Graph Contraction based Self-organizing Networks for Future Internet

Tae Hwan Kim
National Institute for Mathematical Sciences
Daejeon, Korea
thkim@nims.re.kr

Jung Yeun Lee
National Institute for Mathematical Sciences
Daejeon, Korea
jungyeunlee@nims.re.kr

Joon Heo*
National Institute for Mathematical Sciences
Daejeon, Korea
heojoon@nims.re.kr

Sang Heum Cho
Korea Advance Institute of Science and Technology
Daejeon, Korea
ray5640@kaist.ac.kr

Abstract—**This paper is a proposal to arrange the way in which nodes in a future Internet are address according to a hierarchical scheme based on "graph contraction", in which nodes are organized into a hierarchical abstracted graph. We describe a new network architecture scheme─graph contraction based self-organizing networks and topology-aware addressing. Our proposed network design principle abstracts large-scale networks while preserving the topological characteristics of nodes. It also provides a way to define topology-aware address. This network architecture scheme can be a solution to the challenges of scalability, mobility, and the issue of rapid increase in routing table size through aggregatable address information. By means of a three-dimensional topology viewer and a self-organizing network simulator, we analyzed the self-organizing scheme and topology-aware address allocation process of our architecture.**

Keywords—Component; Future Internet; Graph Contraction; Self-organizing networks; Topology-aware architecture.

I. INTRODUCTION

Even though the Internet is such an essential infrastructure, there are many concerns about architectural aspects of the current Internet such as scalability, mobility, security, heterogeneity, and manageability. In recent years, many research organizations and standards bodies have studied future Internet, which is anticipated to provide functionalities and services beyond the limitations of current Internet technology. Without regards to practical progress, they sought to make intellectual progress by focusing on clean slate designs [14]. The clean-slate approach enables the research community to investigate how to address these problems architecturally, even if the changes themselves cannot be incrementally deployed. These clean slate efforts begin to address architectural considerations such as topological aspects because they realize that most of the concerns of the current Internet are caused by lack of topological consideration [4, 9]. For example, in the current IP addressing scheme, network topology is not taken into consideration. The address structure ought to be aggregatable and should contain topological information about the communication nodes. An IP address is a meaningless series of bits that simply expresses the smallest topological region by subnet prefix. We are now beginning to understand the relation between network topology and address. We believe that by focusing on the structure of topological addressing many issues associated with future Internet may be resolved. Consequently, we have tried to find a new addressing structure that can contain the topological information of the physical network. In this paper, we introduce a proposal to arrange the way in which nodes in a future Internet are address according to a hierarchical scheme based on "graph contraction", in which nodes are organized into a hierarchical abstracted graph. We believe that this type of architecture can solve issues such as scalability, mobility, and increases in the size of routing tables being faced by the current Internet. We hope to advance the dialog on future Internet research by reconsidering this relation and observing that although relatively simple, the changes are meaningful and would vastly improve the Internet's evolvability.

Returning to the fundamentals of network systems, we design a network architecture focusing on the two most basic components: *nodes* and *connection* between nodes. All infrastructural constraints of the current Internet (such as domain, AS, gateway, prefix, and addressing) are ignored. Under these assumptions, we discuss two important network architecture matters—design principle and address structure. In particular, we focus on how to organize and govern a large-scale complex network regardless of any topological constraints and how to assign an address that connotes the topological information of nodes. To achieve these objectives, we propose graph contraction based self-organizing networks. This flexible network architecture automatically organizes the network topology and constructs a hierarchical network structure using graph contraction. The network organizing principle—graph contraction—abstracts the network topology, contains topological characteristics, and defines topology-aware addressing. Thanks to this flexible architectural characteristic, network scalability is no longer a problem. Further, the topology-aware addressing defined by the self-organization supports the aggregation of routing information.

The remainder of this paper is organized as follows: Section II introduces previous works closely related to our approach and compares them with other research efforts. Section III gives a detailed account of our proposed network design principle and the topology-aware addressing structure using an illustrated example. Section IV describes the actualization of our network architecture in a simulator and analyzes the results obtained. Finally, we discuss issues remaining and conclude this paper by giving an outline of future works in Section V.

