Research Article

A Source Aware Scheduling Algorithm for Time-Optimal Convergecast

Jing Zhao, Yajuan Qin, Dong Yang, and Ying Rao
National Engineering Laboratory for Next Generation Internet Interconnection Devices,
School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
Correspondence should be addressed to Dong Yang; dyang@bjtu.edu.cn
Received 13 January 2014; Revised 13 May 2014; Accepted 19 May 2014; Published 5 June 2014
Academic Editor: Li Xu
Copyright © 2014 Jing Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most applications of industrial wireless sensor networks (IWSNs) should converge process data generated by each node to the central manager. The data collection operation results in an important communication primitive referred to as convergecast. Convergecast is a many-to-one communication paradigm as a critical functionality deployed for industrial monitoring and control. Delaying of process data may degrade the overall control performance and even lead to the malfunction of industrial applications. Therefore, timeslot and channel resources should be scheduled efficiently for real-time communication. This paper is interested in determining a TDMA schedule that minimizes the number of timeslots and completes convergecast with a limited number of channels. In order to achieve the lower bound derived by theoretical analysis, we proposed a source aware scheduling algorithm for general network. For IWSNs with a fixed number of available channels, we present a source aware scheduling algorithm with constrained channel. According to simulation results, we demonstrate that the performance of our algorithm is close to the lower bound on latency with a limited number of channels. Our algorithm is also scalable for schedules with multiple packets and specific transmission latency of a single packet.

1. Introduction

The emergence of researches on wireless communication has enabled wireless sensor networks (WSNs) applied to industrial applications [1–4]. Several industrial organizations such as ISA [5], WirelessHART [6], and ZigBee [7] have been promoting the development of wireless technology in industrial application. IWSNs improve industrial monitoring and control due to their flexibility, scalability, and efficiency. However, concerns about network latency, reliability, and security along with harsh and noisy industrial environments impose many challenges in designing efficient protocols for IWSNs. The characteristics of WSNs such as limited energy, computing, and storage capability also need to be considered [8, 9]. To operate a control loop, real-time data delivery is instrumental.

The architecture in WirelessHART is designed to be a centralized wireless mesh network which pushes the complexity of ensuring communication performance to a central manager, as in Figure 1. The medium access control (MAC) sublayer of WirelessHART relies on time division multiple access (TDMA) [10] combined with channel hopping to enhance reliable real-time data communication. The contention-free MAC protocols can obtain a deterministic delay bound and eliminate collisions to achieve end-to-end real-time communication. Moreover, TDMA is energy efficient since the amount of idle listening and overhearing can be reduced and retransmissions caused by collisions can be eliminated. Channel hopping makes links with different channels be assigned the same timeslot to scale down the total number of scheduled timeslots. The central manager computes and distributes the transmission schedules of timeslots and channels to all devices based on the global network information. However, the existing standards provide little guidance on how to produce an effective scheduling solution.

In IWSNs, the reliable and real-time communication ensures availability of process data for industrial application. Collection from devices has a designated sample rate to converge these process data to the central manager through multihop transmission. The convergecast transmits a large
burst of packets within a period to the central manager. Two kinds of convergecast, aggregated data and raw data, have been studied in [11–13] for aggregated data and in [14–16] for raw data. For many industrial applications, all raw data should be sent to the central manager for low correlation between the data. How to determine a TDMA schedule is a critical problem in minimal number of timeslots using a limited number of channels. The important contributions of our work include the following.

(i) We present a source aware scheduling algorithm which assigns a dedicated schedule to each link of a source path, which requires at most \(2n_s - 1, N - 1\) timeslots, where \(N\) is the number of nodes in the networks and \(n_s\) is the maximum number of nodes in any subtree. And our algorithm is proven to attain the derived lower bound.

(ii) We present a scheduling algorithm for latency-optimal convergecast with a fixed number of channels. Then we carry out simulations to demonstrate that performance of our algorithm is closed to the derived lower bound on latency for convergecast.

(iii) We present a simple modification of source aware scheduling algorithm to handle some nodes with multiple packets. Our algorithm also can guarantee the delivery time of different packets by setting priorities of transmission.

The remainder of this paper is organized as follows. In Section 2, we discuss related works. In Section 3, we describe the problem formulation of convergecast scheduling. We propose the adaptive cross layer scheduling, the lower bound of convergecast scheduling, and source aware scheduling algorithm in Section 4. Section 5 provides the lower bound with constrained channel and the policies for minimum latency with a limited number of channels. In Section 6, we discuss the multipacket scheduling without limited time and with limited time. Some simulation results are shown in Section 7. Lastly, Section 8 draws our conclusion and discusses future work.

2. Related Works

Unlike traditional sensor networks, IWSNs adopt the centralized scheduling algorithm. The motivation for a centralized scheduling is as follows: (1) it could achieve rigorous analysis for optimal schedule solution based on global information, (2) this schedule pushes the complexity of ensuring good performance for link scheduling to the central manager, (3) the central manager determines convergecast schedules using cross layer information because each layer has an impact on the link scheduling, and (4) centralized scheduling is feasible to adopt different algorithms.

