September 21, 2018

NON-INTERFERING SEQUENCE FOR MASSIVE DATA TRAFFIC IN LOW-POWER LOSSY NETWORKS

Caijun Jiang

Dapeng Zhu

Yiyun Meng

Xiaopu Zhang

Recommended Citation

Jiang, Caijun; Zhu, Dapeng; Meng, Yiyun; and Zhang, Xiaopu, "NON-INTERFERING SEQUENCE FOR MASSIVE DATA TRAFFIC IN LOW-POWER LOSSY NETWORKS", Technical Disclosure Commons, (September 21, 2018)
https://www.tdcommons.org/dpubs_series/1524

NON-INTERFERING SEQUENCE FOR MASSIVE DATA TRAFFIC IN LOW-POWER LOSSY NETWORKS

AUTHORS:
Caijun Jiang
Dapeng Zhu
Yiyun Meng
Xiaopu Zhang

ABSTRACT

Techniques are described herein for generating a non-interfering sequence. This mechanism mainly comprises two steps. The first step involves computing and generating a schedule, and the second step involves receiving metering data from all nodes. After the second step ends, the head end system builds a residual node set containing all nodes that failed to report data in a given schedule, and then the first and second steps are repeated until the residual set is empty. The two steps are performed serially. The techniques described herein optimize these two steps. In particular, first a system mathematical mode is built from a Routing Protocol for Low-Power and Lossy Networks (RPL) p-table and link neighbor, second the constraint condition is constructed, and third the parameter is adjusted and the schedule is implemented.

DETAILED DESCRIPTION

Routing Protocol for Low-Power and Lossy Networks (RPL) is oriented towards supporting multi-point to point (M2P) communications in which multiple points typically communicate with one sink node. Due to the heterogeneous traffic patterns in the network, some sensor nodes may have a much heavier workload in terms of packets forwarded than others. From a global perspective, the connected grid mesh formation network topology is shaped like a funnel in that the closer to the root, the heavier pressure. Fortunately, a large number of algorithms have been proposed to optimize RPL to maintain the unbalanced workload distribution.

Once the connected grid mesh network topology is stable, the sensor nodes need to interact with the remote server. Figure 1 below illustrates a typical collision without consideration of a data traffic sequence. The solid line represents the parent-children relationship from the RPL tree and the dotted line represents the link neighbor relationship.

In case 1, meters 3 and 6 collide in the root node. In case 2, meter 2 appears, transfers, and receives the collision. In case 3, meters 2 and 4 collide as direct connections in the link neighbors.

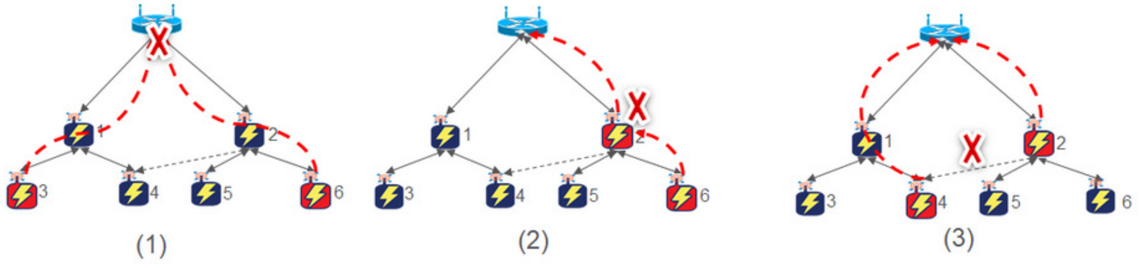

Figure 1: Typical collision for unreasonable data collection sequence

The techniques described herein generate a global planning schedule to make efficient use of the RPL topology instead of improving the protocol. In terms of massive data traffic, such as Advanced Metering Infrastructure (AMI), if data is requested without considering load balance, even the most perfect network topology could not prevent the congestion of packets and guarantee the transfer efficiency.

This mechanism mainly comprises two steps. The first step involves computing and generating a schedule, and the second step involves receiving metering data from all nodes. After the second step ends, the head end system builds a residual node set containing all nodes that failed to report data in given schedule, then repeat the first and second steps until the residual set is empty. The two steps are performed serially. The techniques described herein optimize these two steps. The following steps describe how to generate the non-interfere sequence.

The first step is to build the system mathematical mode from an RPL p-table and link neighbor. An example network topology is illustrated in Figure 2.1 below. The line represents the parent-children relationship from the RPL tree. The solid lines are the preferred parent and the dotted lines are the backup parents. The dotted ellipse selection represents the link neighbor relationship. Figure 2.2 below illustrates the data transition matrix T which is a very sparse matrix and which represents the data traffic transfer direction. Figure 2.3 below illustrates the neighbor connection matrix N which represents the physically connection of any two nodes and which is fairly stable since the site deployment.

