Inter-Cell Channel Time-Slot Scheduling for Multichannel Multiradio Cellular Fieldbuses

Aiping Tan[1,2], Qixin Wang[1,4], Nan Guan[2], Qingxu Deng[2], Xiaobo Sharon Hu[3]

***Abstract*—Recently there is a growing interest of incorporating cellular architecture (with wired base stations and last-hop wireless connections) into fieldbuses to support mobile real-time applications. A promising trend is that such cellular fieldbuses will go multichannel multiradio, due to the wide availability of cheap multichannel commercial-off-the-shelf (COTS) wireless nodes, and the rise of 4G and future cellular technologies. For multichannel multiradio cellular fieldbuses, per-flow real-time schedulability guarantee in the inter-cell level has not yet been well studied. Particularly, unlike 3G cellular networks, which use static FDMA/CDMA to isolate cells, the multichannel multiradio feature allows neighboring cells to use the same radio frequency channel at different time-slots; or the same time-slot at different radio frequency channels. How to carry out channel time-slot scheduling is therefore the focus of this paper. To address this issue, we propose a greedy scheduling algorithm, together with a polynomial time closed-form schedulability test. The relationship between the schedulability test result, greedy scheduling schedulability, and schedulability is explored. We prove the equivalence of the three for chained cellular fieldbus topology, a typical topology with broad applications. This also implies the optimality of greedy scheduling, and the sufficiency and necessity of the schedulability test in the context of chained topology. To demonstrate and validate these schedulability theories, we carry out a case study on a classic admission planning problem. The schedulability test not only serves as a planning constraint, but also guides us to propose an approximation algorithm to solve the NP-hard admission planning problem. Comparisons to exhaustive search corroborate the validity of our schedulability theories.**

I. Introduction

Recently, there is an increasing interest of incorporating cellular architecture into *fieldbuses* (i.e. real-time networks for industrial applications) to support mobile real-time applications. Kang et al. [1] and Wang et al. [2] point out that a cellular network architecture very well suits mobile fieldbuses. The cellular network only deploys wireless in the last hop. This achieves mobility, meanwhile reduces complexity and reliability concerns compared to multi-hop wireless solutions. The cell base stations can be connected by existing wire line fieldbus infrastructures, which is cheaper, simpler, and more reliable. Wei et al. [3] and Leng et al. [4] propose RT-WiFi [3], [4], a wireless fieldbus solution based on WiFi standard [5], to provide high bandwidth as well as high reliability. RT-WiFi can also be regarded as a cellular fieldbus solution. Each RT-WiFi access point serves the role of a cell base station, which provides last hop wireless accesses to its member mobile stations. The access point and its affiliated mobile stations thus form a cell.

1. The Embedded Systems and CPS Lab, Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong S.A.R., China.
2. College of Information Science and Engineering, Northeastern University, Shenyang, China.
3. Dept. of Computer Science and Engineering, University of Notre Dame, IN, USA.
4. Corresponding Author. Email: csqwang@comp.polyu.edu.hk

On the other hand, due to the rapid cost reduction of *commercial-off-the-shelf* (COTS) multichannel wireless nodes (such as WiFi, ZigBee, Bluetooth nodes), people are willing to deploy multiple radio interfaces in a base station, to allow simultaneous access to multiple *radio-frequency* (RF) channels in the cell. For example, any 802.11g/n [5], [6] wireless node can work on three non-overlapping RF channels. An RT-WiFi base station equipped with three of such nodes can thus work on the three RF channels simultaneously. This expands the total RF spectrum available to the cell, hence enhances performance.

This multichannel multiradio abstraction also applies to OFDM [7] and MIMO [8] technologies, two hot topics in 4G and future cellular standards. In OFDM (MIMO), subcarriers (antennas) can be grouped, and each group can be regarded as one radio covering a unique RF channel. We can therefore reasonably expect that in the future, if 4G (or above) technologies are to be incorporated into cellular fieldbuses, multichannel multiradio will still be an inevitable feature.

As multichannel multiradio cellular fieldbus architecture becomes a promising architecture for mobile fieldbuses, a fundamental question is how to carry out real-time scheduling. The simplest answer is TDMA: two spatially overlapping wireless links do not interfere each other if they are scheduled at different time-slots. However, in the case of multichannel multiradio, we have another dimension of freedom: the scheduling of RF channels. Two spatially overlapping wireless links can communicate simultaneously if they are using different channels. Therefore, an important and more specific question arises: how to carry out channel and time-slot (real-time) scheduling in a cellular fieldbus.

Channel time-slot scheduling for traditional cellular networks has been well studied in the *intra*-cell context; but not in the *inter*-cell context [9]. This is partly because in 3G and earlier cellular standards, FDMA/CDMA is deployed between cells, hence no temporal inter-cell scheduling is needed. However, with the rise of 4G cellular standards, neighboring cells may work in the same RF bands at different time-slots . This has triggered an increasing interest in inter-cell scheduling recently [10]–[13]. Yet, how to carry out *inter*-cell channel time-slot scheduling in the cellular *fieldbus* context, which features per flow real-time schedulability guarantee, is still an open problem. This problem is therefore the focus of this paper.

Through this paper, we make the following contributions.

1) We propose a greedy scheduling algorithm for inter-cell channel time-slot scheduling of multichannel multiradio cellular fieldbuses.
2) We introduce a polynomial time closed-form sufficient schedulability test for the greedy scheduling; and dis-

cover the relationship between the test, greedy scheduling schedulability (simplified as "G-schedulability" in the following), and schedulability for general network topologies.

3) For chained topology, a typical cellular fieldbus topology with broad applications, we prove the optimality of greedy scheduling, and find a polynomial time closed-form sufficient and necessary schedulability test.
4) We demonstrate and validate the above schedulability theories with a case study on a classic admission planning problem. The schedulability test not only serves as the planning constraint, but also guides us to devise an approximation algorithm to solve the NP-hard admission planning problem. Comparisons to exhaustive search corroborate the validity of our schedulability theories.

The rest of the paper is organized as follows. Section II discusses related work. Section III formulates the channel time-slot scheduling problem for multichannel multiradio cellular fieldbuses. Section IV presents the greedy scheduling algorithm, the closed-form schedulability test, and discusses the relationship between the test, G-schedulability, and schedulability in the general sense. Section V discusses the equivalence between the closed-form schedulability test, G-schedulability, and schedulability for chained topology cellular fieldbuses, hence proves the optimality of greedy scheduling, and the sufficiency and necessity of the schedulability test. Section VI demonstrates and validates the developed schedulability theories with an admission/resource planning case study. Section VII concludes the paper.

II. Related Work

As a main form of 4G cellular networks [14] [15][1], multichannel multiradio cellular networks' scheduling problems are extensively studied [9]. However, majority of these studies focus on optimizing statistical metrics, such as total capacity/throughput, average delay, and average queue length [11]–[13], [16]. For fieldbus, however, we concern more about *per flow real-time schedulability guarantee*, which is fundamentally different from the aforementioned performance metrics. For example, maximizing the total throughput of a cell does not guarantee every real-time flow is schedulable.

That said, the most relevant subset of multichannel multiradio cellular network scheduling literature is on fairness, i.e. given the total throughput of a cell, each flow is guaranteed a quota of service [9], [10], [17]–[20]. With the service guarantee, we can judge real-time schedulability. However, existing efforts on multichannel multiradio cellular network fairness focus either on intra-cell, or downlink only (which is relevant, as 4G's feature demand/bottleneck is downlink streaming of Internet data, such as video). Furthermore, the fairness considerations are mainly on *proportional fairness scheduling* [21], which is still an opportunistic, rather than guaranteed service[2] for each flow. In contrast, this paper focuses on inter-cell scheduling, uplink and downlink traffic, and per flow real-time schedulability guarantee.

There are other multichannel multiradio cellular related technologies, such as CDMA [22] and beamforming [23]. However, these technologies are not yet widely available in cheap COTS wireless sensors/atuators; neither are they going to be implemented in cellular fieldbuses soon. Therefore, we do not discuss them in this paper.

Besides scheduling multichannel multiradio cellular networks, there are studies on scheduling wireless sensor/mesh/mobile-ad-hoc networks [24]–[44]. However, wireless sensor/mesh/mobile-ad-hoc network research typically focuses on a peer-to-peer multi-hop wireless network topology, or a multi-hop wireless network plus a single sink topology. Such topologies are quite different from this paper's wired base stations cellular network topology, where wireless is only deployed in the last hop. Due to the above difference, for most multichannel multiradio wireless sensor/mesh/mobile-ad-hoc networks, even under given workloads, the channel scheduling problem alone (i.e., given the workload of each flow, how to assign RF channels to each wireless link) is NP-hard. For more general cases, the corresponding optimal scheduling algorithms and schedulability tests are generally open problems [33], [34]. In contrast, this paper studies channel and time-slot scheduling together, proposes an optimal scheduling algorithm, and a closed-form schedulability test for a typical cellular fieldbus topology.

There are works on *intra*-cell scheduling of wireless cellular fieldbuses [3], [4], [32], [45], while this paper's focus is *inter*-cell scheduling. In this sense, this paper complements the intra-cell scheduling papers.

III. Problem Formulation

We assume each base station in the cellular fieldbus is equipped with F radio interfaces, therefore can simultaneously work on F RF channels. Besides, all base stations are well synchronized by the wire line backbone, hence can carry out TDMA scheduling synchronously. The TDMA schedule spans T time-slots, i.e. every T time-slots the TDMA schedule repeats itself. We call the T-time-slot TDMA schedule the *TDMA scheduling superframe*. For narrative simplicity, unless explicitly denoted, we focus on uplink communications in the rest of the paper. Downlink communications analysis can be derived following the same approaches.