TDMA-based protocols could achieve energy-efficient and collision-free packet delivery with deterministic delay bound. Timeslots and channels could be scheduled without interference. These schedules attempt to minimize the number of timeslots required for convergecast communication by few channels. Most of existing algorithms aim to minimize the transmission delay and eliminate interference by time sharing reuse. The scheduling problem required to complete convergecast has been studied in [11–13] for aggregated data and in [14–16] for raw data.

In many applications, the data aggregation is not used, and all the raw data should be sent to the gateway [17]. The raw-data convergecast is useful because the correlation between the data is low and is not predictable. In addition, the devices in the network have no need for complex function for data aggregation. At present, scheduling problems have been widely studied and many formulations have proven that the optimal scheduling is NP-complete [18, 19].

Gandham et al. proved that the latency bound of convergecast scheduling for tree networks is max \(3n_s - 1, N - 1\) timeslots [15]. Reference [15] also proposed the distributed convergecast scheduling algorithm which achieves the lower bound of the time-optimal schedule length. And this algorithm requires sensor nodes to buffer no more than two packets. However, later studies could support much shorter latency and hence yield much faster convergecast.

Zhang et al. designed scheduling algorithms to establish the lower bound on convergecast by max \(2n_s - 1, N - 1\) timeslots and the lower bound on the number of channels for line [20], binary tree [21], and multiline [22] networks. They proved that these scheduling algorithms of special architectures always achieve the lower bound on convergecast latency with the least number of channels. However, these architectures could not effectively represent general matters. Reference [16] presented a latency-optimal scheduling algorithm of tree networks, but its policies are complex and difficult to extend to guarantee a specific delivery time of a packet.

Most existing TDMA scheduling algorithms are based on single-channel communication. Recently, many researches have started to emerge which focus on how to efficiently use multiple channels to relieve radio collisions and limited
bandwidth [23–26]. Reference [23] showed that the problem is NP-complete which is “the problem of assigning a minimum number of frequencies to receivers such that all the interference links in an arbitrary network are removed.” Multichannel communication supports multiple packet transmissions at the same time, avoids interference and increases throughput. Multichannel communication is also an essential prerequisite for the lower bound on schedule. References [24, 25] proposed node-based channel assignment schemes which coordinate channel switching and transmission. Reference [26] proposed a tree-based multichannel protocol (TMCP) for channel assignment. TMCP partitions the network into multiple subtrees and assigns different channels to each subtree.

3. Problem Formulation

3.1. System Model. In this paper, we construct the convergecast schedule of a multihop network comprising a central manager and field devices. For a multihop path, the earlier hop should be scheduled earlier. These devices have full-function devices (FFD) defined in IEEE 802.15.4 [27] which can serve as routers or sensors. The maximum number of available channels is 16. Time is divided into timeslots with the same size. All the devices are equipped with a single omnidirectional transceiver so that each device can only be scheduled to transmit/receive once a timeslot. Channel hopping proposed by the WirelessHART standard means that the parallel transmissions scheduled at the same timeslot cannot use the same channel. We assume that the network connectivity is fixed over time and every device has an end-to-end convergecast path from the node to the central manager as a source path. We consider application wherein the manager has to receive every data packet sent by field devices in time.

We model the topology of a multihop network as a graph $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_N\}$ is the set of nodes and $E = \{e_{i,j}, \ldots, e_{m,n}\}$ represents the set of links, as shown in Figure 2(a). The $e_{i,j}$ means that $v_i$ can communicate reliably with node $v_j$. Specially, $v_0$ represents the gateway and $V - v_0$ represent the set of sensor nodes. We define an end-to-end path $P_{v_0}$ from a source node $v_{i_0}$ to the sink $v_0$ as a source path, $P_{v_0} = \{e_{i_0, i_1}, \ldots, e_{i_n, 0}\} \subseteq E$, where the packet originated by $v_{i_0}$ will be transmitted by $v_{i_0}, v_{i_1}, \ldots, v_{i_n}, v_0$ and the transmission is $T_{i_n}$. We use $R(V, E')$ with $E' \subseteq E$ to represent a routing tree. The routing tree is rooted at the gateway constructed based on the graph $G$ and all packets can be routed to the gateway.
along the source path in routing tree, as shown in Figure 2(b). The notation $R_i$ refers to the subtree rooted at $v_i$ and $v_j$, which is a child of gateway, shown in Figure 2(c). We use $n_j$ to represent the number of devices in $R_j$. The convergecast can be decomposed into multiple transmissions in the network.

In our paper, we consider devices in IWSNs; the gateway is static and the network connectivity is fixed over time. The maximum length of a packet is fixed and every packet can be transmitted in a timeslot. Each device generates one data packet at the beginning of a convergecast. We assume the drift in the clock of a node is bounded all the time so that synchronization can be achieved. These devices support communication in multiple nonoverlapping channels but only can be scheduled a channel in a decided timeslot. The central manager in these applications has to receive all data packets in time.

