# A Novel Routing and Data Transmission Method for Stub Network of Internet of Things based on Percolation

Xiangming Li, Jihua Lu, Jie Yang, and Jianping An
School of Information and Electronics
Beijing Institute of Technology
Beijing, China
{xmli, lujihua, yangjie, an}@bit.edu.cn

***Abstract*—A new network architecture and routing method based on percolation for the Machine-to-Machine (M2M) stub network of Internet of things is proposed. The proposed network architecture is router-free, in which efficient routing can be operated with percolations based on the six degrees of separation or small world network. A file transmission will be divided into two phases: routing phase and data transmission phase. In the routing phase, probe packets will be transmitted and forwarded in the network thus path selections are performed based on small-world strategy. In the second phase, the file will be encoded and transmitted using the paths selected at the first phase. In such a way, an efficient routing and data transmission mechanism can be built, with which we can construct a low-cost, flexible, ubiquitous stub network.**

***Keywords- Internet of things; percolation network; Machine-to-Machine; routing***

## I. INTRODUCTION

Machine-to-machine (M2M) is a set of technologies that enables data communications between machines, devices, systems, and people. As defined in [1-2], M2M uses a device to capture an event, which is relayed through a network to an application, that translates the captured event into meaningful information [1-2]. Meanwhile, Internet of things (IoT) [3], referring to the networked interconnection of everyday objects, can be regarded as an extension of the existing interaction between humans and applications through the new dimension of "things" communication and integration [4]. In IoT, devices are clustered together to create a stub M2M network, and are then connected to its infrastructure, *i.e.*, the traditional "Internet of people".

In a stub M2M network of IoT, the devices are usually very small, and interconnect over wireless communications. Therefore, both the power consumption and processing ability of the nodes are restricted. Moreover, the scale of network may vary dynamically [4]. In such a stub network, it is costly to use the traditional Internet protocol. First of all, the bandwidth of the stub network is limited and its data packet size is usually small, while the TCP/IP protocol and its ARQ mechanism consume a lot of extra bandwidth. Moreover, TCP/IP protocol requires a large number of gateways and routers, which increases the complexity of the network structure, leading to high cost in network deployment and maintenance.

Ad hoc network works without any gateway and special routers [5-6]. However, in Ad hoc, each node must be informed of the complete structure of the entire network to calculate its routing tables. As a result, a large amount of information needs to be broadcasted among the network nodes to maintain the routing. Any change in the network structure, as nodes' joining in or departure, may generate significant overhead. Furthermore, to manage and operate a large routing table, a node requires rather high level of storing and processing ability.

An IoT stub network is usually composed of hundreds to thousands of, or even more nodes, and each node can directly communicate with several other nodes. Such a network should have the small-world properties [7]. On the basis of such an observation, we build a small world routing strategy: if a node wants to communicate with a non-conterminous node, it simply hands its data to a conterminous node that may be neighbored to the destination, and the conterminous node repeats the same procedure until the data is successfully delivered to the destination. To avoid the bandwidth waste, a probe mechanism is used in path selection. Moreover, to prevent the frequent feedback signals in the data transmission, fountain coding is used to encode the source packets. Such a routing and data transmission strategy will be referred to as percolation network scheme.

The rest of the paper is organized as follows. The network architecture is introduced in Section II and the percolation routing and data transmission method is described in Section III. Simulations about the IoT based on percolation are specified in Section IV. And finally, in Section V we conclude the work.

## II. A NEW NETWORK ARCHITECTURE OF THE INTERNET OF THINGS BASED ON PERCOLATION

An IoT stub with small-world properties of M2M network is usually composed of hundreds to thousands of, or even more nodes, and each node can router-freely communicate with several other neighbor nodes. Each node in the network connected to several neighbor nodes randomly. In such a way, a small-world network is formed, in which efficient routing can be operated with percolations based on the six degrees of separation.

 

### *A. Network Topology of the StubNnetwork of IoT*

Due to the signal-attenuation nature of wireless signals, a node can directly communicate with its neighbor nodes while it is hard to communicate with the other nodes that are out the scope of radio range. Therefore we take the network shown in Fig. 1 as an example of the stub network of IoT. In the network, each node is directly connected to one node or several other nodes. A node's neighbor nodes will be referred to as close-neighbors, if they are directly connected to the node. For example, the nodes C, G and L are close-neighbors of the node F.

