Abstract—Convergecast, in which data from a set of sources is routed toward one data sink, is a critical functionality for wireless networks deployed for industrial monitoring and control. We address the joint link scheduling and channel assignment problem for convergecast in networks operating according to the recent WirelessHART standard. For a linear network with $N$ single-buffer devices, we demonstrate that the minimum time to complete convergecast is $2N-1$ time-slots, and that the minimum number of channels required for this operation is $\lceil N/2 \rceil$. When the devices are allowed to buffer multiple packets, we prove that the optimal convergecast time remains the same while the number of required channels can be reduced to $\lceil N - \sqrt{N(N-1)/2} \rceil$. For both cases, we present jointly time- and channel-optimal scheduling policies with complexity $O(N^2)$. Numerical results demonstrate that our schemes are also efficient in terms of memory utilization.

I. INTRODUCTION

There is a strong current interest in migrating substantial parts of the traditionally wired industrial infrastructure to wireless technologies to improve flexibility, scalability, and efficiency [5]. However, concerns about network latency, reliability and security along with the lack of device interoperability have hampered the deployment rate. To address these concerns, WirelessHART is the first open and interoperable wireless communication standard specially designed to address the critical needs of real-world industrial applications. This standard was recently approved and released [1]. The WirelessHART architecture is designed to be an easy-to-use wireless mesh networking protocol, leveraging on advanced techniques such as time diversity, frequency diversity and path diversity to achieve the level of reliability and latency required to support advanced process monitoring and control applications.

Real-time data delivery is a critical issue in WirelessHART networks in order to guarantee the performance of closed-loop controllers. However, the general time-constrained scheduling problem is NP-complete even in linear networks [2]. Time-optimal convergecast, in which all devices in the network send data to a central control device in minimum time, is a special case of the general time-constrained scheduling problem. Much work has been done on designing efficient convergecast schemes for wireless sensor/ad-hoc networks. In [3], Choi et al. proved that the decision version of convergecast problem is NP-complete in a weak sense. In [4], Gandham et al. proposed distributed algorithms to generate near-optimal schedule which can complete convergecast using at most $\max\{3n_k - 1, N\}$ time slots in a tree network, where $n_k$ is the maximum number of nodes in any subtree and $N$ is the total number of nodes. The methodology applied in this paper also demonstrates that understanding the performance for linear topology is instrumental for the analysis of other (more complex) topologies. However, in most existing work, it is assumed that all devices communicate using a single channel, which can not provide reliable and timely communication with high data rate requirement because of radio collisions and limited bandwidth. Recently, proposals for multi-channel convergecast protocols have started to emerge [6]–[9]. However, the existing multi-channel convergecast schemes cannot be used in WirelessHART networks due to the special characteristics of the standard. Firstly, WirelessHART performs channel hopping on a per-transaction (packet-acknowledgement) basis [1], while most existing TMDA convergecast schemes do not support this feature. Secondly, the current WirelessHART physical layer supports 16 non-overlapping channels, making the channel a scarce resource, demanding that convergecast schemes for WirelessHART networks should be efficient in terms of channel utilization. Last but not the least, the field devices are generally memory-constrained, and the convergecast schemes for WirelessHART networks must also be memory-efficient.

In this paper, we present a complete understanding of the joint link scheduling and channel assignment convergecast problem for WirelessHART networks with linear topology in terms of time-efficiency, channel-efficiency and memory-efficiency. Although the general time-constrained scheduling problem is NP-complete, we demonstrate that the time- and channel-optimal convergecast schedule for linear networks can be obtained in polynomial time. We show that understanding the exact mechanisms for the linear topology is non-trivial and provides deep insight into the solution for the WirelessHART networks with multi-line and tree topologies. These natural extensions of this work are currently investigated. The main contributions of this work are:

- For a linear network composed of $N$ devices with single-packet buffering capability, we demonstrate that the minimum time to complete convergecast is $2N-1$ time slots, and the minimum number of channels required for this operation is $\lceil N/2 \rceil$. We present an algorithm with time complexity $O(N^2)$ to generate the time- and channel-optimal convergecast schedule.
- For a linear network in which devices are allowed to buffer multiple packets, we prove that the optimal convergecast time remains the same while the number of re-
quired channels can be reduced to \([N - \sqrt{N(N - 1)/2}].\) We give an algorithm with time complexity \(O(N^2)\) for generating the optimal convergecast schedule.