Suppose there are totally I cells, which are arbitrarily identified as cell 1, 2, …, i, …, I. A mobile station can move to anywhere inside its cell[3]. We assume there is no real-time monitoring of the interference relations between each

[1]The multichannel multiradio characteristics of 4G cellular networks are typically embodied via OFDM [7] and MIMO [8] technologies.

[2]Unless explicitly denoted, in this paper, when we talk about "service guarantee", we are talking about service guarantee when the wireless medium condition is benign: e.g. when the bit error rate remains below a preset threshold, say 0.1%, so that bit errors are acceptable, or fixable via correction coding.

[3]But not outside of its cell. Otherwise, the per cell workload is changed, and a new scheduling process should be carried out.

pair of wireless links, which is true for most low-cost COTS wireless devices. However, worst case interference range of every RF transmitter can be known offline. Based on this knowledge, we can define offline $\mathsf{nbr}(i) \overset{\text{def}}{=} \{j|j \neq i$; and the ith cell's wireless communications can interfere the jth cell's wireless communications, or vice vesa $\}$ ($i = 1, \ldots, I$). Note, here we do *not* assume interference relation to be symmetric: ith cell interferes jth cell does not necessarily mean jth cell interferes ith cell. But as per the definition, nbr is a symmetric relation: $j \in \mathsf{nbr}(i) \Leftrightarrow i \in \mathsf{nbr}(j)$. Such definition makes sense, because regardless of whether ith cell interferes $j$the cell, or vice versa, ith and jth cell should not be transmitting in the same time-slot via the same RF channel. We denote $H \overset{\text{def}}{=} \max_{i=1,\ldots,I}\{|\mathsf{nbr}(i)|\}$; apparently $H \leqslant I$.

On the other hand, TDMA scheduling implies all data packets are divided into same size *data fragments*, simplified as "*fragments*" in the following. Each fragment takes one TDMA time-slot to transmit over one RF channel. Let $L(i)$ denote the workload of the ith cell. That is, in each T-time-slot TDMA superframe, the ith cell needs to serve $L(i)$ fragments.

Our channel time-slot scheduling addresses the following problem: given I, nbr, L, T, and F, how to schedule L over F RF channels within the T-time-slot superframe, so that every cell finishes its workload, and no two interfering cells (i.e. $i \in \mathsf{nbr}(j)$) ever transmit at the same RF channel in a same time-slot. For narrative simplicity, we denote the above channel time-slot scheduling problem as $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$; and call the aforementioned schedule a *valid schedule*.

Let an $I \times T \times F$ dimension integer array $s[i][t][f]$ ($1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$, $1 \leqslant f \leqslant F$) represent a schedule. $s[i][t][f] = 1$ means the ith cell shall transmit during the tth time-slot in the fth RF channel; any other value means the ith cell shall not transmit during the tth time-slot in the fth RF channel (more specifically, in the following, we use 0 to mean "idle"; and negative value to mean "forbidden"). Then the notion of valid schedule can be formalized as

Definition 1 (Valid Schedule): Schedule s is valid iff the following two conditions both hold:
Condition 1: $\forall i, t, f \in \mathbb{N}^+$ (where $1 \leqslant i \leqslant I$, $1 \leqslant t \leqslant T$, and $1 \leqslant f \leqslant F$), such that $s[i][t][f] = 1$, we have

$$\forall j \in \mathsf{nbr}(i), s[j][t][f] \neq 1. \tag{1}$$

Condition 2: $\forall i \in \{1, \ldots, I\}$,

$$\sum_{t=1}^{T}\sum_{f=1}^{F} \mathsf{cnt}(s[i][t][f]) = L(i), \tag{2}$$

$$\text{where } \mathsf{cnt}(x) = \begin{cases} 1 & \text{(when } x = 1) \\ 0 & \text{(otherwise)} \end{cases}.$$

If a valid schedule exists, we say $\mathcal{S}$ is *schedulable*; and otherwise *unschedulable*.

IV. Greedy Scheduling

In this section, we first analyze the complexity of channel time-slot scheduling problem; and then introduce a greedy scheduling algorithm to solve this problem.

A. Complexity Analysis

In the following, we show that channel time-slot scheduling problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is NP-hard by proving a special case of $\mathcal{S}$ is NP-complete.

Lemma 1 (NP-Completeness of $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$): Determining whether $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ (i.e. a special case of channel time-slot scheduling problem, where $T \equiv 3$ and $F \equiv 1$) is schedulable is NP-complete.

Proof: The problem is NP, and we can reduce the well-known NP-hard problem of graph 3-coloring [46] to this problem. See Appendix A for the proof details. ∎

Lemma 1 implies Theorem 1.

Theorem 1 (NP-Hardness of $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$): Determining whether $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is schedulable is NP-hard.

Proof: As per Lemma 1, the special case of $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ problem, $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$, is already NP-hard. Thus the more general problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is also NP-hard. ∎

B. A Greedy Scheduling Algorithm

As channel time-slot scheduling problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is NP-hard, we do not aim to find a polynomial time algorithm. Instead, we propose a pseudo polynomial time greedy algorithm, G-schedule (see Fig. 1) to solve $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$. The high level idea of G-schedule is to fill the schedule for one cell at a time, using any channel time-slots still available (i.e. not yet interfered by previous cells), and ignoring impacts to the remaining cells.

By counting the level of nested loops in Fig. 1, we see that G-schedule has a pseudo polynomial time-cost of $O(ITFH)$ (hence $O(I^2TF)$, as $H \leqslant I$). Note if T and F are given constants, then G-schedule's time complexity becomes polynomial $O(IH)$ (hence $O(I^2)$, as $H \leqslant I$). The validity of G-schedule is proven by Theorem 2.

Theorem 2 (G-schedule Validity): Upon claiming success (see line 22 of Fig. 1), s returned by G-schedule is a valid schedule.

Proof: $s[i][t][f] = 1, 0, -1$ respectively mean the corresponding slot is reserved, idle, and forbidden. We can then prove the resulted schedule satisfies **Condition 1** and **2** of Definition 1. See Appendix B for details. ∎

As per Theorem 2, if G-schedule algorithm creates a schedule (i.e. claims success), the schedule must be a valid schedule. Thus, in the following, we do not differentiate "G-schedule creates a schedule" and "G-schedule creates a valid schedule".

```
1.  G-schedule(I, nbr, L, T, F) {
2.      int s[I][T][F]; // a I × T × F schedule table
3.      Initialize all elements in s to 0; //0 for "idle"
4.      for (int i ← 1 to I){ // start of I-loop
5.          int l ← L(i);
6.  loop:
7.          for (int t ← 1 to T){ // start of T-loop
8.              for (int f ← 1 to F){ // start of F-loop
9.                  if (l ⩽ 0) break loop;
                    //nothing to schedule
10.                 if (s[i][t][f] = 0) {
11.                     s[i][t][f] ← 1; //1 for "reserved"
12.                     l ← l − 1;
13.                     foreach (j ∈ nbr(i) and j > i)
14.                         s[j][t][f] ← −1; //-1 for "forbidden"
15.                 } // else do nothing
16.             } // end of F-loop
17.         } // end of T-loop
18.         if (l > 0) {
19.             Claim failure; return null;
20.         }
21.     } // end of I-loop
22.     Claim success; return s as the schedule;
23. }
```

Fig. 1. Pseudo Code of G-schedule Algorithm

Furthermore, for a given channel time-slot scheduling problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$, if G-schedule can create a schedule (i.e. claims success), then we say $\mathcal{S}$ is *G-schedulable*. Otherwise, we say $\mathcal{S}$ is *G-unschedulable*.

We are interested in studying the relationship between the following three claims:

C1: $\forall i \in \{1, \ldots, I\}$,

$$L(i) + \sum_{j \in \mathsf{nbr}(i) \wedge j < i} L(j) \leqslant FT,$$

where we define special case $\sum_{j \in \mathsf{nbr}(1) \wedge j < 1} L(j)$ to be of value 0; "$\wedge$" means logical "and".

C2: $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is G-schedulable.

C3: $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ is schedulable.

First, we have

Theorem 3 (G-schedulability Test): Claim **C1** $\Rightarrow$ **C2**.

Proof: We prove **C1** $\Rightarrow$ **C2** with induction.

Apparently when $I = 1$, **C1** $\Rightarrow$ **C2**.

Suppose when $I = k$ ($k \in \mathbb{N}^+$), **C1** $\Rightarrow$ **C2**. $\quad (*)$

When $I = k + 1$, suppose **C1** holds. Let us denote the $(k+1)$ cells channel time-slot scheduling problem as $\mathcal{S}$. Then

$$\forall i \in \{1, \ldots, k+1\}, L(i) + \sum_{j \in \mathsf{nbr}(i) \wedge j < i} L(j) \leqslant FT$$

$\Rightarrow$ $\forall i \in \{1, \ldots, k\}$,
$$L(i) + \sum_{j \in \mathsf{nbr}(i) \setminus \{k+1\} \wedge j < i} L(j) \leqslant FT$$

$\Rightarrow$ Disregarding the $(k+1)$th cell, consider the channel time-slot scheduling problem $\mathcal{S}'$ that only involves the first k cells. As per $(*)$, $\mathcal{S}'$ is G-schedulable.

$\Rightarrow$ For the original $(k+1)$ cells problem $\mathcal{S}$, the first k iterations of G-schedule I-loop (see Fig. 1 line 4) can finish successfully.

