important part in both ensuring network stability and making routing decisions.

1 Dynamics of input queue $Q_{\text{in},\ell}^f$: If $b(\ell)$ is the source of flow $f$, for each path $\mathcal{P}^f$, the arrivals of the flow will be buffered in $Q_{\text{in},\ell}^f$ with probability $\eta^{f,i}(t)$; otherwise, if $b(\ell)$ is not the source of flow $f$, $Q_{\text{in},\ell}^f$ buffers packets from $B_{\ell}^{f,i}$. The routing controller of $\ell$ randomly checks $Q_{\text{in},\ell}^f$ with probability $\frac{1}{\Gamma}$, where $\Gamma \geq 1$. At the beginning of the check-point slot (say, slot $t$), if $Q_{\text{in},\ell}^f(t) > \Gamma w_{\ell}$, amount of packets will be transferred from the input queue to the corresponding output queue $Q_{\text{out},\ell}^f$ with probability $\frac{1}{\Gamma w_{\ell}} \mu^{f,i}_{\ell}(t)$, where $\mu^{f,i}_{\ell}(t) \in \{0, 1, \ldots, |\mathcal{L}^{f,i}| - 1\}$ is the hop count. Otherwise, nothing will be delivered from $Q_{\text{in},\ell}^f$ to $Q_{\text{out},\ell}^f$. At slot $t$, for each link $\ell$, the expected amount of packets actually delivered from $Q_{\text{in},\ell}^f$
to \(Q_{\text{out,}t}^{f,i}\) is

\[
\mathbb{E}[D_{\ell}^{f,i}(t)] = \frac{1}{\Gamma} \sum_{i} 1_{\{Q_{m,\ell}^{f,i}(t) > \Gamma w_{\ell}\}} \frac{\mu_{h_{\ell}^{f,i}}^{f,i}(t)}{w_{\ell}},
\]

(1)

where \(1_{\text{condition}}\) is 1 if “condition” holds and 0 if otherwise. Further, define \(T_\ell\) as the total amount of transferred data out from the routing controller of link \(\ell\), i.e.,

\[
D_{\ell}(t) = \sum_{f \in F, i \in P_f} D_{\ell}^{f,i}(t).
\]

(2)

Then, the dynamics of the input queue can be written as:

\[
Q_{\text{in,}t}^{f,i}(t + 1) = \left[Q_{\text{in,}t}^{f,i}(t) - D_{\ell}^{f,i}(t)\right]^{+} + T_{B_{\ell}}^{f,i}(t) 1_{b(t) \neq s(f)} + \lambda_{\ell}(t)\eta_{\ell}^{f,i}(t) 1_{b(t) = s(f)}.
\]

(3)

**Remark 1**: The input queues can be viewed as buffers to cope with arrival fluctuations. By using \(\Gamma\), the arrivals are assembled into blocks of packets which provide more stable input to the output queues. \(\Gamma\) can be set to 1, which allows that each input queue is checked in every time slot. Other transmission rules are also allowed as long as the average amount of packets delivered from input to output queues is controlled by the decision variable \(\mu_{h_{\ell}^{f,i}}^{f,i}\). The condition \(Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell}\) may also be dropped, in which case the queue stability theorem still holds.

2) **Dynamics of output queue \(Q_{\text{out,}t}^{f,i}\)**: Define the total length of the output queues of link \(\ell\) as

\[
Q_{\text{out,}t}^{f,i}(t) = \sum_{f \in F, i \in P_f} Q_{\text{out,}t}^{f,i}(t).
\]

(4)

A link \(\ell\) is said backlogged if its total output queue length exceeds \(\Gamma w_{\ell}\), i.e., \(Q_{\text{out,}t}^{f,i}(t) > \Gamma w_{\ell}\), where \(\Gamma > 0\). At each slot, only backlogged links will be considered for scheduling. Define a binary variable \(\pi_\ell\) as follows. If \(\ell\) is not scheduled, \(\pi_\ell(t) = 0\) and nothing will be transmitted over link \(\ell\) at slot \(t\). If \(\ell\) is backlogged and scheduled at slot \(t\), \(\pi_\ell(t) = 1\) and totally \(w_{\ell}\) amount of data will be transmitted from \(Q_{\text{out,}t}^{f,i}\) to the incoming queues of next-hop links. The detailed strategy can be as follows. For each pair of \((f, i)\) such that \(\ell \in L_{f,i}\),

\[
\frac{Q_{\text{out,}t}^{f,i}(t)}{Q_{\text{in,}t}^{f,i}(t)} w_{\ell}\text{ amount of data will be transmitted from } Q_{\text{out,}t}^{f,i}\text{ to } Q_{\text{in,}t}^{f,i},
\]

It is worth noticing that, as indicated later by the proof of Theorem 1, how \(w_{\ell}\) is distributed to the transmissions between \((Q_{\text{out,}t}^{f,i}, Q_{\text{in,}t}^{f,i})\) does not affect the network queue length stability.

At slot \(t\), if \(\ell\) is backlogged, the amount of packets actually transmitted from \(Q_{\text{out,}t}^{f,i}\) to \(Q_{\text{in,}t}^{f,i}\) over \(\ell\) is

\[
T_{\ell}^{f,i}(t) = \frac{Q_{\text{in,}t}^{f,i}(t)}{Q_{\text{out,}t}^{f,i}(t)} w_{\ell}\pi_\ell(t),
\]

(5)

\[
T_{\ell}(t) = \sum_{f \in F, i \in P_f} T_{\ell}^{f,i}(t) = w_{\ell}\pi_\ell(t).
\]

(6)

C. **Scheduling and routing**

Based on the above framework, the scheduling and routing decisions are made by the routing controllers. Specifically, a distributed scheduling policy should be able to decide \(\pi_\ell(t)\) based on only local information (particularly the information shared within \(\mathcal{I}_\ell\)). A routing algorithm should determine the values for \(\{\mu_{h_{\ell}^{f,i}}^{f,i}(t)\} \ell \in L_{f,i}\) and \(\{\eta_{\ell}^{f,i}(t)\}\). The scheduling and routing methods must ensure the network queue stability in mean sense, i.e., \(\lim_{T \to \infty} \frac{\Gamma}{T} \sum_{t=1}^{T} \mathbb{E}[Q_{\text{out,}t}^{f,i}(t)] < \infty\) and \(\lim_{T \to \infty} \frac{\Gamma}{T} \sum_{t=1}^{T} \sum_{i \in E} \mathbb{E}[Q_{\text{in,}t}^{f,i}(t)] < \infty\). In the following two sections, we develop a maximal scheduling based distributed scheduling algorithm and a delay-aware routing algorithm, respectively.

III. **DISTRIBUTED SCHEDULING ALGORITHM**

In this section, we first propose a maximal-scheduling-based distributed link scheduling policy for multi-hop MR-MC networks with a generic routing strategy. We then study the throughput and the capacity efficiency of the proposed scheduling policy. Based on the MDCG, the distributed maximal scheduling for SR-SC networks can be readily extended to that for MR-MC networks as follows.

