A Smart Meter Firmware Update Strategy Through Network Coding for AMI Network

Syed Qaisar Jalil[1(✉)], Stephan Chalup[1], and Mubashir Husain Rehmani[2]

[1] The University of Newcastle, Callaghan, Australia
syedqaisar.jalil@uon.edu.au
[2] Waterford Institute of Technology (WIT), Waterford, Ireland

Abstract. With the introduction of communication infrastructure into the traditional power grids, smart power grids are emerging to meet the future electricity demands. In smart grid, advanced metering infrastructure (AMI) is one of the main components that enables bi-directional communication between home area networks and utility providers. In an AMI network, one of the crucial operations is to update the firmware of the smart meters. In this paper, we propose a new forwarding strategy for the firmware updates in AMI network. Our simulation results show that the completion time of the smart meter firmware update process can be reduced significantly by using the proposed new strategy.

Keywords: Network coding · Firmware updates · Smart grid
Neighbourhood area networks · Advanced metering infrastructure

1 Introduction

In traditional power grids, electric power flows only in one direction, i.e. from the power generating stations to the consumers via large interconnected networks. Information is only monitored in the distribution networks that distribute the electric power to the individual consumers [1]. These power grids will not be able to support the future electricity demands due to the ageing infrastructure, growing energy demands, emerging renewable energy sources and security problems. Smart grid (SG) paradigm has been introduced to meet the future electricity demands. SG is envisioned to integrate the bi-directional communication, control technology, and sensing into the power system to achieve significant improvements in reliability, sustainability, stability and security of the electrical grid [2].

In SG, AMI is the architecture that enables two-way communication between home area networks and the utility provider [3]. AMI includes smart meters, communication system and a meter data management system. AMI plays a significant role in the working of SG by measuring, collecting and analysing energy usage patterns. To provide these functionalities, AMI must support a

A.-S. K. Pathan et al. (Eds.): SGIoT 2018, LNICST 256, pp. 68–77, 2019.
https://doi.org/10.1007/978-3-030-05928-6_7

variety of traffic generated from different sources (utility, data concentrators and smart meters) along with the constraints like limited bandwidth and time-critical applications [4].

The smart meter is an advanced energy meter that provides functions like data collection, data storing, load control, display and billing [5]. A smart meter comprises the network interface card (NIC), processor and other electronic parts which can process the information and communicate it over the communication network. It has the ability to run TCP/IP suite as well as can use TCP or UDP. Moreover, the operating system of a smart meter can support a range of applications which enables it to perform tasks like measurement, database management and communication [6]. The software which runs on the smart meter to control, monitor and manipulate the data is known as firmware. Smart meter vendors develop the firmware and update it regularly to improve the functionality, fix the detected bugs, and add new functionality to their smart meters [7]. Moreover, firmware is also useful for utility companies when they update their applications for functionality improvement, bug fixing or due to the changes of legal requirements.

Firmware updates are required to have 98% reliability and in some cases, are required to have a latency of 2 mins [8]. Such reliability is needed because of the fact that if the firmware of the smart meters is compromised, then the real-time monitoring and other functionalities of the SG are compromised too. Also, if there is a bug in the smart meters which hinders the process of demand response management, then the utility provider would want to update the firmware of the smart meters as soon as possible, otherwise it can lead to the high economic loss. Moreover, there is a considerable security threat involved in the firmware update process. For instance, a terrorist organisation can launch the firmware update in the SG which cannot only give them access to shut off the electricity to the customers but also damage the power generating facilities and SG infrastructure [9]. Therefore, in the event of a malicious code attack, some fallback measure must be present such that the smart meter vendors or utility companies can revert all smart meters to the safe mode and this should be done remotely and quickly.