For $\mathcal{S}$, in its jth ($1 \leqslant j \leqslant k$) iteration of G-schedule I-loop, if $j \in \mathsf{nbr}(k+1)$, then $L(j)$ elements of $s[k+1][\ldots][\ldots]$ is set to -1 as per line 14. Therefore, right before the start of the $(k+1)$th iteration of I-loop, at the most $\sum_{j \in \mathsf{nbr}(k+1) \wedge j < k+1} L(j)$ elements of $s[k+1][\ldots][\ldots]$ have been set to -1.

Meanwhile, for $\mathcal{S}$, the first k iterations of G-schedule I-loop never assign any element of $s[k+1][\ldots][\ldots]$ to 1. Therefore, for $\mathcal{S}$, right before the $(k+1)$th iteration of G-schedule I-loop, there are at least $FT - \sum_{j \in \mathsf{nbr}(k+1) \wedge j < k+1} L(j) \geqslant L(k+1)$ (as per **C1**) elements of $s[k+1][\ldots][\ldots]$ with value 0. Hence the $(k+1)$th iteration of I-loop will succeed. Therefore, the G-schedule will claim success. That is, **C1** $\Rightarrow$ **C2** for the case of $I = k + 1$.

Induction holds. ■

Note, the converse of Theorem 3 in general does not hold, i.e. **C2** $\not\Rightarrow$ **C1**. For example, cell $\imath$ and $\jmath$ may not interfere each other, but both interfere cell I. When executing the $\imath$th and $\jmath$th I-loop iteration, they may set the same elements of $s[I][\ldots][\ldots]$ to -1. This results in less than $\sum_{j \in \mathsf{nbr}(I) \wedge j < I} L(j)$ elements of $s[I][\ldots][\ldots]$ set to -1. Consequently, a workload $L(I)$ of more than $FT - \sum_{j \in \mathsf{nbr}(I) \wedge j < I} L(j)$ for the Ith cell can be G-scheduled, causing $L(I) + \sum_{j \in \mathsf{nbr}(I) \wedge j < I} L(j) > FT$.

As for **C2** and **C3**, Theorem 2 indeed implies **C2** $\Rightarrow$ **C3**. For whether **C3** $\Rightarrow$ **C2**, we have the following.

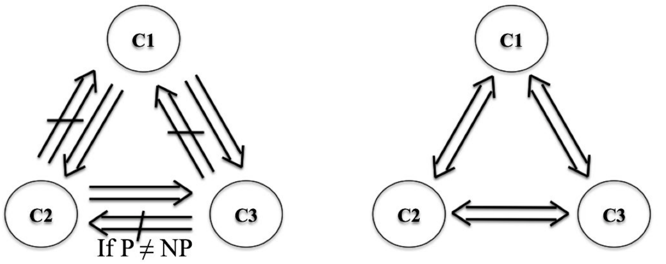
Theorem 4 (General G-schedule Optimality): If P $\neq$ NP, then **C3** $\not\Rightarrow$ **C2**. Or equivalently, if **C3** $\Rightarrow$ **C2**, then P = NP.

Proof: We prove if **C3** $\Rightarrow$ **C2**, NP-complete problem of whether $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is schedulable can be decided in polynomial time, see Appendix C for details. ■

Finally, because **C1** $\Rightarrow$ **C2** and **C2** $\Rightarrow$ **C3**, we have **C1** $\Rightarrow$ **C3**. But **C3** $\not\Rightarrow$ **C1**, because otherwise, **C2** $\Rightarrow$ **C3** and **C3** $\Rightarrow$ **C1** would imply **C2** $\Rightarrow$ **C1**.

The above relations are summarized by Fig. 2(a).

Fig. 2(a) inspires us to question under what conditions **C1**, **C2**, and **C3** are equivalent (see Fig. 2(b)). We find that for a category of very useful cellular fieldbus topology, **C2** $\Rightarrow$ **C1**



(a) (b)

Fig. 2. Relationship between **C1**, **C2**, **C3**: (a) General Case; (b) Special Case.

and **C3** $\Rightarrow$ **C2**. That is **C1**, **C2**, and **C3** are equivalent. We shall discuss this in the next section.

V. Chained Topology Schedulability

A typical cellular fieldbus topology is what we call a *chained topology*. Mathematically, it means the following.

Definition 2 (Chained Topology): I cells of a cellular fieldbus form a chained topology, iff $\forall i \in \{1, \ldots, I\}$, $\forall \imath < i$ such that $i \in \mathsf{nbr}(\imath)$, then $\forall \jmath$ such that $\imath < \jmath < i$, there is $\jmath \in \mathsf{nbr}(\imath)$.

Intuitively, a simple and typical chained topology cellular fieldbus is exemplified by Fig. 3(a), where cells are geographically lined up one by one as per ascending order of their IDs. Suppose for each cell, interference range is always its Θ-hop neighbors (for the case of Fig. 3(a), $\Theta \equiv 2$), then for all $i \in \{1, \ldots, I\}$, for all $\imath < i$ such that $i - \imath \leqslant \Theta$, we have $i \in \mathsf{nbr}(\imath)$. Furthermore, for all $\jmath$ such that $\imath < \jmath < i$, we also have $\jmath \in \mathsf{nbr}(\imath)$. Therefore, the cellular fieldbus of Fig. 3(a) conforms to a chained topology.

In practice, many cellular fieldbuses can have a chained topology: e.g. when a cellular fieldbus runs along a underground mining tunnel (see Fig. 3(b)), along an assembly line, along a transportation track, or along a chemical plant pipe. Basically the chained topology is the wireless equivalent of the well-known "daisy chain" topology, which is defined in the widely adopted FOUNDATION wired fieldbus standard [47].

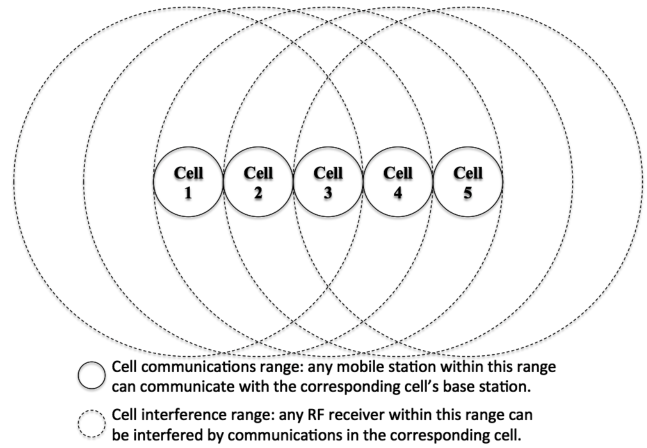
We have the following lemmas:

Lemma 2 (Number of Forbidden Elements in G-schedule): Given a channel time-slot scheduling problem $\mathcal{S}$ for a chained topology cellular fieldbus, if the G-schedule execution reaches right before the ith iteration ($i \in \{1, \ldots, I\}$) of I-loop (see Fig. 1), then at that moment, exactly $\sum_{j \in \mathsf{nbr}(i) \wedge j < i} L(j)$ elements of subarray $s[i][\ldots][\ldots]$ are -1; and the other elements of $s[i][\ldots][\ldots]$ are 0.

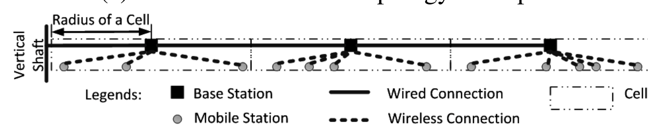
Proof: The lemma trivially holds for $i = 1$. The following thus considers cases where $i > 1$.

If G-schedule execution reaches right before the ith I-loop iteration, then $\forall \imath \in \{1, \ldots, i-1\}$, we have two cases:

Case 1: If $i \notin \mathsf{nbr}(\imath)$, then the $\imath$th I-loop iteration has set no element of subarray $s[i][\ldots][\ldots]$ to -1.



(a) an abstract chained topology example



(b) cellular fieldbus laid along an underground mining tunnel

Fig. 3. Chained topology cellular fieldbus examples

Case 2: If $i \in \mathsf{nbr}(\imath)$, then the $\imath$th I-loop iteration has set $L(\imath)$ different elements of subarray $s[i][\ldots][\ldots]$ to -1.

Furthermore, if both $\imath$th and $\jmath$th ($\imath \neq \jmath$) I-loop iteration belong to Case 2 (without loss of generality, assume $1 \leqslant \imath < \jmath < i$), then they must have set non-overlapping elements of $s[i][\ldots][\ldots]$ to -1. $(\star)$

Otherwise, $\exists t_0$, f_0 ($1 \leqslant t_0 \leqslant T$ and $1 \leqslant f_0 \leqslant F$), such that $s[\imath][t_0][f_0] = s[\jmath][t_0][f_0] = 1$. $(\star\star)$

Assumption $(\star\star)$ implies that in the $\imath$th I-loop iteration, line 11 sets $s[\imath][t_0][f_0]$ to 1. Therefore, in line 14 of the $\imath$th I-loop iteration, $s[\jmath][t_0][f_0]$ is set to -1. This is because the cellular fieldbus conforms to chained topology. As per Definition 2, $i \in \mathsf{nbr}(\imath)$ and $\imath < \jmath < i \Rightarrow \jmath \in \mathsf{nbr}(\imath)$.

Assumption $(\star\star)$ also implies in the $\jmath$th I-loop iteration, line 11 sets $s[\jmath][t_0][f_0]$ to 1, this contradicts the fact that $s[\jmath][t_0][f_0]$ is already set to -1 (hence the evaluation in line 10 is false) in the $\imath$th I-loop iteration.