---

[*] J. Heo is the corresponding author.

978-3-901882-50-0

II. Related Work

A. New Addressing Structure for Future Internet

Recently, the ID/locator split address concept has been extensively discussed as one of the solutions that can support scalability and mobility in future Internet [8, 10, 11, 12, 23]. Currently, an IP address is used in the network layer as a locator to locate the destination host and is also used as an identifier in the transport and application layers to identify the host or communication sessions. This dual use of IP addressing as ID and locator makes it difficult to design efficient solutions for mobility, multihoming, renumbering, and scalable routing because such solutions require the capability of changing locators without changing the ID. The ID/locator address split concept is aimed at eliminating the above problems and limitations by separating the two roles. The concept has gained increasing popularity: with a number of approaches presently in existence [5, 7, 12, 20]. It has also been recently introduced in the standardization activities of ITU-T for integration in future network architectures [13, 23]. We agree with the ID/locator split concept but believe that the locator, which indicates node location and functions as routing information, should be more meaningful. Many of the current approaches pay full regard to node identification (for example, XIA [1] and MobilityFirst [11] adopt the ID-based cryptosystem [6, 22] for node identification) but relatively less to the locator. Teraoka et al. [8], as well as Jung and Koh [10], do not contemplate a locator property; they simply use IP address or prefix as a locator without any reconsideration. As we mentioned above, the locator is primary information for routing; therefore, the locator property should be taken into consideration in order to avoid making the same mistake as was made with the current Internet.

B. Understanding Network Topology

The previous topological considerations in networks can be easily found in [4, 9] and there is effort to organize topology automatically in sensor networks and mobile ad-hoc networks. Since they have less infrastructure supporting communication, they have been trying to solve the technical issues by organizing the network topology themselves. Various research studies that focus on organizing the network currently exist. A cluster-based network [16, 18, 24] is a typical approach for organizing networks in these fields. It provides a method for building and maintaining hierarchical network topologies in ad hoc networks. Krishna et al. [19] presented a clustering algorithm in which clusters are formed by a clique. The basic concept underlying the clique-based clustering mechanism is quite similar to our graph contraction approach. However, one thing that sets us apart from them is that the goal of clustering is to cluster nodes but ours is creates abstracted graph which contains connectivity of original graph. In addition, complex and heavy processes are required to retain clique-based clustering graphs with node dynamics. Moreover, in clique-based clustering a node can overlap, which results in nodes having multiple locator information.

To simplify and organize networks, various mechanisms, such as hypergraph [21], clique-based clustered network [19] and, hierarchical structure [2, 15], have been proposed. However, because they assume small networks and target specific topologies, these mechanisms are not suitable for the Internet (they are not robust enough to tolerate network dynamics and scaling). In addition, using them to build hierarchical network structures would be quite inefficient as they preserve the constricted organizing rules such as full mesh, two steps graph, and clique-based clustering. To overcome the limitations of the proposed mechanisms, we propose graph contraction based self-organizing networks. The proposed networks can be organized into any type of topology and can also scale. In the next section, we introduce the architectural details: network design principle, self-organizing scheme, network organizing example scenario, and topology-aware addressing.

III. Architectural Details

It is essential that the appropriate topology be considered in the design development process of network architecture. Network topology can be related to almost all components comprising the network architecture: node location, physical capabilities of the network devices, and peripheral settings. Potential network topologies are made according to these components and the network topology can be intricate if there is no design principle. Current Internet topologies are primarily influenced and formed according to the policies of ISPs and in line with economics. To the best of our knowledge, there is a few scientific approaches to understand network topology in network organization and deployments [4, 9]. Since the Internet is a collection of thousands of smaller networks which has various forms of topology, we would like to find a principle to organize and abstract any forms of topology over holding the topological characteristics of the physical nodes. To overcome the limitations of previous topology organizing mechanisms, we thus employ graph contraction to obtain the simple form. This is a reasonably simple technique and can be applied to abstract complex networks without any topological constraints. It can abstract any form of topology and any scale network.