In an AMI network, one of the critical operations is to update the firmware of the smart meters. However, there are only few publications on updating the firmware of smart meters, and most of the work in the literature has focused on the security aspects. Authors in [10] proposed a network service management system and firmware update management system and they also introduced remote firmware update process in AMI networks. To avoid malicious firmware updates, authors in [11] proposed a secure firmware update method based on the pre-defined pattern of changes in the base frequency. In [12], authors have proposed an attribute-based multicast-over-broadcast protocol for firmware updates in AMI network. This protocol makes use of ciphertext-policy attribute-based signcryption to provide access control, confidentiality and message authentication. Authors extended their work in [7] by employing random linear network

coding to overcome the issue of reduced reliability imposed by the use of signcryption.

In this paper, we propose the firmware update process under a new cooperation strategy among the smart meters namely: most served neighbour forwarding firmware update (MSNFFU) and most neighbour forwarding firmware update with network coding (MSNFFU-NC). In our strategies, we utilise neighbour information and random linear network coding to speed up the process of the firmware updates. We compare our strategies with four other cooperation strategies namely: blind forwarding firmware update (BFFU), blind forwarding firmware update with network coding (BFFU-NC), selective forwarding firmware update (SFFU) and selective forwarding firmware update with network coding (SFFU-NC).

The organisation of the paper is as follows. Section 2 provides the background information about the NAN, AMI and RLNC. Section 3 introduces system model and problem definition. In Sect. 4, we present the solution and implementation details along with different cooperation strategies. Simulation results are also presented and discussed in this section. Finally, we summarise our work and conclude the paper in Sect. 5.

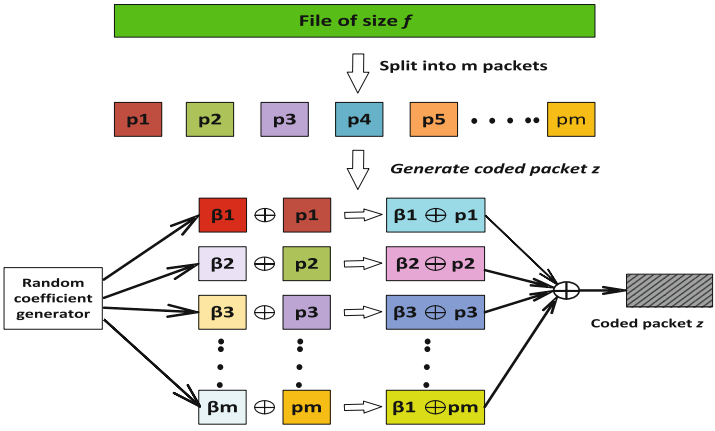

Fig. 1. In this figure, generation of the coded packet through RLNC is shown. $\beta 1, \beta 2, ..., \beta m$ are the random coefficients whereas, *p1, p2, ..., pm* are the packets (chunks) of the original file of size *f*. Addition operation ($\oplus$) is used to combine the random coefficients and chunks. Then, again addition is performed over the combined packets to generate the coded packet *z*.

2 Preliminaries

Neighborhood Area Networks and Advance Metering Infrastructure. Communication in SG can be broken down into following three main network types. Firstly, home area network (HAN) which is responsible for the communication of sensors and devices inside the home. Secondly, neighbourhood area

network (NAN) which is responsible for connecting smart homes (smart meters) and concentrators. Finally, wide area network (WAN) which is responsible for the communication between concentrators and the utility control centres [1].

In NAN, smart meters are connected in a mesh topology along with a concentrator such that total connectivity is ensured. Smart meters generate data which is then collected by the concentrators and forwarded to the control centre using WAN. Moreover, pricing and control messages are delivered in the reverse direction by the concentrators. Since smart meters are nodes, we will use the term "smart meter" and "node" interchangeably. It is worth mentioning that NANs consists of thousands of nodes which are deployed in complex and large geographical areas. Therefore, NANs play an essential role in SG communication [13].

