Scheduling in Wireless Networks with Full-Duplex Cut-through Transmission

Yang Yang*, and Ness B. Shroff†

*Department of ECE, The Ohio State University, Columbus, OH
†Department of ECE and CSE, The Ohio State University, Columbus, OH

***Abstract*—The recent breakthrough in wireless full-duplex communication makes possible a brand new way of multi-hop wireless communication, namely full-duplex cut-through transmission, where for a traffic flow that traverses through multiple links, every node along the route can receive a new packet and simultaneously forward the previously received packet. This wireless transmission scheme brings new challenges in the design of MAC layer algorithms that aim to reap its full benefit. First, the MAC layer rate region of the cut-through enabled network is directly a function of the routing decision, leading to a strong coupling between routing and scheduling. Second, it is unclear how to dynamically form/change cut-through routes based on the traffic rates and patterns. In this work, we introduce a novel method to characterize the interference relationship between links in the network with cut-through transmission, which decouples the routing decision with the scheduling decision and enables a seamless adaptation of traditional half-duplex routing/scheduling algorithm into wireless networks with full-duplex cut-through capabilities. Based on this interference model, a queue-length based CSMA-type scheduling algorithm is proposed, which both leverages the flexibility of full-duplex cut-through transmission and permits distributed implementation.**

***Index Terms*—Wireless Full-duplex, Cut-through Transmission, Dynamic Routing, Scheduling**

I. Introduction

Recent development in wireless radio technology shows that by using advanced signal processing techniques together with new RF circuit designs, a wireless device can transmit and receive on the same frequency band, achieving full-duplex transmission [1]–[3]. The key idea that enables this full-duplex capability is that a node has complete knowledge of the digital packet it is transmitting, and therefore, it could potentially predict and actively cancel the impact of the transmitted signal onto its own receive antenna.

The full-duplex technology eliminates the conventional constraint in wireless networking that a node can either transmit or receive at any time but not both for any frequency band, and as a result it enlarges the capacity region of wireless networks. Take the five node tandem network shown in Figure 1 as an example. Assume that the nodes interfere only with their closest neighbors, and there exists a single flow from node A to node E. Under the half-duplex constraint, each link along the route of the flow can be activated only one third of the time.

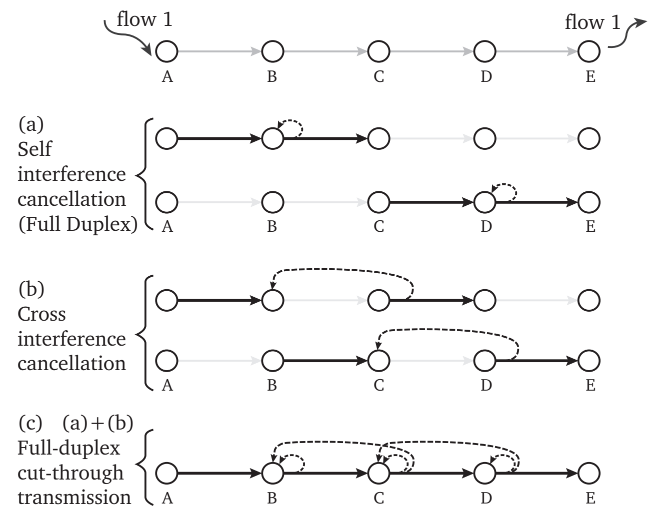


Fig. 1: Full-duplex cut-through transmission

However, when the nodes are capable of canceling their self interference, i.e., they are full-duplex enabled, then each link can be activated half of the time, as shown in Figure 1(a). Further, since there is only a single traffic flow from node A to node E, we know that any packet transmitted by node C is previously received by C from its upstream node B. If we assume that node B keeps a copy for every packet that it has received, then it has the potential to decode any packet that is collided with the transmission from node C. We use the term *cross interference cancellation* to describe the case when a node can withstand a stream of interference that carries a known packet. From Figure 1(b) we can see that each link can be activated half of the time if the nodes are capable of performing cross interference cancellation.

Interestingly, as is evident from Figure 1(c), if the nodes are capable of canceling both their self interference and a stream of known cross interference, then all the link can be activated at the same time, effectively forming a *cut-through* route with each node along the route simultaneously receiving a new packet from its upstream node and forwarding a previously received packet to its downstream node. The concept of wireless cut-through

This work was supported in part by NSF grant CNS-1314538.

transmission was first envisioned in [1], in which it was argued that as a full-duplex node starts to receive a packet it can simultaneously start to forward it without having to decode the entire packet, while the downstream interference can be digitally canceled since the interference is simply a delayed version of the received packet. A preliminary implementation is presented in [4]. In this work, we make a somewhat less restrictive assumption that *a full-duplex node can only start to forward a packet after the packet is completely decoded, and that it has the capability to cancel a single stream of interference from one of its immediate neighbors given that the neighbor is forwarding a packet previously received from it.* While wireless cut-through transmissions is a promising way to increase the capacity of the network, as we will see shortly, it also introduces new challenges in the design of MAC layer dynamic scheduling policies.

In a multi-hop wireless network, it is critical to have a resource allocation algorithm that efficiently utilizes the limited wireless spectrum. The seminal work of [5] develops a joint dynamic routing and scheduling algorithm, namely back-pressure, which is proven to be throughput-optimal, i.e., it can stabilize any network load that can be stabilized by some joint routing and scheduling algorithm. Later, through a utility maximization framework [6], it is shown that the wireless resource allocation problem can be optimally decomposed into three parts: transport layer rate control, network layer routing, and MAC layer scheduling, with minimal coupling among the layers. This cross-layer decomposition suggests that the MAC layer scheduling component is the bottleneck of this problem, as it requires solving a difficult combinatorial optimization problem which is NP-complete in general. There has been a plethora of work that focus on devising low-complexity and/or distributed scheduling algorithms. At the same time, with the continuous evolution of wireless physical layer technologies, many new transmission schemes emerge, such as wireless full-duplex, interference alignment, distributed multi-user MIMO, noisy network coding, etc. On the one hand, these new schemes keep breaking conventional transmission constraints and expanding the rate region of wireless networks. For example, the rate regions under different combinations of full-duplex and MIMO techniques are compared in [7]. On the other hand, they inherently come with more sophisticated interference relationship among wireless links and bring many challenges in the design of scheduling algorithms. Regarding full-duplex cut-through transmission, the challenges are the following: (i) the cut-through route naturally involves links that are multiple hops away from each other, which makes it hard to design scheduling algorithms based only on local information at each node. (ii) the MAC layer scheduling decision becomes closely coupled with the network layer routing decision, as a cut-through route is usually formed to serve a specific flow.

Our contributions are as follows:

- We introduce a novel way to model the interference relationship in wireless networks with full-duplex cut-through capability, which decouples the routing decision from the scheduling decision in a scalable and efficient manner.
- A queue-length based CSMA-type algorithm, similar to the one in [8], is proposed, which can dynamically form/change full-duplex cut-through routes in the network and achieve throughput-optimality.