The above contradiction proves assumption $(\star\star)$ cannot hold. Therefore $(\star)$ holds.

To summarize, right before the execution of the ith I-loop iteration, exactly $\sum_{j \in \mathsf{nbr}(i) \wedge j < i} L(j)$ elements of $s[i][\ldots][\ldots]$ are set to -1. Because the only other element value assignment statement is line 11, which has not affected subarray $s[i][\ldots][\ldots]$ yet, so all the other elements of subarray $s[i][\ldots][\ldots]$ maintain initial value of 0. ∎

Based on the property given in Lemma 2, we can show that channel time-slot scheduling problem $\mathcal{S}$ for chained topology is G-schedulable iff **C1** holds, as implied by Lemma 3.

Lemma 3 (G-schedulability Test Tightness): For a chained topology cellular fieldbus, **C2** $\Rightarrow$ **C1**.

Proof: We prove by induction. All line numbers in the following refer to those of Fig. 1.

Apparently, when $I = 1$, **C2** $\Rightarrow$ **C1**.

Suppose when $I = k$ ($k \in \mathbb{N}^+$), **C2** $\Rightarrow$ **C1**. $\quad (\dagger)$

When $I = k + 1$, suppose **C2** holds. Let us denote the $(k+1)$ cells channel time-slot scheduling problem as $\mathcal{S}$. Then $\mathcal{S}$ is G-schedulable

- $\Rightarrow$ For $\mathcal{S}$, the first k iterations of G-schedule I-loop can all finish successfully.
- $\Rightarrow$ Ignoring the $(k+1)$th cell, consider the channel time-slot scheduling problem $\mathcal{S}'$ that only involves the first k cells, then $\mathcal{S}'$ is G-schedulable.
- $\Rightarrow$ as per ($\dagger$), $\forall i \in \{1, \ldots, k\}$,
$$L(i) + \sum_{j \in \mathsf{nbr}(i) \wedge j<i} L(j) \leqslant FT. \quad (\dagger\dagger)$$

Due to Lemma 2, for G-schedule to succeed in the $(k+1)$th I-loop iteration (as $\mathcal{S}$ is G-schedulable), we must have sufficient elements of $s[k+1][\ldots][\ldots]$ with value 0. That is, $L(k+1) + \sum_{j \in \mathsf{nbr}(k+1) \wedge j<k+1} L(j) \leqslant FT$.

This, together with ($\dagger\dagger$), imply **C1** holds for the case of $I = k + 1$. ■

Next, we prove **C3** $\Rightarrow$ **C2** for chained topology cellular fieldbuses. The proof would need the help of algorithm PartialValidateSchedule (see Fig. 4), which cell by cell checks a given channel time-slot schedule's compliance to **Condition 1** and **2** of Definition 1.

Note that PartialValidateSchedule algorithm is similar to G-schedule algorithm, but fundamentally different. G-schedule creates a schedule; while PartialValidateSchedule partially checks the validity of a given schedule. That is, passing PartialValidateSchedule is a necessary condition for any valid schedule, as described by Lemma 4.

> *Lemma 4 (Necessity of Passing PartialValidateSchedule):* If s is a valid schedule for a channel time-slot scheduling problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$, then PartialValidateSchedule(s, I, nbr, L, T, F) shall claim success.

Proof: Keep in mind $s[i][t][f]$ (or $s'[i][t][f]$) $= 0, 1, -1$ respectively mean the slot is idle, reserved, and forbidden; and **Condition 1** and **2** of Definition 1 must be preserved if the schedule is valid. Details of the proof is in Appendix D. ■

The source code resemblance between G-schedule and PartialValidateSchedule does imply similar properties. Particularly, similar to Lemma 2, we have Lemma 5.

```
PartialValidateSchedule(s, I, nbr, L, T, F) {
    // s is the given I × T × F schedule to validate.
    int s'[I][T][F]; // a temporary schedule
    Initialize all elements in s' to 0;
    for (int i ← 1 to I){ // start of I-loop
        int l ← L(i);
loop:
        for (int t ← 1 to T){ // start of T-loop
            for (int f ← 1 to F){ // start of F-loop
                if (l ⩽ 0) break loop;
                // nothing to schedule
                if (s[i][t][f] = 1) { // told to transmit
                    if (s'[i][t][f] = −1) { // jammed
                        Claim failure; return;
                    }
                    s'[i][t][f] ← 1;
                    l ← l − 1;
                    foreach (j ∈ nbr(i) and j > i)
                        s'[j][t][f] ← −1;
                } // else do nothing
            } // end of F-loop
        } // end of T-loop
        if (l > 0) {
            Claim failure; return;
        }
    } // end of I-loop
    Claim success; return;
}
```

Fig. 4. Pseudo Code of PartialValidateSchedule Algorithm

> *Lemma 5 (Number of Forbidden Elements in PartialValidateSchedule):* Given a channel time-slot scheduling problem $\mathcal{S}(I, \mathsf{nbr}, L, T, F)$ for a chained topology cellular fieldbus, if s is a valid schedule for $\mathcal{S}$, if the PartialValidateSchedule(s, I, nbr, L, T, F) execution reaches right before the ith iteration ($i \in \{1, \ldots, I\}$) of I-loop (see Fig. 4), then at that moment, exactly $\sum_{j \in \mathsf{nbr}(i) \wedge j<i} L(j)$ elements of subarray $s'[i][\ldots][\ldots]$ are -1; and the other elements of $s'[i][\ldots][\ldots]$ are 0.

Proof: The proof is similar to that for Lemma 2. The details are given in Appendix E for reader's convenience. ■

Based on Lemma 4 and 5, we can prove Lemma 6.

> *Lemma 6 (G-schedule Optimality):* For a chained topology cellular fieldbus, **C3** $\Rightarrow$ **C2**.

Proof: Let us prove the equivalent contrapositive statement: $\neg$**C2** $\Rightarrow$ $\neg$**C3**.

If the chained topology cellular fieldbus channel time-slot scheduling problem $\mathcal{S}$ is G-unschedulable, then Theorem 3 $\Rightarrow$ $\exists i \in \{1, \ldots, I\}$, such that $L(i) + \sum_{j \in \mathsf{nbr}(i) \wedge j<i} L(j) > FT$. ($\diamondsuit$)

Assume $\mathcal{S}$ is schedulable, i.e. a valid schedule s exists. ($\diamondsuit\diamondsuit$)

Let us execute PartialValidateSchedule (see Fig. 4) upon s. Lemma 5 implies that right before the execution of the ith I-loop iteration, there are exactly $\sum_{j\in\mathsf{nbr}(i)\wedge j<i} L(j)$ elements of subarray $s'[i][\ldots][\ldots]$ set to -1, and the rest $FT-\sum_{j\in\mathsf{nbr}(i)\wedge j<i} L(j) < L(i)$ (due to $(\diamondsuit)$) elements set to value other than -1. $(\diamondsuit\diamondsuit\diamondsuit)$

Lemma 4 implies that during the ith I-loop iteration, line 16 is executed at least $L(i)$ times. Each such execution corresponds to a distinct element of $s'[i][\ldots][\ldots]$ with value other than -1. So right before the execution of the ith I-loop iteration, there are at least $L(i)$ elements of $s'[i][\ldots][\ldots]$ with value other than -1. This contradicts $(\diamondsuit\diamondsuit\diamondsuit)$. Therefore, assumption $(\diamondsuit\diamondsuit)$ does not hold. That is, $\mathcal{S}$ is unschedulable.

Therefore, we have $\neg$**C2** $\Rightarrow$ $\neg$**C3**. ∎

Theorem 5 (Equivalence): For a chained topology cellular fieldbus, Claim **C1**, **C2**, **C3** are equivalent.

Proof: This is a direct outcome of Theorem 2 (i.e. **C2** $\Rightarrow$ **C3**), Theorem 3 (i.e. **C1** $\Rightarrow$ **C2**), Lemma 3 (i.e. **C2** $\Rightarrow$ **C1**), and Lemma 6 (i.e. **C3** $\Rightarrow$ **C2**). ∎

Theorem 5 has several exciting implications. (i) G-schedule is an *optimal scheduling algorithm*, i.e. any schedulable $\mathcal{S}$ is G-schedulable, because **C3** $\Rightarrow$ **C2**. (ii) G-schedule has a polynomial time ($O(IH)$, hence also $O(I^2)$, as $H \leqslant I$) closed-form and tight schedulability test: **C1** $\Leftrightarrow$ **C2**. (iii) the original scheduling problem $\mathcal{S}$ has a polynomial time ($O(IH)$, hence also $O(I^2)$, as $H \leqslant I$) closed-form and tight schedulability test: **C1** $\Leftrightarrow$ **C3**.

Note that G-schedule's optimality does not mean the schedule produced by G-schedule is the *only* valid schedule for $\mathcal{S}$. If $\mathcal{S}$ is schedulable, it may have other valid schedule(s) than the one produced by G-schedule. Furthermore, this optimal scheduling algorithm has a pseudo polynomial time complexity: $O(ITFH)$ (also $O(I^2TF)$, as $H \leqslant I$). In case T and F are fixed constants, the time complexity becomes polynomial: $O(IH)$ (also $O(I^2)$, as $H \leqslant I$).

VI. Case Study: Admission Planning

G-schedule algorithm and the associated schedulability test provide powerful tools for admission/resource planning of multichannel multiradio cellular fieldbuses. In this section, we demonstrate and validate these tools with a typical admission planning problem. We still assume chained topology cellular fieldbuses, due to the popularity of chained topology, the optimality of the corresponding G-schedule, and the polynomial-time tight schedulability test. We show that our typical admission planning problem is NP-hard, and give an approximation algorithm solution based on the aforementioned tools (Theorem 5 in particular).