Random Linear Network Coding (RLNC). Random linear network coding is utilized in the broadcast based communication because it produces close-to-optimal throughput and does not require a centralized controller for encoding/decoding operations [14]. In RLNC, data is divided into generations by the data source for broadcasting. There are m number of packets in a generation denoted by p_k where $k = 1, 2, ..., m$. Each packet has a size of q bytes. To obtain an encoded packet, a random linear combination of all the packets are computed in a generation. The encoded packet for each generation can be written as

$$z = \sum_{k=1}^{m} \beta_k p_k \tag{1}$$

where $\boldsymbol{\beta_k}$ is encoding vector. The encoding process of RLNC is illustrated in Fig. 1. In order to decode a packet, an encoding vector is required, which is included in every packet. Upon receiving an encoded packet, the receiver checks if the received packet is useful or not by checking the linear dependency of the encoding vector with all other possessed encoded packets. If the linear dependency is found then the received packet is discarded. On the other hand, the receiver keeps the packet (also known as the innovative packet) in the buffer and tries to decode the packets stored in the buffer.

Each entry of the encoding vector $\boldsymbol{\beta}$ is chosen randomly from Galois field $GF(2^i)$ where i is a positive integer. It is important to note that the Galois field is a finite field which means that the number of elements in it is finite and all the operations defined in it are closed. Therefore, the resultant of any operation from this field on two elements will be in the same field.

3 System Model

We consider a scenario of NANs where n fixed smart meters connected in a mesh topology are deployed in a square of area a. Each node has a well defined transmission radius r. To ensure the total connectivity among nodes, we chose the values of a, n and r by following the inequality $n \geq \frac{10 \cdot a}{\pi \cdot r^2}$ presented in [15].

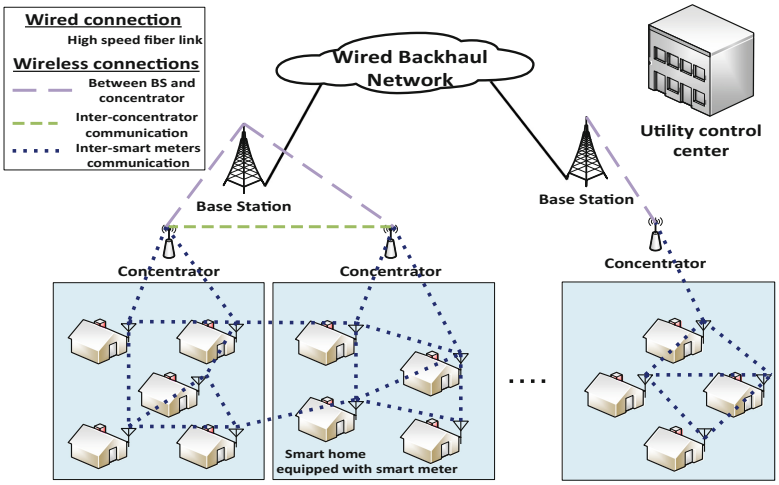

Fig. 2. Illustration of the smart grid communication architecture. Smart homes equipped with smart meters are connected in a mesh topology using wireless medium. Some of the smart homes are connected to concentrators that ensure the total connectivity of the network. Concentrators are connected to the base stations which in turn are connected to the utility control centre through WAN.

There is a number of c_t concentrators in the system which are connected to the control center through WAN. Here, it is important to mention that concentrator placement itself is a research problem [16] and this is beyond the scope of this paper. Therefore, we have only considered the case of concentrators at the extreme of the area a. The considered network model is illustrated in Fig. 2.

Concentrators have a firmware update patch of size f which is divided into $|m|$ chunks and needs to be disseminated in the system. γ is the available rate at which nodes can transmit file chunks over the wireless medium. The time required to download one single chunk and the entire firmware patch at rate b are defined as *one round* and *one unit of time*, respectively. Therefore, if there is a file of size f divided into |m| equivalent chunks then one round will be equal to $\frac{1}{|m|}$ unit of time.

We assume that the utility provider or smart meter vendors can determine the firmware of smart meters which needs to be updated. In the case of an update, the concentrator(s) will be notified and required to release the firmware update patch in the network such that the firmware is updated as soon as possible. A firmware update is a crucial operation because it might be happening in the case of a serious bug which, if delayed, can result in enormous economic loss. Therefore, the firmware update process should be performed in a time efficient manner.