**Distributed maximal scheduling algorithm**: starting with the set \(\mathcal{L}\), we randomly pick up a link, say \(\ell\), and check whether it is backlogged or not. If so, \(\ell\) is scheduled for data transmission; meanwhile, all the links in \(\mathcal{I}_\ell\) are removed from \(\mathcal{L}\) and will not be considered for scheduling. Then, we run the above process based on the residual set \(\mathcal{L} \setminus \mathcal{I}_\ell\). The algorithm stops when the residual set becomes empty.

1) **Stability conditions**: By the above maximal scheduling policy, if \(\ell\) is backlogged, at least one of the links in \(\mathcal{I}_\ell\) is scheduled, i.e.,

\[
\sum_{t \in \mathcal{I}_\ell} \pi_\ell(t) \geq 1.
\]

(8)

**Theorem 1 (Queue stability)**: For a multi-hop MR-MC network, if \(\exists a_{\max} \in (0, 1)\) and \(\exists a_{\min} > 0\) such that the following conditions hold for all \(t\):

\[
\sum_{\ell \in \mathcal{I}_\ell} \sum_{f,i \in P_f} \frac{\mu_{h_{\ell}^{f,i}}^{f,i}(t)}{w_{\ell}} 1_{Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell}} \leq a_{\max}, \quad \forall \ell;\]

(9)

\[
\mu_{0}^{f,i}(t) > \lambda_{\ell}\eta_{\ell}^{f,i}(t), \quad \forall f, i;\]

(10)

\[
\mu_{h_{\ell}^{f,i}}^{f,i}(t) \geq a_{\min} \forall f, i, \ell \text{ such that } Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell} \text{ and } Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell};\]

(11)

\[
\mu_{h_{\ell}^{f,i}}^{f,i}(t) - \mu_{h_{\ell}^{f,i}}^{f,i}(t) \geq a_{\min}, \quad \forall f, i, \ell \text{ such that } Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell} \text{ and } Q_{\text{in,}t}^{f,i}(t) > \Gamma w_{\ell};\]

(12)

then, the network is queue-length stable under the above distributed maximal scheduling algorithm.

The proof is provided in Appendix A. Basically, the proof consists of three steps. In the first step, the input queues of
each flow’s source node are proven stable. Then, we show that both \( Q_{\text{out}} \) and \( Q_{\text{in},N} \) are stable in mean sense, where \( Q_{\text{out}} \) and \( Q_{\text{in},N} \) are the vectors constructed by all \( \{Q_{\text{out},\ell}\} \) and all \( \{Q_{\text{in},N,f,i}\} \), respectively. This is achieved by analyzing the drift of the following Lyapunov function:

\[
V(Q_{\text{in},N}, Q_{\text{out}}) = V_1(Q_{\text{out}}) + \xi V_2(Q_{\text{in},N}, Q_{\text{out}}),
\]

where \( \xi > 0 \) and

\[
V_1(Q_{\text{out}}) \triangleq \frac{1}{2} \sum_{\ell \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}_f} \left( \frac{Q_{\text{out},\ell} + Q_{\text{in},N,f,i}}{w_\ell} \right)^2.
\]

Apparentl, \( V(Q_{\text{in},N}, Q_{\text{out}}) \) is radially unbounded and strictly positive unless \( Q_{\text{out},\ell} = 0 \) and \( Q_{\text{in},N,f,i} = 0 \) for all \( f, i, \ell \).

2) Scheduling efficiency: As compared to centralized optimal scheduling algorithms, the above distributed maximal scheduling which relies only on local information is suboptimal. To evaluate its performance, we define the following terms and performance metrics. Define the capacity region of a scheduler as the set of input \( \lambda \) under which the network remains stable, where \( \lambda = \{\lambda_1, \ldots, \lambda_{|\mathcal{F}|}\} \). The optimal capacity region \( \Lambda_{\text{opt}} \) of the network is defined as the supremum of all schedulers. A scheduler is throughput-optimal if it can achieve the optimal capacity region. It is thus interesting to investigating its scheduling efficiency. For such a suboptimal scheduler, define the capacity efficiency ratio as the largest \( \gamma \) such that the network is stable under any \( \lambda \in \gamma \Lambda_{\text{opt}} \). For the proposed scheduling algorithm, its scheduling efficiency is captured in the following Theorem. The proof can be found in Appendix B.

**Theorem 2 (Scheduling efficiency):** For a multi-hop MRMC network, the tuple-based distributed maximal scheduling can achieve a capacity efficiency ratio of \( \frac{1}{\gamma} \).

IV. DISTRIBUTED ROUTING ALGORITHM

In this section, we propose a distributed routing algorithm where the routing decisions are made at routing controllers of the source nodes. The routing decisions are based on solving a linear programming obtained by using the Lyapunov optimization technique.

A. Routing based on Lyapunov Optimization

For ease of presentation, \( \forall f, i, \ell \), let

\[
\begin{align*}
\dot{Q}_{\text{out},\ell}(t) &= Q_{\text{out},\ell}(t) + \sum_{f \in \mathcal{F}} Q_{\text{in},N,f,i}(t) \rightarrow \Gamma_{\ell,w_\ell} \\
\dot{Q}_{\text{out},\ell}(t) &= Q_{\text{out},\ell}(t) + \sum_{f \in \mathcal{F}} Q_{\text{in},N,f,i}(t) \rightarrow \Gamma_{\ell,w_\ell} \\
\dot{\mu}_{h_{\ell,i}} &= \mu_{h_{\ell,i}}(t) + \sum_{f \in \mathcal{F}} Q_{\text{in},N,f,i}(t) \rightarrow \Gamma_{\ell,w_\ell}.
\end{align*}
\]

Consider the Lyapunov function \( V \) as in (13), Combining (40), (41) and (43) in the proof of Theorem 1, we get

\[
\begin{align*}
\mathbb{E}[\Delta V(t)] &= \sum_{\ell \in \mathcal{L}} \dot{Q}_{\text{out},\ell}(t) \left( \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \sum_{\ell' \in \mathcal{P}_{\ell}} \frac{\mu_{h_{\ell,i}}(t) - \dot{\mu}_{h_{\ell,i}}(t)}{w_{\ell'}} - 1 \right) \\
&+ \xi \sum_{\ell \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \sum_{\ell' \in \mathcal{P}_{\ell}} Q_{\text{in},N,f,i}(t) + Q_{\text{in},N,f,i}(t) \\
&\times \left( \frac{\mu_{h_{\ell,i}}(t) - \dot{\mu}_{h_{\ell,i}}(t)}{w_{\ell'}} + C_0 \right) \\
&= \sum_{\ell \in \mathcal{L}} \dot{Q}_{\text{out},\ell}(t) \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \sum_{\ell' \in \mathcal{P}_{\ell}} \left( \frac{Q_{\text{out},\ell}(t) + Q_{\text{in},N,f,i}(t)}{w_{\ell'}} \right) \\
&\quad - \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \sum_{\ell \in \mathcal{L}} \dot{\mu}_{h_{\ell,i}}(t) \frac{Q_{\text{out},\ell}(t) + Q_{\text{in},N,f,i}(t)}{w_{\ell'}} + C_0. \quad (17)
\end{align*}
\]