- We design an algorithm with time complexity \(O(NC_S)\) \((C_S \text{ is the number of channels used for convergecast})\) for transforming the global schedule into sub-schedules with time-slot and channel assignments for each device.
- Through numerical simulations, we demonstrate that the schedule generated without imposing buffer constraints is still memory-efficient. We also give the approach to apply our scheme in wirelessHART networks with general topology, and discuss the possibility of extending the solution for linear networks to other networks with more complex topologies.

The paper is organized as follows. Section II gives a brief introduction to WirelessHART. Problem formulations are given in Section III and solutions are presented in Section IV. Section V details the sub-schedule extraction algorithm, while Section VI presents numerical results. The paper is concluded in Section VII.

## II. WIRELESSHART

WirelessHART is an extension of wired HART, a transaction-oriented communication protocol for process control applications. WirelessHART is a complete wireless mesh networking protocol based on radios compliant with the IEEE 802.15.4-2006 physical layer standard, supporting 16 channels in the 2.4 GHz ISM band and a total data rate of up to 250 kbit/s. To meet the requirements for control applications, WirelessHART uses TDMA technology to arbitrate and coordinate communications between network devices. The TDMA data-link layer establishes links and specifies the time slot and channel to be used for communication between devices. WirelessHART has several mechanisms to promote network-wide time synchronization and maintains time slots of 10 ms length. To enhance reliability, TDMA is combined with channel hopping on a per-transaction (time slot) basis. In a dedicated time slot, only a single device can be scheduled for transmission in each channel (i.e., no spatial re-use is permitted).

The basic elements of a WirelessHART network include:

- **Field Devices** are connected to the process equipments. All field devices are able to source and sink packets and capable of routing packets on behalf of other devices in the network.
- **Gateways** enable communication between host applications and field devices. A gateway may have more than one access point. Device-to-device communication is not supported: all data must pass through the gateway.
- A **Network Manager** is responsible for configuring the network, scheduling communication between devices, managing the routing tables and monitoring and reporting the health of the network.

Besides the above components, a wirelessHART network may also include adapters used for connecting to existing HART-compatible field devices and handheds used to configure, maintain or control plant assets.

It is worth noting that the network manager is responsible for scheduling all communications between wirelessHART network devices. Thus, the convergecast schedule must be first computed centrally by the network manager (which maintains global network information) and then disseminated to all network devices. A more complete description of the WirelessHART standard can be found in [1].

## III. PROBLEM FORMULATION

We model the WirelessHART network as a graph \(G = (V, E)\) where the vertices \(V = \{v_0, v_1,\ldots,v_N\}\) represent the network devices and the edges \(E\) denote the device pairs that can sustain reliable communication. There is only one gateway (GW) which is denoted by \(v_0\) while \(v_1,\ldots,v_N\) are the \(N\) field devices. An edge \((v_i, v_j)\) exists in the graph if and only if node \(v_i\) and node \(v_j\) can communicate reliably with each other. Time is synchronized and slotted, and the length of a time slot allows transmitting exactly one packet and its associated acknowledgement. Channel hopping is performed on a per-time slot basis, and parallel transmissions scheduled at the same time slot must use different channels. We also assume that nodes cannot transmit and receive at the same time. This final assumption is not imposed by the standard, but is included since it is an important common limitation of most current radio platforms.

