Cluster approach for network routing in hierarchical systems

S V Timofeeva¹, D M Sarkisyan¹, N A Alzinskaya² and A M Sukhov²

¹Physics and Technology Institute, Crimea Federal University, pr. Vernadskogo 4, Simferopol, Republic of Crimea, Russia, 295007
²Samara National Research University, Moskovskoe shosse 34A, Samara, Russia, 443086

e-mail: sukhov@ssau.ru

Abstract. This paper considers routing in a hierarchical space based on the principle of greedy forwarding. In order to avoid overloading the hierarchical tree, we suggested to take into account horizontal communication channels based on the cluster principle. We introduced notion of a cluster and the metric for describing the level of the intercluster transition. The modernised method of greedy forwarding involves the transfer of data with a deviation from the main hierarchical tree.

1. Introduction

New technologies of computer networks, such as the Internet of Things, require new approaches to routing. The main problem with routing protocols is the rapid growth of routing table sizes [1, 2]. Global routing tables must be stored on each BGP router of each autonomous system. The sizes of the BGP tables, in spite of compression measures, represent the biggest problem of global routing.

Limiting the size of these tables or completely rejecting them is the goal of many modern routing studies [3]. The main results in this field have been obtained by developing technologies for self-organising networks. Particular mention should be made of the method of greedy forwarding and its variations, which do not require routing tables [4, 5].

In this paper, a cluster-based routing technology will be introduced that complements hierarchical routing based on the method of greedy forwarding. When referring to the hierarchical system, we mean a set of nodes with the possibility of organising communication between them. For this system, we can build a hierarchical connection graph with the selection of the central node and the child nodes. All other connections between nodes, except for hierarchical ones, are ignored.

Based on the hierarchical approach, it is suggested that different types of addressing be built, relying on the hyperbolic transformation [6, 7]. Routing between nodes in such spaces is based on the principle of greedy forwarding [8, 9]. As a distance, different functions are selected, starting from the distance in the hyperbolic space and ending with the number of hops in the hierarchical graph.
Numerous experiments on building a route in hierarchical space have shown that metrics for greedy advancement are almost completely non-functional. More than 95% of the routes are laid on a tree-like hierarchical connection graph [10]. More specifically, the possibility of transitions between different branches of a tree-like hierarchical structure is extremely small. In order to use the transition mechanism, it is necessary to modernise the notion of distance between nodes. To achieve this, in the present paper we propose the application of the cluster approach.

2. Addressing problems
We previously proposed a new system of addressing in decimal form [11], which is easily built based on a hierarchical graph. The process of assigning addresses is illustrated, and is realised in the form of successive processes of determining the central point, partitioning into neighbourhoods, and constructing a tree. However, the proposed bit address form has significant drawbacks and must be improved. Previously, such addressing has been proposed in [12], but this addressing is not based on hyperbolic transformation, and routing has not used the principle of greedy forwarding.

In this section, an attempt will be made to find a new bit form for recording the address and for estimating the address length and the size of the address space to fully cover the needs of technologies for the Internet of Things. The proposed form of writing an address \( m_1.m_2...m_i \) with zero as a separating sign requires \( N \) bits

\[
N = m_1 + m_2 + ...m_i + i, \tag{1}
\]

which is a very uneconomical form of recording.

In order to shorten the length of the address, we could write down the constituent parts of the address \( m_k \) in the binary system. Following this, to write each component of the address \( m_k \), an \( N_k \) bit is used

\[
N_k = \lfloor \log_2 m_k \rfloor + 1 \tag{2}
\]

and rounded to the nearest whole number from the top.

However, this approach does not consider the presence of separating signs between different parts of the address hierarchy. If for each address component a fixed bit number were allocated, as in the IP address space, then the introduction of special signs for the delimiters would not be required. To reduce the total length of the address, a variable number of bits is used to describe the constituent part of the address; because of this, it is necessary to transfer the exact place of its beginning.

It should be noted that there are two ways to transfer data related to the beginning and end of the constituent part of the address. The first method involves writing a separating sign using a fixed bit set. This set should be excluded from the bit form of the constituent part of the address, which leads to a significant decrease in the address space.