3.2. Interference Model. A WirelessHART network should ensure that wireless links whose transmissions may interfere cannot be allocated the same timeslots. The interferences in wireless communication are typically referred to as primary and secondary interference [28]. Two links interfere with each other by primary interference if they share a common device in a single timeslot; that is, the node receives packet from two different transmitters, forwards a packet to two different receivers, or receives a packet from a transmitter and forwards a packet to a receiver at the same time. Two links with no common device interfere with each other by secondary interference if a receiver of a link could receive the packet of another link which does not intend to transmit a packet to the receiver. In order to avoid the occurrence of interferences, WirelessHART allows only a single transmission on any given channel in any dedicated timeslot. In compliance with WirelessHART, all links are scheduled in dedicated timeslots with different channels to deal with interference.

3.3. Problem Formulation. Based on the above models, we focus on providing a time-optimal convergecast scheduling. The schedule finds a source path to the gateway from every device in the network and assigns a timeslot and a channel to each link in the path. The objective of convergecast scheduling is to minimize the total number of timeslots and forward all packets to gateway successfully. We use $t_s$ to denote the number of timeslots for completing convergecast, $c_t$ to denote the number of channels, and $C$ to denote the set of channels. The schedule is computed by the central manager and disseminated to all field devices; then each device executes its schedule repeatedly. The schedule contains a table with entries \((\text{timeslot } t, \text{channel } c, \text{state})\), where $t$ is the transmission timeslot, $c$ is the transmission channel, and state is the ratio state as follows:

(i) $T$: the device may transmit a packet to a neighbor node.

(ii) $R$: the device may receive a packet from a neighbor node.

(iii) $I$: the device neither transmits nor receives.

Let $x(i, j) \in \{0, 1\}$ represent the transmission state of the directed link $e_{ij}$ in timeslot $t$; $x(i, j) = 1$ if node $v_i$ transmits a packet to $v_j$ in timeslot $t$ and $x(i, j) = 0$ otherwise. Let $p_i(v_j)$ represent the number of packets in the node $v_i$ at the end of timeslot $t$, while $p_0(v_j)$ represent the number of packets generated by node $v_i$, $p_i(v_j) \in \{0, 1, \ldots, N\}$. And $p_i$ represents the packet originated by node $v_i$. The convergecast problem is formulated as follows:

\[
\begin{align*}
\min \ t_s & \ \\
\text{s.t.} & \sum_{v_j \in V} \sum_{v_e \in V} x(i, j) \leq 1, \quad c \in C, \quad \forall t \in \{1, \ldots, t_s\}, \\
& \sum_{v_j \in V} \sum_{v_e \in V} x(i, j) \leq c_t, \quad \forall \mathcal{I} \in \{1, \ldots, t_s\}, \\
& \sum_{c \in C} \sum_{v_j \in V} x(i, j) \leq 1, \quad \forall \mathcal{I} \in \{1, \ldots, t_s\}, \\
& \mathcal{P}_i(v_j) = \sum_{v_e \in \mathcal{V} \setminus v_i} p_0(v_j), \\
& p_0(v_j) = 0, \quad \mathcal{P}_0(v_j) = 1 \quad \forall v_j \in \mathcal{V} \setminus v_0, \\
& \mathcal{P}_i(v_j) = \mathcal{P}_{i-1}(v_j) + \sum_{c \in C} x(i, j, c) - \sum_{c \in C} x(i, c) .
\end{align*}
\]

The objective must be required for schedule to minimize the number of timeslots $t_s$. Constraint 1 states that only a single transmission can communicate on any given channel in any timeslot. Constraint 2 states that the number of transmissions cannot exceed the maximum number of available channels $c_t$ at the same time. Constraints 3 and 4 restrict a node to communicate with at most one neighbor in the same timeslot. Constraint 5 means that all packets can reach the gateway by the end of convergecast. The devices except for gateway have a packet to converge at the beginning of schedule as constraint 6 and constraint 7 state the conservation of data packets.

4. Source Aware Scheduling Algorithm

In this section, we focus on how to schedule packets from all devices to the gateway as soon as possible and design optimal scheduling policy with minimum latency bound in combination with the characteristics of convergecast.

4.1. Adaptive Cross Layer Scheduling. The convergecast scheduling should take into account cross layer information because each layer could impact scheduling performance, as shown in Figure 3. The physical layer provides available wireless channels for multichannel scheduling. The schedule includes the ordered timeslots and wireless channels from the 16 available channels of 2.4 GHz defined by IEEE 802.15.4. Our scheduling algorithm should deal with the primary interference, because one device cannot perform
more than one action in any timeslot due to the inherent property of half-duplex transmission. WirelessHART allows a single transmission-receiver pair on a dedicated channel in any given timeslot to avoid the occurrence of secondary interference.