Based on this model, we focus on solving the link scheduling and channel assignment problems for convergecast in WirelessHART networks. During convergecast, each field device generates one packet and transmits the packet to the GW. The first problem to be addressed is to design efficient algorithms for generating a convergecast schedule which completes the convergecast in a minimal number of time slots. Informally, we pose this problem as one of finding a feasible
convergecast schedule $S$ that solves the problem
\[
\begin{align*}
\text{minimize} & \quad l_S, \\
\text{subject to} & \quad S \text{ satisfies buffer constraints}
\end{align*}
\]
where $l_S$ is the length of $S$ (i.e., the number of time slots in $S$).

Since memory is a scarce resource for embedded devices, we consider both the cases where (a) each node can only buffer a single packet at a time slot, and (b) nodes have unlimited buffering capabilities. Note that the single-buffer constraint implies that a node with a data packet in its buffer must be scheduled for transmission before it can receive a new packet.

A single WirelessHART network can use a maximum of 16 parallel channels. Since WirelessHART employs channel blacklisting to avoid unsuitable channels which have consistently high interference levels (e.g. due to co-existence with 802.11 devices), the number of available channels may be significantly less than 16. Thus channel is a scarce resource which should be carefully managed in the convergecast operation.

The second problem to be addressed in this paper is the design of a jointly time- and channel-optimal convergecast scheme with the objective to minimize both the number of time slots and the number of channels required to complete convergecast. To this end, we let $C_S$ be the number of channels used in schedule $S$ and informally state this problem as
\[
\begin{align*}
\text{minimize} & \quad C_S, \\
\text{subject to} & \quad l_S \text{ is minimized.}
\end{align*}
\]

Also this problem is considered for the cases with different buffering capabilities.

IV. TIME AND CHANNEL OPTIMAL CONVERGECAST SCHEDULING FOR LINEAR NETWORKS

In this section, we focus on solving link scheduling for convergecast in a line network as shown in Fig. 2. Without loss of generality, the GW is placed at the right end of the line, and the $N$ field devices ($v_1, \ldots, v_n$) are placed from right to left.

![fig2](image)

Fig. 2. A line network with one gateway and $N$ field devices

A. Lower bound on $l_S$

Lemma 1: Given a schedule $S$ for convergecast in a line network with $N$ field devices and one GW, the lower bound on $l_S$ is $2N - 1$.

Proof: As shown in Fig. 2, link $(v_2, v_1)$ and link $(v_1, v_0)$ cannot be scheduled simultaneously since each node has only one half-duplex radio transceiver. To complete convergecast, node $v_1$ needs $N - 1$ time slots to receive packets generated by the other $N - 1$ nodes, and needs another $N$ time slots to transmit the received $N - 1$ packets as well as the packet generated by itself to the GW. Therefore, the lower bound on $l_S$ is $N - 1 + N = 2N - 1$.

B. Scenario I: Single-packet buffering capability

If each node can buffer at most one packet, a node is eligible to receive a packet only when its buffer is empty. The following lemma gives the sufficient and necessary condition for time-optimal convergecast in a line network with single-packet buffering constraint.

Lemma 2: With single-buffer constraint, convergecast in a line network can be completed in $2N - 1$ time slots if and only if there is one packet scheduled to be transmitted from node $v_1$ to the GW at any time slot $t = 2k - 1$, where $1 \leq k \leq N$.

Proof:
(1): $\rightarrow$
To complete convergecast, node $v_1$ needs to be scheduled for transmission in $N$ time slots. Since node $v_1$ has only one buffer, it cannot work in transmit state in two consecutive time slots. Therefore, the only way to complete convergecast in $2N - 1$ time slots is that node $v_1$ transmits one data packet to the GW at any time slot $t = 2k - 1$, where $1 \leq k \leq N$.

(2): $\leftarrow$
If node $v_1$ transmits one data packet to the GW at any time slot $t = 2k - 1$ where $1 \leq k \leq N$, obviously, the total number of time slots for convergecast is $2N - 1$.