4 AMI Firmware Updates Strategies

We consider six different AMI firmware updates strategies among nodes where three of these strategies make use of network coding. We adopted the first four strategies from the study [17] where authors have used the network coding for file sharing application in the wireless mesh network. In the first two strategies, we use the concept of flooding with and without network coding. Whereas, in the last four strategies, we assume that every node in the network has a neighbour information table, i.e. a list of available chunks at neighbours. Considering that smart meters have the capability to manage the database [6], and importance of the firmware updates, it is reasonable to assume such strategies where smart meters can maintain a table. However, building and maintaining such tables is beyond the scope of our paper. These strategies will ensure when and how network coding can be used in the process of smart meter firmware updates.

1. **Blind Forwarding Firmware Update (BFFU)** is a primitive cooperation strategy which is based on flooding. In this strategy, when a node receives a chunk, it tries to get access to the wireless medium. Upon getting access to the wireless medium, a node transmits the chunk to its neighbours without considering if any neighbour is interested in this chunk or not. Before performing a single transmission, a node may receive multiple chunks from the neighbours due to the shared wireless medium. Moreover, the order of reception of the chunks determines the transmission sequence of the chunks, i.e. first received chunks will be transmitted first.
2. **Blind Forwarding Firmware Update with Network Coding (BFFU-NC)** is same as the BFFU, but it uses network coding before transmission. In this strategy, upon receiving a new combination and getting access to the wireless medium, a node generates combination of all chunks (that are in the possession of this node) and forwards it to the neighbour nodes.
3. **Selective Forwarding Firmware Update (SFFU)** make use of the assumption that every node contains a table that has information about the neighbours, i.e. the list of chunks available at the neighbours. In this strategy, every node checks its table continually, and if it finds a chunk which is required by one of its neighbours, then this node transmits that particular chunk.
4. **Selective Forwarding Firmware Update with Network Coding (SFFU-NC)** is similar to the SFFU but it utilises network coding during transmission. Identical to the SFFU, every node checks its table continually, and upon finding a chunk of interest to its neighbour, this node generates the combination of all the information that it posseses and transmits it.
5. **Most Served Neighbour Forwarding Firmware Update (MSNFFU)** is similar to the SFFU, i.e. every node has a table which has the information of available chunks at its neighbours. In this strategy, a node is selected for transmission based on the number of neighbours it can satisfy, i.e. how many of its neighbours require a chunk that is available at this node. A priority queue of nodes, where the node that can serve most neighbours is first in

the queue, is created in each round. If there are many nodes with the same priority, then the first from the queue is selected.

6. **Most Served Neighbour Forwarding Firmware Update with Network Coding (MSNFFU)-NC** is similar to the MSNFFU, i.e. every node has a table which has the information of available chunks at its neighbours, but it utilises network coding during transmission. Similar to the MSNFFU, a priority queue is created in each round based on the number of neighbours a node can satisfy, but the transmission is different from MSNFFU. When a node is selected for transmission, it generates the combination of all the information that it posseses and transmits it. Pseudo code for MSNFFU-NC is presented in Algorithm 1.

4.1 Implementation Details

The proposed cooperation strategies were implemented in C++. We have taken the implementation code from [17] and modified it according to our scenario. Here, it is important to mention that the implementation considers a collision-free medium access control (MAC) protocol that gives the same probability to all the competing nodes to access the medium.

As stated earlier, the time-space is divided into rounds which enables us to analyse the distribution of the firmware patch among the nodes in the network. Transmission can only occur at the beginning of the round. The following four steps are performed in each round for transmission of chunks among the nodes.