The second method involves transferring the bit length of the constituent parts of the address. In the IP routing system, this role is played by the "Netmask" parameter. Said parameter shows how many bits in the address remain unchanged; it also sets the parent node and describes the corresponding cluster. In our case, we can transfer the number of bits \( p_k \) that is equal to the number of bits \( N_k \) of the corresponding value of the hierarchical address \( m_k \), where \( p_k = \lfloor \log_2 N_k \rfloor + 1 \).

There is also a combined way of transmitting information about the length of the constituent part of the address. The bit form of the record assumes the use of several blocks of fixed length. The length of such a block can be three or four bits, and two service bits are allocated before the constituent part of the address to determine the length of this address part (see Table 1). Thus, the minimum address length is six bits, and the maximum length is 18 bits, if the elementary block is set at four bits.
Table 1. Example of a hierarchical address record.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 11 | m_1 (16 bits) | 10 | m_2 (12 bits) | ... | 01 | m_{i-1} (8 bits) | 00 | m_i (4 bits) |

Now we can begin to estimate the bit length of the address that will be required for the needs of IoT technologies. The first step is to estimate the depth of the address space. To achieve this, we divide the address space into two parts, namely global and local ones. Global addresses serve to bind a local segment to a global network; this role was mainly performed by IPv4 addresses, but they are long gone, and so this address space was extended by 2 bytes (16 bits) using NAT technology.

The depth of the hierarchical address space can be estimated since the hop number of the ping command is limited to 30. Appropriate selection of the central node can reduce this number by half. As a central node, it is advisable to select the most popular traffic exchange point located in New York, USA. This choice makes it possible to set a limit at 15 levels of hierarchy to reach each household or another local node, to which local networks will be added to meet local needs.

The maximum length of a hierarchical address with this approach can be estimated at 60 bits, but this is without dividing into hierarchy levels. Given such separation, it is still approximately 30 bits. This is more than the existing IPv4 address space. If we discuss the average length of the address, it will be much shorter. The greater the distance from the central node, the more the length of the constituent part of the address will decrease. The absolute number of addresses will be up to 32 bits. Another plus of this routing is the absence of huge routing tables.

For the local delivery of information, the number of hierarchy levels can vary, as can the number of devices at each level. Thus far, three hierarchy levels seem to be sufficient, with eight bits per level. However, the advantage of this approach is that the address length is not fixed and we can add hierarchy levels as needed and change the number of devices at each level.

3. Basic provisions of intercluster routing

When talking about the cluster we mean the aggregate of the device and all its child nodes (see Figure 1). On a hierarchical graph, this is a separate skeleton branch starting with the parent node. In our notation, a cluster can be written as a parent node $m_1.m_2...m_i$.

The task of defining a route between two nodes can be completed in several ways. The first of these is the standard method of greedy forwarding, described in [1]. According to this method, we pass the packet to that of the neighbouring nodes, which is closer to the destination node by the tree hierarchy. A promising direction of the route is also the search for a direct transition between two clusters, tied to the beginning and end of the route.

In the standard method of greedy forwarding, we always move up the hierarchy, while the cluster method is based on finding possible routes through the child nodes. The question is how such transitions can be found without significantly increasing the computational complexity of the algorithm and the connection establishment time. An illustration of the construction of an alternative route is shown in Figure 1.

The variant of assigning addresses, which we described in [8], imposes strong restrictions on intercluster transitions. In the first stage of assigning addresses, we cut the neighbourhood from the beginning of the route, and at the second stage clusters are built. Therefore, an intercluster transition can occur between nodes of the same neighbourhood or neighbouring ones. If the node is at the bottom of the older cluster, then it can be connected to the node of the junior cluster of the same neighbourhood or the higher neighbourhood. The nodes of the junior cluster can be associated with the nodes of the older cluster of the same neighbourhood or with a decrease
A cluster is called a senior one if its first number after the matching part of the address is less than that of the other cluster. More specifically, of two clusters \( m_1\ldots m_k, m_{k+1}\ldots m_i \) and \( m_1\ldots m_k, n_{k+1}\ldots n_l \) are older than the one with the smaller \( k+1 \) term. Thus, the node \( m_1\ldots m_k, n_{k+1}\ldots n_l \) from the older cluster can be associated with nodes \( m_1\ldots n_k, n_{k+1}\ldots n_l \) or \( m_1\ldots n_{k-1}, n_{l+1}\ldots n_l \) from the junior cluster.