A. Design Principle: Graph Contraction

Graph contraction is made up of two contractions: node contraction and edge contraction. Note that our contraction is different from the graph contraction in graph theory textbooks. Our contraction is based on the design principle of the network architecture. We first pick a node that has the maximum degree in its neighborhood (we call this node the core node), then contract the neighborhood into this core node. We regard the set of contracted nodes (the closed neighborhood and the core node) as a new node in the upper level topology. We call this process neighborhood contraction. Graph contraction is a process that produces a new graph, upper level topology, by iterating the neighborhood contraction process. We adopt the neighborhood graph contraction with maximum degree node since it can effectively contract the form of any scale-free network that has a hub (we will explain in detail the reason why we adopted neighborhood graph contraction below). Other contraction rules such as two-step contraction, dominating set contraction, and full mesh graph contraction can be applied according to the purpose of the network.

In the neighborhood-based graph contraction, we are given an undirected graph $G = (V, E)$ representing a network topology, where the vertices are the nodes in the network and the edges are the communication links. Let u be a core node and $N[u]$ be a closed neighborhood of u. The node contraction $NC(u)$ contracts the edges between u and $N[u]$ and makes a new vertex V' of G' that contains u and $N[u]$. In Fig. 1 for example, core node e and its neighborhood c, d, f, g are contracted into the single vertex β in the third graph. (We call β the parent node and c, d, f, g its child nodes.) After node contraction, the edges between $N[u]$ and the other nodes except u are contracted as a single edge if multiple edges existed between them. This edge contraction removes the multiple edges while maintaining the connectivity between nodes. For example, the edges ad and ac are converted into double edges $\alpha\beta$ after node contraction. These parallel edges are contracted as a single edge. If a node is left in G, we choose another core node among the nodes that have not been contracted and contract the neighborhood. The graph contraction process is repeatedly carried out until there is no un-contracted node in G. The graph contraction process eventually produces a graph G' in which adjacent nodes u and $N[u]$ are replaced with a single node and its connections are maintained. Fig. 1 depicts our graph contraction process. The process is defined as follows:

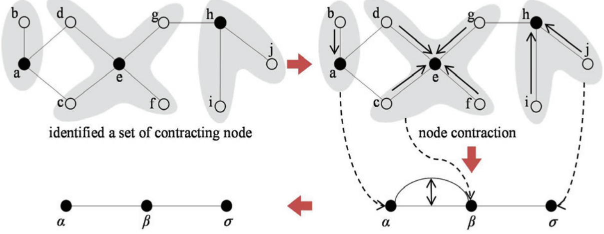


Figure 1. Neighborhood-based graph contraction

GraphCont($G = (V, E)$) :

1. Identify a set of contracting nodes: u and $N[u]$.
2. *Node contraction:* Remove the edges between u and $N[u]$ and replace u and $N[u]$ with a single vertex V' in G'.
3. *Edge contraction:* Contract the edges between $N[u]$ and other nodes except u if multiple edges exist between them and replace single edge E' in G'.
4. If all nodes in G are not covered then GraphCont($G' = (V - V', E')$).
5. If all nodes in G are covered then GraphCont($G = (V, E)$) = $G' = (V', E')$.

The number of parent nodes varies greatly depending on how the core nodes are selected. In this work, we state that contraction is performed within the nodes that are in a neighborhood relationship, and the node that has maximum degree in the neighborhood is chosen as the core node. This means that a large number of nodes can be contracted at once without the loss of topological characteristics. All nodes in the disjoint node set that are contracted as one node are regarded as being directly connected in the set. For example, nodes c, d, e, f, and g are contracted to the single node β and can be considered as being directly connected to each other. Therefore, this intra-routing protocol within the disjoint node set is very simple, since it can be considered that there is full mesh connectivity among the contracted nodes.

The new graph G' can also be contracted, leading to a multi-level hierarchy, with each graph in each level containing the topological characteristics of the low level graph. Consequently, the graph contraction process causes network complexity to fall into a desired range through the reasonable union of components. In the following subsections, we explain the self-organizing algorithm based on the graph contraction process with the aid of an illustrative example.