In the first step, nodes which have chunks to send are identified and saved in the candidates list. Initially, only the concentrators have the patch; therefore, they are placed in the candidates list but later on, when the file chunks propagate in the network then other nodes will be added to this list. Assuming the collision-free MAC protocol, transmission can only be performed by some of the nodes in the candidate list. Therefore, we select the nodes that can perform transmission in the second step. We begin this step by randomly choosing a node from the candidates list and place it in a new list called the transmitters list. To ensure collision-free transmission, we look for the neighbours of the selected node and delete them from the candidates list. We also remove the neighbours of the neighbours of the selected node to avoid the hidden terminal situation.

It can be seen that the transmission selection criteria are fair and simple and gives the same chance, to all the nodes, to access the medium. The selection of the nodes from the candidates list is performed until it becomes empty. Then, in the third step, nodes from transmitter list perform transmission, i.e. transmit one chunk each. Finally, an update of the list of chunks at all nodes is performed in the fourth step. The above mentioned steps are performed continuously until the firmware patch is completely disseminated in the network.

4.2 Performance Evaluation

We conducted simulations for randomly created topologies of 100, 200, 300, 400 and 500 nodes. In every case, there was one concentrator that was placed at

Algorithm 1. MSNFFU-NC

```
while update patch is not disseminated in the network do
    for every node in network do
        check the neighbors table
        if node has chunk(s) required by neighbors then
            candidates_list ← node
        end if
    end for
    Based on the number of neighbours a node can serve, sort the candidates_list in
descending order
    while candidates_list is not empty do
        node_ ← select first node from candidates_list
        delete neighbors and neighbors of neighbors of selected node_
        transmitters_list ← node_
        delete selected node_ from candidates_list
    end while
    for every node in transmitter_list do
        generate random coefficient from Galois field
        create coded packet of all chunks
        perform transmission
        update nodes
    end for
end while
```

the extreme of area *a*. The reason for considering different numbers of nodes with one concentrator is to generalise our results for rural and urban areas, i.e. in urban areas, nodes are deployed more densely compared to rural areas. We divided the firmware update patch into 10 chunks. We considered completion time as a performance metric, i.e. the amount of time required to distribute the firmware patch in the system. We conducted simulations with 500 runs, and average values are shown in Fig. 3.

In Fig. 3, the completion time of different nodes with $|m| = 10$ for all strategies is plotted. We can see that BFFU has the worst completion time compared to other schemes and the reason is that it is based on flooding. In BFFU, a node will transmit the chunks even when none is interested in receiving the chunks; therefore, a lot of useless chunks are forwarded and this makes the process, of disseminating the actually required chunks, slow. When NC is incorporated in BFFU, its performance gets better. Now, the completion time of SFFU and SFFU-NC shows similar behaviour and the reason is that in SF only those chunks are transmitted by a node which is of interest to any of its neighbours. When NC is incorporated in SF, it results in an increased number of neighbours that are interested in each transmission. The completion time of MSNFFU is better than all above strategies because in this strategy nodes are selected for transmission based on the number of neighbours they can serve. Therefore, in each transmission the maximum number of nodes gets served; hence, the firmware update patch is disseminated quickly. Now, MSNFFU-NC has the most reduced

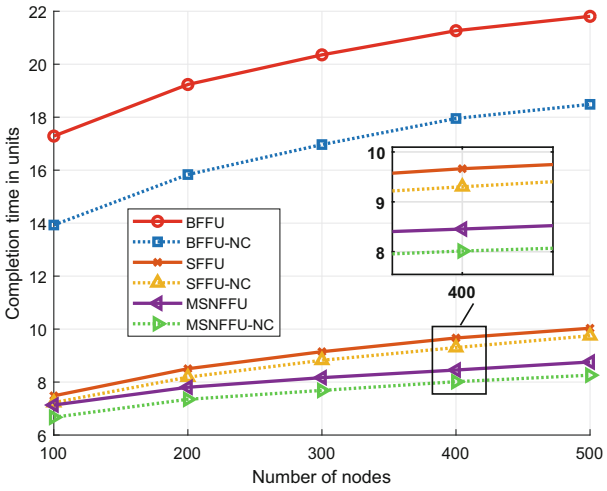

Fig. 3. Completion time vs number of nodes, where $|m| = 10, r = 5$

completion time compared to other strategies. It is even better than MSNFFU because, upon getting access to the medium, a node will generate the combination of all the information that it posseses and transmits. Whereas, in MSNFFU a node can only transmit one chunk at a time.