The rest of the paper is organized as follows: In Section II, we introduce the network model, review the back-pressure algorithm under half-duplex or full-duplex networks, and then explain the difficulty in devising a scheduling algorithm for networks with cut-through capability. In Section III, we propose a new and efficient method of modeling the interference relationship for cut-through transmission. In Section IV, we develop a throughput-optimal queue-length based CSMA-type algorithm that leverages the full-duplex cut-through capability. The algorithm is evaluated in Section V and the paper is concluded in Section VI.

II. System Model

A. Network model and half-duplex/full-duplex constraint

We consider a multi-hop wireless network that can be described by a network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ together with an interference graph[1] $\mathcal{G}_{\rm I} = (\mathcal{V}, \mathcal{E}_{\rm I})$, where $\mathcal{V}$ denotes the set of wireless nodes, $\mathcal{E}$ denotes the set of wireless links, and $\mathcal{E}_{\rm I}$ denotes the interference relationship between wireless nodes. For any two nodes A, B in $\mathcal{V}$, $(A, B) \in \mathcal{E}_{\rm I}$ if the transmission of node A interferes with the reception of node B, while $(A, B) \in \mathcal{E}$ if a direct data-link can be established from node A to node B. In other words, $\mathcal{E}$ captures the communication region of every node in the network and $\mathcal{E}_{\rm I}$ captures the interference region the nodes in the network. We assume that $\mathcal{E}$ is a subset of $\mathcal{E}_{\rm I}$, and the edges in both $\mathcal{G}$ and $\mathcal{G}_{\rm I}$ are bidirectional. An example network is shown in Figure 2

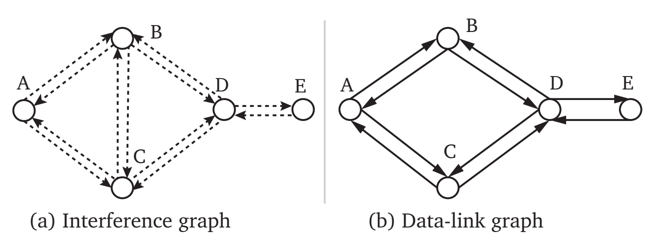


Fig. 2: The interference graph $\mathcal{G}_{\rm I}$ and the data link graph $\mathcal{G}$ of an example wireless network with five nodes. $\mathcal{I}(B) = \{A, C, D\}$.

For any node A in $\mathcal{V}$, we denote $\mathcal{I}(A) \triangleq \{B | (A, B) \in \mathcal{E}_{\rm I}\}$ as the set of nodes that interfere with node A. For any

[1] The use of an interference graph gives a more accurate modeling of the interference relationship between wireless nodes, compared with the conventional link-centric hop-count based interference model.

subset $\mathcal{S}$ of data-links in $\mathcal{E}$, we denote $\mathcal{R}(\mathcal{S})$ and $\mathcal{T}(\mathcal{S})$ as the set of receiver nodes and the set of transmitter nodes under the subset of links $\mathcal{S}$, respectively. More precisely, $\mathcal{R}(\mathcal{S}) = \{B|(A,B) \in \mathcal{S}\}$ and $\mathcal{T}(\mathcal{S}) = \{A|(A,B) \in \mathcal{S}\}$. We also call a subset of data links $\mathcal{S}$ a schedule.

Given these notations, we can describe the half-duplex interference constraint in a wireless network as follows:

Definition 1 (Half-duplex feasibility conditions)**.** A subset $\mathcal{S}$ of links in $\mathcal{E}$ is half-duplex feasible if for any link (A,B) in $\mathcal{S}$, all of the following conditions hold

- $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}) = \{B\}$
- $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}) = \{A\}$
- $A \notin \mathcal{R}(\mathcal{S})$ and $B \notin \mathcal{T}(\mathcal{S})$

We denote the set of all half-duplex feasible schedules in the network as $\mathbb{S}_{\text{HD}}$. For any $\mathcal{S} \in \mathbb{S}_{\text{HD}}$ and any link $(A,B) \in \mathcal{S}$, the first condition in the above definition implies that there is no receiver node within node A's interference range, other than node A's intended receiver B. Similarly, the second condition says that there is no transmitter node within node B's interference range, except its intended transmitter A. While the first two constraints guarantee that there is no cross-interference between links, the last constraint makes sure that there is no self-interference in $\mathcal{S}$, i.e., any node under $\mathcal{S}$ cannot be both a transmitter and a receiver.

From the half-duplex feasibility conditions, It is straightforward to derive the full-duplex feasibility conditions, which are described in Definition 2, since we only need to remove the last condition in Defintion 1 that forbids the existence of self-interference in a schedule.

Definition 2 (Full-duplex feasibility conditions)**.** A subset $\mathcal{S}$ of links in $\mathcal{E}$ is full-duplex feasible (denote as $\mathcal{S} \in \mathbb{S}_{\text{FD}}$) if for any link (A,B) in $\mathcal{S}$, all of the following conditions hold:

- $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}) = \{B\}$
- $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}) = \{A\}$

B. Traffic model

Let $\mathcal{F}$ denote the set of data flows in the network. For any flow f in $\mathcal{F}$, we further denote the source and the destination node of that flow as f_s and f_d, respectively. We assume that each node maintains a set of next-hop nodes for each flow, in such a way that there is no loop for any flow[2]. Based on this routing table, for each data link $(A,B) \in \mathcal{E}$, we define $\mathcal{F}_{(A,B)}$ as the set of flows that it carries. In other words, $\mathcal{F}_{(A,B)}$ denotes the set of flows that have node B as a valid next-hop from node A.

We assume a time-slotted system, where each data-link, if scheduled at a certain time-slot, can transmit a single packet. We denote $a^f[t]$ as the number of packets that arrive at the source node of flow f at time-slot t, $\mathcal{S}[t]$ as the set of scheduled links at time-slot t, and $f_{(A,B)}[t]$ as the index of the flow that link (A,B) chooses to serve if it is scheduled at time-slot t. We also assume that each node keeps a queue for each flow, and denote $Q_A^f[t]$ as the queue length of flow f right before the start of the t^{th} time-slot. Based on the model, we know that the queue Q_A^f evolves as

$$\begin{aligned} Q_A^f[t] = Q_A^f[t-1] &+ a^f[t-1]\mathbf{1}(f_s = A) \\ &+\textstyle\sum_{B:(B,A)\in\mathcal{S}[t-1]}\mathbf{1}(f = f_{(B,A)}[t-1]) \\ &-\textstyle\sum_{B:(A,B)\in\mathcal{S}[t-1]}\mathbf{1}(f = f_{(A,B)}[t-1]), \end{aligned}$$

if $f_d \neq A$, and $Q_A^f[t] = 0$ otherwise, where $\mathbf{1}(.)$ is the indicator function.