A. A Typical Admission Planning Problem

For narrative simplicity, in the following, all time units are "time-slots" and all data size units are "fragments".

Given a chained topology cellular fieldbus, suppose there are a set of J demanded uplink (downlink) flows:

$$\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_J\}. \tag{3}$$

Φ is further divided into I subsets: $\phi(i)$ ($i = 1, \ldots, I$) represents the set of demanded flows in the ith cell. Every demanded flow φ_j ($j = 1, \ldots, J$) is shaped by a token bucket [48] at its source end. The token bucket is of size $c(\varphi_j)$ (fragment), and has a token refilling rate of $c(\varphi_j)/p(\varphi_j)$ (fragment/time-slot). We denote this as $\varphi_j \sim \mathsf{tb}(p(\varphi_j), c(\varphi_j))$. To simplify our model, we require $\forall j$, $c(\varphi_j) \in \mathbb{N}^+$, $p(\varphi_j) \in \mathbb{N}^+$.

The token bucket traffic model is very generic. For example, if a periodic flow φ_j generates $c(\varphi_j)$ fragments of traffic at the beginning of each $p(\varphi_j)$ time-slot period, then it can be regarded as a token bucket shaped flow: $\varphi_j \sim \mathsf{tb}(p(\varphi_j), c(\varphi_j))$.

If $\varphi_j \sim \mathsf{tb}(p(\varphi_j), c(\varphi_j))$, and it is served $\ell(\varphi_j)$ fragments every T-time-slot TDMA superframe, then according to network calculus [48], a *necessary and sufficient condition to upper bound the flow's queueing delay and queue length* is

$$\ell(\varphi_j) \geqslant \frac{c(\varphi_j)}{p(\varphi_j)}T. \tag{4}$$

As long as Ineq. (4) holds, we have queueing delay upper bound $\bar{d}_j = T + \frac{c(\varphi_j)}{\ell(\varphi_j)}T \leqslant T + p(\varphi_j)$, and queue length upper bound $\bar{q}_j = c(\varphi_j) + \frac{c(\varphi_j)}{p(\varphi_j)}T$.

Note in our TDMA cellular fieldbus, $\ell(\varphi_j) \in \mathbb{N}^0$. Therefore, the necessary and sufficient condition of Ineq. (4) becomes

$$\ell(\varphi_j) \geqslant \left\lceil \frac{c(\varphi_j)T}{p(\varphi_j)} \right\rceil. \tag{5}$$

Meanwhile, every demanded flow φ_j corresponds to a reward value $v(\varphi_j) \in \mathbb{N}^+$.

Our admission planning problem asks, given I, nbr, T, F, ϕ, p, c, and v, which demanded flows in Φ should be admitted, so that the total reward is maximized, meanwhile every admitted flow has upper bounded queueing delay and queue length, and the cellular fieldbus is schedulable. We denote this *admission planning problem* as $\mathcal{P}(I, \mathsf{nbr}, T, F, \phi, p, c, v)$.

Based on the necessary and sufficient condition of Ineq. (5), every admitted flow φ_j must be served $\ell(\varphi_j) \geqslant \left\lceil \frac{c(\varphi_j)T}{p(\varphi_j)} \right\rceil$ fragments per T-time-slot superframe to have upper bounded queueing delay and queue length. Let $\phi_{\mathsf{adt}}(i)$ denote the set of admitted flows in the ith cell. Then

$$L(i) = \sum_{\varphi_j \in \phi_{\mathsf{adt}}(i)} \ell(\varphi_j). \tag{6}$$

In case the cellular fieldbus conforms to chained topology, as per Theorem 5, **C3** $\Leftrightarrow$ **C1**. Therefore, admission planning problem $\mathcal{P}(I, \mathsf{nbr}, T, F, \phi, p, c, v)$ can be formulated as a

$(0,1)$-integer planning problem.

$$\max_{x_j\in\{0,1\}} \sum_{\varphi_j\in\Phi} x_j v(\varphi_j) \tag{7}$$

$$\text{s.t.} \qquad \ell(\varphi_j) \geqslant \left\lceil \frac{c(\varphi_j)T}{p(\varphi_j)} \right\rceil \qquad (\forall \varphi_j \in \Phi), \tag{8}$$

$$\ell(\varphi_j) \in \mathbb{N}^0, \tag{9}$$

$$\sum_{j\in\Psi(i)} x_j \ell(\varphi_j) \leqslant FT \qquad (\forall i \in \{1,\ldots,I\}). \tag{10}$$

Here $\Psi(i) \overset{\mathsf{def}}{=} \phi(i) \cup \left(\bigcup_{\imath\in\mathsf{nbr}(i)\wedge\imath<i} \phi(\imath)\right)$; and x_j ($j=1$, …, J) is the $(0,1)$ planning variable: $x_j=1$ meaning flow φ_j is admitted and $x_j=0$ meaning flow φ_j is rejected.

Formulation (7) $\sim$ (10) can be rewritten as follows[4]

$$\max_{x_j\in\{0,1\}} \sum_{\varphi_j\in\Phi} x_j v(\varphi_j) \tag{11}$$

$$\text{s.t.} \qquad \sum_{\varphi_j\in\Psi(i)} x_j \left\lceil \frac{c(\varphi_j)T}{p(\varphi_j)} \right\rceil \leqslant FT \quad (\forall i \in \{1,\ldots,I\}). \tag{12}$$

Now the challenge is how to solve the $(0,1)$-integer planning problem defined by (11) and (12).

B. Solution to $\mathcal{P}$

To solve $\mathcal{P}$, we first show that the admission planning problem defined by (11) and (12) is NP-hard, as summarized in the proposition below.

Proposition 1 (NP-Hardness of $\mathcal{P}$): Admission planning problem $\mathcal{P}(I,\mathsf{nbr},T,F,\phi,p,c,v)$ is NP-hard.

Proof: We prove that the well-known NP-hard knapsack problem [46] can be reduced to $\mathcal{P}(I,\mathsf{nbr},T,F,\phi,p,c,v)$. See Appendix F for the details. ■

Because of NP-hardness, it is impractical to devise polynomial solution for $\mathcal{P}$. Instead, we propose an approximation algorithm solution. The idea is as follows. For chained topology, Theorem 5 let us exploit the closed-form schedulability test **C1** (rewritten as Constraint (12)) to propose a new size metric for flows and flow-sets. With this new metric, we can generalize a well-known knapsack problem approximation algorithm to solve $\mathcal{P}$.

We first propose the following *size* metric for a flow $\varphi_j \in \Phi$.

$$\mathsf{size}(\varphi_j) \overset{\mathsf{def}}{=} (\mathsf{size}_1(\varphi_j),\ldots,\mathsf{size}_I(\varphi_j))^{\mathsf{T}}, \tag{13}$$

where for $i=1$, …, I,

$$\mathsf{size}_i(\varphi_j) \overset{\mathsf{def}}{=} \begin{cases} \left\lceil \frac{c(\varphi_j)T}{p(\varphi_j)} \right\rceil & \text{if } \varphi_j \in \Psi(i), \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Note the above implies $\mathsf{size}(\varphi_j)$ is an $\mathbb{N}^0_{I\times 1}$ vector.

[4]It is easy to prove that Constraint (8) $\sim$ (10) is equivalent to Constraint (12): existence of $\ell(\varphi_j)$ satisfying Constraint (8) $\sim$ (10) implies Ineq (12); and vice versa.

For a set of flows $S \subseteq \Phi$, we define

$$\mathsf{size}(S) \overset{\mathsf{def}}{=} \begin{cases} \sum_{\varphi_j\in S} \mathsf{size}(\varphi_j) & \text{if } S \neq \varnothing, \\ \underbrace{(+\infty,\ldots,+\infty)^{\mathsf{T}}}_{I \text{ components, each is } +\infty} & \text{if } S = \varnothing. \end{cases} \tag{15}$$

Note $\mathsf{size}(S)$ is also an $\mathbb{N}^0_{I\times 1}$ vector; and for the $S\neq\varnothing$ case, the right hand side is a vector sum.

Given $x = (x_1,x_2,\ldots,x_I)^{\mathsf{T}} \in \{0,1\}_{I\times 1}$, let $S(x) \overset{\mathsf{def}}{=} \{\varphi_j | \varphi_j \in \Phi \text{ and } x_j = 1\}$, we can rewrite Constraint (12) as[5]

$$\left|\mathsf{size}(S(x))\right|_\infty \leqslant FT, \tag{16}$$

where $|\xi|_\infty$ is the $\mathcal{L}^\infty$ norm of vector $\xi \in \mathbb{R}_{I\times 1}$: for $\xi = (\xi_1,\xi_2,\ldots,\xi_I)^{\mathsf{T}} \in \mathbb{R}_{I\times 1}$, $|\xi|_\infty \overset{\mathsf{def}}{=} \max_i\{|\xi_i|\}$.

We also define the *reward* metric for an $S\subseteq\Phi$ as

$$\mathsf{reward}(S) \overset{\mathsf{def}}{=} \sum_{\varphi_j\in S} v(\varphi_j). \tag{17}$$

With the above proposed definitions, problem $\mathcal{P}$ as defined by (11) and (12) becomes finding a set S, such that $\mathsf{reward}(S)$ is maximized under the constraint of Ineq. (16). With such transformation, we can generalize a classic knapsack approximation algorithm [49] to solve problem $\mathcal{P}$, as follows.