To find intercluster transitions, special algorithms must be developed. Schematically speaking, it is possible to present the structure of intercluster transitions as follows:

The vertical structure of the cluster links completely repeats the hierarchical structure seen in Figure 1. The most important thing is the accounting of horizontal (intercluster) transitions and their use in routing. This approach will relieve the hierarchical structure, especially its upper levels, while also providing additional routing stability.

The number of intercluster transitions is limited, and such transitions must occur less frequently the further the node is from the centre. Therefore, the routing algorithm must first record such transitions.

Each intercluster transition can be compared to its level. The intercluster transition level will
be defined as the number of route hops along the hierarchical structure between two neighbouring
nodes. More specifically, for two nodes \( m_1 \ldots m_k \ldots m_{k+1} \ldots m_l \) and \( m_1 \ldots m_k \ldots m_{k+1} \ldots m_l \), the intercluster transition level \( Y \) is equal to

\[
Y = i + l - 2k
\]  

(3)

Note here that \( i \) and \( l \) cannot differ by more than 1, as shown above.

Particular attention should be paid to the search for intercluster transitions on the first four
levels of the hierarchy, as shown in Figure 2. These levels are the most loaded, and for them it
is necessary to use alternative intercluster routes. Therefore, it is also necessary to store, in a
special way, information about transitions for which \( k \) from equation (3) is less than or equal to
3.

We can propose the following algorithm for accounting for such intercluster transitions. Each
node \( m_1 \ldots m_2 \ldots m_4 \) polls its neighbours about addressing. For each neighbour, the intercluster
transition level is calculated and the common parent cluster \( m_1 \ldots m_k \) and the number \( k \) are
located. If the number \( k \leq 2 \), then information about such a transition and its source should
be stored on the parent nodes of the clusters \( m_1 \ldots m_{k+1} \).

Thus, all nodes of the first three levels of the hierarchy will store information about possible
intercluster transitions. Based on this information, a routing table can be built. This table will
bypass the most downloaded top levels of the hierarchy. The size of such a table will be severely
limited due to the limited number of nodes of the first three levels of the hierarchy.

4. New metrics for greedy forwarding

In order to enable intercluster transitions, it is necessary to use horizontal links that are not
included in the hierarchical tree. To achieve this, there is a need to propose new metric functions.
The first candidate for the role of the new metric function is the Poincar metric, which is
calculated as the hyperbolic distance between the two nodes \( z_1 \) and \( z_2 \).

\[
d(i,j) = arccosh \left(1 + \frac{2|z_i - z_j|^2}{(1 + |z_i|^2)(1 + |z_j|^2)}\right)
\]  

(4)

Note that for a hierarchical address \( m_1 \ldots m_2 \ldots m_i \), we can write its polar coordinates

\[
r_i = 2^i - \frac{1}{2^i}
\]  

(5)

\[
\varphi_i = 2\pi \left(1 - \frac{1}{2m_i-1}\right) + \frac{2\pi}{2m_1} \left(1 - \frac{1}{2m_{i-1}}\right) + \cdots + \frac{2\pi}{2m_1 + m_2 + \cdots + m_{i-1}} \left(1 - \frac{1}{2m_{i-1}}\right)
\]  

(6)

Consider the routing between nodes \( z_1 = 2.1.2 \) and \( z_2 = 3.2.3.2.3 \), for this we find the
hyperbolic distance from node \( z_1 \) to the nearest neighbours of node \( z_2 \), nodes \( z_3 = 3.2.3.2.3 \) and
\( z_4 = 3.2.3.2.3.3 \). Note that node \( z_3 \) is located higher in the hierarchy. Let us write the polar
coordinates for these nodes, which are calculated according to the equations given above.

\[
r(z_1) = 2^2, \varphi(z_1) = \frac{9}{8}\pi
\]

\[
r(z_2) = 104/61, \varphi(z_2) = 143471/8192\pi
\]

\[
r(z_3) = 31, \varphi(z_3) = 3455/512\pi
\]

\[
r(z_4) = 15/512, \varphi(z_4) = 15883/16384\pi
\]

For the method of greedy forwarding it is necessary to compare the distances \( d(z_1, z_3) \) and
\( d(z_1, z_4) \). Accordingly, the data packet will be transferred from node \( z_2 \) to node, for which the
distance from \( z_1 \) is less. In our case, \( d(z_1, z_3) < d(z_1, z_4) \). Such a metric obviously does not
provide either a shortest route or a movement along the hierarchy.