Based on Lemma 2, we design an algorithm with time complexity $O(N^2)$ to generate the time-optimal schedule for convergecast in a line network in which each node can buffer at most one packet at a time slot. The algorithm works as follows:

- For node $v_1$, its schedule is generated based on the criteria given in Lemma 2. At each time slot $t = 2k - 1$ where $1 \leq k \leq N$, node $v_1$ is scheduled for transmission.
- For node $v_i$ ($i > 1$), at any time slot $t$, if node $v_i$ has a packet to transmit and the buffer at node $v_{i-1}$ is empty, node $v_i$ is scheduled for transmission.

The detailed algorithm is given in Algorithm 1. The schedule is stored in a two-dimensional array $Sch[1..2N - 1][1..C_S]$, where $Sch[t][1..C_S]$ ($1 \leq t \leq 2N - 1$) records the nodes scheduled for transmission at time slot $t$. An example of the optimal schedule generated by Algorithm 1 for a line network with 5 nodes is given in Fig. 3.

![Algorithm 1](image)

Algorithm 1: Convergecast_Optimal_Scheduling

Input: $G = (V, E)$
Output: $Sch[1..2N - 1][1..C_S]$

1 begin
2 for $i \leftarrow 1$ to $N$ do
3 for $j \leftarrow i$ to $2N - i$ do
4 if $j - i + 1 \mod 2 = 1$ then
5 $Sch[j][i] \leftarrow Sch[j][i] + i$
6 end
7 end

Corollary 1: The schedule generated by Algorithm 1 is the only optimal schedule that can complete convergecast in a line network with $2N - 1$ time slots if each node can buffer at most one packet at a time slot.
The number of channels required to complete convergecast in a line network with multi-buffering capability for each node in terms of minimizing both the number of time slots and the number of channels.

**Theorem 2:** Given any schedule $\mathcal{S}$ which can complete convergecast in $2N - 1$ time slots in a line network with $N$ nodes and without buffering constraint, the lower bound on the number of channels used in $\mathcal{S}$ is $\lceil N - \sqrt{N(N - 1)/2}\rceil$.

**Proof:** Let $C_\mathcal{S}$ be the number of channels used in schedule $\mathcal{S}$, and $PT_\mathcal{S}(t)$ denote the number of parallel transmissions scheduled in time slot $t$ ($1 \leq t \leq 2N - 1$). When $2N - 2(C_\mathcal{S} - 1) \leq t \leq 2N - 1$, at most $\lceil N(N - 1)/2 \rceil$ nodes can be scheduled for transmission in time slot $t$; otherwise, convergecast can not be completed in $2N - 1$ time slots. Thus

$$PT_\mathcal{S}(t) \leq \begin{cases} C_\mathcal{S}, & \text{if } 1 \leq t < 2N - 2(C_\mathcal{S} - 1); \\ \lceil N(N - 1)/2 \rceil, & \text{if } 2N - 2(C_\mathcal{S} - 1) \leq t \leq 2N - 1. \end{cases}$$

Let $ST_{\max}$ be the maximum number of transmissions that can be scheduled in $2N - 1$ time slots using at most $C_\mathcal{S}$ channels. By inequality (3),

$$ST_{\max} = \sum_{t=1}^{2N-1} \max(PT_\mathcal{S}(t)) = \sum_{t=1}^{2N-2(C_\mathcal{S} - 1) - 1} C_\mathcal{S} + \sum_{t=2N-2(C_\mathcal{S} - 1)}^{2N-1} \frac{2N - t}{2}.$$

Since $\sum_{t=2N-2(C_\mathcal{S} - 1)}^{2N-1} \frac{2N - t}{2} = N - (C_\mathcal{S} - 1) + N - (C_\mathcal{S} - 1), ..., +2 + 2 + 1 + 1 = 2 \cdot \frac{(N - (C_\mathcal{S} - 1) + (N - (C_\mathcal{S} - 1))) \cdot (N - (C_\mathcal{S} - 1))}{2}$,

$$ST_{\max} = -C_\mathcal{S}^2 + 2N \cdot C_\mathcal{S}$$

(4)

Let $ST_\mathcal{S}$ be the total number of transmissions scheduled in $\mathcal{S}$. Then $ST_\mathcal{S} = N(N + 1)/2$.