The network layer provides an end-to-end path in which earlier link should be scheduled earlier. The number of timeslots required for a packet depends on the source path from the source device to the gateway. The order of scheduling should follow the routing order when the source device $v_i$ transmits a packet to the gateway $v_g$. If the scheduling is disordered, the convergecast cannot be completed in a superframe. In some industrial applications, the data will be discarded if they cannot arrive the gateway at the end of superframe because new data should be transmitted in next superframe.

### 4.2. Lower Bound on Convergecast Scheduling

Consider the following theorem.

**Theorem 1.** For a routing tree network, the minimum number of timeslots required for convergecast schedule is $\max\{2n_s - 1, N - 1\}$, where $N$ is the number of nodes in the networks and $n_s$ is the maximum number of nodes in any subtree [12].

**Proof.** For the routing tree network, the subtree $R_{v_i}$ rooted at $v_i$ has the maximum number of devices $n_s$. The device $v_i$ needs $n_s$ timeslots to forward all packets of the subtree to the gateway and $n_s - 1$ timeslots to receive these packets generated by other devices in the subtree. Therefore, $2n_s - 1$ timeslots are required to complete the convergecast of the largest subtree. The gateway needs $N - 1$ timeslots to receive packets from all other devices for convergecast schedule. □

### 4.3. Source Aware Scheduling Algorithm

The central manager could maintain the hop-count $h_i$ from every node to the gateway. Each node in the network has three ratio states, and one of the states should be assigned in any timeslot, as shown in Figure 4. To avoid primary interferences in a subtree, when the root node of a subtree with hop-count 1 is assigned a state $T$, one of the nodes in the subtree with hop-count $h$ could be assigned state $T$ if $h \mod 2 = 1$, then the corresponding father nodes should be assigned state $R$.

The convergecast scheduling could be regarded as the transmissions from each device to the gateway, as illustrated in Figure 2(d). Each transmission belongs to a subtree with a root node $v_i$, and all root nodes should be scheduled first. The length of transmission decreases along with its assigned timeslots and its packet is always located at the last hop of the transmission. The initial transmission of a packet is $T_{v_i} = \{e_{V_i}^0, e_{V_i}^1, \ldots, e_{V_i}^{m}\}$, and the packet is located at node $v_{i_0}$. After $m$ timeslots are assigned for this transmission, $T_{i_0}$ is shortened to $\{e_{V_i_{0,m}}^0, \ldots, e_{V_i}^0\}$ and the packet is located at the node $v_{i_m}$. The level of the located node $d_{i_m}$ represents the location of the packet in the routing tree and $d_{i_0}$ is the level of the node which generates this packet. When a packet reaches the gateway, the convergecast of this packet is completed. The set of transmissions is initialized as the paths of all nodes in the network and completed as null. We present the source aware scheduling algorithm which attains the lower bound of convergecast latency. The following gives the policy for which packet will be scheduled in the set of transmissions.

1. The node $v_i$ located in level 1 is a child of gateway and its subtree is sorted by the number of packets left. The node $v_i$ can be scheduled at timeslot $t$ only if its subtree has the highest priority and it is not scheduled at timeslot $t - 1$. If there are multiple subtrees with the same number of packets left, the subtree with higher depth is given the highest priority.

2. At timeslot $t$, if the root node of this subtree has been scheduled, a node $v_i$ of this subtree will be assigned state $T$ only if (1) the buffer of this node is not empty, (2) the depth $d$ mod 2 is 1, and (3) $v_i$ has the most packets left of nodes in the same level. Then the corresponding node will be assigned state $R$.

3. At timeslot $t$, if the root node of this subtree has not been scheduled, a node $v_i$ will be assigned state $T$ only if (1) the buffer of this node is not empty, (2) the level $d$ is not assigned state $T$ at timeslot $t - 1$, when $t = 1$ the level $d$ mod 2 is 0, and (3) $v_i$ has the most packets left of nodes in the same level. Then the corresponding father node will be assigned state $R$.

Algorithm 1 assigns timeslots and channels for the schedule chosen by scheduling policies. For each timeslot, the channel 0 should be assigned to the node at the level 1, and the rest of channels could be assigned to other eligible nodes. The time complexity of Algorithm 1 is $O(N^2)$. Table 1 gives
an example of the source aware scheduling of the routing tree network in Figure 2.

**Theorem 2.** The schedule generated by Algorithm 1 can complete convergecast in a routing tree network in \(\max\{2n_s - 1, N - 1\}\) timeslots.