where \( C_0 \) is a constant. In the above, for convenience, we define \( Q_{\text{out},\ell}(t) = 0 \) if \( b(f) = s(f) \) and \( Q_{\text{in},N,f,i}(t) = 0 \) if \( c(\ell) = d(f) \). Supposing that the MDCG is un-directional, we have that \( \forall f, \ell \in \mathcal{L}, \ell' \in \mathcal{I}_f \), then \( \ell \in I_{\ell'} \). Thus, we obtain (18) (which is shown in the next page), where \( \alpha_{f,i}(t) = \omega_1 - \epsilon_2 \omega_2 \) is a function of \( Q_{\text{out},\ell}(t) \) and \( Q_{\text{in},N,f,i}(t) \). For convenience, we further define \( J_{\ell}^{f,i}(t) = \omega_{\ell} \frac{Q_{\text{out},\ell}(t) + Q_{\text{in},N,f,i}(t)}{w_{\ell}} \).

**Remark 2:** In (18), \( \forall (f,i), \sum_{\ell \in \mathcal{L}} J_{\ell}^{f,i}(t) \) can be interpreted as follows.

- The term \( \omega_1 \) can be further written as

\[
\omega_1 = \frac{Q_{\text{out},\ell}(t)}{w_{\ell}} + \sum_{\ell' \in \mathcal{I}_f \setminus \{\ell\}} \frac{Q_{\text{out},\ell}(t)}{w_{\ell'}}, \quad (19)
\]

where the first part represents the queueing delay in the output queues (since the transmission rate \( w_{\ell} \) is limited and we assume the queues are first-in-first-out) while the second represents the delay due to our distributed scheduling policy.

- For each link \( \ell \) such that \( \ell \in \mathcal{L}^{f,i} \), let \( \tilde{Q}_{\ell}^{f,i} = Q_{\text{out},\ell} + Q_{\text{in},N,f,i} \) be the total length of both input and output queues associated with \( \ell \) with respect to flow \( f \) and path \( P^{f,i} \). Then, the input and output of \( \tilde{Q}_{\ell}^{f,i} \) are managed by the routing controllers with parameters \( \mu_{h_{\ell,i}}^{f,i} \) and \( \mu_{h_{\ell,i}}^{f,i} \).

We can view the routing controller with parameter \( \mu_{h_{\ell,i}}^{f,i} \) connecting between \( Q_{\text{out},\ell} \) and \( Q_{\text{in},N,f,i} \) as a virtual link \( \ell \).

Thus, the term \( \omega_2 \) in (18) can be viewed as the queue length (or backlog) over the virtual link \( \ell \). Based on (1), \( \dot{\mu}_{h_{\ell,i}}^{f,i} \) can be viewed as the link rate of \( \ell \). Thus, according to [17], the throughput-optimal policy by only
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2474695, IEEE Transactions on Vehicular Technology

\[
\mathbb{E}[\Delta V(t)|\mathbf{Q}_{\text{in},N}(t), \mathbf{Q}_{\text{out}}(t)] \\
\leq \sum_{f \in F} \sum_{i \in P_f} \sum_{\ell \in L^{f,i}} \frac{\mu^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}} \left\{ \sum_{\ell' \in L_{\ell}} \frac{Q_{\text{out},\ell'}(t)}{w_{\ell'}} - \xi \left( \frac{Q_{\text{out},B^{f,i}_{\ell}}(t) + Q_{\text{in},\ell}(t)}{w_{B^{f,i}_{\ell}}} - \frac{Q_{\text{in},N^{f,i}_{\ell}}(t)}{w_{\ell}} \right) \right\} \\
- \sum_{\ell \in L} \frac{Q_{\text{out},\ell}(t)}{w_{\ell}} + C_6 \\
\triangleq \sum_{f \in F} \sum_{i \in P_f} \sum_{\ell \in L^{f,i}} \sum_{\ell' \in L^{f,i}} \frac{\alpha^{f,i}(t) Q_{\text{out},\ell'}(t) > \Gamma w_{\ell}}{w_{\ell}} \mu^{f,i}_{h^{f,i}_{\ell}}(t) - \sum_{\ell \in L} \frac{Q_{\text{out},\ell}(t)}{w_{\ell}} + C_6, \quad (18)
\]

considering the virtual links can be expressed as:

\[
\max : \sum_{i \in P_f} \sum_{\ell \in L^{f,i}} \frac{\tilde{\mu}^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}} \omega_2 \\
\Leftrightarrow \min : - \sum_{i \in P_f} \sum_{\ell \in L^{f,i}} \frac{\tilde{\mu}^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}} \omega_2.
\]

In this sense, \(-\omega_2\) can be viewed as a measure of the virtual link queuing delay which is caused by both the routing decisions and queuing in the input queue.

Since \(\frac{\tilde{\mu}^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}}\) is the probability of data transferring from the input queue to the output queue of \(\ell\), the term \(\sum_{\ell \in L^{f,i}} \frac{\tilde{\mu}^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}} \omega_2 = \sum_{\ell \in L^{f,i}} \frac{\tilde{\mu}^{f,i}_{h^{f,i}_{\ell}}(t)}{w_{\ell}}(\omega_1 - \xi \omega_2)\) can be viewed as the expected end-to-end delay along the path \(P^{f,i}\), which consists of the delay due to distributed scheduling, routing as well as queuing in both the input and output queues of the links. Thus, by minimizing this value, we can improve the overall end-to-end delay performance.

With the aim to minimize the Lyapunov drift, we obtain a routing method by solving the following optimization problem.

\textbf{Problem 1 (Centralized routing optimization problem):}
Find the optimal \(\{\mu^{f,i}_{h^{f,i}_{\ell}}(t) | Q_{\text{in},\ell} > \Gamma w_{\ell}\}^3\) and \(\{\eta^{f,i}(t)\}\) to

\[
\min \sum_{f \in F} \sum_{i \in P_f} \sum_{\ell \in L^{f,i}} J^{f,i}(t) \quad (20) \\
\text{s.t.} \quad (9)-(12) \text{ hold,} \\
\sum_{i \in P_f} \eta^{f,i}(t) = 1, \quad \forall f, \\
\mu^{f,i}_{h^{f,i}_{\ell}}(t) > 0, \quad \forall f, i, \ell, Q_{\text{in},\ell} > \Gamma w_{\ell}, \\
\eta^{f,i}(t) \geq 0, \quad \forall f, i.
\]

\textbf{Remark 3:} In the literature, there are two representative routing algorithms based on the distributed maximal scheduling for MR-MC networks. In [16], for the traffic of flow \(f\), the multi-path routing algorithm aims at minimizing the congestion cost which is the summation of the queue lengths along the paths. Alternatively, the objective of the cross-layer control based routing algorithm in [17] is to minimize the traffic incurred within the network to support all commodity flows. The algorithm is further reduced to select the optimal path for each flow such that the weighted sum of the path length and the queue lengths along this path is minimized. However, these algorithms only focus on queuing delay in making routing decisions, while the scheduling delay induced by the distributed scheduling policy has not been considered.