To guarantee that schedule $\mathcal{S}$ can complete convergecast in $2N - 1$ time slots with $C_\mathcal{S}$ channels, it must satisfy that $ST_{\max} \geq ST_\mathcal{S}$, i.e., $-C_\mathcal{S}^2 + 2N \cdot C_\mathcal{S} \geq N(N + 1)/2$. Thus $C_\mathcal{S} \geq \lceil N - \sqrt{N(N - 1)/2}\rceil$.

In Theorem 2, we give the lower bound on the number of channels required to complete convergecast in $2N - 1$ time slots, but do not prove that the lower bound is always achievable. In the following, we design an algorithm that can always generate time- and channel-optimal schedule for convergecast in line networks with unlimited buffering capabilities, thereby demonstrating the tightness of this lower bound.

The basic idea of the algorithm is to schedule as many transmissions as possible at each time slot to make full use of the available channels. At each time slot $t$, the schedule is generated in 2 steps: **forward scheduling** and **backward scheduling**.

In the **forward scheduling** step, the algorithm searches the eligible nodes that can be scheduled for transmission in the
direction from node $v_1$ to node $v_n$. For node $v_1$, if it has a packet in its buffer, node $v_1$ is scheduled for transmission at this time slot. For node $v_i$ ($i > 1$), if node $v_i$ has a packet in its buffer and node $v_{i-1}$ does not have a packet in its buffers, node $v_i$ is scheduled for transmission at this time slot. Therefore, a node which does not has a packet to transmit at the beginning of time slot $t$ receives one packet at time slot $t$.

When the forward scheduling step is finished, the backward scheduling step is started if the number of nodes scheduled for transmission in the forward scheduling step is less than the maximum transmissions that can be scheduled in this time slot. Let $end\_node$ be the node which is farthest from the GW among all nodes that have packets to transmit. In the backward scheduling step, the algorithm searches the eligible nodes for transmission in the direction from $end\_node$ to node $v_1$.

Let $max(PT_S(t))$ be the maximum number of nodes that can be scheduled for transmission at time slot $t$. By inequality (3) and Theorem 2,

$$\max(PT_S(t)) = \left\{ \begin{array}{ll}
C^*, & 1 \leq t < 2N - 2(C^* - 1); \\
\left\lceil \frac{2N-1}{2} \right\rceil, & 2N - 2(C^* - 1) \leq t \leq 2N - 1,
\end{array} \right. \quad (5)$$

where $C^* = \left\lfloor N - \sqrt{N(N - 1)/2} \right\rfloor$. Let $N_p(i)$ be the number of packets in the buffers of node $v_i$. The node $v_i$ that satisfies the following conditions is scheduled:

- The number of nodes that have been scheduled for transmission at time slot $t$ is less than $\max(PT_S(t))$.
- Node $v_i$ is not scheduled in this time slot and $N_p(i) > 0$.
- If $i < N$, node $v_{i+1}$ is not scheduled for transmission in this time slot as a device can not transmit and receive at the same time slot.
- Similarly, if $i > 1$, node $v_{i-1}$ is not scheduled for transmission in this time slot.

Thus the backward scheduling step maximizes the number of nodes scheduled for transmission at each time slot.

The detailed algorithm is given in Algorithm 2. Since the time complexity for both forward scheduling and backward scheduling is $O(N)$, the time complexity of Algorithm 2 is $O(N^2)$. Fig. 4 gives the time- and channel-optimal convergecast schedule in a line network with 5 nodes.

![Optimal convergence schedule for a line network (N = 5) with multi-packet buffering capability at each node](image)

**Fig. 4.** Optimal convergence schedule for a line network $(N = 5)$ with multi-packet buffering capability at each node

**Theorem 3:** For a line network with $N$ nodes and unlimited buffering capabilities, the schedule generated by Algorithm 2 can always complete convergecast in $2N - 1$ time slots using only $\left\lfloor N - \sqrt{N(N - 1)/2} \right\rfloor$ channels.