*Proof.* If \(2n_s > N\), for the maximum subtree, there are \(n_s\) timeslots that the root node \(v_s\) forwards packets to the gateway. At the same time, one of the nodes in the level \(d = 2k - 1\) \((1 < k < d_{\text{max}}/2 + 1)\) is also assigned the state \(T\) and the corresponding node will be assigned state \(R\). And in other \(n_s - 1\) timeslots, one of the nodes in the level \(d = 2k\) \((0 < k < d_{\text{max}}/2)\) is also assigned the state \(T\). After \(t = 2k\) \((1 \leq k \leq n_s - 1)\) timeslots, there are \(n_s - k\) packets left in the subtree and \(N - n_s - k\) packets in the other subtrees. Since \(2n_s > N\), the subtree always has most packets left and its root node can be scheduled to forward a packet to the gateway. At the beginning of a superframe, every node originates a packet, and each subtree has one or more packets in every level. After two timeslots scheduled, there are one or more packets located in level 1 and the root node always has a packet to forward at the timeslot \(t = 2k - 1\) \((1 \leq k \leq n_s - 1)\). Therefore, after \(2n_s - 1\) timeslots, there are \(n_s\) packets forwarded to the gateway and the schedule of this subtree has been completed.

The gateway can receive packets from other subtrees in the remaining \(n_s - 1\) slots and these root nodes also can keep one or more packets to forward in these timeslots. Above all, the source aware schedule can complete convergecast in \(2n_s - 1\) timeslots. If \(2n_s < N\), every root node of subtree will get \(n_s\) timeslots to forward a packet to the gateway and have at least one packet to forward in every two timeslots. Other subtrees in the network also have at least one packet to forward in these two timeslots. Therefore, every subtree can complete its convergecast in at least \(2n_s - 1\) timeslots. The source aware schedule can complete convergecast in \(N - 1\) timeslots. 

\[\blacksquare\]

**5. Source Aware Scheduling Algorithm with Constrained Channel**

The precious communication resources include not only timeslots but also channels. The source aware scheduling algorithm mentioned above gives no consideration to available channels. As shown in Table 1, channel 4 is wasteful to be only scheduled one transmission. The transmission \(p_7 : v_7 \rightarrow v_9\) can be scheduled in timeslots 5–10, because the next hop of \(p_7\) is scheduled in the timeslot 11. And the number of available channels is limited in IWSNs. Under the premise

```
(1) Input: \(R(V,E)\)
(2) Output: \((\text{timeslot } t, \text{channel } c, \text{state})\)
(3) for \(t \leftarrow 1\) to \(\max\{2n_s - 1, N - 1\}\) do
(4) \(c \leftarrow 0\)
(5) Schedule \(e_{i,j}\) by Policy 1
(6) \(\langle t, c, R \rangle \leftarrow v_i\)
(7) \(\langle t, c, T \rangle \leftarrow v_j\)
(8) Choose available schedule set \(S\) by Policy 2 and 3, \(e_{i,j} \in S\)
(9) while \(S \neq \emptyset\) do
(10) \(c \leftarrow c + 1\)
(11) \(\langle t, c, R \rangle \leftarrow v_i\)
(12) \(\langle t, c, T \rangle \leftarrow v_j\)
(13) endwhile
(14) endfor
```

**Algorithm 1:** Source aware scheduling algorithm.

| \(t \setminus c\) | 1 | 2 | 3 | 4 |
|-----------------|---|---|---|---|
| 1               | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) |
| 2               | \(p_2: v_2 \rightarrow v_0\) | \(p_2: v_2 \rightarrow v_0\) | \(p_2: v_2 \rightarrow v_0\) | \(p_2: v_2 \rightarrow v_0\) |
| 3               | \(p_3: v_1 \rightarrow v_0\) | \(p_3: v_1 \rightarrow v_0\) | \(p_3: v_1 \rightarrow v_0\) | \(p_3: v_1 \rightarrow v_0\) |
| 4               | \(p_4: v_2 \rightarrow v_0\) | \(p_4: v_2 \rightarrow v_0\) | \(p_4: v_2 \rightarrow v_0\) | \(p_4: v_2 \rightarrow v_0\) |
| 5               | \(p_5: v_1 \rightarrow v_0\) | \(p_5: v_1 \rightarrow v_0\) | \(p_5: v_1 \rightarrow v_0\) | \(p_5: v_1 \rightarrow v_0\) |
| 6               | \(p_6: v_2 \rightarrow v_0\) | \(p_6: v_2 \rightarrow v_0\) | \(p_6: v_2 \rightarrow v_0\) | \(p_6: v_2 \rightarrow v_0\) |
| 7               | \(p_7: v_1 \rightarrow v_0\) | \(p_7: v_1 \rightarrow v_0\) | \(p_7: v_1 \rightarrow v_0\) | \(p_7: v_1 \rightarrow v_0\) |
| 8               | \(p_8: v_2 \rightarrow v_0\) | \(p_8: v_2 \rightarrow v_0\) | \(p_8: v_2 \rightarrow v_0\) | \(p_8: v_2 \rightarrow v_0\) |
| 9               | \(p_9: v_1 \rightarrow v_0\) | \(p_9: v_1 \rightarrow v_0\) | \(p_9: v_1 \rightarrow v_0\) | \(p_9: v_1 \rightarrow v_0\) |
| 10              | \(p_{10}: v_2 \rightarrow v_0\) | \(p_{10}: v_2 \rightarrow v_0\) | \(p_{10}: v_2 \rightarrow v_0\) | \(p_{10}: v_2 \rightarrow v_0\) |
| 11              | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) | \(p_1: v_1 \rightarrow v_0\) |

**Table 1:** Source aware schedule.
of limited channels, how to achieve the latency-optimal is an important issue in convergecast scheduling.