The optimal solution to Problem 1 can only be obtained in a centralized way, which is inefficient for network with many flows. In the following, we propose a distributed routing algorithm to approximate the centralized optimal solution.

\textbf{B. Distributed Routing Algorithm Design}

For each path \(P^{f,i}\), in calculating the summation of \(J^{f,i}(t)\), we only need to consider those links on the paths of flow \(f\), i.e., \(\ell \in L^{f,i}\). The other information needed in calculating \(J^{f,i}(t)\) that relates to \(Q_{\text{in},\ell}\) and \(Q_{\text{out},\ell}\) can be collected by \(\ell\) itself and then transmitted to the source node. Therefore, the objective function \(\sum_{\ell \in L^{f,i}} J^{f,i}(t)\) can be evaluated locally by the source node of flow \(f\).

Although the conditions (10)-(12) are fully distributed, the condition (9) requires global information; that is, each source node of a flow needs to know the decisions of other flows. In the following, we propose a distributed method to decide \(\{\mu^{f,i}_{h^{f,i}_{\ell}}(t) | Q_{\text{in},\ell} > \Gamma w_{\ell}\}\) to satisfy condition (9) sufficiently.

\[\forall f, i \text{ and } \ell \text{ such that } Q_{\text{in},\ell} > \Gamma w_{\ell}, \text{ let}
\]

\[
e^{f,i}_{h^{f,i}_{\ell}}(t) \triangleq \mu^{f,i}_{h^{f,i}_{\ell}}(t) - \mu^{f,i}_{h^{f,i}_{\ell}-1}(t)1_{Q_{\text{in},\ell} > \Gamma w_{B^{f,i}_{\ell}}}.
\]

Corresponding to (11) and (12), \(e^{f,i}_{h^{f,i}_{\ell}}(t) \geq \epsilon_{\min}\), where \(\epsilon_{\min}\) can be chosen sufficiently small. Meanwhile, we assume that \(e^{f,i}_{h^{f,i}_{\ell}}(t) \leq \epsilon_{\max}(t)\) with \(\epsilon_{\max}(t)\) as a global value which will be discussed later. Let \(\mu^{f,i}_{h^{f,i}_{\ell}}(t) = \lambda_f e^{f,i}_{h^{f,i}_{\ell}}(t)\). Thus, along the path \(P^{f,i}\), (26) and (27) as shown in the next page can be
Thus, our distributed optimization problem can be formulated as follows.

Problem 2 (Distributed routing optimization problem):

For each flow $f$, find the optimal routing parameters $\{\eta^f(i)|i \in \mathcal{P}^f\}$ and $\{\epsilon^f_{h,i}(t)|Q_{in,f,t} > \Gamma w_t\}$ to

$$
\min \sum_{i \in \mathcal{P}^f} \sum_{t \in L} J^{f,i}_{\epsilon}(t)
$$

s.t. (25) holds, $\forall i, \epsilon, Q_{in,f,t} > \Gamma w_t$,

R.H.S. of (28) $< \epsilon_{\text{max}}$, $\forall \ell \in \mathcal{L}$,

$\mu_{0}^{f,i}(t) = \lambda_f \eta^{f,i}(t)$, $\forall i \in \mathcal{P}^f$

$\sum_{i \in \mathcal{P}^f} \eta^{f,i}(t) = 1$,

$\epsilon^{f,i}_{h,t}(t) \in [\epsilon_{\text{min}}, \epsilon_{\text{max}}(t)]$, $\forall i \in \mathcal{P}^f$

$\eta^{f,i}(t) \geq 0$, $\forall i \in \mathcal{P}^f$

To solve this problem, the first step is to determine the global value $\epsilon_{\text{max}}(t)$ in a distributed manner.

Step 1: For each flow $f$, let $\mathcal{F}_f$ be the set of flows whose paths traverse link $\ell$. Thus, for the source node of each flow $f$, it can obtain a local estimate of $\epsilon_{\text{max}}(t)$, denoted as $\epsilon^{f}_{\text{max}}(t)$, such that the R.H.S. of (28) is less than or equal to $\epsilon_{\text{max}}$ for all $\ell$ on the paths of $f$. Specifically, each link $\ell$ collects all available $\lambda_{\ell}$ $\epsilon_{\text{max}}(t)$ and $\{Q_{in,f,t}(t), Q_{out,f,t}(t)| \ell \in \mathcal{L}^{f,i}\}$ and sends them to the source nodes of the flows traversing it. Thus, $\epsilon^{f}_{\text{max}}(t)$ can be obtained locally by the source node of $f$ based on (31). Then, all the source nodes can apply a minimum-consensus algorithm (which runs in a fully distributed manner) [23] as follows to achieve a globally feasible $\epsilon_{\text{max}}(t) = \min_{f \in F} \{\epsilon^{f}_{\text{max}}(t)\}$.

Minimum-consensus algorithm for computing $\epsilon_{\text{max}}(t)$: The algorithm runs iteratively. Initially, each source node decides its own $\epsilon^{f}_{\text{max}}(t)$ based on the above discussion. In each iteration, each source node broadcasts its current $\epsilon^{f}_{\text{max}}(t)$ to its immediate neighbors. Upon receiving an $\epsilon^{f}_{\text{max}}(t)$, a source node compares the received $\epsilon^{f}_{\text{max}}(t)$ and that of its own and uses the smaller one of them to update its own value; while a non-source node will simply forward the smallest $\epsilon^{f}_{\text{max}}(t)$ it has received to its neighbors. The algorithm will finally converge to the maximum of the initial values held by the source nodes.

Step 2: With $\epsilon_{\text{max}}(t)$ obtained in Step 1, the source node of each flow can make routing decisions by solving Problem 2. Once the problem is solved, the source node will calculate $\{\mu_{0}^{f,i}(t)|h^{f,i}_t \geq 1, Q_{in,f,t} > \Gamma w_t\}$ based on (27) and then transmit them along each path $\mathcal{P}^{f,i}$ to the corresponding routing controllers. For those links with $Q_{in,f,t} \leq \Gamma w_t$, since the corresponding routing controller does nothing in this case, we can simply assign $\mu_{h^{f,i}_t}^{f,i}(t) = 0$.