**Algorithm 2: ConvergeCast_Line_Multiple_Buffer**

```plaintext
Input: $G = (V, E); C^*; N_p(1..N)$
Output: $Sch[1..2N-1][C^*]
1 begin
2 end_node $\leftarrow$ N;
3 for $i \leftarrow 1$ to $2 \times N - 1$ do
4 $PT \leftarrow 0;
5 /* Forward Scheduling */
6 for $j \leftarrow 1$ to $end\_node$ do
7 if $N_p[j] = 1$ then
8 $Sch[i][] \leftarrow Sch[i][] + j;
9 N_p[j] = -1; PT +=;
10 else if $N_p[j] > 0 \&\& N_p[j-1] = 0 \&\& j \notin Sch[i][] then
11 $Sch[i][] \leftarrow Sch[i][] + j;
12 PT +=;
13 N_p[j] = -1; N_p[j-1] +=;
14 if $N_p[j] = 0 \&\& j = end\_node$ then
15 end_node $\leftarrow j - 1;
16 /* Backward Scheduling */
17 $j \leftarrow end\_node;
18 while $PT < \max(PT_S(i)) \&\& j > 0$ do
19 if $N_p[j] > 0 \&\& (j > 1, j, j+1) \notin Sch[i][] then
20 $Sch[i][] \leftarrow Sch[i][] + j;
21 PT +=;
22 $N_p[j] = -1; N_p[j-1] +=;
23 $j \leftarrow j - 1;
24 end
```

**Proof:** In both the forward scheduling step and the backward forwarding step, a node can be scheduled for transmission at time slot $t$ only if the number of nodes that have been scheduled in this time slot is less than $\max(PT_S(t))$. By Equation (5), the number of channels used in the schedule generated by Algorithm 2 does not exceed $\left\lfloor N - \sqrt{N(N - 1)/2} \right\rfloor$. In the following, we prove that the schedule generated by Algorithm 2 can always complete convergecast in $2N - 1$ time slots.

![Two special cases of network status at a time slot](image)

**Fig. 5.** Two special cases of network status at a time slot

As shown in Fig. 5, the line network is divided into two parts. For node $v_i$, if $i \leq 2C^* = 2\left\lfloor N - \sqrt{N(N - 1)/2} \right\rfloor$, node $v_i$ is located in Region A; otherwise, node $v_i$ is located...
in Region B. Case I and Case II give two special cases of the network state at a time slot, and the digits in the two cases represent the number of packets in each corresponding node.

The forward scheduling step in Algorithm 2 schedules nodes from \( v_1 \) to \( v_n \). If \( C^* \) is large enough, the forwarding scheduling step can always guarantee that there is a packet transmitted from node \( v_1 \) to the GW at any time slot \( t = 2k-1 \), where \( 1 \leq k \leq N \). Therefore, if the schedule generated by Algorithm 2 can not complete convergecast in \( 2N - 1 \) time slots, one of the two cases given in Fig. 5 must happen at some time slot \( t \) \( (2C^* < t < 2N - 2C^*) \). In both cases, Region A contains \( C^* \) nodes with exactly one packet at each node, and there must be one node that has a packet in any two adjacent nodes. By Lemma 2, to guarantee that convergecast can be complete in \( 2N - 1 \) time slots, there must be a packet transmitted from node \( v_1 \) to the GW at any time slot \( t = 2k-1 \), where \( 1 \leq k \leq N \). Thus the \( C^* \) nodes which are located in Region A and have one packet at each node must be scheduled for transmission at time slot \( t \), and node \( v_{i+1} \) in Region B also must be scheduled at time slot \( t \). If Case I occurs, at least \( C^* + 1 \) channels are required to complete convergecast in \( 2N - 1 \) time slots. In the following, we prove that Case I never occurs.