5.1. Lower Bound with Constrained Channel. The convergecast scheduling algorithm needs to complete the transmissions from each node to the gateway. All these schedules should be completed in the constrained channels for minimum latency or in the given timeslots by minimum channels. The number of schedules \( N_s \) is shown as follows:

\[
N_s = \sum_{i=1}^{N} d_i^0,
\]

where \( d_i^0 \) is the level of the packet generating and the number of schedules for the packet converging and \( N_s(t) \) is the number of schedules in timeslot \( t \). These schedules should be completed in \( t_s \) timeslots and \( c_i \) channels. Thus

\[
N_s(t) = \begin{cases} 
    c_i, 1 \leq t \leq t_s - c_i; \\
    t_s - t + 1, t_s - c_i + 1 \leq t \leq t_s.
\end{cases}
\]

**Theorem 3.** For \( c_i \) constrained channels, the minimum number of timeslots for convergecast is \( \max ([N_s/c_i] + (c_i/2) - (1/2), 2n_{m} - 1, N]) \) [16].

**Proof.** The schedules complete convergecast in \( \max [2n_{m} - 1, N - 1] \) timeslots without constrained channel. All schedules should be completed as follows:

\[
N_s = \sum_{i=1}^{t_s} N_s(t)
\]

\[
\leq c_i (t_s - c_i) + \sum_{t=1}^{t_s} (t_s - t + 1) c_i
\]

\[
\leq c_i \left( t_s + \frac{1}{2} \right) - \frac{c_i^2}{2}.
\]

Thus,

\[
t_s \leq \frac{N_s}{c_i} + \frac{c_i}{2} - \frac{1}{2}.
\]

5.2. Source Aware Scheduling Algorithm with Constrained Channel. The root nodes of subtree should be scheduled first which means that one packet has been converged. One of the root nodes will be scheduled so long as there are one or more packets in their buffers. Policies 2 and 3 of source aware scheduling algorithm choose the set of eligible nodes which can be scheduled in this timeslot. When the number of available channels is not enough to schedule these transmissions, we give the following policies to select nodes scheduled from the set of eligible nodes.

(1) All the nodes have their remaining packets, and the remaining packets of every node in the transmission set can be recorded. The more the remaining packets of a device, the higher the priority, since these nodes have more packets to transmit.

(2) If some nodes have the same amount of remaining packets, we will sort the number of packets left in the subtree. We give higher priority to nodes located in the subtree with more packets left.

(3) If some nodes are the same in Policies 1 and 2, the transmission with higher \( d_i \) will be prioritized for the transmission needs more schedules.

If Policies 1, 2, and 3 select more than one node, one of the nodes is optionally scheduled. Repeat these policies with the remaining channels until all channels are used or no node is eligible. If there are some channels left, the rest schedules are included into a rest set. In order to fully use channels the rest schedule will be started if there are still unused channels and no node is eligible. The transmission with higher \( d_i \) of the rest set is scheduled first so long as the neighbors of the transmission are not scheduled in this timeslot (i.e., child nodes of transmission are not scheduled and parent of transmission is also not scheduled for the constraint on half-duplex transmission). This process is repeated until all channels are used or the rest set is empty. Table 2 gives an example of source aware schedule of the routing tree network with two channels in Figure 2.

6. Source Aware Scheduling Algorithm with Multiple Packets

In some industrial applications, some nodes may have more than one packet to transmit in a superframe. In some cases, the data generated by some nodes is larger than the maximum packet size which IWSNs could transmit. Each device has a designed sample rate to converge its process data and the sample rate may be different. Therefore, multipacket transmissions also exist in a schedule. The central manager knows exactly that the number of packets should be scheduled. The multipacket transmissions are classified into transmission without limited time and transmission with limited time. We modify our convergecast scheduling algorithm to handle these multipacket transmissions.

6.1. Multipacket Scheduling without Limited Time. Consider a routing tree network shown in Figure 5; node \( v_7 \) in the network generates three packets in a superframe and other nodes
6.2. Multipacket Scheduling with Limited Time. In some scenarios where nodes with designed sample rate should transmit their packet to the central manager before their next packet is generated, the multipacket scheduling should guarantee that these multiple packets are scheduled in time. As shown in Figure 5, multiple packets of node \( v_7 \) are generated at a certain time, where the first is generated in 0 timeslot, the second in \( \text{int}(t_s/3) \) timeslot, and the last in \( \text{int}(2t_s/3) \) timeslot. Therefore, the set of scheduled transmissions should add the transmission \( T_7 \) at the timeslot \( \text{int}(t_7/3) \) and \( \text{int}(2t_7/3) \). The transmissions of multipacket will be given higher transmission priorities. Whether or not with constrained channel, as long as there are more than one channel which is used to forward packets located on the subtree root node, the transmissions of multipacket could be scheduled. The number of required timeslots is \( d_{0\text{m}} + 1 \) at most.
7. Performance Evaluation

In this section, we evaluate the performance of our convergecast scheduling algorithm through numerical experiments in MATLAB. The performance of source aware scheduling algorithm has been proven, and it is validated in our experiment. Then we evaluate the source aware scheduling algorithm designed for constrained channel.