Let $R = \max_{\varphi\in\Phi}\{v(\varphi)\}$. Then JR is the upper bound for objective function (11). For each $j\in\{1,\ \ldots,\ J\}$ and $r \in \{1,\ \ldots,\ JR\}$, let

$$S_{j,r} \overset{\mathsf{def}}{=} \underset{S\subseteq\{\varphi_1,\varphi_2,\ldots,\varphi_j\}\wedge\mathsf{reward}(S)=r}{\mathsf{argmin}} \{\left|\mathsf{size}(S)\right|_\infty\}. \tag{18}$$

That is, $S_{j,r}$ is the subset of $\{\varphi_1,\varphi_2,\ldots,\varphi_j\}$, whose reward is exactly r, and whose size-norm (in terms of $\mathcal{L}^\infty$-norm) is the minimal.

We have the following dynamic programming algorithm to construct all $S_{j,r}$s ($j=1,\ \ldots,\ J$; and $r=0,\ \ldots,\ JR$).

First, for each $r=0,\ \ldots,\ JR$, we have

$$S_{1,r} \overset{\mathsf{def}}{=} \begin{cases} \{\varphi_1\} & \text{if } v(\varphi_1) = r, \\ \varnothing & \text{otherwise.} \end{cases} \tag{19}$$

Next, the dynamic programming runs as follows:

$$S_{j+1,r} = \begin{cases} \min_{\mathcal{L}^\infty}\{S_{j,r},\ S_{j,r-v(\varphi_{j+1})} \cup \{\varphi_{j+1}\}\} \\ \quad (\text{if } (v(\varphi_{j+1}) < r \wedge S_{j,r-v(\varphi_{j+1})} \neq \varnothing) \\ \quad \text{or } (v(\varphi_{j+1}) = r)), \\ S_{j,r} \qquad\qquad\qquad\qquad (\text{otherwise}). \end{cases} \tag{20}$$

Here $\min_{\mathcal{L}^\infty}\{\sigma_1,\sigma_2\}$ returns the σ_i ($i\in\{1,2\}$), whose $|\mathsf{size}(\sigma_i)|_\infty$ value is the smallest.

We have the following proposition.

[5]Note assuming $I\in\mathbb{N}^+$, and each demanded flow is schedulable alone (when there are no other demanded flows in the entire cellular fieldbus), the solution to $\mathcal{P}$ can never be $x=(0,0,\ldots,0)^{\mathsf{T}}$, i.e. $S(x)=\varnothing$. This complies with Ineq. (16), as $\left|\mathsf{size}(\varnothing)\right|_\infty = +\infty > FT$.

Proposition 2 (Solution to $\mathcal{P}$): The solution to objective function (11) is

$$r^* = \max\Big\{r \Big| \big|\mathsf{size}(S_{J,r})\big|_\infty \leqslant FT\Big\}. \tag{21}$$

The corresponding admitted flow set is S_{J,r^*}.

Proof: First, as per Eq. (21), S_{J,r^*} is feasible under Constraint (16). Suppose there is another feasible solution $S_{J,r^{**}}$ (i.e. $\big|\mathsf{size}(S_{J,r^{**}})\big|_\infty \leqslant FT$), where $r^{**} > r^*$; then it contradicts the fact of Eq. (21). ■

The above dynamic programming algorithm's time cost is $O(J^3IR)$. The involvement of R implies the time cost is pseudo polynomial instead of polynomial. To make it polynomial, the following classic approximation [49] is carried out.

1) Given $\varepsilon > 0$, let $K = \frac{\varepsilon R}{J}$.
2) For each demanded flow φ_j, revise reward value function as $v'(\varphi_j) = \lfloor\frac{v(\varphi_j)}{K}\rfloor$, and revise the set reward definition Eq. (17) accordingly: now $\mathsf{reward}'(S) \stackrel{\mathsf{def}}{=} \sum_{\varphi_j\in S} v'(\varphi_j)$.
3) Solve the updated admission planning problem $\mathcal{P}'(I, \mathsf{nbr}, T, F, \phi, p, c, v')$ with the dynamic programming algorithm of (19) (20), denote the admitted flow set to be S'.

Based on the the conclusion in [49],

Proposition 3 (Approximation Optimality):

$$\mathsf{reward}(S') \geqslant (1-\varepsilon)r^*. \tag{22}$$

Proof: The original idea of the proof can be found in [49]. We rewrite the proof in Appendix G as per our context for reader's convenience. ■

The time complexity of the approximation algorithm is

$$O(J^3I\lfloor\frac{R}{K}\rfloor) = O(J^3I\lfloor\frac{J}{\varepsilon}\rfloor) = O(J^4I\frac{1}{\varepsilon}).$$

This implies that the approximation algorithm is a *fully polynomial time approximation scheme* (i.e. polynomial in terms of the bit size of the problem and the given constant $\frac{1}{\varepsilon}$) [49].

C. Numerical Example

Below we give a numerical example to demonstrate the aforementioned admission planning methodology, and to validate the schedulability test **C1** implied by Theorem 5.

Consider a chained topology cellular fieldbus consists of three sequentially laid out cells: cell 1, 2, and 3. This implies $I = 3$. The demanded flow set Φ consists of $J = 9$ flows: $\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_9\}$ as shown in Table I.

Suppose cells only interfere one hop neighbor(s); the TDMA superframe consists of $T = 3$ time-slots; each base station is equipped with $F = 2$ RF channels.

Then using *exhaustive search* (Exhst), pseudo polynomial time *optimal algorithm* (Opt) of (19) and (20), and *approximation algorithm* (Aprx) with $\varepsilon = 10\%$ respectively, we get the admission plans and schedules as shown in Table II.

TABLE I
NUMERICAL EXAMPLE DEMANDED FLOWS $\Phi = \{\varphi_1, \varphi_2, \ldots, \varphi_9\}$

φ	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9
host cell	1	1	1	2	2	2	3	3	3
$p(\varphi)$	10	4	7	10	6	7	5	11	9
$c(\varphi)$	1	3	3	1	5	6	3	4	6
$v(\varphi)$	8	39	38	50	29	41	3	20	30

TABLE II
COMPARISONS OF ADMISSION PLANNINGS AND SCHEDULES

	Admitted Flows	Total Reward
Exhst	$\{\varphi_2, \varphi_3, \varphi_4, \varphi_8, \varphi_9\}$	177
Opt	$\{\varphi_2, \varphi_3, \varphi_4, \varphi_8, \varphi_9\}$	177
Aprx $\varepsilon = 0.1$	$\{\varphi_2, \varphi_3, \varphi_4, \varphi_7, \varphi_9\}$	160

(a) Admitted Flows and Total Values

Exhst									
	cell 1			cell 2			cell 3		
	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
f_1	φ_2	φ_2	φ_3	×	×	×	φ_9	φ_8	×
f_2	φ_2	φ_3	×	×	×	φ_4	φ_9	φ_8	×

Opt									
	cell 1			cell 2			cell 3		
	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
f_1	φ_2	φ_2	φ_3	×	×	×	φ_9	φ_8	×
f_2	φ_2	φ_3	×	×	×	φ_4	φ_9	φ_8	×

Aprx $\varepsilon = 0.1$									
	cell 1			cell 2			cell 3		
	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
f_1	φ_2	φ_2	φ_3	×	×	×	φ_9	φ_7	×
f_2	φ_2	φ_3	×	×	×	φ_4	φ_9	φ_7	×

×: the slot is idle or forbidden.

(b) Schedules ($t_1 \sim t_3$: TDMA time slots; $f_1 \sim f_2$: RF channels)

Note the exhaustive search (Exhst) method exhaustively tries all possible combinations of admitted flow sets; and for each admitted flow set, it exhaustively tries all possible combinations of schedules. The schedulability determined by Exhst is henceforth trustworthy, the schedule produced by Exhst is henceforth guaranteed to be optimal, and Exhst does not rely on the schedulability test **C1** at all.

In contrast, Opt method relies on the schedulability test **C1**. As Opt reaches the same schedulability decisions as Exhst, and produces the same optimal schedule, the validity of schedulability test **C1** is corroborated.

VII. CONCLUSION

In this paper, we study the inter-cell channel time-slot scheduling of multichannel multiradio cellular fieldbus. A greedy scheduling algorithm is proposed, together with a polynomial time closed-form schedulability test. We reveal the relationship between the schedulability test, greedy schedulability, and schedulability in the general sense. Furthermore, we prove the equivalence of the three in the context of chained topology, a typical cellular fieldbus topology with broad applications. This also implies the optimality of greedy scheduling, and the sufficiency and necessity of the schedulability test. We demonstrate and validate these schedulability theories with a classic admission planning problem, where the schedulability test not only serves as the planning constraint, but also guides us to propose an approximation algorithm for the NP-hard admission planning problem. Comparisons with exhaustive search corroborate the validity of the theories.

ACKNOWLEDGEMENT

The research project related to this paper is supported in part by Hong Kong RGC General Research Fund (GRF) PolyU 152164/14E, RGC Early Career Scheme (ECS) PolyU 5328/12E, The Hong Kong Polytechnic University internal fund A-PJ80, A-PK46, A-PL82, G-YN37, Department of Computing start up fund; and by China NSF 61472072, National Basic Pre-research Program of China 2014CB360509. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of sponsors.

The authors thank anonymous reviewers for their efforts/advice on improving this paper.

REFERENCES

[1] K. Kang *et al.*, "Medical-grade quality of service for real-time mobile healthcare," *IEEE Computer*, vol. 48, no. 2, pp. 41–49, Feb. 2015.