Let \( P_B \) be the number of packets in the nodes located in Region B at the beginning of time slot \( t \). Since the number packets that the GW receives before time slot \( t \) is \( \frac{1}{2} (t \) must be odd if Case I occurs), \( P_B = N - \frac{1}{2} C^* \). Obviously, in the best case all the \( P_B \) packets are in the buffers of node \( v_{i+1} \), and the minimum number of transmissions needed to deliver the \( P_B \) packets to the GW is \( P_B (2C^* + 1) \). For the \( C^* \) packets in Region A, each node \( v_j \) where \( j = 1, 3, 5, ..., 2C^* - 1 \) holds a packet, and \( j \) transmissions are needed to transmit the packet at node \( v_j \) to the GW. Therefore, the number of transmissions needed to deliver the \( C^* \) packets to the GW is \( 1 + 3 + 5 + ... + (2C^* - 1) = C^* \). Since there is at least one packet in the nodes located in Region B, the number of transmissions scheduled at any time slot before time slot \( t \) must be \( C^* \). Therefore, the minimum number of transmissions required to complete convergecast is

\[
P_A(2C^* + 1) + C^* (t - 1)
= (N - \frac{1}{2} C^*) (2C^* + 1) + C^* (t - 1);
= -C^* + 2NC^* + (N - \frac{1}{2} C^*).
\]

As there is at least one packet in the nodes located in Region B, \( N - \frac{1}{2} C^* > 0 \). By Equation (4) in the proof of Theorem 2, the maximum number of transmissions (i.e., \( ST_{\text{max}} \)) that can be scheduled in \( 2N - 1 \) time slots using \( C^* \) channels is \( -C^* + 2NC^* \). Therefore, Case I never happens. For Case II, the proof is the same as that for Case I.

Therefore, the schedule generated by Algorithm 2 can achieve both the lower bound on the number of time slots and the lower bound on the number of channels required to complete convergecast on a line network.

V. SUB-SCHEDULE EXTRACTION AND CHANNEL ASSIGNMENT

In wirelessHART networks, the convergecast schedule is firstly computed at the network manager, and then disseminated to all field devices in the networks. In our schemes, the convergecast schedule is stored in a compact structure \( sch[1..2N - 1][C_S] \) as it only records which nodes should be scheduled for transmission at a time slot. Once a field device receives the compact schedule, it should extract the useful information from \( sch[1..2N - 1][C_S] \) and generate a sub-schedule for itself.

WirelessHART supports channel hopping on a per-transmission (time slot) basis. Each transmission is associated with a channel offset which represents the logical channel to be used for this transmission. In wirelessHART, the logical channel can be easily mapped to the actual channel.

For each transmission, the transmitter and the receiver should use the same channel. In our scheme, the channel for each transmission can be easily obtained based on \( sch[1..2N - 1][C_S] \) using the following policy:

- At time slot \( t \) \( (1 \leq t \leq 2N - 1) \), let \( v_k \) be the node recorded in \( Sch[t][ch] \). Both \( v_k \) and \( v_{k-1} \) have channel offset with value of \( ch \).

At every time slot, each node can work in 3 states: Transmit (T), Receive (R) and Sleep (S). The sub-schedule and channel offset for each field device are stored in a 2-dimensional array \( S_{\text{sch}}[1..2N - 1][1..2] \), where \( S_{\text{sch}}[t][1] \) records the state that the field device should work in time slot \( t \) and \( S_{\text{sch}}[t][2] \) records the channel offset for the filed device at time slot \( t \).

The algorithm for generating the sub-schedule and assigning channel offset for each filed device \( v_i \) is given in Algorithm 3.

The time complexity of Algorithm 3 is \( O(NC_S) \) where \( C_S \) is the number channels used for convergecast. Fig. 6 gives the sub-schedule for node \( v_2 \) based on the convergecast schedule given in Fig. 4.