Let us assume that the arrival process is i.i.d. across different flows, and denote the arrival rate of flow f as λ_f. The *capacity region* of the network is then defined as the set of rate vectors $\vec{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{F}|}]$ under which all the queues in the network can be stabilized[3] by some scheduling policy. An algorithm is called *throughput-optimal* if it can stabilize the queues in the network for any arrival rates within the capacity region. It is well known from the seminal work [5] that a joint routing and scheduling algorithm, called back-pressure algorithm, is throughput optimal. The algorithm is restated below:

$$\textbf{Routing:}\ f_{(A,B)}[t] = \underset{f\in\mathcal{F}_{(A,B)}}{\arg\max}\Big(Q_A^f[t-1] - Q_B^f[t-1]\Big)$$

$$W_{(A,B)}[t] = \max_{f\in\mathcal{F}_{(A,B)}}\Big(Q_A^f[t-1] - Q_B^f[t-1]\Big)$$

$$\textbf{Scheduling:}\ \mathcal{S}[t] = \underset{\mathcal{S}\in\mathbb{S}}{\arg\max}\textstyle\sum_{(A,B)\in\mathcal{S}}W_{(A,B)}[t], \tag{1}$$

where $\mathbb{S}$ can either be $\mathbb{S}_{\text{HD}}$ or $\mathbb{S}_{\text{FD}}$. A nice feature of the above algorithm is that there exists only a loose coupling between the routing decision and the scheduling decision: the scheduling component only needs to obtain the value of the maximum queue differential W, and operates irrespective of which flow gets served on each link. The fundamental reason for the loose coupling is that *the routing decision does not affect the physical layer interference relationship of the data-links in the network, and therefore does not alter the MAC layer rate region of the network.* While this is true for both half-duplex and full-duplex wireless network, this claim no longer holds for wireless networks with full-duplex cut-through capability. Indeed, there exists a direct coupling between the routing decision and the physical layer capability of the network.

To support our claim in the previous paragraph, let us look at the network shown in Figure 3(a). There are two flows running on the network, where flow 1 runs from node A to node E, and flow 2 runs from node D to node E. Now assume that the two links (A,B) and (B,D) are already activated to serve flow 1, in which case node B is in full-duplex mode, then, whether link (D,E) can be activated together with (A,B) and (B,D)

[2]The assumption that there is no loop in the routing table does not preclude the possibility for there to be cycles in the network graph.

[3]We assume that the queueing dynamic can be captured by a Markovian process and stability refers to the Markov chain being positive recurrent [9].

or not depends on which flow it chooses to serve: if it picks flow 1, then node B is capable of canceling the cross-interference from node D and form a cut-through route, since the packet transmitted by node D is previously received by node B. If it, on the other hand, picks flow 2, then the link cannot be activated together with (A, B). Therefore, *whether a link can be scheduled or not depends on which flow it chooses to serve*, leading to a direct coupling between the routing decision and the scheduling decision. Furthermore, if the flows have multiple routes, then *whether a link can be scheduled or not may even depends on which packet it chooses to transmit*. For example, in the scenario shown in Figure 3(b) where there are two paths for flow 1, link (D, E) can be activated simultaneous with (A, B) and (B, D) only when it chooses to serve flow 1 with a packet it received previous from node B.

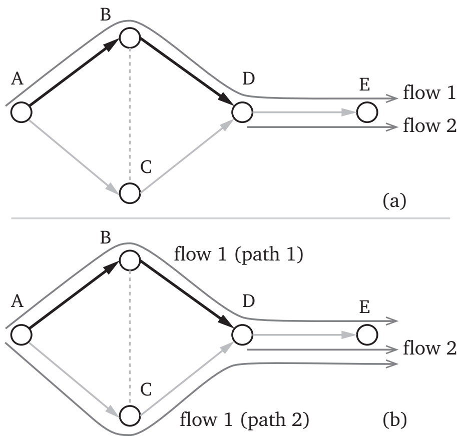


Fig. 3: There are two flows that run on the network shown in Figure 2. The links (A, B) and (B, D) are already scheduled to serve flow 1. (a) Both flow 1 and flow 2 have only a single path. (b) Flow 1 has two alternative paths.

Given the inherent direct-coupling of routing and scheduling decisions and the complicated flow-dependent interference relationship between data-links in the network, two natural questions arise: (i) Is there a way to efficiently model the interference relationship among data-links in wireless networks with full-duplex cut-through transmission capability, and at the same time circumvent the problem of the coupling between routing and scheduling? (ii) How to devise a distributed algorithm that dynamically forms/changes cut-through routes based on the patterns and the arrival-rates of traffic flows? We answer these two key questions in the rest of the paper.

III. Modeling of interference relationship with full-duplex cut-through capability

In the previous section, we showed that cut-through transmission introduces a direct-coupling of the routing decision with the schedule decision. From the point of view of the back-pressure algorithm, this coupling means that the set of all feasible schedules $\mathbb{S}$ in Equation (1) at time-slot t is a function of the set of routing decisions $\{f_{(A,B)}[t]\}_{(A,B)\in\mathcal{E}}$.

In this section, we will focus on developing feasibility conditions that incorporate the routing decision on each link. The attempt is to restore the decoupled structure shown in Equation (1) by revising the state-space of feasible schedules. We will first discuss two straightforward, yet inefficient and unscalable, methods, and then propose a new and efficient way to describe the cut-through interference relationship using a *neighbor-expanded graph*.

A. Modeling using per-flow virtual links

From the previous section, we showed that even under the case when each flow has a single route, the physical layer interference relationship among data-links is dictated not only by their topological relationship in the network graph, but also by which flow each link chooses to serve. Then, a straightforward way to construct the interference relationship is to extend each data-link into a number of virtual links, with each virtual-link designated to a single specific flow. More precisely, we can denote the virtual-link of (A, B) that is committed to a flow f as a triplet (A, B, f), and thus the set of data-links is extended to the set of virtual links $\{(A, B, f)|(A, B) \in \mathcal{E}, f \in \mathcal{F}_{(A,B)}\}$, based on which a feasibility condition of cut-through transmission can be derived.

The drawbacks of this approach are clear. First, this approach works only when each flow has only a single fixed route. Second, the state space of the virtual links can be very large, depending on the number of flows that get admitted. Third, the interference relationship among virtual links has to be re-derived each time new traffic flows enter the system.

B. Modeling using per-route virtual links

This approach exploits the fact that cut-through transmission can activate a group of consecutive data-links to serve a flow as if the whole group of data-links is a single link. We can exhaustively find all possible routes from any node to any other node in the network, and view each route we find as a per-route virtual link. A per-route virtual link can be denoted as a sequence of nodes that the route it represents traverses. For example, we can denote the route that corresponds to the path 1 of flow 1 in Figure 3(b) as a virtual link (A, B, D, E).

The drawbacks of this approach are also clear. First, an exhaustive search of all possible routes in the network is required every time there is a topology change. Second, the number of all possible routes in a network increases exponentially as the network size grows, which makes this approach unscalable and only suitable for small networks. Third, since links several hops away may form a single virtual link and have to be activated simultaneous when that virtual link is scheduled, this per-route virtual link is not suitable for distributed implementations of the scheduling algorithm, as a message passing mechanism has to be added to make sure that two ends of a virtual link coordinates with each other.