Theorem 3: If at some time $t$ there holds that $\alpha^{f,i}(t) > 0$ for all $f, i, \ell$, then the optimal solutions for Problem 1 and Problem 2 coincide at this time. Moreover, the optimal solution is: $\forall f \in \mathcal{F}$ and $i \in \mathcal{P}^f$,

$$
\eta^{f,i}(t) = \begin{cases} 
1, & \text{if } i = \arg \min_{\ell \in L_f} \sum_{i \in \mathcal{P}^f} \alpha^{f,i}_w(t) \Xi^{f,i}_{\ell,t}(t), \\
0, & \text{otherwise}.
\end{cases}
$$

$$
\mu^{f,i}_h(t) = \eta^{f,i}_h(t) \lambda_f,
$$

$$
\epsilon^{f,i}_{h,t}(t) = \epsilon_{\text{min}}, \text{ if } Q_{in,f,t}^{f,i} \geq \Gamma w_t.
$$

The proof of the above theorem can be found in Appendix C. Notice that (36) implies that the optimal choice is to route the input through the path with minimum delay (see Remark 2), which matches our intuition. (38) can be interpreted as follows. When $\alpha^{f,i}_h(t) > 0$ for all $f, i, \ell$, the output queues are long; therefore we shall use the input queues to buffer more flow, i.e., to deliver as small amount of flow from input to corresponding output queues as possible. Based on the definition of $\alpha^{f,i}_h$ as in (18), the condition $\alpha^{f,i}_h(t) > 0$ can be easily satisfied if $\xi$ is small enough. $\alpha^{f,i}_h(t)$ becomes non-positive only when $Q_{in,f,t}^{f,i}$ is much larger than other queues in $w_1$ and $w_2$. In this case, the solution in the above theorem is: $\forall f \in \mathcal{F}$ and $i \in \mathcal{P}^f$,

$$
\eta^{f,i}(t) = \begin{cases} 
1, & \text{if } i = \arg \min_{\ell \in L_f} \sum_{i \in \mathcal{P}^f} \alpha^{f,i}_w(t) \Xi^{f,i}_{\ell,t}(t), \\
0, & \text{otherwise}.
\end{cases}
$$

$$
\mu^{f,i}_h(t) = \eta^{f,i}_h(t) \lambda_f,
$$

$$
\epsilon^{f,i}_{h,t}(t) = \epsilon_{\text{min}}, \text{ if } Q_{in,f,t}^{f,i} \geq \Gamma w_t.
$$
does not hold any longer. Instead, since the queueing delay in $Q_{in,f,i}(t)$ becomes a dominant factor, the better choice is to use a larger $\mu_{h,f,i}$ to deliver more flow from $Q_{in,f,i}(t)$ to $Q_{out,f,i}(t)$.

C. Discussions

Problem 2 is a linear programming with continuous decision variables which is not difficult to solve. The problem scale depends on the number and lengths of the paths as the number of decision variables is $|\{Q_{in,f,i}(t)\}| + |\{Q_{out,f,i}(t)\}| = |\mathcal{P}| + \sum_{i \in \mathcal{F}} |\mathcal{L}_{f,i}|$. Moreover, if $e_{f,i}^g > 0$ for all $f,i,\ell$, the optimal solution shown in Theorem 3 only requires simple calculations. In case that $Q_{in,f,i} > 0$ for all $f,i,\ell$, $\mu_{h,f,i}$ $\in$ $\mathcal{Q}$ which will further simply the solution in Theorem 3. However, for practical implementations, several issues need to be considered.

1) Convergence time: In each time slot $t$, the proposed algorithm runs iteratively until converging, and the convergence time is determined by the minimum-consensus algorithm. Providing that the network is connected, the convergence is guaranteed and the maximum convergence time is characterized by the network diameter [24].

2) Communication overhead: Both Step 1 and 2 incur extra communication overhead. Consider the links of non-source nodes. In Step 1, the information about $\{\lambda_f| \ell \in \mathcal{L}\}$ and $\{Q_{in,f,i}(t), Q_{out,f,i}(t)| \ell \in \mathcal{L}_{f,i}\}$, which amounts to a total of at most $\sum_{f,e} (1 + |\mathcal{P}^{f,i}|)$ values, is reported from the links to the source nodes. On average, each link should transmit and also forward at most $\sum_{f,e} \sum_{i \in \mathcal{F}} (1 + |\mathcal{L}_{f,i}|) / 2 \sum_{f,e} (1 + 2|\mathcal{P}^{f,i}|)$ values, where $|\mathcal{L}_{f,i}|$ is the average number of hops averaged over all links along $\mathcal{P}^{f,i}$. Also, the nodes are involved in exchanging the values of $\{e_{f,i}\}$ within their neighborhoods in order to run the minimum-consensus algorithm, which amounts to a total of at most $d_{\max} LD$ times of transmissions per link, where $d_{\max}$ and $L_D$ are the maximum node degree and the network diameter, respectively. In Step 2, each link should forward the routing decisions $\{y_{f,i}^{\ell}(t)\}$. Similarly as above, the average number of values a link should transmit is $\sum_{f,e} \sum_{i \in \mathcal{F}} (\mathcal{L}_{f,i} + 1)$. Suppose on average transmitting and receiving a value consume $E_{\text{comm}}$ amount of energy. Then the average communication overhead per link of each non-source node is at most

$$E_{\text{comm}} \sum_{f,e} \sum_{i \in \mathcal{F}} \left(1 + \frac{|\mathcal{L}_{f,i}|}{2}\right) \left[1 + \sum_{f \in \mathcal{F}} \left(1 + 2|\mathcal{P}^{f,i}|\right)\right],$$

which depends on the number of flows and the number and lengths of the paths.

3) Impact of delay: All the above decision-making related communications also incur some delay of response (i.e., each link has to wait some time for the source nodes’ decisions after reporting its collected data), which may vary the overall performance. However, our method can still guarantee network queue-length stability if the decision-making related communications can be performed on a separated channel and without interrupting the ongoing data transmissions. Assume network synchronization is achieved and there is no packet loss. Based on the path lengths, the maximum delay of all links can be evaluated. Thus, we can allocate a period of time (at least equals to the maximum delay) before the end of each time slot for the links to collect data and conduct the decision-making related communications. Since the data delivery from an input queue to its corresponding output queue can be completed instantly, the lengths of the input queues will remain unchanged during the allocated period; however, the lengths of output queues reported to the source nodes may still vary. Since the constraints of Problem 2 involve the lengths of the input queues but no output queues, the decisions made by the source nodes with accurate information about the input queues can guarantee those constraints and hence guarantee the network stability according to Theorem 1.