Such a network should have the property of the small-world network. When one node is not a close-neighbor of another node and if the two nodes want to communicate with each other, they can communicate with each other through relay. For example, when F wants to communicate with the node N, it may first "guess" that the node N is a close-neighbor of the node L, so it hands its data to the node L. Similarly, the node L "guesses" that the node O may directly connects to the node N, so it forwards the data to the node O. Finally, the node N delivers the data to the node N. Based on the six degrees of separation principle, any node can reach everywhere in finite steps of relaying with a very high probability [8][9].

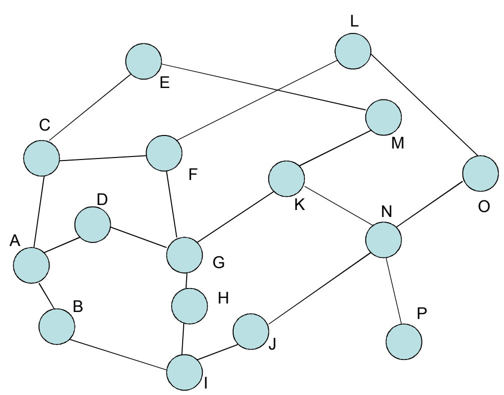


Figure 1. An example of small-world network

In the network, each node acts as relay for the other nodes, it simultaneously transmits its own data packet, thus no dedicated routers will be deployed. As stated, the routing may be based on "guess", which are best presented as likelihood. That is, a node will hand its data to its close-neighbors which can forward the data to the destination with high probability. For this reason, each node in the network should have a table which contains all or part of the following messages.

- Number of close-neighbors. This field describes the number of nodes available for direct communication without relay.
- Names of the close neighbors.
- Addresses of the close-neighbors. An address of a node is the identity number of a node. Generally, the name and the address can be the same thus the name and the address fields can be merged into one field.
- Bandwidth of the direct links from me to close-neighbors. Here we define "me" to be the current node.
- Measures from me to other nodes via close neighbors. Starting from me, one may have one or more paths connecting to another node with the help of close-neighbors. Some close-neighbors may take fewer hops while some other close-neighbors may take more hops to reach the destination. The measures from me to a node via a close neighbor reflect the likelihood that a path can be built from me to a destination node through a particular close-neighbor node. To an unknown destination node, one may initialize these values as zero at the first time choosing path for it.
- Bandwidth from me to other nodes via close-neighbors. These values can be initialize to a predetermined minimum values.

TABLE I. ROUTING INFORMATION TABLE OF NETWORK NODES

| | |
|---|---|
| **Number of close-neighbor nodes** | **N** |
| **Names of close-neighbor nodes** | $L_1, L_2, \ldots, L_N$ |
| **Addresses of close-neighbor nodes** | $A_1, A_2, \ldots, A_N$ |
| **Bandwidth of the direct links from me to close-neighbors** | $R_1, R_2, \ldots, R_N$ |
| **Measures from me to the node $D_1$ via the close neighbors** | $M_{C\_L_1\_D_1}$ |
| | … |
| | $M_{C\_L_N\_D_1}$ |
| … | |
| **Measures from me to the node $D_L$ via the close neighbors** | $M_{C\_L_1\_D_L}$ |
| | … |
| | $M_{C\_L_N\_D_L}$ |
| **Bandwidth from me to the node $D_1$ via the close neighbors** | $R_{C\_L_1\_D_1}$ |
| | … |
| | $R_{C\_L_N\_D_1}$ |
| … | |
| **Bandwidth from me to the node $D_L$via the close neighbors** | $R_{C\_L_1\_D_L}$ |
| | … |
| | $R_{C\_L_N\_D_L}$ |

In Table I, we suppose that the current node (me) is C, the other nodes (not the close-neighbor nodes) are $\mathbf{D_1}$, $\mathbf{D_2}$,…, $\mathbf{D_L}$, and $M_{C\_L_i\_D_k}$ is the measure from me to the node $\mathbf{D_K}$ via the close neighbor $\mathbf{L_i}$.