C. Modeling using neighbor-expanded graph

Now we propose a novel and scalable way to describe the interference relationship in wireless networks with cut-through capability. The key is to realize that full-duplex cut-through capability is no more than the capability for each node to cancel its self interference and a stream of cross interference, if that cross interference carries a packet the node has previously received. Given our assumption that there is no loop in the routing table for any routes, we observe that *node A can cancel a stream of cross-interference from node B, if B transmits a packet that it has previously received from node A*. Based on this observation, we can think of node B as a cluster of sub-nodes denoted as $\{B_C|(C,B) \in \mathcal{E} \text{ or } C = B\}$, where sub-node B_A stores the packets node B has obtained directly from node A (B_B stores the packets that exogenously arrived at node B). In this case, node A can cancel a stream of cross interference from node B only if B_A is the activated sub-node among all the sub-nodes of node B. According to the definition of sub-node, if link $(A,B) \in \mathcal{E}$ is scheduled, B_A is the activated receiver sub-node.

Given the concept of per-neighbor sub-node, we can construct an extended data-link graph, which we call *the neighbor-expanded graph*. Specifically, we denote the neighbor-expanded graph as $\widehat{\mathcal{G}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$, where

$$\widehat{\mathcal{V}} = \{A_C|(C,A) \in \mathcal{E} \text{ or } C = A\},$$
$$\widehat{\mathcal{E}} = \left\{(A_C, B_A)|A_C \in \widehat{\mathcal{V}}, \text{ and } (A,B) \in \mathcal{E}\right\}.$$

Figure 4 shows the neighbor-expanded graph of the network shown in Figure 2. Note, that the wireless network can now be fully represented by $\mathcal{G}_{\text{I}}$ and $\widehat{\mathcal{G}}$. For the rest of the paper, we reserve the term *schedule* and the notation $\mathcal{S}$ only to refer to a subset of links in $\widehat{\mathcal{E}}$. For any schedule $\mathcal{S}$, we define, similar as before,

$$\mathcal{R}(\mathcal{S}) = \{B|(A_C, B_A) \in \mathcal{S} \text{ for some } C, A \in \mathcal{V}\}$$
$$\mathcal{T}(\mathcal{S}) = \{A|(A_C, B_A) \in \mathcal{S} \text{ for some } C, B \in \mathcal{V}\}$$
$$\widehat{\mathcal{T}}(\mathcal{S}) = \{A_C|(A_C, B_A) \in \mathcal{S} \text{ for some } B \in \mathcal{V}\}.$$

By the help of the neighbor-expanded graph, we can derive the feasibility condition under full-duplex cut-through transmission as the following.

Definition 3 (Cut-through feasibility conditions)**.** A subset $\mathcal{S}$ of links in $\widehat{\mathcal{E}}$ is cut-through feasible (denote as $\mathcal{S} \in \mathbb{S}_{\text{CT}}$) if for any link (A_C, B_A) in $\mathcal{S}$, all of the following conditions hold:

(C1) $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}) = \{B\}$ or
$\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}) = \{B, C\}$ with $\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}) = \{A\}$

(C2) $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}) = \{A\}$ or
$\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}) = \{A, D\}$ for some D with $D_B \in \widehat{\mathcal{T}}(\mathcal{S})$

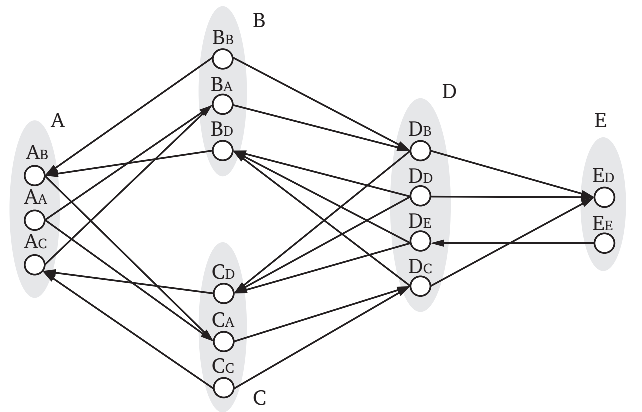


Fig. 4: Neighbor-expanded graph of the data-link graph shown in Figure 2(b).

(C3) $(A_D, B_A) \notin \mathcal{S}$ for any $D \in \mathcal{V} \backslash \{C\}$

By comparing the above definition with Definition 2, we can see that the full-duplex feasibility conditions are a subset of the cut-through ones. Condition (C1) in the above definition implies that (A_C, B_A) may be activated even when C is a receiver node, given that A is the only transmitter among all C's neighbors. Condition (C2) says that (A_C, B_A) may be activated even when one of B's neighbors D is a transmitter node, given that D_B is the activated sub-node (which means the packet that D transmits is received from node B). The last condition implies that there cannot exist two simultaneously activated transmitter sub-nodes that belong to the same node.

Based on Definition 3, in Figure 4, it is easy to check that the set of links $\{(A_A, B_A), (B_A, D_B), (D_B, E_D)\}$ can be activated simultaneously, while $\{(A_A, B_A), (B_A, D_B), (D_D, E_D)\}$ and $\{(A_A, B_A), (B_A, D_B), (D_C, E_D)\}$ are not feasible schedules. This can be mapped back to our previous examples in Figure 3, where we claim that link (D,E) can be activated together with (A,B) and (B,D) only when D serves a flow 1 packet that it previous obtained from node B.

There are two advantages in using neighbor-expanded graph to characterize the interference relationship, compared with the other two approaches. (i) since the number of sub-nodes that a node has in the neighbor-expanded graph equals the degree of that node in the data-link graph, we know that the size of $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{E}}$ is bounded by the size of $\mathcal{V}$ and $\mathcal{E}$ times the maximum degree of the data-link graph, which is quite scalable as the network size grows. (ii) the feasibility constraint is not a function of the routing decision, nor does it involve links that are more than 2-hop away from each other, which not only restores the decoupled structure of the back-pressure algorithm, but also permits the development of distributed scheduling algorithms.

Since the minimal scheduling entity is now the sub-node, in order to apply the back-pressure algorithm in Equation (1), for every sub-node A_C and any flow f, we need to keep a packet queue $Q^f_{A_C}$. For the rest of the paper, we will use (A_C, B_A) and l interchangeably to index the links in $\widehat{\mathcal{E}}$.

IV. Queue-length based Cut-through CSMA algorithm

In this section, we develop a queue-length based CSMA algorithm, similar to the one proposed in [8], that can achieve throughput optimality in wireless networks with cut-through capability.

We should point out that there exists an important difference between the interference model adopted in [8] and the interference relationship we have derived in Definition 3 for full-duplex cut-through enabled network, which prohibits a direct application of the algorithm in [8] onto the cut-through scenario. The dynamic CSMA algorithm developed in [8] depends on the assumption that the interference relationship between different links can be captured by an conflict graph[4] with each independent set being a feasible schedule. However, under our cut-through feasibility conditions, such a conflict graph cannot be formed. In other words, given three links l_1 l_2 and l_3, the assumption that any two links can be activated together does not necessarily imply that the three links form a feasible schedule. For example, in Figure 5, among the three links $(A_A, B_A), (C_B, D_C)$, and (E_B, F_E), any two can be activated at the same time, however, a schedule that include all three of them would violate condition (C2) in Definition 3, since node B_A can only cancel one stream of cross-interference. We circumvent this obstacle by introducing the concept of a *trimmed decision schedule*, whose definition is provided in Definition 5.