B. Graph Contraction based Self-organization

In this section, we explain the self-organization network process based on graph contraction and the allocation methods in topology-aware addressing. The following assumptions are made:

- *i)* The networks are assumed to have no isolation nodes that are connected graphs. As in practical environments, the network graph is assumed to be undirected unless stated otherwise.
- *ii)* Infrastructures are disregarded in this assumption, e.g., border gateway routers and sub-network structures; only nodes with global unique identifiers and their connectivity exist.
- *iii)* All nodes have equivalent roles and they only relay data to other nodes.

The self-organizing scheme starts from a randomly chosen node. From this node, it automatically covers the entire network topology and builds a hierarchical structure containing multiple contracted graphs. (As a result, we call this scheme a self-organizing scheme.) The basic outline of the self-organizing scheme is as follows:

1. A node is randomly chosen in the network and the scheme specifies a node that has maximum degree in its neighborhood as the core node. If there is no node in its neighborhood with a degree greater than itself, it becomes the core node.
2. Graph contraction is performed with the core node chosen.
3. Steps 1 and 2 are performed repeatedly until the entire network is covered and then the level 1 topology is created.
4. Graph contraction is performed on the newly created topology (level 1 topology).
5. Steps 1, 2, 3, and 4 are performed repeatedly until the newly created topology has only one node. This node then represents the entire network.
6. The top layer node assigns its ID itself and allocates IDs to the nodes in the layer immediately below it (its child nodes). These IDs are locally unique within a subset of their parent node. (Note that only physical nodes in the lowest layer have globally unique IDs.)
7. The nodes in each lower layer allocate IDs to its child nodes.
8. Step 6 is performed repeatedly for each layer until the penultimate layer from the lowest level (physical layer).
9. The lowest nodes, physical nodes, are allocated topology-aware addresses—a sort of parent node ID.

Random selection in the first step, node choice, does not yield optimal topology abstraction. However, this does not make a substantial difference through the multiple graph contraction processes without node choice optimization. (We prove this supposition in Section IV by our experimental results.) The proposed self-organizing algorithm using graph contraction is outlined in Fig. 2. Fig. 3 depicts an illustrative self-organization scenario.

```
Graph Contraction Algorithm
%Input: Graph $G=(V,E)$, where $V=\{v_1,v_2,\dots,v_m\}$
%Output: Graph $A(G)=(V',E')$
fun $graphCont(G=(V,E))=$
let
    $W \leftarrow V$
    For $(k=0, W \neq \phi, k \leftarrow k+1)$
        $v'_k = nodeCont(v,W)$ for some $v \in W$
        $V' = V' \cup \{v'_k\}$
        $E' = E' \cup (edgeConn(v'_k))$
        $W \leftarrow W - \{v \mid v \in v'_k\}$
Return: $A(G)=(V',E')$
```

```
Node Contraction Algorithm
%Input: Graph $G=(V,E)$ and $v \in W \subset V$ is selected vertex
%Output: $c = nodeCont(v,W) = (C,cc)$,
%          where $C$ is a contracting node and $cc$ is a core node in $C$
fun $nodeCont(G=(V,E),v)=$
    $S \leftarrow N[v] = \{u \in W \mid uv \in E\} \cup \{v\}$
    $N \leftarrow N[v]$
    $c \leftarrow (N[v],v)$
    while$(S \neq \{v\})$
        $N[x] = \{u \in W \mid xu \in E\} \cup \{x\}$, where $x \in S - \{v\}$
        if $n(N[x]) \geq n(N)$, $c \leftarrow (N[x],x)$ and $N \leftarrow N[x]$
        $S \leftarrow S - \{x\}$
Returen: $c$
```

```
Edge Contraction Algorithm
%Input: Graph $G=(V,E)$ and $V'=\{v_1, v_2,\dots,v_{k-1}\}$
%Output: $AE = edgeConn(v_k)$ = the set of edges incident to $v_k$
fun $edgeConn(v_k)=$
    $AE = \varnothing$
    For $i=1$ to $k-1$
        If $\exists\, u \in v_i, w \in v_k$ such that $uv \in E$,
        $AE \leftarrow AE \cup \{v_i v_k\}$
Returen: $AE$
```