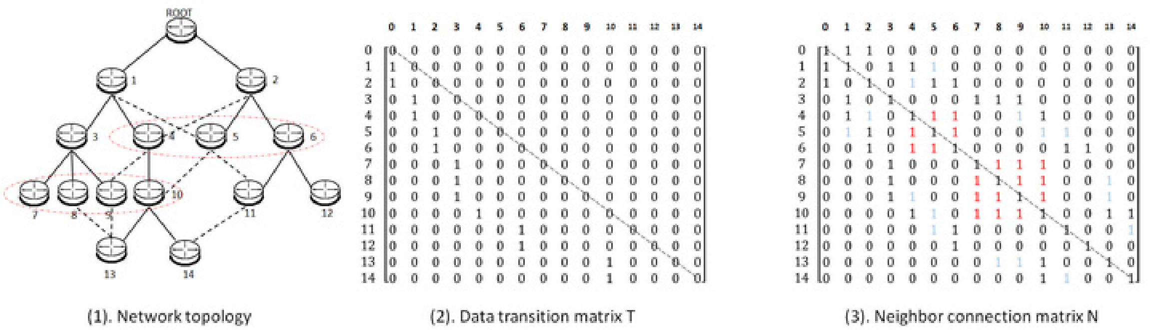
(1). Network topology (2). Data transition matrix T (3). Neighbor connection matrix N

Figure 2: System mathematical mode in LLNs

The following two formulas represent the fatal mathematical mode.

$$S_{j+1} = S_j\, T \qquad (1)$$

$$I_j = S_j\, \mathrm{N} \qquad (2)$$

S_j represent the active transmitting nodes at time j, (e.g., a 1xn and value 0-1 matrix). I_j represent the degree of interference for the active transmitting nodes.

The second step involves constructing the constrain condition. Three rules from RPL topology may prevent the collision in Figures 1.1 and 1.2. Even though two transmitting missions are independent from an RPL topology perspective, interfering with each other is highly probable from the link neighbor perspective. Thus this constraint condition must be considered in the final sequence. The constraint formula is as follows.

$$\max\big(S * (SN)\big) < 2 \qquad (3)$$

The Hadamard product, denoted by *, is different from the common matrix product. For example, if S is [0,0,0,0,0,1,0,0,0,0,1,0,0,0,0], and nodes 5 and 10 are selected, the according to above inequality S*(SN)=[0,0,0,0,0,2,0,0,0,0,2,0,0,0,0]. Obviously, the inequality relation is not true as the sequence should not contain them simultaneously.

A single node should avoid simultaneously sending and receiving. In order to increase the robustness of the sequence, a new parameter gap, denoted by G, is added. $G_{1,10}$ represents the hop difference between node 1 and 10 ($G_{1,10} = 2$, $G_{1,14}=3$ in figure 2.1). The larger the gap values, the better the sparsity of the sequence. The constraint formula is improved as follows.

$$(S_i+X_i) * T^k \sum_{n=0}^{\max(G)} (T^n) \le C \qquad (4)$$

The parameter may be adjusted and the schedule implemented. The implementation of the schedule depends on the mechanism of obtaining data by a remote server (e.g., Head End System (HES)). Currently, request-response and service listen are two typical approaches of obtaining data in an HES, and the selection may affect the generation of the schedule. Request-response requires the HES to start the data process actively, and each request contains a group with maximum elements. Service listen requires the mesh nodes to report data actively to the HES, and the nodes should report accurately at the scheduled time.

In addition to the data obtaining mechanism, generating a reliable schedule requires consideration of the bandwidth, data size, network quality, and so on. The parameter adjustment determines the final generation of the sequence. Experimental results show that the parameter of the schedule generator should be adjusted dynamically according to the actual implementation results.

A non-sequential generator makes each X_i in a schedule not depend on former X_{i-1}, which means state matrix S_i is always zero before X_i is calculated. For example, if max(G) is 2 in non-sequential mode, nodes 1, 6, and 7 are enabled in X_0, nodes 2, 3, 11, and 13 are enabled in X_1, and each X_i represents a dependent group.

A sequential generator calculates in each time cycle, and thus each X_i relies on state matrix S_i. For example, if max(G) is 2, nodes 1, 6, and 7 are enabled in X_0. But in the next time cycle, nodes [1, 6, 7] translate to nodes [0, 2, 3], and thus node 11 is enable in X_1. And then nodes [0, 2, 3, 11] are active and translate to [0, 1, 6] in the next time cycle. In the second time cycle, node 7 and 14 are enable in X_2.

In summary, techniques are described herein for generating a non-interfering sequence. This mechanism mainly comprises two steps. The first step involves computing and generating a schedule, and the second step involves receiving metering data from all nodes. After the second step ends, the head end system builds a residual node set containing all nodes that failed to report data in a given schedule, and then the first and second steps are repeated until the residual set is empty. The two steps are performed serially. The techniques described herein optimize these two steps. In particular, first a system mathematical mode is built from a Routing Protocol for Low-Power and Lossy Networks

(RPL) p-table and link neighbor, second the constraint condition is constructed, and third the parameter is adjusted and the schedule is implemented.