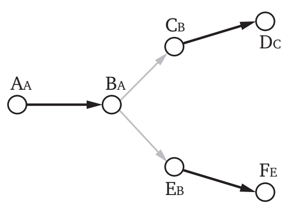


Fig. 5: An example neighbor-expanded graph.

To differentiate between different cut-through feasible schedules in $\mathbb{S}_{\text{CT}}$, we use a subscript to denote the index of a specific schedule. For example, $\mathcal{S}_{(x)}$ is the x^{th} schedule in $\mathbb{S}_{\text{CT}}$.

Definition 4 (Decision schedule)**.** For any $\mathcal{M} \subseteq \widehat{\mathcal{E}}$, we say that $\mathcal{M}$ is a valid decision schedule if for any $(A_C, B_A) \in \mathcal{M}$, all of the following conditions hold:

- $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}) = \{B\}$ • $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{M}) = \{A\}$
- $(A_D, B_A) \notin \mathcal{M}$ for any $D \in \mathcal{V}\backslash\{C\}$

Definition 5 (Trimmed decision schedule)**.** For any cut-through feasible schedule $\mathcal{S}_{(x)} \in \mathbb{S}_{\text{CT}}$ and any valid decision schedule $\mathcal{M}$, we define a new schedule $\mathcal{M}_{(x)} = \mathcal{M}^1_{(x)} \cup \mathcal{M}^2_{(x)} \cup \mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}$, where

$$\mathcal{M}^1_{(x)} = \Big\{(A_C, B_A) \in \mathcal{M} \cap \mathcal{S}_{(x)} \big| \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)}) = \{B\}\Big\},$$

$$\mathcal{M}^2_{(x)} = \Big\{(A_C, B_A) \in \mathcal{M} \cap \mathcal{S}_{(x)} \big| \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)}) = \{B, C\}, \mathcal{I}(C) \cap \mathcal{T}(\mathcal{M}) = \{A\}\Big\},$$

$$\mathcal{M}^3_{(x)} = \Big\{(A_C, B_A) \in \mathcal{M}\backslash\mathcal{S}_{(x)} \big| \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)}) = \varnothing, \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)}) = \varnothing \text{ or } \{D\} \text{ for some } D \text{ with } D_B \in \widehat{\mathcal{T}}(\mathcal{S}_{(x)})\Big\},$$

$$\mathcal{M}^4_{(x)} = \Big\{(A_C, B_A) \in \mathcal{M}\backslash\mathcal{S}_{(x)} \big| \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)}) = \{C\}, |\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)})| = 1, \mathcal{I}(C) \cap \mathcal{T}(\mathcal{M}) = \{A\}, \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)}) = \varnothing \text{ or } \{D\} \text{ for some } D \text{ with } D_B \in \widehat{\mathcal{T}}(\mathcal{S}_{(x)})\Big\}.$$

Since $\mathcal{M}_{(x)}$ is a subset of $\mathcal{M}$ and also a function of $\mathcal{S}_{(x)}$, we say that $\mathcal{M}_{(x)}$ is the result of $\mathcal{M}$ *trimmed* by $\mathcal{S}_{(x)}$.

From the above definitions, we can see that whether a link in the decision schedule should be trimmed or not depends on the status of the links that are no more than two-hops away. Given the definitions of the decision schedule and trimmed decision schedule, we obtain the following two lemmas, whose proofs can be found in the Appendix.

Lemma 1. For any cut-through feasible schedule $\mathcal{S}_{(x)} \in \mathbb{S}_{\text{CT}}$ and any valid decision schedule $\mathcal{M}$, $\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}$ is a cut-through feasible schedule.

Lemma 2. For any two cut-through feasible schedules $\mathcal{S}_{(y)}, \mathcal{S}_{(z)} \in \mathbb{S}_{\text{CT}}$ and any valid decision schedule $\mathcal{M}$, if $\mathcal{S}_{(y)}\backslash\mathcal{M}_{(y)} = \mathcal{S}_{(z)}\backslash\mathcal{M}_{(y)}$, then $\mathcal{M}_{(y)} = \mathcal{M}_{(z)}$.

Now we are ready to introduce the algorithm. Let $\mathbb{M}$ be a set of valid decision schedules. At the beginning of each time-slot, the system chooses a decision schedule $\mathcal{M} \in \mathbb{M}$ with probability $P_{\mathcal{M}}$, where $P_{\mathcal{M}} > 0$ for any $\mathcal{M} \in \mathbb{M}$ and $\sum_{\mathcal{M}\in\mathbb{M}} P_{\mathcal{M}} = 1$. Assume, w.l.o.g., that the schedule used in the network at the $(t-1)^{\text{st}}$ time-slot is $\mathcal{S}[t-1] = \mathcal{S}_{(y)}$ for some y. At the start of time-slot t, right after a decision schedule $\mathcal{M}$ is picked, the system obtains a trimmed schedule $\mathcal{M}_{(y)}$ as a function of $\mathcal{M}$ and $\mathcal{S}_{(y)}$. For any link $l \in \mathcal{M}_{(y)}$, it is included in the updated schedule $\mathcal{S}[t]$ with probability P_l and rejected with probability $1 - P_l$, while any link in $\mathcal{S}_{(y)}\backslash\mathcal{M}_{(y)}$ is included in $\mathcal{S}[t]$. From the Lemma 1 we know that $\mathcal{S}[t]$ is a cut-through feasible schedule, since $\mathcal{S}[t]$ is a subset of $\mathcal{S}_{(y)} \cup \mathcal{M}_{(y)}$. This algorithm is summarized in Algorithm 1. All the steps in the algorithm rely solely on local information and thus permit a distributed implementation.

[4]In a conflict graph (also called link-contention graph), the vertices represent data-links and an edge between two vertices indicates that the two data-links cannot be activated simultaneously.

Algorithm 1: Basic Cut-through CSMA Algorithm

Assume that the schedule at time slot $t-1$ is $\mathcal{S}[t-1] = \mathcal{S}_{(y)}$

Before the start of time-slot t:

Randomly pick a decision schedule $\mathcal{M}$ from $\mathbb{M}$ with probability $P_{\mathcal{M}}$.

The picked decision schedule $\mathcal{M}$ is then trimmed by $\mathcal{S}_{(y)}$ to form a trimmed schedule $\mathcal{M}_{(y)}$.

During time-slot t:

for *any link* $(A_C, B_A) \in \widehat{\mathcal{E}}$ **do**

  If (A_C, B_A) is in $\mathcal{M}_{(y)}$, then with probability $P_{(A_C,B_A)}$, the link gets included in $\mathcal{S}[t]$.

  If (A_C, B_A) is in $\mathcal{S}_{(y)} \backslash \mathcal{M}_{(y)}$, then the link is included in $\mathcal{S}[t]$.

Along with the same lines as in [8], [10], we can obtain the following proposition.