Figure 2. The proposed graph contraction based self-organizing algorithm

C. *Topology-aware Addressing Structure*

Perkins [3] observed that aggregating routing information is the key to Internet scalability. This connotes that the addressing structure should be able to aggregate topologically. This key solves not only the scalability problem but also the drastic growth of routing tables. In fact, the current IP addressing structure is quite irrelevant to the Internet topology, although it can express the smallest topological region by subnet prefix. In addition, the BGP is designed with a flat routing structure. Every AS treats every other AS equally, which brought about the exponential increase in the size of the routing tables and impaired BGP's scalability. These complications encouraged us to propose topology-aware addressing, which denotes topology characteristics and provides aggregatable routing information. With the self-organizing scheme, complex networks can have a hierarchical structure and address structures denoting this multi-layer topology.

Fig. 4 describes the addressing structure consisting of node identifier and locator, which is topology information designated by a sort of parent node identifier. For example, in the scenario depicted in Fig. 3, the address of node 26 is "$26 \colon a.X.Z$." This addressing style represents a hierarchical topology and connotes aggregatable routing information.

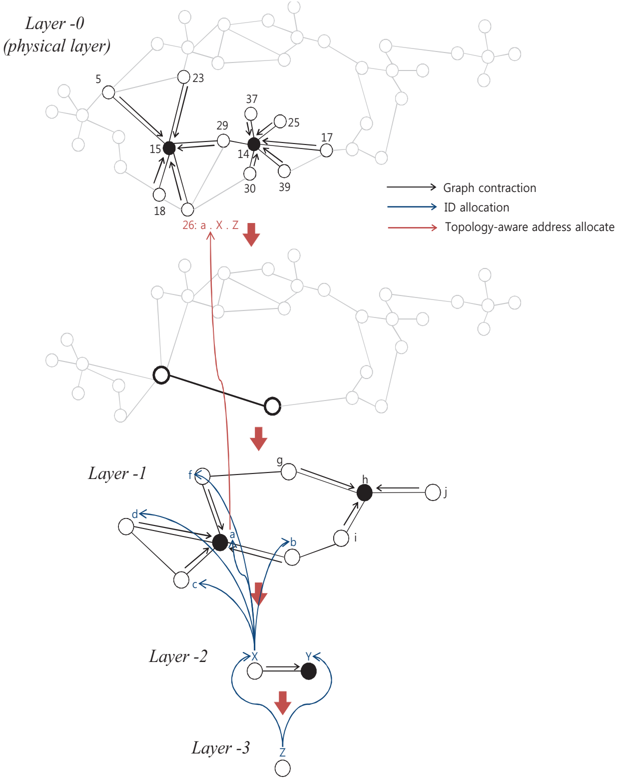


Figure 3. Illustrative self-organizing scenario

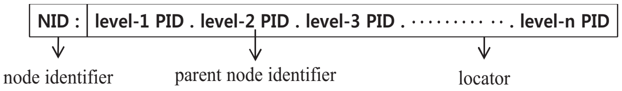


Figure 4. Proposed topology-aware addressing structure

The locator is a representation or notation of the types of parent nodes identifier in the hierarchy. As mentioned above, a parent node identifier (PID) in the locator is unique in the neighborhood. The locator provides information on how to reach the node. This is temporal, and therefore due to node mobility or network re-organization. On the other hand, the node identifier (NID) is independent of location and is used to identify communicating end points. Both NID and PID may exist in the form of a character string or bit-stream and can be adopted by the ID-based cryptosystem [6, 22] for secure communication. After self-organization, each node in the hierarchy can be assigned a hierarchical address that indicates its position in each layer of the hierarchy.

IV. ANALYSIS OF OUR ARCHITECTURE

In this section, we describe the simulator used to analyze our proposed network architecture and our experimental results. The simulator was implemented using 3D graph visualization tool Ubigraph [25]. It was able to describe the self-organization and address assignment processes. Taking into consideration practical Internet environments, we performed the experiments based on the scale-free networks created by the SFNG function [17] in Matlab and were able to change the scale of the networks and the mesh degree. In the next subsection, we describe the simulator and our experimental results.