V. SIMULATION RESULTS

In this section, we present simulation results to evaluate the performance of the proposed joint scheduling and routing method. Applying the same distributed maximal scheduling algorithm, we also compare the proposed distributed delay-aware routing (DDAR), the single-path (SP) and multi-path (MP) routing algorithms proposed in [16] and the cross-layer control (CLC) based routing algorithm proposed in [17]. We developed C++ codes to implement all these algorithms. Simulations are conducted based on a random network topology with 25 nodes deployed in a 900m x 900m area, as shown in Fig. 2. The transmission and interference ranges of each node are set to 250m and 500m, respectively. To simulate the channel diversity, in each slot, the capacity of each link is uniformly generated in the range [0.5, 1.5] with an average of 1 unit (packets/slot). For a fair comparison, we fix $\Gamma = 1$ in implementing both DDAR and CLC. As shown in Fig. 2, there are three multi-hop commodity flows to be served by the network, each of which has the same average input rate $\lambda$. To account for flow dynamics, we assume the instantaneous input varies within [80%, 120%] $\lambda$. In the figure, the source and destination nodes for each flow are marked as $s(f)$ and $d(f)$, respectively. We gradually increase the average flow input rate $\lambda$ and observe the average per-node backlog (averaged over those nodes involved in flow transmissions). With a given input rate $\lambda$, i.e., the total amount of input traffic is given, the larger the average backlog is, the lower throughput the network can achieve. Also, the average backlog can serve as a measure of the network delay performance.

First, we study the performance of the proposed joint scheduling and routing algorithm by examining the average per-node queues. In Fig. 3, $Q_{out}$ and $Q_{in}$ are the output and input queue lengths, respectively, averaged over all nodes that participate in the flow transmission, i.e.,

$$Q_{in} = \frac{1}{|\Omega|} \sum_{k \in \Omega} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{F}} Q_{in,f,i}(t),$$

$$Q_{out} = \frac{1}{|\Omega|} \sum_{k \in \Omega} \sum_{f \in \mathcal{F}} Q_{out,f,i}(t),$$

where $\Omega$ is the set of nodes each of which has at least one incident link that lies on a candidate path of some flow. For better visibility, we compare $Q_{out}$ and $10Q_{in}$. From Fig. 3,
Comparing the two cases (where each node has 2 radios operating with 3 available channels in cases 1 and has 3 radios with 5 channels in case 2), we can observe that the performance is improved by using more radios and channels. As shown in both figures in Fig. 3, when the per-flow input rate increases, \( \bar{Q}_{\text{out}} \) starts to climb quickly when the input rate reaches a turning point (which is 0.35 as in Fig. 3(a) and 0.42 as in Fig. 3(b)). Such a turning point indicates that, with a higher input rate, the network queues \( Q_{\text{out}, \ell} \) will become unstable. Therefore, it also indicates the network capacity region under the distributed maximal scheduling. On the other hand, \( \bar{Q}_{\text{in}} \) remains almost stable as the flow input rate increases. A major reason is that, for each link \( \ell \), the data transferring from \( Q_{\text{in}, \ell}^{(i)} \) to \( Q_{\text{out}, \ell}^{(f)} \) is conducted within the corresponding tuple instantly and thus the amount is constrained by neither neighbors’ activities nor the scheduling policy. Scrutinizing the curves of \( \bar{Q}_{\text{in}} \), we can find an interesting phenomenon that \( \bar{Q}_{\text{in}} \) reaches a local maximal value at the turning point. Before reaching the turning point, the flow input rate is lower than the capacity that the scheduling algorithm can support; hence the output queue lengths remain low because their buffered packets will be served with high probabilities. When the input rate equals the turning point, the network capacity is reached, and thus the network reaches equilibrium and the input and output queues are balanced. As the input rate keeps increasing, the network becomes more and more congested and the throughput only increases slightly. Thus, the average input queue length \( \bar{Q}_{\text{in}} \) increase a little.

Based on the same distributed maximal scheduling policy, the performance of DDAR, SP, MP and CLC are compared. In DDAR and CLC, the set of candidate paths at the tuple-link level is obtained through the off-line planning technique [17]. In MP, the set of physical link-based candidate paths is extracted from the above tuple-based path set. The SP algorithm uses the shortest path for each flow. From Fig. 4, we can clearly observe that, in both cases, our DDAR algorithm outperforms the existing ones as the average backlog is always lower than those of the others and the turning point lies to the right of the others. As explained in Remark 3, compared with the other three algorithms, our routing algorithm takes more
accurate expectation of flow end-to-end delay into account and can achieve a better tradeoff between delay and path rate.

VI. CONCLUSION

We have proposed a cross-layer framework for studying distributed scheduling and routing in general MR-MC networks. Our scheduling policy is based on the distributed maximal scheduling implemented on the tuple-based MR-MC network model. Under this policy, the network stability is analyzed and a set of sufficient conditions are discovered. A distributed delay-aware routing method is proposed based on minimizing the Lyapunov drift. As different from existing distributed routing methods which only take partial end-to-end delay in making routing decisions, our method accounts for all the delay due to scheduling and queueing in both input and output queues of the links. Through extensive simulations, we showed that the queues maintained at the routing controllers are almost stable as the flow input rate increases. We also showed that the proposed joint scheduling and routing algorithm outperforms existing SP, MP and CLC algorithms in terms of average backlog.

APPENDIX A

Proof of Theorem 1: First, consider the stability of the input queues of incident tuple-links of flow source nodes. Based on (3), \( \forall \ell \in \{ \ell \mid f_\ell \in F, b(\ell) = s(f) \} \), we have

\[
E[\Delta Q_{in,\ell}^f(t)|Q_{in,\ell}^f(t)] \leq E[Q_{in,\ell}^f(t + 1) - Q_{in,\ell}^f(t)|Q_{in,\ell}^f(t)] = E[A_{\ell}^f(t) - D_{\ell}^f(t)|Q_{in,\ell}^f(t)] = \lambda f \eta^f_{\ell} - \mu^f_{\ell}(t)Q_{in,\ell}^f(t) + \Gamma w_{\ell} \leq \begin{cases} 0, & \text{if } Q_{in,\ell}^f(t) > \Gamma w_{\ell}, \\ \lambda f \eta^f_{\ell}(t), & \text{otherwise,} \end{cases}
\]

where we have used the condition (10). Thus, the stability of \( Q_{in,\ell}^f(t) \) is guaranteed in mean sense as the expectation of \( Q_{in,\ell}^f(t) \) will decrease or remain unchanged once \( Q_{in,\ell}^f(t) \) exceeds \( \Gamma w_{\ell} \).