[2] Q. Wang *et al.*, "Building robust wireless LAN for industrial control with the DSSS-CDMA cell phone network paradigm," *IEEE Trans. on Mobile Comp.*, vol. 6, no. 6, pp. 706–719, Jun. 2007.

[3] Y.-H. Wei, Q. Leng, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "Rt-wifi: Real-time high-speed communication protocol for wireless cyber-physical control applications," *Proc. of IEEE RTSS*, 2013.

[4] Q. Leng, Y.-H. Wei, S. Han, A. K. Mok, W. Zhang, and M. Tomizuka, "Improving control performance by minimizing jitter in rt-wifi networks," *Proc. of IEEE RTSS*, pp. 63–73, 2014.

[5] IEEE-SA Standards Board, *IEEE Std 802.11g-2003*. IEEE, 2003.

[6] ——, *IEEE Std 802.11n-2009*. IEEE, 2009.

[7] J. Heiskala and J. Terry, *OFDM Wireless LANs: A Theoretical and Practical Guide*, Dec. 2001.

[8] G. Foschini and M. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 331–335, Mar. 1998.

[9] A. Asadi and V. Mancuso, "A survey on opportunistic scheduling in wireless communications," *IEEE Surveys and Tutorials on Communications*, vol. 15, no. 4, pp. 1671–1688, 4th Quarter 2013.

[10] A. Hooshmand *et al.*, "Joint inter-cell interference coordination and forced cooperative scheduling for the downlink of lte systems," *Proc. of IEEE WCNC'14 Track 2*, pp. 1880–1884, 2014.

[11] W. Yu, T. Kwon, and C. Shin, "Multicell coordination via joint scheduling, beamforming, and power spectrum adaptation," *IEEE Trans. on Wireless Communications*, vol. 12, no. 7, pp. 3300–3313, Jul. 2013.

[12] X. Lin *et al.*, "Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks," *IEEE/ACM ToN*, vol. 17(6):1874-1887, Dec. 2009.

[13] W. Choi *et al.*, "The capacity gain from intercell scheduling in multi-antenna systems," *IEEE TWC*, vol. 7(2):714-725, Feb. 2008.

[14] H. Ekstrom *et al.*, "Technical solutions for the 3g long-term evolution," *IEEE Communications Magazine*, vol. 44(3):38-45, Mar. 2006.

[15] IEEE-SA Standards Board, *IEEE Std 802.16-2004*. IEEE, 2004.

[16] M. Alicherry *et al.*, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE JSAC*, vol. 24(11):1960-1971, Nov. 2006.

[17] B. A. Salihu *et al.*, "Scheduling in interference-limited environment for lte-a systems," *Intl. J. of Future Generation Communication and Networking*, vol. 7(5):179-190, 2014.

[18] D. Parruca *et al.*, "On semi-static interference coordination under proportional fair scheduling in lte systems," *Proc. of European Wireless Conf.*, pp. 1, 8, 16–18, April 16-18 2013.

[19] S. Bi and Y. J. Zhang, "Outage-optimal TDMA based scheduling in relay-assisted MIMO cellular networks," *IEEE Trans. on Wireless Communications*, vol. 11(4):1488-1499, 2012.

[20] Y. Ma, "Proportional fair scheduling for downlink ofdma," *Proc. of ICC*, pp. 4843–4848, 2007.

[21] A. Jalali *et al.*, "Data throughput of cdma-hdr a high efficiency-high data rate personal communication wireless system," *Proc. of IEEE 51st Vehicular Tech. Conf. (VTC)*, vol. 3, pp. 1854–1858, 2000.

[22] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley, 1995.

[23] B. D. V. Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, pp. 4–24, Apr. 1988.

[24] C.-T. Chiang *et al.*, "A decentralized minislot scheduling protocol (dmsp) in tdma-based wireless mesh networks," *Journal of Network and Computer Applications*, vol. 37:206-215, 2014.

[25] K. S. Prabh, "Bandwidth allocation in hexagonal wireless sensor networks for real-time communications," *Proc. of Intl. Conf. on Networking, Sensing and Control*, pp. 451–456, Apr. 11-13 2011.

[26] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16:985-997, 2010.

[27] X. Liu, Q. Wang, W. He, M. Caccamo, and L. Sha, "Optimal real-time sampling rate assignment for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2(2):263-295, May 2006.

[28] L. Tassiulas *et al.*, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. on Auto. Control*, vol. 37(12):1936-1948, 1992.

[29] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," *Proc. of IEEE RTSS*, pp. 150–159, 2010.

[30] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "A spatiotemporal communication protocol for wireless sensor networks," *IEEE TPDS*, vol. 16(10):995-1006, 2005.

[31] T. Facchinetti, L. Almeida, G. C. Buttazzo, and C. Marchini, "Real-time resource reservation protocol for wireless mobile ad hoc networks," *Proc. of IEEE RTSS*, 2004.

[32] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, "An implicit prioritized access protocol for wireless sensor networks," *Proc. of IEEE RTSS*, 2002.

[33] R. Soua and P. Minet, "Multichannel assignment protocols in wireless sensor networks: A comprehensive survey," *Pervasive and Mobile Computing*, vol. (Preprint), May 2014.

[34] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks," *J. Parallel Distrib. Comput.*, vol. 70, pp. 505–524, 2010.

[35] L.-W. Chen *et al.*, "Exploiting spectral reuse in routing, resource allocation, and scheduling for ieee 802.16 mesh networks," *IEEE Trans. on Vehicular Tech.*, vol. 58(1), Jan. 2009.

[36] S. Liu *et al.*, "Slot allocation algorithms in centralized scheduling scheme for ieee 802.16 based wireless mesh networks," *Computer Communications*, vol. 32(5):943-953, 2009.

[37] F. Jin *et al.*, "Routing and packet scheduling for throughput maximization in ieee 802.16 mesh networks," *Proc. of the IEEE Intl' Conf. on Broadband Communications, Networks and Systems*, 2007.

[38] Y. Lu *et al.*, "Maintaining routing tree in ieee 802.16 centralized scheduing mesh networks," *Proc. of ICCCN*, pp. 240–245, 2007.

[39] M. Cao, V. Raghunathan, and P. R. Kumar, "A tractable algorithm for fair and efficient uplink scheduling of multi-hop wimax mesh networks," *Proc. of IEEE WiMesh*, pp. 93–100, 2006.

[40] J. Chen *et al.*, "An odd-even alternation mechanism for centralized scheduling in wimax mesh networks," *Proc. of IEEE GLOBECOM*, 2006.

[41] B. Han *et al.*, "Performance evaluation of scheduling in IEEE 802.16 based wireless mesh networks," *Proc. of IEEE MASS*, 2006.

[42] H. Viswanathan *et al.*, "Throughput-range tradeoff of wireless mesh backhaul networks," *IEEE JSAC*, vol. 24(3):593-602, Mar. 2006.

[43] L. Fu, Z. Cao, and P. Fan, "Spatial reuse in ieee 802.16 based wireless mesh networks," *Proc. of ISCIT'05*, 2005.

[44] H.-Y. Wei *et al.*, "Interference-aware ieee 802.16 wimax mesh networks," *Proc. of IEEE VTC*, pp. 3102–3106, 2005.

[45] B. Andersson, N. Pereira, and E. Tovar, "Analyzing tdma with slot skipping," *IEEE TII*, vol. 4(4):225-236, 2008.

[46] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

[47] Fieldbus Foundation, *FOUNDATION Fieldbus*. http://www.fieldbus.org.

[48] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer, 2001.

[49] V. V. Vazirani, *Approximation Algorithms*. Springer, 2003.

Appendix A
Proof of Lemma 1

First, it is trivial to see when $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is schedulable, a valid schedule can be verified within polynomial time: $O(I \times I)$. Therefore, $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is NP.

Next, we prove that the well-known NP-hard problem of graph 3-coloring $\mathcal{G}(V, E)$ [46] can be reduced (in polynomial time) to a $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ schedulability determination problem.

The graph 3-coloring problem $\mathcal{G}(V, E)$ is as follows: given a graph $G = (V, E)$, where V and E are respectively G's vertex and edge set, can we color the vertices with 3 colors, such that the endpoints of every edge are colored differently? A 3-coloring is a function $f : V \mapsto \{1, 2, 3\}$ such that for every edge $(u, v) \in E$, we have $f(u) \neq f(v)$. If there exists such a function, we say the graph is 3-colorable.

Given a graph 3-coloring problem $\mathcal{G}(V, E)$, where $V = \{v_1, v_2, \ldots\}$ is the set of vertices and E is the set of edges, we can construct a $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ problem in polynomial time as follows:

1) There are $I = |V|$ cells.
2) For each cell i, we have $j \in \mathsf{nbr}(i)$ iff there is an edge in E connecting v_i and v_j in G.
3) For each cell i, set $L(i) = 1$.

Denote the above constructed problem as $\mathcal{S}_\mathcal{G}$.

In the following, we show that $\mathcal{S}_\mathcal{G}$ is schedulable iff graph $G(V, E)$ is 3-colorable.

We first prove the "if" direction. If G is 3-colorable, then there exists a function f that maps each vertex to a color in $\{1, 2, 3\}$. Then we can construct a schedule s for $\mathcal{S}_\mathcal{G}$ according to the coloring: if vertex v_i is colored $k \in \{1, 2, 3\}$, then we let cell i transmit in the kth time-slot. Since the colors of a pair of connected vertices in G have different colors, so any pair of cells i and j with $i \in \mathsf{nbr}(j)$ (and hence $j \in \mathsf{nbr}(i)$) do not transmit in the same time-slot. This implies **Condition 1** in Definition 1 holds. Since each vertex in G is assigned a color, so each cell is assigned a time-slot, which implies **Condition 2** in Definition 1 holds. In summary, s is a valid schedule for $\mathcal{S}_\mathcal{G}$, i.e. $\mathcal{S}_\mathcal{G}$ is schedulable.