A. The Self-organizing Network Simulator

```
c1Fis uncontracted as it self. It becomes ucbl
pJis uncontracted as it self. It becomes f066
cOis uncontracted as it self. It becomes m7Vp
PawaJis uncontracted as it self. It becomes OMZm7
E20ikis uncontracted as it self. It becomes HT
XWis uncontracted as it self. It becomes WNPR
Y1iis uncontracted as it self. It becomes i5V2
yGbis uncontracted as it self. It becomes qFebj
qSis uncontracted as it self. It becomes fK
```

Index	Node	Degree	Contracting Node	Address
1	NGRL	31	GrH7	NGRL:36jXa.GrH7.0BsDf.ls46
2	3cQ4	23	GrH7	3cQ4:36jXa.GrH7.0BsDf.ls46
3	QFbtK	15	GrH7	QFbtK:LiPR.GrH7.0BsDf.ls46
4	XZ	34	GrH7	XZ:LiPR.GrH7.0BsDf.ls46
5	uQPSC	5	GrH7	uQPSC:36jXa.GrH7.0BsDf.ls46
6	7N	5	GrH7	7N:nObDF.Z8Lt.0BsDf.ls46
7	RLr4	20	GrH7	RLr4:36z.GrH7.0BsDf.ls46
8	1eOjE	7	GrH7	1eOjE:LiPR.GrH7.0BsDf.ls46
9	JX	1	GrH7	JX:Tld.eQ0.0BsDf.ls46
10	NDJ	17	GrH7	NDJ:rPS4L.GrH7.0BsDf.ls46
11	3ox	6	GrH7	3ox:ceNKj.58e1m.0BsDf.ls46
12	nvDyz	2	GrH7	nvDyz:36jXa.GrH7.0BsD

Level	Number of Nodes	Contracting(%)	Completion
Physic...	500	–	Drawn
Level 1	233	53.4%	Completed
Level 2	113	51.5%	Completed
Level 3	6	94.69%	Completed
Level 4	1	83.33%	Completed

Figure 5. The self-organizing network simulator

Fig. 5 depicts the simulator, which comprises a 3D topology viewer and a command window. The viewer describes the network states and all the topology organization and address assignment processes. By means of this viewer, we were able to observe our network architecture and the organizing scheme—how the topology was organized by the self-organizing scheme and how addresses were assigned to the nodes. In addition, the command window gave comprehensive information about the networks—self-organizing operation state, node information, and topology information about each layer. Using this simulator, we evaluated our topology organizing scheme with various network topologies.

B. Experimental Results

We observed the self-organizing process at various scales and using networks with various shapes (100 to a maximum of 5000 nodes of scale-free and mesh networks) and the resultant hierarchical topologies. All experiments were performed a total of 10 times and data recorded. Figs. 6–9 show average data from the experimental results. Since graph contraction is performed from a randomly selected node, each experiment gave slightly different results even though the self-organization was performed on the same topology. The self-organizing scheme constructed topologies with a maximum of six layers in the 5000 nodes scale-free networks. In this case, every physical node was assigned a topology-aware address consisting of seven properties: a node identifier and six parent node identifiers. On the whole, the larger the scales of the networks, the more layers were constructed in the topologies, regardless of the shape of the networks. In addition, the number of layers in the topologies of the mesh networks was less than that of the scale-free networks as the connectivity between nodes was more complex.