In the second step, consider the Lyapunov function \( V_1(Q_{out}) \) as defined in (14). The drift is:

\[
\Delta V_1(t) = V_1(Q_{out}(t + 1)) - V_1(Q_{out}(t)) = \frac{1}{2} \sum_{\ell \in L} \sum_{\ell' \in T_{\ell}} \frac{Q_{out,e}(t) + D_{\ell'}(t) - T_{\ell'}(t)}{w_{\ell}} \times \left( \sum_{\ell' \in T_{\ell}} \frac{Q_{out,e}(t) + D_{\ell'}(t) - T_{\ell'}(t)}{w_{\ell}} - \frac{1}{2} \sum_{\ell' \in T_{\ell}} \frac{Q_{out,e}(t)}{w_{\ell}} \sum_{\ell' \in T_{\ell}} \frac{Q_{out,e}(t)}{w_{\ell}} \right) = \sum_{\ell \in L} \frac{Q_{out,e}(t)}{w_{\ell}} \sum_{\ell' \in T_{\ell}} \frac{D_{\ell'}(t) - T_{\ell'}(t)}{w_{\ell}} + \frac{1}{2} \sum_{\ell \in L} T_{\ell}(t) - D_{\ell}(t) \sum_{\ell' \in T_{\ell}} \frac{D_{\ell'}(t) - T_{\ell'}(t)}{w_{\ell}} \leq \sum_{\ell \in L} \frac{Q_{out,e}(t)}{w_{\ell}} \sum_{\ell' \in T_{\ell}} \frac{D_{\ell'}(t) - T_{\ell'}(t)}{w_{\ell}} + C_1
\]

\[
= \sum_{\ell \in L} \frac{Q_{out,e}(t)}{w_{\ell}} \sum_{\ell' \in T_{\ell}} \left( \frac{D_{\ell'}(t)}{w_{\ell'}} - \pi_{\ell'}(t) \right) + C_1.
\]
where \( C_1 \geq 0 \) is a constant. Inequality (39) holds because both \( T^{f,i}_\ell(t) \) and \( D^{f,i}_\ell(t) \) are finite values as constrained by \( w_\ell \) and \( \Gamma_\ell \), respectively. For any link \( \ell \), \( \sum_{\nu \in I_\ell} \pi_\nu \geq 0 \). In particular, if \( \ell \) is backlogged, we have (8). Thus, \( \Delta V_1(t) \leq \sum_{\ell \in L} \frac{Q_{\text{out},\ell}(t)}{w_\ell} \left( \sum_{\nu \in I_\ell} \sum_{f \in F_\ell} \sum_{i \in P_f} \mu_{h^{f,i}_\ell}^-(t) \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') \right) \left( \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') - 1 \right) + C_2, \) where \( C_2 \geq 0 \) is another constant. Furthermore, based on (1), \[ \mathbb{E}[\Delta V_1(t) | Q_{\text{out}}(t)] \leq \sum_{\ell \in L} \frac{Q_{\text{out},\ell}(t)}{w_\ell} \left( \sum_{\nu \in I_\ell} \sum_{f \in F_\ell} \sum_{i \in P_f} \mu_{h^{f,i}_\ell}^+(t) \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') \right) \left( \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') - 1 \right) + C_2, \] where the last inequality holds due to condition (9). Therefore, based on the Lyapunov drift theory [25], \( Q_{\text{out}} \) is stable in mean sense.

Finally, to prove the stability of \( Q_{\text{in},N} \), consider the Lyapunov function \( V(Q_{\text{in},N}, Q_{\text{out}}) \) as defined in (13). Similar to (39), \[ \Delta V_2(t) = V_2(Q_{\text{in},N}(t + 1)) - V_2(Q_{\text{in},N}(t)) \leq \sum_{\ell \in L} \sum_{f \in F_\ell} \sum_{i \in P_f} \frac{Q^{f,i}_{\text{out},\ell}(t) + Q^{f,i}_{\text{in},Nf,i}(t)}{w_\ell} \left( D^{f,i}_\ell(t) - D^{f,i}_{Nf,i}(t) \right) + C_3, \] where \( C_3 \) is a constant. Above we have used the fact that \( h^{f,i}_{Nf,i} = h^{f,i}_\ell + 1 \). Thus, \[ \mathbb{E}[\Delta V_2(t) | Q_{\text{in},N}(t), Q_{\text{out}}(t)] \leq C_3 + \sum_{\ell \in L} \sum_{f \in F_\ell} \frac{Q^{f,i}_{\text{out},\ell}(t)}{w_\ell} \left( \sum_{\nu \in I_\ell} \sum_{i \in P_f} \frac{\mu_{h^{f,i}_\ell}^+(t) Q^{f,i}_{\text{in},Nf,i}(t)}{w_\ell} \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') \right) \left( \sum_{\nu' \in I_\ell} D^{f,i}_\ell(t, \nu') - 1 \right) + C_4, \] where \( \rho_{\text{max}} \triangleq \max_{f,i,\ell} \left\{ \frac{w_\ell^2 Q^{f,i}_{\text{in},Nf,i}(t)}{w_\ell^2} \right\} \), \( \mu_{\text{max}}(t) \triangleq \max_{f,i,\ell} \left\{ \frac{\mu_{h^{f,i}_\ell}^+(t)}{w_\ell^2} \right\} = \max_{f,i,\ell} \left\{ \frac{\mu_{h^{f,i}_\ell}^+(t)}{w_\ell^2} \right\} \). In the above, (45) holds due to conditions (11) and (12). Combining the above inequality with (42), we get \[ \mathbb{E}[\Delta V(t) | Q_{\text{in},N}(t)] \leq - \mathbb{E} \left[ \sum_{\ell \in L} \frac{1}{w_\ell} \left( 1 - a_{\text{max}} - \xi \frac{\mu_{\text{max}}(t)}{w_\ell^2} \right) Q_{\text{out},\ell} \right] - \xi \sum_{\ell \in L} \sum_{f \in F_\ell} \sum_{i \in P_f} \frac{\epsilon_{\text{min}}}{\rho_{\text{max}} w_\ell^2} Q^{f,i}_{\text{in},Nf,i}(t) + C_2 + C_4, \] and that \( \sum_{\ell \in L} \sum_{f \in F_\ell} \sum_{i \in P_f} \frac{\epsilon_{\text{min}}}{\rho_{\text{max}} w_\ell^2} Q^{f,i}_{\text{in},Nf,i}(t) + C_2 + C_4, \)
The last inequality holds because of the stability of $Q_{\text{out}}$ which has been proven above. Let $\xi < (\frac{1}{2} - \frac{\xi_{\text{max}}}{\mu_{\text{max}}})W$, we will have $1 - \frac{1}{2} - \frac{\xi_{\text{max}}}{\mu_{\text{max}}} > 0$. Therefore, the Lyapunov function $V(t)$ will be negative if $Q_{f,i}^{f,i}$ is large enough, which implies that every $Q_{m,N}^{f,i}$ is stable in mean sense. Thus, the whole theorem is proven.