7.1. Experimental Setup. The performance of the scheduling algorithm for convergecast is affected by different network topologies with different parameters in both depth and size. The network topology is determined by two parameters: \( N \) (the number of nodes in the network except for gateway) and \( D \) (the maximum depth of routing tree network). In our experiments, \( N \) is varied from 35 to 70, and \( D \) is varied from 2 to 8. In the real WirelessHART network, most of routing trees are with the depth 4 [6]. For different parameters, we average each data over 1,000 runs. We test the performance of scheduling algorithms by different networks.

7.2. Simulation Results. The first simulation tests the schedules generated by source aware scheduling algorithm which has been proven optimal. Figure 6(a) shows the number of scheduled timeslots and \( N \) is varied from 35 to 70 with different depth by 1,000 runs. We can see that the bounded scheduling delay and our algorithm can attain the minimum latency. Figure 6(b) shows the number of channels with varied \( N \). The number of channels increases along with the increased \( N \). The explanation of the increase is that more nodes increase the depth and the number of subtrees and then more nodes are eligible to be scheduled. Figure 6(c) shows

![Figure 6: Performance of source aware scheduling algorithm with constrained channel.](image-url)
We design an adaptive cross-layer scheduling and source aware scheduling algorithm to sustain the derived lower bound with 2 channels. More timeslots in the network. Thus timeslots without constrained channels while the schedule with 3 channels is close to the lower bound of 3 channels. When $N=4$, the number of timeslots decreases along with the increase of $D$ due to the fact that more nodes are attached to a subtree and the number of subtrees decreases. The increase of channels between $D=6$ and $D=8$ is the result of the increase of depth because eligible nodes of a subtree become more.

We evaluate the performance of source aware scheduling algorithm with constrained channel by the second simulation. Figure 7(a) shows the number of timeslots with 2 channels and 3 channels. When $N$ is varied from 35 to 70, the schedule with 3 channels is close to the lower bound of timeslots without constrained channels while the schedule with 2 channels needs more timeslots in the network. Thus source aware scheduling algorithm with constrained channel could get a balance between channel resource and transmission delay. Figure 7(b) compares the number of timeslots increasing along with the depth of routing tree where $N=50$. Obviously, the number of timeslots increases along with the growth of depth. The scheduled timeslots with 2 channels increase faster than 3 channels. In order to find the number of channels to achieve minimum latency, Figure 7(c) evaluates the delay performance with different channels. The number of scheduled timeslots decreases along with the increase of available channels. When the number of channels is more than 4, the delay performance does not change obviously.

We have computed the lower bound on latency with constrained channel in Section 5. Then we compare the schedules generated by our algorithm with the lower bound. We first compute the lower bound of timeslots $T_b$ by theorem and then use it as a baseline to evaluate the schedules generated by our algorithm. Figure 8(a) plots the average difference between our algorithm and the lower bound where the number of available channels is varied from 1 to 7 and $D$ is varied from 2 to 6, which is computed by $(\sum_{i=1}^{N}(T_i - T_b))/N$. Figure 8(b) shows the average ratio of difference between our algorithm and the lower bound, which is computed by $(\sum_{i=1}^{N}(T_i - T_b)/T_b))/N$. We can see that the result of our algorithm is close to the lower bound of analysis and most schedules can achieve the optimal result.

### 8. Conclusion

Convergecast is a critical communication for industrial monitoring and control. This paper studies how to achieve time-optimal scheduling using multiple channels. First, we design an adaptive cross-layer scheduling and source aware scheduling algorithm to sustain the derived lower bound.
on convergecast scheduling. Then a scheduling algorithm is proposed for latency-optimal convergecast with a fixed number of channels. Simulations show that the performance of our algorithm is close to the derived lower bound. And we modify source aware scheduling algorithm to handle convergecast scheduling with multiple packets. Different packets of a device can be delivered before different deadlines.

In our future work, we will discuss the latency convergecast scheduling of aggregated data. We should derive a lower bound on latency and propose an efficient scheduling scheme for aggregated data convergecast. And in some cases, multipath routing is adopted to maximize the probability that the packet is delivered. How to design an optimal scheduling policy is also an important issue for multipath routing. The interference model becomes more complex for the redundant path. At last our algorithms will be applied in real large-scale manufacturing factory and will face more challenges and problems in industrial environments. We should further improve our algorithms to achieve better performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant no. 61201204, no. 61271201, and no. 61100217), the Beijing Higher Education Young Elite Teacher Foundation of China (Grant no. 61201204, no. 61271201, and no. 61201204), the Beijing ChaoChengZhi Training Program (Grant no. YETP0533 and no. YETP0534), and the SRF for ROCS, SEM.