### B. *Network Initialization Process*

The network initialization process is as follows.

- A node sends the query signal of close-neighbor nodes. Any node which receives this query signal sends back a reply. The nodes involved in these query and reply signals establish and renew their information table, including the following fields: number of close-neighbor nodes, names and addresses of close-neighbor nodes, bandwidth of the direct links from me to close-neighbor, etc.
- Set all the measures from me to other nodes via close neighbors to 0. Set all the bandwidth from me to other nodes via close-neighbors to a predetermined minimum values. For those unknown destination nodes to me, reserve adequate memory for the measures and bandwidth.

## III. Routing And Data Transmission Method Based on Percolation

We propose a routing method for the stub network of IoT based on the six degrees of separation principle, which is referred to as percolation routing. Suppose that the stub of IoT have many nodes, and every node has connections with several close-neighbor nodes. The scale of this type of network may change with time, that is, its connection relationship may change with the increasing and disappearing of nodes and link bandwidth may change dynamically with the battery consumption, movement in places of the nodes, and so on.

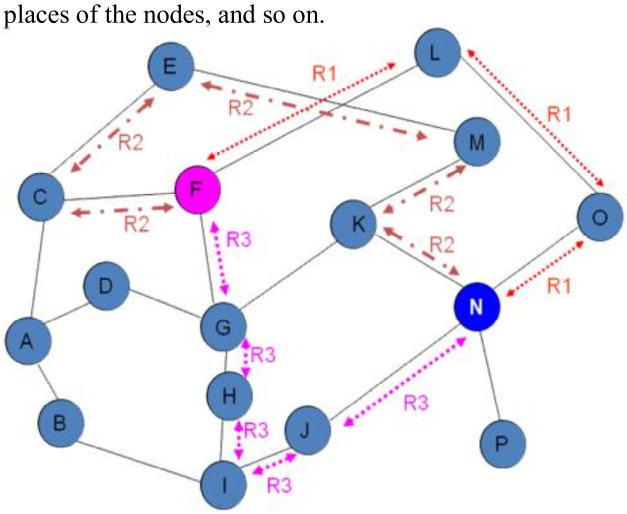


Figure 2. A routing with multiple paths based on percolation

When a network scale is large and if a node is randomly connected to several nodes in the network, this network can be modeled as a small world network as long as a node is randomly connected to several other nodes. In such a network, any node can establish communication links with other nodes through finite steps of relaying (generally no more than six steps) with a very high probability. When a node wants to send data packets to another node and if the source can not reach its destination directly, it simply hands the data packets to a close-neighbor which is likely to close to the destination. The selected close-neighbor repeats the procedure until the data reaches the destination. In such a way, multiple paths between the source and the destination nodes may be built. Notice that some paths will terminate early before they connect to the destination due to the network jam and path halt. Thus part of the data packets will be lost. To avoid frequent feedback in an ARQ mechanism, the source may encode the source file using fountain codes [10][11]. With fountain codes, the destination is then capable of successfully decoding the received file if a sufficient number of packets is received, regardless of which packets have been exactly received. This process of routing and data transmission is just like the water seeping in a porous material. As little trickle becomes a stream, and stream becomes torrent, multiple parallel data transmission paths may merge to a high-speed route. Due to this reason, the routing and data transmission based on six degree separation and fountain codes are referred to as percolation routing and data transmission method.

### A. *Percolation routing*

It is seen that percolation routing is based on "guess" thus some paths may reach the destination finally whereas other paths will terminate at other nodes except the destination. Thus direct data packets based on percolation scheme is inefficient. To avoid the bandwidth loss, we divide the data delivery procedure into two phases: routing phase and data transmission phase. The former phase corresponds to path selection in which probing packets will be transmitted and relayed in the network. In the later phase, paths are available and data transmission will be performed efficiently. To prevent flooding, the number of hops of a path will be restricted up to Y. Moreover, to balance the network resource, we limit the number of maximum paths to X.

Assume that the network initialization is complete, i.e., each node has its initialized routing information table. Then the routing procedure is as follows.