To prove the "only if" direction, we suppose $\mathcal{S}_\mathcal{G}$ is schedulable. Suppose a valid schedule is s. Then we can color vertex v_i in G with the kth ($k \in \{1, 2, 3\}$) color iff cell i is transmitting in the kth time-slot of s. Because s is a valid schedule for $\mathcal{S}_\mathcal{G}$, according to Definition 1, every vertex is colored, and only colored by one of the 3 colors; and no two connected vertices are assigned the same color. ■

Appendix B
Proof of Theorem 2

In the following, all line numbers refer to those of Fig. 1.

First, due to line 12 and 19, if G-schedule is to claim success, in the ith ($i \in \{1, \ldots, I\}$) iteration of I-loop (see line 4), line 11 is guaranteed to be executed $L(i)$ times. The assignment of value 1 by line 11, once happened, can never be changed, as future execution of line 14 only affects cells of bigger IDs. Therefore, when line 22 is executed, subarray $s[i][\ldots][\ldots]$ has $L(i)$ elements with value 1. Therefore, the returned s satisfies **Condition 2** of Definition 1.

Second, assume when G-schedule claims success, $\exists i$, t, f, $j \in \mathsf{nbr}(i)$ (i.e. $i \in \mathsf{nbr}(j)$, as nbr is a symmetric relationship), such that $s[i][t][f] = s[j][t][f] = 1$.

As nbr is a symmetric relationship, without loss of generality, suppose $i < j$. Then in the ith iteration of I-loop that sets $s[i][t][f]$ to 1, line 14 should have set $s[j][t][f]$ to -1; and after that, $s[j][t][f]$ can never be assigned any other value (0 or 1 to be specific). This contradicts the assumption that in the end $s[j][t][f] = 1$. Therefore, the original assumption is wrong. This means s satisfies **Condition 1** of Definition 1. ■

Appendix C
Proof of Theorem 4

As Theorem 2 already proves **C2** ⇒ **C3**, if in addition **C3** ⇒ **C2**, then we have **C3** ⇔ **C2**.

Due to Lemma 1, the problem on determining whether $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is schedulable is NP-complete. However, if **C3** ⇔ **C2**, then whether $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is schedulable can be determined in polynomial time $O(I^2)$, by simply checking whether $\mathcal{S}(I, \mathsf{nbr}, L, 3, 1)$ is G-schedulable (note here as $T \equiv 3$ and $F \equiv 1$, G-schedule time complexity reduces from pseudo polynomial $O(I^2TF)$ to polynomial $O(I^2)$). That is, an NP-complete problem can be determined in polynomial time. Therefore P = NP. ■

Appendix D
Proof of Lemma 4

The line numbers in this proof all refer to those of Fig. 4.

It is trivial to prove that if s is a valid schedule, then the 1st I-loop iteration never executes line 13, nor line 23. (‡)

For the ith ($i \in \{2, \ldots, I\}$) I-loop iteration, as s is a valid schedule, due to **Condition 2** of Definition 1, there must be $L(i)$ times that line 11 evaluates to true, respectively for $L(i)$ different $s[i][\ldots][\ldots]$ elements.

Suppose for t_0, f_0 ($1 \leqslant t_0 \leqslant T$, $1 \leqslant f_0 \leqslant F$) line 11 evaluates to true, i.e. $s[i][t_0][f_0] = 1$. Then the following line 12 must evaluate to false, i.e. $s'[i][t_0][f_0] \neq -1$. This can be proven by contradiction. (‡‡)

Suppose $s'[i][t_0][f_0] = -1$, then $\exists \imath$ ($1 \leqslant \imath < i$), such that in the $\imath$th I-loop iteration, $s'[i][t_0][f_0]$ is set to -1 by line 18. That is, $i \in \mathsf{nbr}(\imath)$. Also, for line 18 to execute, the corresponding line 11 must evaluate to true, i.e. $s[\imath][t_0][f_0] = 1$. So we have $s[i][t_0][f_0] = s[\imath][t_0][f_0] = 1$ and $i \in \mathsf{nbr}(\imath)$. This contradicts the given fact that s is a valid schedule, which must comply with **Condition 1** of Definition 1. Proposition (‡‡) holds. Therefore, in the ith I-loop iteration, line 13 can never be executed.

Meanwhile, as in the ith I-loop iteration, line 11 is evaluated to true for $L(i)$ times, and the following line 12 always evaluate to false, therefore line 16 is executed $L(i)$ times. This means line 22 cannot evaluate to true. Hence line 23 is never executed.

Therefore, in the $\imath$th ($i \in \{2, \ldots, I\}$) I-loop iteration, neither line 13, nor line 23 can ever be executed. This, combined with proposition (‡), implies algorithm PartialValidateSchedule must return via line 26, i.e. claim success. ■

Appendix E
Proof of Lemma 5

All line numbers refer to those of Fig. 4.

The lemma trivially sustains for $i = 1$. The following thus considers cases where $i > 1$.

If PartialValidateSchedule execution reaches right before the ith I-loop iteration, then $\forall \imath \in \{1, \ldots, i-1\}$, we have two cases:

Case 1: If $i \notin \mathsf{nbr}(\imath)$, then the $\imath$th I-loop iteration has set no element of subarray $s'[i][\ldots][\ldots]$ to -1.

Case 2: If $i \in \mathsf{nbr}(\imath)$, then the $\imath$th I-loop iteration has set $L(\imath)$ different elements of subarray $s'[i][\ldots][\ldots]$ to -1, as s is a valid schedule, hence satisfies **Condition 2** of Definition 1.

Furthermore, if both $\imath$th and $\jmath$th ($\imath \neq \jmath$) I-loop iteration belong to Case 2 (without loss of generality, assume $1 \leqslant \imath < \jmath < i$), then they must have set non-overlapping elements of $s'[i][\ldots][\ldots]$ to -1. (♣)

Otherwise, $\exists t_0$, f_0 ($1 \leqslant t_0 \leqslant T$ and $1 \leqslant f_0 \leqslant F$) such that $s'[\imath][t_0][f_0] = s'[\jmath][t_0][f_0] = 1$. (♣♣).

Assumption (♣♣) implies in the $\imath$th I-loop iteration, line 15 sets $s'[\imath][t_0][f_0]$ to 1. Therefore, in line 18 of the $\imath$th I-loop iteration, $s'[\jmath][t_0][f_0]$ is set to -1. This is because the cellular fieldbus conforms to chained topology. As per Definition 2, $i \in \mathsf{nbr}(\imath)$ and $\imath < \jmath < i \Rightarrow \jmath \in \mathsf{nbr}(\imath)$.

Assumption (♣♣) also implies in the $\jmath$th I-loop iteration, line 15 sets $s'[\jmath][t_0][f_0]$ to 1, this contradicts the fact that $s'[\jmath][t_0][f_0]$ is already set to -1 (hence the evaluation in line 12 is true, and line 15 cannot happen) in the $\imath$th I-loop iteration.

The above contradiction proves assumption (♣♣) cannot hold. Therefore (♣) holds.

To summarize, right before the execution of the ith I-loop iteration, exactly $\sum_{j \in \mathsf{nbr}(i) \wedge j < i} L(j)$ elements of $s'[i][\ldots][\ldots]$ are set to -1. Because the only other element value assignment statement is line 15, which has not affected subarray $s'[i][\ldots][\ldots]$ yet, so all the other elements of subarray $s'[i][\ldots][\ldots]$ maintain initial value of 0. ■

Appendix F
Proof of Proposition 1

We prove the well-known NP-hard knapsack problem [46] can be reduced to $\mathcal{P}(I, \mathsf{nbr}, T, F, \phi, p, c, v)$ in polynomial time.

Given a knapsack problem, we can construct an admission planning problem, with $I = 1$, $\mathsf{nbr}(1) = \varnothing$, T equals the size of the knapsack B, $F = 1$. For each knapsack candidate item u_j, we construct a demanded flow φ_j, where

1) $v(\varphi_j)$ equals the value of item u_j;
2) $c(\varphi_j)$ equals the size of item u_j; and $p(\varphi_j) \equiv B$.

All the above constructed flows together form set $\phi(1)$.

In this way, admission planning problem $\mathcal{P}(I, \mathsf{nbr}, T, F, \phi, p, c, v)$ in the form of (11) and (12) is equivalent to the original knapsack problem. ■

Appendix G
Proof of Proposition 3

The original idea of the proof can be found in [49]. We rewrite the proof here according to our context for reader's convenience.

Based on the definition of K, v', and reward', for any $\sigma \subseteq \Phi$, we have

$$K \cdot \mathsf{reward}'(\sigma) \leqslant \mathsf{reward}(\sigma) \leqslant K \cdot \mathsf{reward}'(\sigma) + JK.$$

Let S^* denote the optimal admitted flow set using the original reward values v. Then

$$\mathsf{reward}(S') \geqslant K \cdot \mathsf{reward}'(S') \geqslant K \cdot \mathsf{reward}'(S^*)$$
$$\geqslant \mathsf{reward}(S^*) - JK = r^* - \varepsilon R \geqslant r^* - \varepsilon r^* = (1-\varepsilon) r^*.$$

■