Proposition 1. *$\{S[t]\}_t$ in Algorithm 1 evolves as an irreducible and aperiodic Markov chain with the state space being $\mathbb{S}_{CT}$, if the set of decision schedules $\mathbb{M}$ satisfies*

$$\cup_{\mathcal{M}\in\mathbb{M}} \{(A_C, B_A) \in \mathcal{M} | \mathcal{I}(C) \cap \mathcal{T}(\mathcal{M}) = \{A\}\} = \widehat{\mathcal{E}}.$$

If the above condition is satisfied, then the stationary distribution of the Markov chain $\{S[t]\}_t$ is

$$\begin{aligned}\pi\left(\mathcal{S}_{(x)}\right) &\triangleq P\left(\mathcal{S}[\infty] = \mathcal{S}_{(x)}\right) \\ &= \tfrac{1}{Z}\left(\prod_{l\in\mathcal{S}_{(x)}} P_l\right)\left(\prod_{l\notin\mathcal{S}_{(x)}} (1-P_l)\right),\end{aligned}$$

where $Z = \sum_{\mathcal{S}_{(y)}\in\mathbb{S}_{CT}} \left(\prod_{l\in\mathcal{S}_{(y)}} P_l\right)\left(\prod_{l\notin\mathcal{S}_{(y)}} (1-P_l)\right)$.

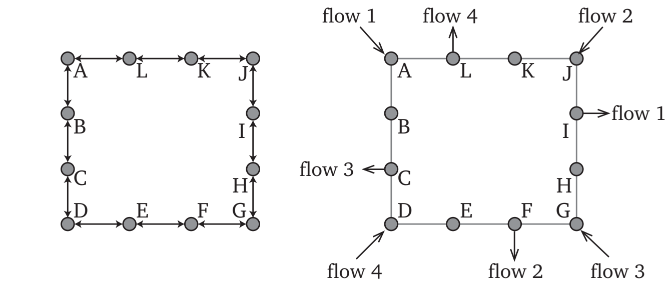
The stationary distribution of the cut-through feasible schedule in Proposition 1 is obtained by verifying the local balance equation of the Markov chain $\{S[t]\}_t$. The result in Lemma 2 is essential in guaranteeing that the Markov chain that describes Algorithm 1 is reversible, and thus has a product-form stationary distribution.

Given this product form distribution, the throughput-optimality of the algorithm can be established using standard method. For example, we can set P_l at time-slot t to be $\frac{\exp(W_l[t])}{\exp(W_l[t])+1}$, and then prove the throughput-optimality by the aid of a time-scale separation assumption [8], [10], [11], or without such an assumption [12].

V. Numerical Simulation

In this section, we provide simulation results to (i) verify the effectiveness of using neighbor-expanded graph to capture the interference relationship in networks with cut-through capability; (ii) show the performance of the proposed distributed queue-length based CSMA algorithm.

Specifically, we study a 12-node ring network, where for each node in the network, we assume that it can communicate and interfere with only its two immediate neighbors, which makes the network graph $\mathcal{G}$ and the interference graph $\mathcal{G}_{\text{I}}$ to be the same, as depicted in Figure 6(a). There are four flows running in the network, with the source and destination node of each flow indicated in Figure 6(b). Since the network has full-duplex cut-through capability, it is not hard to see that we can divide the four flows into two groups {flow 1, flow 3} and {flow 2, flow 4}, where at each time-slot two cut-through routes can be formed to serve both flows in a single group without causing interference to each other. In other words, the network can simultaneously support the four flows each with a packet arrival rate of 0.5 packet/time-slot. In our simulation, we first obtain the neighbor-expanded graph of the 12-node ring network, and then apply the proposed distributed queue-length based CSMA algorithm. We choose P_{decision} to be 0.2 and P_l to be $\frac{\exp(0.2W_l[t])}{\exp(0.2W_l[t])+1}$. The packet arrivals to each flow follow a poisson process with the same rate λ. Packets in each node are routed towards the direction with the shortest path to its destination.



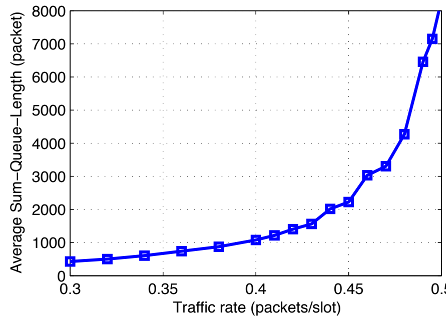
(a) Network graph/Interference graph (b) Flow pattern

Fig. 6: Simulation scenario

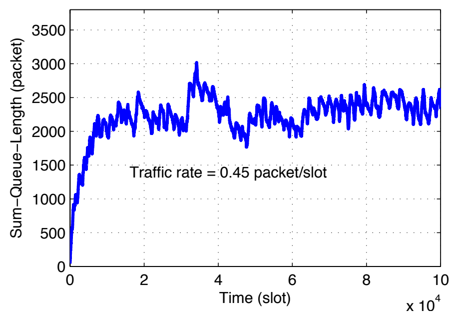
In Figure 7(a) we show the time-average of the sum-queue-length in the network (averaged across 10^5 time-slots) as a function of the traffic rate λ. In Figure 7(b) we focus on the case when $\lambda = 4.5$ packet/time-slot and plot the evolution of sum-queue-length in the network. From these figures we can see that, indeed, the proposed distributed CSMA algorithm can support any traffic rate λ that is less than 0.5 packet/time-slot.

VI. Conclusion

The full-duplex cut-through transmission technique in wireless networks introduces a direct coupling between the network-layer routing decision with the MAC-layer rate-region, leading to a complicated route/flow-dependent interference relationship between data-links in the network, which makes it hard to fully exploit the potential of cut-through transmission in an efficient way. In this paper, we circumvent this difficulty by introducing the concept of neighbor-expanded network graph, which allows us to derive simple interference conditions that capture the full-duplex cut-through constraint in a scalable and low-complexity manner. The neighbor-expanded graph also enables us to devise algorithms that use only local information to form/change cut-through routes,



(a) Average sum-queue-length vs. packet arrival rate λ.



(b) Evolution of sum-queue-length when $\lambda = 0.45$.

Fig. 7: Simulation results

with the proposed queue-length-based CSMA algorithm being an example.

References

[1] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving Single Channel, Full Duplex Wireless Communication," in *MobiCom 2010*. ACM, 2010, pp. 1–12.

[2] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, Real-time, Full Duplex Wireless," in *ACM MobiCom 2011*, 2011, pp. 301–312.

[3] A. Sahai, G. Patel, and A. Sabharwal, "Pushing the limits of full-duplex: Design and real-time implementation," *CoRR*, vol. abs/1107.0607, 2011.

[4] B. Chen, G. K. Tummala, Y. Qiao, and K. Srinivasan, "In-band wireless cut-through: Is it possible?" in *HotWireless 2014*. ACM, 2014, pp. 1–6.

[5] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, Dec 1992.

[6] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1452–1463, Aug 2006.

[7] Y. Yang, B. Chen, K. Srinivasan, and N. Shroff, "Characterizing the achievable throughput in wireless networks with two active rf chains," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 262–270.

[8] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-Length-Based CSMA/CA Algorithms for Achieving Maximum Throughput and Low Delay in Wireless Networks," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 825–836, June 2012.

[9] M. Bramson, "Stability of queueing networks," *Probability Surveys*, vol. 5, no. 1, pp. 169–345, 2008.