Step 1: The source node sends probe packets to its close-neighbor nodes. Probe packet contains the source address, destination address, and a hop counter. Let us reserve $Y$ positions in the probe for the relay nodes. Set the counter number to zero.

Step 2: A close-neighbor node receives the probe packet. If the node is the destination node, go to step 3; Otherwise, check the hop counter and switch to the following two cases if the hop counter is less than Y :

Case 1: If there doesn't exist a close-loop close-neighbor node in this selecting path, increase the hop counter by one and the address of the current node is written into the probe packet. Forward the renewed probe packet to a close-neighbor of the maximum measures. Repeat step 2 until the destination is reached, go to step 3;

Case 2: If no any close-neighbor forms a non-closed path, we drop this probe data packet and terminate this routing process.

Step 3: The destination node calculates and selects maximum $X$ distinct or parallel paths as the routing. The rule to select paths may be based on the least hops, the maximum throughput, and so on.

Step 4: The destination node feeds back the acknowledgement packets composed of the selected path information to the transmitter node through the selected paths. When the feedback information passes through the intermediate nodes, each intermediate node records the routing information and renews its routing information table by increasing each measure from me to any involved node on the path via close neighbors with a unit value.

Thus, the routing procedure is complete. At this point, it is ready to transmit the data packets.

### B. Percolation Data Transmission

Then the data file will be transmitted in the following procedure.

Step 1: The source node encodes the data file to be transmitted. Assign the loads to the paths proportional to their throughput capacity. Feed the encoded packets to the selected paths according to their load capacity.

Step 2: An intermediate node receives the data packets. If the node is the destination node, go to step 3; otherwise, the data packets will be forwarded to the next node on the path, until the destination is reached.

Step 3: The destination node will assemble the data packets received and decode them. An ACK signal will be sends back to the transmitter node as soon as the data file is successfully recovered.

Step 4: The source node receives the ACK from the destination node and go to above step 1 to encode and transmit the next file.

By the way, one may use fountain codes, as LT codes and Raptor codes in the source data encoding in Step 1.

Consider the stub network in Fig. 1 again. No direct connections between node F and node N are available while they want to create communicate links. Suppose that each node in the network contains a routing information Table as shown in Table I. We create three parallel paths using the percolation routing, $R_1$, $R_2$ and $R_3$, as shown in Fig. 2. Then the data files can be transmitted efficiently using these paths.

It is seen that the proposed network routing method either works when a node has the full network architecture or it does when a node contains only local network structure. Since the storage and processing ability of a node in M2M as stub network of IoT is limited, it works with limited information of local close-neighbors. Thus with the proposed method, it may greatly save the cost of the equipments and network deployment of a stub network of IoT. Moreover, it has the advantages of efficiency, self-management, and self-maintaining.

## IV. Simulations and Results

In this section, we give brief descriptions of an example of our stub IoT networking process in Matlab. The network construction progress may produce many small disconnected subgraphs with $10^{\wedge 4}$ is the network dimension which is more realistic and consistent to the practical values. The curvature value(or the clustering coefficients) of a node was defined as how many pairs of its neighbors are themselves connected, and dividing this number by maximum possible number of connections. The curvature value may be computed as: diag(A^3)/(sum(A).*(sum(A)-1)). The curvature values of nine different random models are listed in Table I through simulations.

TABLE I. Nine Random Model and Their Curvatures

| Model name | Curvature Value |
|---|---|
| Erdrey | 8.1009e-004 |
| Gilbert | 0.0012 |
| Small world | 0.4616 |
| Geometric | 0.5825 |
| Pref | 0.0065 |
| Renga | 0.3925 |
| Kleinberg | 0.2001 |
| Lockandkey | 0.0021 |
| Sticky | 0.0134 |

We choose the Geometric random model as our simulation model. Fig. 3 illustrates the process of a geometric random graph with n=100 and r=0.2. The routing with multiple paths based on percolation are shown below.