References

[1] A. Willig, "Recent and emerging topics in wireless industrial communications: a selection," IEEE Transactions on Industrial Informatics, vol. 4, no. 2, pp. 102–122, 2008.
[2] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pp. 4258–4265, 2009.
[3] J. Song, S. Han, A. K. Mok et al., "WirelessHART: applying wireless technology in real-time industrial process control," in Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS ’08), pp. 377–386, IEEE, April 2008.
[4] A. Willig, K. Matheus, and A. Wolisz, "Wireless technology in industrial networks," Proceedings of the IEEE, vol. 93, no. 6, pp. 1130–1151, 2005.
[5] "Isa-100.11a-2009 wireless systems for industrial automation: process control and related applications," http://www.isa.org/ISA100-11a.
[6] "Wirelesshart," http://en.hartcomm.org/.
[7] "Zigbee alliance," http://www.sigbee.org/.
[8] J. Zhao, Y. Qin, D. Yang, and J. Duan, "Reliable graph routing in industrial wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 2013, Article ID 758217, 15 pages, 2013.
[9] J. Zhao, D. Yang, Y. Qin, T. Zheng, J. Duan, and M. Gidlund, "Issues of routing protocol for wireless industrial sensor networks," in Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON ’12), pp. 3225–3230, IEEE, October 2012.
[10] R. Nelson and L. Kleinrock, "Spatial tdma: a collision-free multihop channel access protocol," IEEE Transactions on Communications, vol. 33, no. 9, pp. 934–944, 1985.
[11] A. Ghosh, O. D. Incel, V. S. A. Kumar, and B. Krishnamachari, "Multichannel scheduling and spanning trees: throughput-delay tradeoff for fast data collection in sensor networks," IEEE/ACM Transactions on Networking, vol. 19, no. 6, pp. 1731–1744, 2011.
[12] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 11, no. 1, pp. 86–99, 2012.
[13] Ö. D. Incey and B. Krishnamachari, "Enhancing the data collection rate of tree-based aggregation in wireless sensor networks," in Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’08), pp. 569–577, IEEE, June 2008.
[14] W.-Z. Song, F. Yuan, R. Lahusen, and B. Shirazi, "Time-optimum packet scheduling for many-to-one routing in wireless sensor networks," International Journal of Parallel, Emergent and Distributed Systems, vol. 22, no. 5, pp. 355–370, 2007.
[15] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," Computer Networks, vol. 52, no. 3, pp. 610–629, 2008.
[16] H. Zhang, P. Soldati, and M. Johansson, "Performance bounds and latency-optimal scheduling for convergecast in wireless HART networks," IEEE Transactions on Wireless Communications, vol. 12, no. 6, pp. 2688–2696, 2013.
[17] Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks," in Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC ’03), pp. 177–188, ACM, June 2003.
[18] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-time scheduling for WirelessHART networks," in Proceedings of the 31st IEEE Real-Time Systems Symposium (RTSS ’10), pp. 150–159, IEEE, December 2010.
[19] J. K. Lenstra and A. H. G. R. Kan, "Complexity of scheduling under precedence constraints," Operations Research, vol. 26, no. 1, pp. 22–35, 1978.
[20] H. Zhang, P. Soldati, and M. Johansson, "Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks," in Proceedings of the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt ’09), pp. 1–8, IEEE, June 2009.
[21] P. Soldati, H. Zhang, and M. Johansson, "Deadline-constrained transmission scheduling and data evacuation in WirelessHART networks," in Proceedings of the European Control Conference (ECC ’09), 2009.
[22] H. Zhang, P. Soldati, and M. Johansson, "Time- and channel-efficient link scheduling for convergecast in WirelessHART networks," in Proceedings of the 13th IEEE International Conference on Communication Technology (ICICT ’11), pp. 99–103, IEEE, September 2011.
[23] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. K. Chintalapudi, "Multi-channel scheduling for fast convergecast in
wireless sensor networks,” Tech. Rep., Department of Computer Science, University of Twente, 2008.

[24] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, “MMSN: multi-frequency media access control for wireless sensor networks,” in Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM ’06), vol. 6, pp. 1–13, April 2006.

[25] J. Zhang, G. Zhou, C. Huang, S. H. Son, and J. A. Stankovic, “TMMAC: an energy efficient multi-channel MAC protocol for ad hoc networks,” in Proceedings of the IEEE International Conference on Communications (ICC ’07), pp. 3554–3561, IEEE, June 2007.

[26] Y. Wu, J. A. Stankovic, T. He, and S. Lin, “Realistic and efficient multi-channel communications in wireless sensor networks,” in Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM ’08), pp. 1193–1201, IEEE, April 2008.

[27] “IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs),” 2006.

[28] S. Ramanathan and E. L. Lloyd, “Scheduling algorithms for multihop radio networks,” IEEE/ACM Transactions on Networking, vol. 1, no. 2, pp. 166–177, 1993.