[10] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," *Networking, IEEE/ACM Transactions on*, vol. 18, no. 3, June 2010.

[11] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *Networking, IEEE/ACM Transactions on*, vol. 13, no. 2, pp. 411–424, April 2005.

[12] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: An efficient randomized protocol for contention resolution," in *SIGMETRICS 2009*. New York, NY, USA: ACM, 2009, pp. 133–144.

Appendix A
Proof of Lemma 1

To proof that $\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}$ is a cut-through feasible schedule, we need to show that for any link $(A_C, B_A) \in \mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}$, $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)})$ and $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)})$ satisfy the conditions (C1), (C2) and (C3) in Definition 3.

First, we prove that condition (C3) always holds for any (A_C, B_A) by contradiction. Assume to the contrary that we can find another link $(A_D, B_A) \in \mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}$, then based on Definition 3 and Definition 4, the two links cannot be both in $\mathcal{S}_{(x)}$ or $\mathcal{M}_{(x)}$. Assume w.l.o.g. that $(A_C, B_A) \in \mathcal{S}_{(x)}$ and $(A_D, B_A) \in \mathcal{M}_{(x)}$, then it is clear that $(A_D, B_A) \in (\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)})$ since $\mathcal{M}^1_{(x)}, \mathcal{M}^2_{(x)} \in \mathcal{S}_{(x)}$. However, according to Definition 5, (A_D, B_A) cannot be included in $\mathcal{M}^3_{(x)}$ or $\mathcal{M}^4_{(x)}$ if $(A_C, B_A) \in \mathcal{S}_{(x)}$. Thus, a contradiction is reached and condition (C3) is satisfied.

For any $(A_C, B_A) \in \mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}$, we prove that conditions (C1) and (C2) also hold by discussing the following three cases: (i) $(A_C, B_A) \in \mathcal{M}^1_{(x)}$, (ii) $(A_C, B_A) \in \mathcal{M}^3_{(x)}$, (iii) $(A_C, B_A) \in \mathcal{S}_{(x)} \backslash \mathcal{M}_{(x)}$. The discussion for the cases when $(A_C, B_A) \in \mathcal{M}^2_{(x)}$ and $(A_C, B_A) \in \mathcal{M}^4_{(x)}$ are similar to (i) and (ii) and, therefore, are omitted for brevity.

(i) If $(A_C, B_A) \in \mathcal{M}^1_{(x)}$, then

$$\begin{aligned}
&\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}) = \mathcal{I}(A) \cap \big(\mathcal{R}(\mathcal{S}_{(x)}) \cup \mathcal{R}(\mathcal{M}_{(x)})\big) \\
&= \big(\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)})\big) \cup \big(\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}_{(x)})\big) \\
&\overset{(a)}{=} \big(\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)})\big) \cup \{B\},
\end{aligned}$$

and $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}) = \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)})\big) \cup \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{M}_{(x)})\big) \overset{(b)}{=} \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)})\big) \cup \{A\}$, which, by combing the fact that $\mathcal{M}^1_{(x)} \subset \mathcal{S}_{(x)}$, imply that the condition (C1) and (C2) in Definition 3 are trivially satisfied. Equality (a) and (b) in the above equations follow from the definition of valid decision schedule in Definition 4.

(ii) If $(A_C, B_A) \in \mathcal{M}^3_{(x)}$, then

$$\begin{aligned}
&\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}) \\
&= \big(\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(x)})\big) \cup \big(\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}_{(x)})\big) = \varnothing \cup \{B\},
\end{aligned}$$

and $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}) = \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)})\big) \cup \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{M}_{(x)})\big) = \big(\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(x)})\big) \cup \{A\}$, which, according to the definition of $\mathcal{M}^3_{(x)}$, either equals to $\{A\}$, or $\{A, D\}$ for some D with $D_B \in \widehat{\mathcal{T}}(\mathcal{S}_{(x)})$. Therefore, condition (C1) and (C2) are satisfied.

(iii) Now we focus on the case when $(A_C, B_A) \in \mathcal{S}_{(x)} \backslash \mathcal{M}_{(x)}$. Since $\mathcal{M}^1_{(x)}, \mathcal{M}^2_{(x)} \in \mathcal{S}_{(x)}$, we have

$$\mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{S}_{(x)} \cup \mathcal{M}_{(x)}\right) = \mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{S}_{(x)} \cup \mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}\right). \quad (2)$$

We use a two-step argument to show that condition (C1) holds. In the first step, we claim that $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}) = \varnothing$ or $\{C\}$. In the second step, we claim that if $|\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)})| > 1$, then $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}) = \varnothing$. The proof of condition (C2) can be obtained in a similar manner and is omitted here.

For the first claim, let us assume to the contrary that we can find a node $D \in \mathcal{E} \backslash \{C\}$ such that $D \in \mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)})$. Since $D \in \mathcal{I}(A)$ and $A \in \mathcal{T}(\mathcal{S}_{(x)})$, we know that $\{A\} \subseteq \mathcal{I}(D) \cap \mathcal{T}(\mathcal{S}_{(x)})$. On the other hand, since $D \in \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)})$, we have, according to the definition of $\mathcal{M}^3_{(x)}$ and $\mathcal{M}^4_{(x)}$, that $\mathcal{I}(D) \cap \mathcal{T}(\mathcal{S}_{(x)}) = \{A\}$ implies $A_D \in \widehat{\mathcal{T}}(\mathcal{S}_{(x)})$. However, since $(A_C, B_A) \in \mathcal{S}_{(x)}$ and $A_D \in \widehat{\mathcal{T}}(\mathcal{S}_{(x)})$, $\mathcal{S}_{(x)}$ cannot be a cut-through feasible schedule. Therefore, a contradiction is reached, and the first claim holds.

For the second claim, again, let us assume to the contrary that $|\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)})| > 1$ and $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}) = \{C\}$. However, from the definition of $\mathcal{M}^3_{(x)}$ and $\mathcal{M}^4_{(x)}$ we know that if $C \in \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)})$ then $|\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)})| \leq 1$. Therefore, a contradiction is reached and the second claim holds.

Now, let us combine the two claims. If $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}) = \varnothing$, then it is easy to see that condition (C1) holds trivially. Otherwise, according to the two claims, we must have $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}) = \{C\}$ with $|\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)})| \leq 1$. In other words, $\mathcal{I}(C) \cap \mathcal{T}(\mathcal{S}_{(x)}) = \{A\}$, and $\mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{S}_{(x)} \cup \mathcal{M}^3_{(x)} \cup \mathcal{M}^4_{(x)}\right) = \{B, C\}$, which, by combing Equation (2), completes the proof.