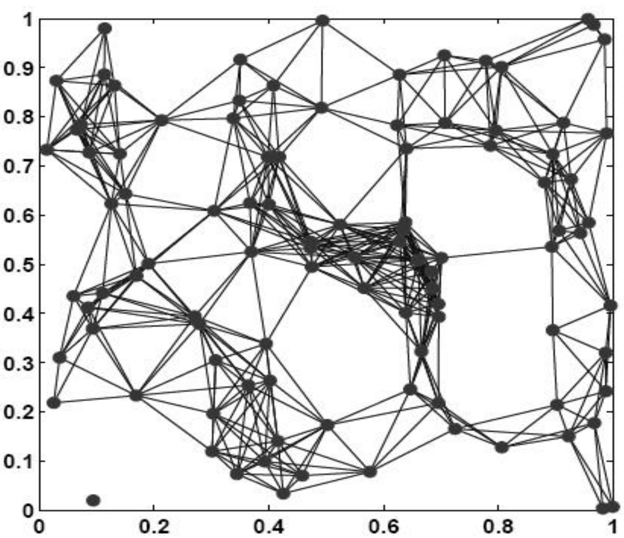


Figure 3. Construction of a geometric random graph, Here, n=100 and r=0.2.

The Mean path lengths under different Radius of geometric random model are listed in Table II.

TABLE II. The Mean Pathlengths under Different Radius of Geometric Random Model

| Conditions(n=100) | Mean-Pathlength |
|---|---|
| r=0.186 | 4.0988 |
| r=0.192 | 3.5716 |

| | |
|---|---|
| ***r=0.198*** | ***3.656*** |
| ***r=0.204*** | ***3.654*** |
| ***r=0.21*** | ***3.4685*** |
| ***r=0.216*** | ***3.2436*** |

## V. CONCLUSIONS

We proposed a new routing and data transmission method for the stub network of internet of things based on the six degrees of separation. The proposed scheme consists of two phases: routing phase and data transmission phase. In the probe phase, probes packets are transmitted and flowed in the network, building multiple paths which form a route. After that, the data file will be fountain encoded and then transmitted. The proposed method has the advantages of efficiency, self-management, self-maintaining and low cost.

## VI. ACKNOWLEDGMENT

This work was supported in part by NSF of China with grants 60972017, 61002014, 60972018, 61072050 and 61072048, the Excellent Young Teachers Program of MOE, PRC with grant 20091101120028, the Research Fund for the Doctoral Program of Higher Education with grants 20091101110019 and 20070007019, the Important National Science& Technology Specific Projects with grant 2010ZX03003-004-03, and the Beijing Natural Science Fund of China with grant 4101002.

## References

[1] Stephen M. Dye, "End-to-End M2M", Third Edition, Mind Commerce LLC, Jan 2010, pp. 222-224.

[2] Bob Emmerson, "M2M: The Internet of 50 Billion Devices", WinWin Magazine, Jan. 2010, pp.19-22.

[3] Commission of the European communities, Internet of Things in 2020, EPoSS, Brussels, 2008.

[4] Antoine de Saint-Exupery, Internet of Things – Strategic Research Roadmap, Sep. 15, 2009.

[5] C.K. -K Toh, "Ad Hoc Mobile Wireless Networks : Protocols and Systems", Prentice Hall PTR Upper Saddle River, NJ, USA, pp.324-326.

[6] G. Jayakumar, and G. Gopinath, Ad Hoc Mobile Wireless Networks Routing Protocols – A Review, Journal of Computer Science, vol. 3, No. 8, pp. 574–582, August 2007

[7] G. Gianini, E. Damiani; "Do Neighbor-Avoiding Walkers Walk as if in a Small-World Network? " INFOCOM IEEE Conference on Computer Communications Workshops 2009, pp: 1–6.

[8] A. Mei; J. Stefa, "SWIM: A Simple Model to Generate Small Mobile Worlds", INFOCOM 2009, pp: 2106–2113.

[9] O. Bakun, G. Konjevod, "Adaptive Decentralized Routing in Small-World Networks", INFOCOM IEEE Conference on Computer Communications Workshops , 2010 , pp: 1–6.

[10] M. Luby, "LT codes," Proc. of 43rd IEEE Symp. Foundations of Computer Sciences, Vancouver BC, Canada, pp. 271-280, Nov. 2002.

[11] A. Shokrollahi, "Raptor codes," IEEE Tranactions on Information Theory, vol. 52, no. 6, pp. 2551-2567, June 2006.