Algorithm 3: Sub-schedule Generation\((v_i)\)

| Input: \( Sch[1..2N - 1][C_S] \) |
| Output: \( S_{\text{sch}}[1..2N - 1][1..2] \) |

begin

  for \( t \leftarrow 1 \) to \( 2N - 1 \) do

    for \( ch \leftarrow 1 \) to \( C_S \) do

      if \( Sch[t][ch] = v_{i+1} \) then

        \( S_{\text{sch}}[t][1] \leftarrow T; \)

        \( S_{\text{sch}}[t][2] \leftarrow ch; \)

      else if \( Sch[t][ch] = v_i \) then

        \( S_{\text{sch}}[t][1] \leftarrow R; \)

        \( S_{\text{sch}}[t][2] \leftarrow ch; \)

      else \( S_{\text{sch}}[t][1] \leftarrow S; \)

end
VI. EVALUATION AND DISCUSSION

In this section, we evaluate the performance of our algorithms through numerical simulations. We first demonstrate the memory efficiency of our solutions, and then discuss the relationship between available number of channels and the size of the network. We also discuss the way to apply our scheme in WirelessHART networks with general topology, and the way to extend our scheme to more complex topologies.

A. Memory efficiency

For the schedule generated by Algorithm 1, each node is required to buffer at most one packet. Therefore, the schedule generated by Algorithm 1 is also optimal in terms of memory utilization, but is not channel optimal. For the schedule generated by Algorithm 2, some nodes in the network need to buffer more than one packet. Fig. 7 shows the maximum number of packets buffered at each node under different settings of the network size $N$. It can be seen that most nodes are required to buffer only a small number of packets. For example, only three nodes ($v_{15}$, $v_{16}$ and $v_{17}$) need to buffer more than 3 packets for a line network with $N = 28$.

![Fig. 7. Maximum packets buffered $v.s. N$](image)

In practice, the buffering capability for each node is limited. At each time slot, an eligible node is not scheduled for transmission if the buffers at the receiver are all full. Fig. 8 plots the number channel required to complete convergecast in $2N - 1$ time slots under different settings of $N$ and buffer size. It is worth noting that, if each node can buffer at most 2 packets at a time slot, only a line network with 26 nodes needs one more channel than the lower bound when $N \leq 32$. This also demonstrates the memory efficiency of our scheme.

![Fig. 8. Minimum number of channels $v.s. N$ and the buffering capability (buf)](image)

B. Number of channels vs. network Size

The WirelessHART architecture is based on IEEE 802.15.4 standard which can support 16 channels. By Theorem 1, the maximum length of a line network is 32 if each node can buffer only one packet at a time slot. From Fig. 8, the size of the network can be much larger if each node can buffer multiple packets. By Theorem 2, the number of nodes in a line network can be as large as 53 if each node has unlimited buffering capability.

C. Application and extension

A wirelessHART network can have more than one gateway, and each gateway can have multiple access points. Give a network with general topology, a multi-line structure, in which each line is connected to different gateways or access points, can be constructed firstly, and then our scheme can be applied directly for convergecast in each line.

The solution for line networks also provides deep insights into the solution for networks with more complex topologies. For example, the lower bound on number channels required for convergecast in multi-line or tree networks can be easily obtained using the same approach, and the schedule algorithms for line networks can also be extended to multi-line or tree networks. The only difference between a line network and a multi-line network is that the GW has multiple children in a multi-line network. To minimize the number of time slots for convergecast, the GW should be optimally scheduled to receive packets from different lines, which can be done by assigning higher priority to lines with large number of nodes. Similarly, each node in a tree network may have multiple children, and the same policy can be applied in tree networks to generate time-optimal convergecast schedule. For more details, please refer to [10].
VII. Conclusion

In this paper, we study the problems of optimal link scheduling and channel assignment for convergecast in wirelessHART networks with line topology, and we discuss these problems for two scenarios: each device can buffer at most one packet at a time slot and each device has unlimited buffering capability. For both scenarios, we establish the lower bound on the number of channels required to complete convergecast using the minimum number of time slots, and we design efficient algorithms for generating optimal convergecast schedule in terms of minimizing both the number of time slots and the number of channels required to complete convergecast. Numerical results demonstrate that our schemes are also memory-efficient. Future work is to evaluate the designed schemes on real wirelessHART testbed and to extend the schemes to networks with other topologies.

References