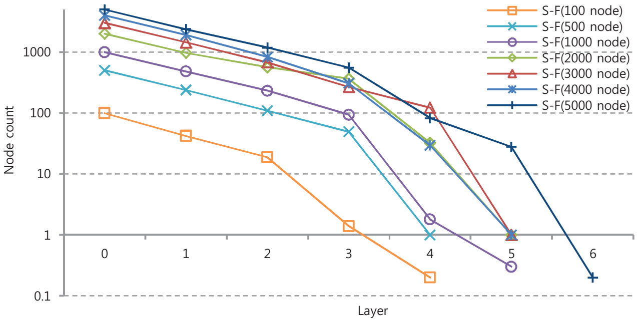


Figure 6. Result of self-organization in scale-free networks

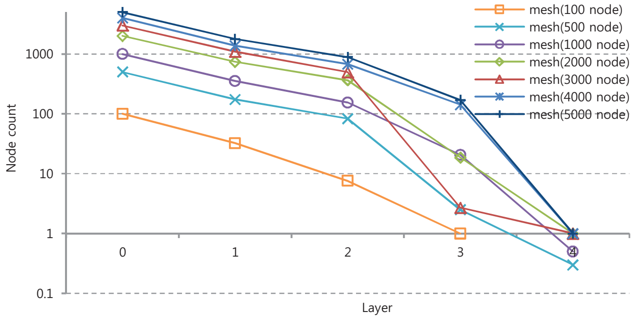


Figure 7. Result of self-organization in mesh networks

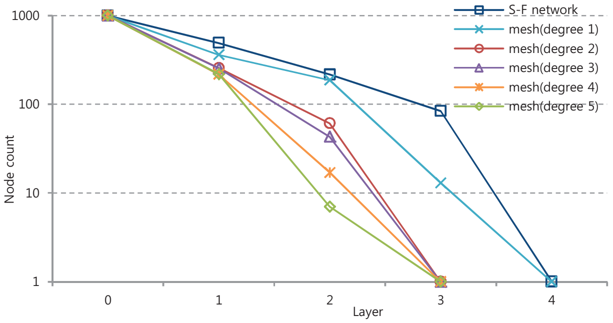
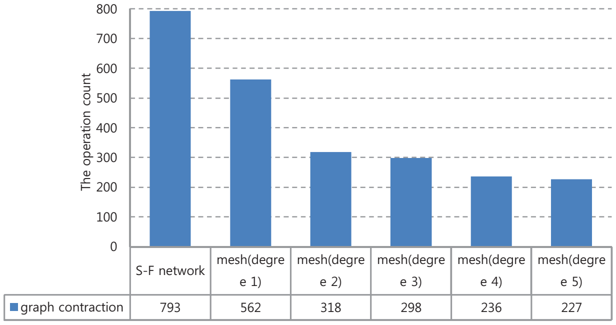


Figure 8. Result of self-organization according to the connection complexity of the network topology



	S-F network	mesh(degree 1)	mesh(degree 2)	mesh(degree 3)	mesh(degree 4)	mesh(degree 5)
■graph contraction	793	562	318	298	236	227

Figure 9. Graph contraction operation count according to the shape of the network topology

To observe the degree to which network complexity affects the self-organizing scheme, we gradually increased the connection complexity in the scale-free networks in terms of the mesh degree. Consequently, we discovered that the more complex the networks got, the lower the number of layers constructed in the topology and the lower the number of graph contraction operations performed.

V. CONCLUSIONS

In this paper, we gave an overview of our proposed graph contraction based self-organizing network scheme. In particular, we focused on the network design principle—self-organizing mechanism and topology-aware addressing. The self-organizing scheme provides a method for organizing a network topology regardless of topological constraints. On the basis of this scheme, topology-aware addresses are defined and allocated to nodes. Topology-aware addressing may solve many future Internet issues and function as the locator property for ID/locator split addressing, which has been extensively discussed as a solution for future Internet. We analyzed our proposed scheme using various network scales and network topologies with a variety of shapes.

This paper introduces a new direction that involves fundamental changes to networks, not just the aspects that are easy to change in the current Internet. We hope that these changes will be applicable to various aspects of network innovation and will bring a new perspective to networking technology. We do not claim that our particular network design scheme is right in all its details since it is an early-stage proposal. Instead, we merely propose it here to initiate a broader discussion on how the goal of architectural innovation can be achieved by the topological approach. There are some possible extensions to our work that may be needed before it can be deployed in the future Internet. In future work, we will enhance the graph contraction rule for other specific networks (in this work, we used a neighborhood contraction for effective topology organization in scale-free networks). We will also examine additional self-organizing schemes for supporting network dynamics and mobility. To communicate between nodes, we used a routing strategy on this hierarchical network architecture using aggregatable topology-aware address. We believe that this interpretation of networked systems from a new angle may lead to a new direction and a fresh possibility for the architecture of future Internet.