Let us consider one more possible routing upgrade based on the method of greedy forwarding,
which can lead to the downloading of communication channels that are not part of the hierarchy.
The classic principle of greedy forwarding involves sending a packet with data to the node that is closest to the destination in the hierarchical addressing system. In this case, intercluster transitions occur relatively rarely, and represent no more than 5% of the total number of transitions. However, it is possible to modernise the scheme as follows:

- If there are several nodes of the top level of the hierarchy among the neighbours of this node, then in the first stage we consider a metric function. This is equal to the number of transitions in the hierarchical tree to the end node.
- A data packet is transferred to the node whose metric function is minimal.
- If the minimum of the metric function is the same for several nodes, then the packet is passed to the node with a deviation from the hierarchical tree. For example, a packet must be transferred from node 4.3.2.2 to node 4.3.1 in the event that its metric value coincides with node 4.3.2.

Note that the last node is located on a hierarchical tree.

5. Conclusions
In this paper, routing is considered based on the principle of greedy forwarding in hyperbolic space. To prevent overflow of the channels of the higher hierarchy levels, it is suggested that the cluster approach be used. The definition of cluster is given and the directions of intercluster transitions are described.

For each node and its neighbours, it is suggested that the metric of the intercluster transition be found. If this metric describes an intercluster transition at the first three levels of hierarchical addressing, then the parent node of the cluster maintains a special account of such transitions and their sources.

The problem of writing hierarchical addresses in bit form is discussed in detail. It is proposed to use a variable-length address in which each component of the address contains up to four blocks of three or four bytes.

New metrics for greedy forwarding have been analysed using calculations for specific nodes. The modernised method of greedy promotion involves the transfer of data with a deviation from the main hierarchical tree.

6. References
[1] Krioukov D, Fall K and Brady A 2007 On compact routing for the internet ACM SIGCOMM Computer Communication Review 37(3) 41-52
[2] Narayan H, Govindan R and Varghese G 2003 The impact of address allocation and routing on the structure and implementation of routing tables Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications 125-136
[3] Agafonov A and Myasnikov V 2016 Method for the reliable shortest path search in time-dependent stochastic networks and its application to GIS-based traffic control Computer Optics 40(2) 275-283 DOI: 10.18287/2412-6179-2016-40-2-275-283
[4] Finn G G 1987 Routing and Addressing Problems in Large Metropolitan-Scale Internetworks ISI Research Report. 87-180
[5] Bose P, Morin P, Stojmenovic I and Urrutia J 2001 Routing with guaranteed delivery in ad hoc wireless networks Wireless networks 7(6) 609-616
[6] Kleinberg R 2007 Geographic routing using hyperbolic space INFOCOM 26th IEEE International Conference on Computer Communications 1902-1909
[7] Cvetkovski A and Crovella M 2009 Hyperbolic embedding and routing for dynamic graphs INFOCOM IEEE 1647-1655
Acknowledgements

The work falls within the public tasks of the Ministry of Education and Science of the Russian Federation (2.3129.2017/4.6) and was carried out with the support of grant RFBR 16-07-00218a.

[8] Papadopoulos F, Krioukov D, Bogu M and Vahdat A 2010 March. Greedy forwarding in dynamics scale-free networks embedded in hyperbolic metric spaces *INFOCOM Proceedings IEEE* 1-9

[9] Papadopoulos F, Psomas C and Krioukov D 2015 Network mapping by replaying hyperbolic growth *IEEE/ACM Transactions on Networking (TON)* 23(1) 198-211

[10] Chemodanov D, Esposito F, Calyam H and Sukhov A 2018 REBATE: a REpulsive-BAsed Trafc Engineering for Dynamic Scale-Free Networks Submitted to ICNP

[11] Timofeeva S V, Sarkisian D M, Sukhov A M and Zuev S A 2017 Greedy forwarding for hyperbolic space in MANET 25th Telecommunication Forum (TELFOR) 1-4

[12] Vorobienko P, Tikhonov V, Smirnov I and Soshina U 2010 Algorithm for dynamic addressing of telecommunications network objects *Digital Technologies* 8 11-19 (in Russian)