Appendix B
Proof of Lemma 2

Let us focus on a particular valid decision schedule $\mathcal{M}$, and pick two cut-through feasible schedules $\mathcal{S}_{(y)}, \mathcal{S}_{(z)} \in \mathbb{S}_{\text{CT}}$ that satisfy $\mathcal{S}_{(y)} \backslash \mathcal{M}_{(y)} = \mathcal{S}_{(z)} \backslash \mathcal{M}_{(y)}$. We need to show that for any link $(A_C, B_A) \in \mathcal{M}$, it is contained either in both $\mathcal{M}_{(y)}$ and $\mathcal{M}_{(z)}$, or in neither $\mathcal{M}_{(y)}$ nor $\mathcal{M}_{(z)}$. The proof can be broken down into the following four cases: (i) $(A_C, B_A) \in \mathcal{S}_{(y)} \cap \mathcal{S}_{(z)}$; (ii) $(A_C, B_A) \in \mathcal{S}_{(y)}$ and $(A_C, B_A) \notin \mathcal{S}_{(z)}$; (iii) $(A_C, B_A) \notin \mathcal{S}_{(y)} \cup \mathcal{S}_{(z)}$; (iv) $(A_C, B_A) \notin \mathcal{S}_{(y)}$ and $(A_C, B_A) \in \mathcal{S}_{(z)}$. Since the discussion on case (i) and (ii) is similar to that on case (iii) and (iv), we omit the last two cases for brevity.

Before we start the discussion, it is worth noting, from the definition of the trimmed decision schedule in Definition 5, that whether a link (A_C, B_A) is included in $\mathcal{M}_{(y)}$ or not depends only on the outcome of the following three sets: $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(y)})$, $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(y)})$, and $\mathcal{I}(C) \cap \mathcal{T}(\mathcal{M})$.

(i) Given that $(A_C, B_A) \in \mathcal{M}$ and $(A_C, B_A) \in \mathcal{S}_{(y)} \cap \mathcal{S}_{(z)}$, in order to prove (A_C, B_A) is either in both $\mathcal{M}_{(y)}$ and $\mathcal{M}_{(z)}$ or in neither $\mathcal{M}_{(y)}$ nor $\mathcal{M}_{(z)}$, it suffices to show that the following two claims hold: 1, $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(y)}) = \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)})$. 2, $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(y)}) = \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)})$.

Given that $\mathcal{S}_{(y)} \backslash \mathcal{M}_{(y)} = \mathcal{S}_{(z)} \backslash \mathcal{M}_{(y)}$, it is easy to see that $(\mathcal{S}_{(y)} \backslash \mathcal{S}_{(z)}) \cup (\mathcal{S}_{(z)} \backslash \mathcal{S}_{(y)}) \subseteq \mathcal{M}_{(y)} \backslash (A_C, B_A) \subseteq \mathcal{M} \backslash (A_C, B_A)$, implying

$$\begin{aligned} &\mathcal{I}(A) \cap \mathcal{R}\left((\mathcal{S}_{(y)} \backslash \mathcal{S}_{(z)}) \cup (\mathcal{S}_{(z)} \backslash \mathcal{S}_{(y)})\right) \\ \subseteq &\mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{M}_{(y)} \backslash (A_C, B_A)\right) \\ \subseteq &\mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{M} \backslash (A_C, B_A)\right) \overset{(a)}{=} \varnothing, \end{aligned}$$

which is an equivalent statement of $\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(y)}) = \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)})$, and thus claim 1 holds. Note that equality (a) above followings from Definition 4.

Similarly, the second claim follows by noting that

$$\begin{aligned} &\mathcal{I}(B) \cap \mathcal{T}\left((\mathcal{S}_{(y)} \backslash \mathcal{S}_{(z)}) \cup (\mathcal{S}_{(z)} \backslash \mathcal{S}_{(y)})\right) \\ \subseteq &\mathcal{I}(B) \cap \mathcal{T}\left(\mathcal{M}_{(y)} \backslash (A_C, B_A))\right) \\ \subseteq &\mathcal{I}(B) \cap \mathcal{T}\left(\mathcal{M} \backslash (A_C, B_A)\right) = \varnothing. \end{aligned}$$

(iii) If $(A_C, B_A) \in \mathcal{S}_{(y)}$ and $(A_C, B_A) \notin \mathcal{S}_{(z)}$, then since $\mathcal{S}_{(y)} \backslash \mathcal{M}_{(y)} = \mathcal{S}_{(z)} \backslash \mathcal{M}_{(z)}$, we must have $(A_C, B_A) \in \mathcal{M}_{(y)}$. Moreover, we know that $(A_C, B_A) \in \mathcal{M}^1_{(y)} \cup \mathcal{M}^2_{(y)}$, since, by definition, $\mathcal{M}^3_{(y)}$ and $\mathcal{M}^4_{(y)}$ have no intersection with $\mathcal{S}_{(y)}$. As a result, in this case, it suffices to show that $(A_C, B_A) \in \mathcal{M}_{(z)}$, which can be proved by showing that the following two claims hold: 1, if $(A_C, B_A) \in \mathcal{M}^1_{(y)}$, then $(A_C, B_A) \in \mathcal{M}^3_{(z)}$. 2, if $(A_C, B_A) \in \mathcal{M}^2_{(y)}$, then $(A_C, B_A) \in \mathcal{M}^4_{(z)}$. The proofs of the two claims are similar and we only show the first one.

If $(A_C, B_A) \in \mathcal{M}^1_{(y)}$ and $(A_C, B_A) \notin \mathcal{S}_{(z)}$, then using the argument we have for the previous case, we can obtain

$$\begin{aligned} \mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{S}_{(y)} \backslash (A_C, B_A)\right) &= \mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)}), \\ \mathcal{I}(B) \cap \mathcal{T}\left(\mathcal{S}_{(y)} \backslash (A_C, B_A)\right) &= \mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}), \end{aligned}$$

which, by combing the fact that (A_C, B_A) is a cut-through feasible schedule, yields

$$\mathcal{I}(A) \cap \mathcal{R}(\mathcal{S}_{(z)}) = \mathcal{I}(A) \cap \mathcal{R}\left(\mathcal{S}_{(y)}\right) \backslash \{B\} = \varnothing, \quad (3)$$

$$\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}) = \mathcal{I}(B) \cap \mathcal{T}\left(\mathcal{S}_{(y)}\right) \backslash \{A\} = \varnothing \text{ or } \{D\} \text{ for some node } D \text{ with } D_B \in \widehat{\mathcal{T}}(\mathcal{S}_{(y)}). \quad (4)$$

Finally, if $\mathcal{I}(B) \cap \mathcal{T}(\mathcal{S}_{(z)}) = \{D\}$ with $D_B \in \widehat{\mathcal{T}}(\mathcal{S}_{(y)})$, then D_B must also be in $\widehat{\mathcal{T}}(\mathcal{S}_{(z)})$. Assume to the contrary that $D_B \notin \widehat{\mathcal{T}}(\mathcal{S}_{(z)})$, then from the fact that $\mathcal{S}_{(y)} \backslash \mathcal{M}_{(y)} = \mathcal{S}_{(z)} \backslash \mathcal{M}_{(y)}$, we know that $D \in \mathcal{R}(\mathcal{M}_{(y)})$, which makes $\mathcal{I}(B) \cap \mathcal{R}(\mathcal{M}_{(y)}) = \{A, D\}$, violating the fact that $M_{(y)}$ is a valid decision schedule. This argument, by combining with Equation (3) and (4), implies that $(A_C, B_A) \in \mathcal{M}^3_{(z)}$, which completes our proof.