5 Conclusion

In this paper, we proposed a new forwarding strategy for the firmware updates in AMI networks. We compared our proposed approach with other different cooperation strategies. Through simulation experiments we demonstrated that our proposed MSNFFU-NC outperforms all other strategies and can reduce smart meter firmware update process time.

References

1. Khan, A.A., Rehmani, M.H., Reisslein, M.: Cognitive radio for smart grids: survey of architectures, spectrum sensing mechanisms, and networking protocols. IEEE Commun. Surv. Tutor. **18**(1), 860–898 (2016)
2. Wang, W., Xu, Y., Khanna, M.: A survey on the communication architectures in smart grid. Comput. Netw. **55**(15), 3604–3629 (2011)
3. Mohassel, R.R., Fung, A., Mohammadi, F., Raahemifar, K.: A survey on advanced metering infrastructure. Int. J. Electr. Power Energy Syst. **63**, 473–484 (2014)
4. Ramírez, D.F., Céspedes, S.: Routing in neighborhood area networks: a survey in the context of ami communications. J. Netw. Comput. Appl. **55**, 68–80 (2015)
5. Zheng, J., Gao, D.W., Lin, L.: Smart meters in smart grid: an overview. In: IEEE Green Technologies Conference (GreenTech), pp. 57–64 (2013)

6. Khalifa, T., Naik, K., Nayak, A.: A survey of communication protocols for automatic meter reading applications. IEEE Commun. Surv. Tutor. **13**(2), 168–182 (2011)
7. Tonyali, S., Akkaya, K., Saputro, N., Cheng, X.: An attribute network coding-based secure multicast protocol for firmware updates in smart grid AMI networks. In: 26th International Conference on Computer Communication and Networks (ICCCN), p. 19, July 2017
8. Khan, A.A., Rehmani, M.H., Reisslein, M.: Requirements, design challenges, and review of routing and MAC protocols for cr-based smart grid systems. IEEE Commun. Mag. **55**(5), 206–215 (2017)
9. Kropp, T.: System threats and vulnerabilities (power system protection). IEEE Power Energy Mag. **4**(2), 46–50 (2006)
10. Kim, Y., Oh, D., Ko, J., Kim, Y., Kang, S., Choi, S.-H.: A remote firmware upgrade method of NAN and HAN devices to support AMI's energy services. In: Lee, G., Howard, D., Ślęzak, D. (eds.) ICHIT 2011. CCIS, vol. 206, pp. 303–310. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24106-2_40
11. Katzir, L., Schwartzman, I.: Secure firmware updates for smart grid devices. In: 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies, pp. 1–5 (2011)
12. Tonyali, S., Akkaya, K., Saputro, N.: An attribute-based reliable multicast-over-broadcast protocol for firmware updates in smart meter networks. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 97–102, May 2017
13. Meng, W., Ma, R., Chen, H.H.: Smart grid neighborhood area networks: a survey. IEEE Network **28**(1), 24–32 (2014)
14. Ho, T., Koetter, R., Medard, M., Karger, D.R., Effros, M.: The benefits of coding over routing in a randomized setting. In: IEEE International Symposium on Information Theory (2003)
15. Philips, T.K., Panwar, S.S., Tantawi, A.N.: Connectivity properties of a packet radio network model. IEEE Trans. Inf. Theory **35**(5), 1044–1047 (1989)
16. Aalamifar, F., Shirazi, G.N., Noori, M., Lampe, L.: Cost-efficient data aggregation point placement for advanced metering infrastructure. In: IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 344–349 (2014)
17. Hamra, A.A., Barakat, C., Turletti, T.: Network coding for wireless mesh networks: a case study. In: International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006), pp. 9-114 (2006)