\section*{Appendix B}

Proof of Theorem 2: It is suffice to prove that for any $\lambda$ such that $K\lambda \in \Lambda_{\text{opt}}$, $\lambda$ can stabilize the network under the proposed scheduling algorithm. Since the network is stable under $K\lambda$, based on the definition of network interference degree, a necessary condition is that, $\forall \ell \in \mathcal{L}$,

$$\sum_{\ell' \in \mathcal{L} \setminus \ell} \frac{K_{\ell} \lambda_{\ell}}{w_{\ell'}} \psi_{\ell'} < K$$

\begin{equation}
\Rightarrow \sum_{\ell' \in \mathcal{L} \setminus \ell} \frac{\lambda_{\ell}}{w_{\ell'}} \psi_{\ell'} < 1. \tag{46}
\end{equation}

With $\lambda$ satisfies the above inequality, we are to construct the conditions in Theorem 1 such that the network is stable under $\lambda$ and the proposed scheduling algorithm. $\forall \ell, i$ such that $h_{f,i} > |\mathcal{L}| - 1$, let $\mu_{0}^{f,i}(t) = \lambda_{f} \eta f^{f,i}(t)$ which guarantees condition (10) and

$$\mu_{h_{f,i}+1}^{f,i}(t) - \mu_{h_{f,i}}^{f,i}(t) = \epsilon f^{f,i}(t).$$

Thus, along the path $p^{f,i}$, we have

$$\epsilon f^{f,i}(t) = \mu_{0}^{f,i}(t) + h \epsilon f^{f,i}(t),$$

where $h \in \{0, 1, \ldots, |\mathcal{L}| - 1\}$ is the number of hops away from the source tuple. Then, $\forall \ell \in \mathcal{L}$,

$$\sum_{\ell' \in \mathcal{L} \setminus \ell} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \frac{\mu_{i}^{f,i}(t)}{w_{\ell'}} \psi_{\ell'} \leq \sum_{\ell' \in \mathcal{L} \setminus \ell} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \frac{\lambda_{f} \eta f^{f,i}(t) + (|\mathcal{L}| - 1) \epsilon f^{f,i}(t)}{w_{\ell'}} \psi_{\ell'} \leq \max_{\ell} \left\{ \sum_{\ell' \in \mathcal{L} \setminus \ell} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \frac{\lambda_{f} \psi_{\ell'} \theta_{f,i}^{f,i}(t)}{w_{\ell'}} \psi_{\ell'} + \sum_{\ell' \in \mathcal{L} \setminus \ell} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \frac{(|\mathcal{L}| - 1) \epsilon f^{f,i}(t)}{w_{\ell'}} \psi_{\ell'} \right\}.$$

In the right-hand-side of the above formula, the first term is less than 1 according to (46), while the second term can be controlled to any small value by choosing proper $\epsilon_{\text{min}} > 0$ and $\{\epsilon f^{f,i}(t)\psi_{\ell'} \theta_{f,i}^{f,i}(t) \geq \epsilon_{\text{min}}\}$. Therefore, there exists $\alpha_{\text{max}} \in (0, 1)$ and $\epsilon_{\text{min}} > 0$ such that the conditions (9)-(12) in Theorem 1 are satisfied.

\section*{Appendix C}

Proof of Theorem 3: For the links \{\ell | Q_{f,i}^{f,i}(t) \leq \Gamma w_{\ell}\}, we already discussed that the decisions will be $\mu_{h_{f,i}+1}^{f,i}(t) = 0$. For the rest links, if $\alpha_{f,i}^{f,i}(t) > 0$, their $\mu_{h_{f,i}}^{f,i}(t)$ should be chosen as small as possible in order to minimize the objective functions in both Problem 1 and Problem 2. Thus, we obtain (37) and (38) for both Problems. Then, for each flow $f$, the objective function of Problem 2 reduces to

$$\sum_{i \in \mathcal{P}} \sum_{\ell \in \mathcal{L}} \frac{\alpha_{f}^{f,i}(t)}{w_{\ell}} (\mu_{0}^{f,i}(t) \Xi_{f,i}^{f,i}(t) + \epsilon_{\text{min}} \Xi_{f,i}^{f,i}(t))$$

$$= \lambda_{f} \sum_{i \in \mathcal{P}} \sum_{\ell \in \mathcal{L}} \frac{\alpha_{f}^{f,i}(t)}{w_{\ell}} \Xi_{f,i}^{f,i}(t)$$

$$+ \epsilon_{\text{min}} \sum_{i \in \mathcal{P}} \sum_{\ell \in \mathcal{L}} \frac{\alpha_{f}^{f,i}(t)}{w_{\ell}} \Xi_{f,i}^{f,i}(t),$$

where the equality in the last line holds if (36) is satisfied. Since (36) can be obtained for each flow, it is also the solution for Problem 1. In addition, with (36)-(38), it is easy to verify that both conditions in (9) and (31) are satisfied, which completes the proof of Theorem 3.

\section*{References}


Xianghui Cao (IEEE S’08-M’11) received the B.S. and Ph.D. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2006 and 2011, respectively. From December 2007 to June 2009, he was a Visiting Scholar with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA. From July 2012 to July 2015, he was a Senior Research Associate with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. Currently he is an associate professor with the School of Automation, Southeast University, Nanjing, China. His research interests include wireless sensor and actuator networks, wireless network performance analysis and network security. He serves as Publicity Co-chair for ACM MobiHoc 2015, Symposium Co-chair for IEEE/CIC ICC 2015, and TPC Member for a number of conferences, including IEEE Globecom, IEEE ICC, and IEEE VTC. He also serves as an Associate Editor of several journals, including KSII Transactions on Internet and Information Systems, Security and Communication Networks (Wiley), and the International Journal of Ad Hoc and Ubiquitous Computing. He was a recipient of the Best Paper Runner-Up Award from ACM MobiHoc 2014.

Lu Liu (IEEE S’13) received the B.S. degree in Automation from Tsinghua University, Beijing, China, in 2010, and the M.S. degree in Electrical Engineering from Illinois Institute of Technology, Chicago, IL, USA, in 2012. She is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at Illinois Institute of Technology, Chicago, IL, USA. Her current research interests include energy efficient networking and communications, performance analysis and protocol design of wireless networks.

Wenlong Shen (IEEE S’13) received the B.E. degree in Electrical Engineering from Beihang University, Beijing, China, in 2010, and the M.S. degree in Telecommunication from University of Maryland, College Park, USA, in 2012. He is currently pursuing the Ph.D. degree in the department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA. His research interests include vehicular ad hoc networks, mobile cloud computing, and network security.

Yu Cheng (IEEE S’01-M’04-SM’09) received the B.E. and M.E. degrees in Electronic Engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2003. From September 2004 to July 2006, he was a postdoctoral research fellow in the Department of Electrical and Computer Engineering, University of Toronto, Ontario, Canada. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, Illinois, USA, where he is now an Associate Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Best Paper Award from the conferences QShine 2007 and IEEE ICC 2011, and the Best Paper Runner-Up Award from ACM MobiHoc 2014. He received the National Science Foundation (NSF) CAREER AWARD in 2011 and IIT Sigma Xi Research Award in the junior faculty division in 2013. He served as a Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009, a Co-Chair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011, a Co-Chair for the Signal Processing for Communications Symposium of IEEE ICC 2012, a Co-Chair for the Ad Hoc and Sensor Networking Symposium of IEEE GLOBECOM 2013, and a Technical Program Committee (TPC) Co-Chair for WASA 2011, ICNC 2015, and IEEE/CIC ICC 2015. He is a founding Vice Chair of the IEEE ComSoc Technical Subcommittee on Green Communications and Computing. He is an Associated Editor for IEEE Transactions on Vehicular Technology and the New Books & Multimedia Column Editor for IEEE Network. He is a senior member of the IEEE.