REFERENCES

[1] A. Anand, F. Dogar, D. Han, B. Li, H. Lim, M. Machadoy, W. Wu, A. Akella, D. Andersen, J. Byersy, S. Seshan, and P. Steenkiste, "XIA: An architecture for an evolvable and trustworthy internet," in *Proc. 10th ACM Workshop on Hot Topics in Networks*, 2011.

[2] A. Feldmann, L. Cittadini, W. Muhlbauer, R. Bush, and O. Maennel, "HAIR: Hierarchical architecture for internet routing," in *Proc. the 2009 workshop on Re-architecting the Internet*, pp.43−48, 2009.

[3] C. Perkins, "Ad Hoc Networking, "*Addison-Wesley*, 2001.

[4] D. Alderson, L. Li, W. Wilinger, and J. C. Doyle, "Understanding Internet Topology: Principles, Models, and Validation," IEEE/ACM Trans. on Networking, vol.13, no. 6, 2005.

[5] D. Clark, R. Braden, A. Falk, and V. Pingali, "FARA: Reorganizing the addressing architecture," in *Proc. ACM SIGCOMM Workshop on Future Directions in Network Architecture*, pp. 313–321, 2003.

[6] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet Protocol (AIP)," in *Proc. ACM SIGCOMM*, pp.339−350, 2008.

[7] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6 (RFC 5533)," 2009.

[8] F. Teraoka, S. Kanemaru, K. Yonemura, "ZNA: A network architecture for new generation network − Focusing on the session layer," in *Proc. 3th Int. Conf. on Ubiquitous and Future Networks*, pp.309−314, 2011.

[9] H. Haddadi, M. Rio, A. Moore, and R. Mortier, "Network Topologies: Inference, Modeling, and Generation," IEEE Commun. Surveys, vol. 10, no.2, 2008

[10] H. Jung and S. Koh, "Mobile-Oriented Future Internet (MOFI): Architecture and Protocols," Release 2.2, 2010.

[11] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, "MobilityFirst future Internet architecture project," in *Proc. 7th Asian Internet Engineering Conference*, pp.1−3, 2011.

[12] D. Lewis, V. Fuller, D. Farinacci, and D. Meyer, "Locator/ID Seperation Protocol (LISP)," *IETF Draft*, 2010.

[13] ITU-T, "General requirements for ID/locator separation in NGN," *Draft Recommendation*, 2009.

[14] J. Pan, S. Paul, and R. Jain, "A survey of the research on future Internet Architectures," *IEEE Commun. Magazine*, vol. 49, Issue 7, 2011.

[15] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 3, Issue 1, 2004.

[16] M. Bechler, H.-J. Hof, D. Kraft, F. Pahlke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 4, pp.2393−2403, 2004.

[17] M. George, "B−A scale−free network generation and visualization," http://www.mathworks.com/matlabcentral/fileexchange/11947, 2006.

[18] M. L. Lim and C. Yu, "Does cluster architecture enhance performance scalability of clustered mobile ad hoc networks," in *Proc. Int. Conf. on wireless Networks*, 2004.

[19] P. Krishna , N. H. Vaidya , M. Chatterjee, and D. K. Pradhan, "A cluster-based approach for routing in dynamic networks," in *Proc. ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 49–64, 1997.

[20] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host identity protocol (RFC 5201)," 2008.

[21] S. B. Patkar and H. Sharma, "Efficient network flow based ratio-cut netlist hypergraph partitioning," WSEAS Transactions on Circuit and Systems, vol. 4, Issue 6, 2005.

[22] S. Tsuji and T. Itoh, "An ID-based cryptosystem based on the discrete logarithm problem," *IEEE Journal of Selected Areas in Communications*, vol. 7, Issue 4, pp.467−473, 1989.

[23] V. P. Kafle, H. Otsuki, and M. Inoue, "An ID/locator split architecture of future networks," *IEEE Commun. Magazine*, vol. 48, no. 2, 2010.

[24] Y.P. Chen and A. L. Liestman, "A zonal algorithm for clustering ad hoc networks using minimum connected dominating sets," *International Journal of Foundations of Computer Science*, 2003.

[25] Ubigraph: Free dynamic graph visualization software, http://ubietylab.net/ubigraph/, 2008.