Proposal of Data-Centric Network for Mobile and Dynamic Machine-to-Machine Communication

Daisuke MATSUBARA\(^{a)}\), Hitoshi YABUSAKI\(^{1)}\), Members, Satoru OKAMOTO\(^{1)}\), Naoaki YAMANAKA\(^{1)}\), and Tatsuro TAKAHASHI\(^{1)}\), Fellows

SUMMARY  Machine-to-Machine (M2M) communication is expected to grow in networks of the future, where massive numbers of low cost, low-function M2M terminals communicate in many-to-many manner in an extremely mobile and dynamic environment. We propose a network architecture called Data-centric Network (DCN) where communication is done using a data identifier (ID) and the dynamic data registered by mobile terminals can be retrieved by specifying the data ID. DCN mitigates the problems of prior arts, which are large size of routing table and transaction load of name resolution service. DCN introduces concept of route attraction and aggregation in which the related routes are attracted to an aggregation point and aggregated to reduce routing table size, and route optimization in which optimized routes are established routes to reduce access transaction load to the aggregation points. These allow the proposed architecture to deal with ever-increasing number of data and terminals with frequent mobility and changes in data.

**key words:** future networks, M2M, mobility, data-centric network, information-centric network

1. Introduction

Networks operating today are mainly used for voice communication between humans or data communication between computers. Recent advancement and proliferation of network-enabled embedded devices such as mobile phones, vehicle telematic devices, and wireless sensors has opened new possibilities for machine-to-machine (M2M) communication [1], where individual M2M terminals collect real-world data and the data is exchanged inside the network and used by different services or other M2M terminals. The number of these M2M terminals is expected to grow rapidly and will be massive compared to conventional telephone or computer terminals, and the need for micro-mobility, dynamic data updates, non-unicast communication, and constraint on the computational and storage capacity of the terminals provide great challenge to the network. This will be likely to lead to significant change in paradigm, where priorities of the network requirements are altered.

In this paper, we propose a network architecture called Data-centric Network (DCN) [2]–[4] where the network supports routing of dynamic data using data identifiers (IDs). We argue that DCN accommodates the challenges mentioned above, and maximizes potential of M2M communication and its applications by providing an optimal architecture. We also propose route establishment method and route optimization method that will mitigate the number of route information and concentration of transactions, which were critical problems of previous approaches.

In Sect. 2, we define the requirements for M2M communication and propose DCN that is optimized for these requirements. In Sect. 3, we discuss related prior arts and assess the problems it will face when trying to fulfill the requirements stated in Sect. 2. In Sect. 4 we explain the DCN architecture and procedures for the architecture, and in Sect. 5 we show the experiment results and evaluate the results.

2. Requirements and Proposed Architecture

2.1 Requirements for M2M Communication

The requirements of M2M communication are as follows. Note that in this context, the term ‘data’ or ‘data object’ refers to any digital information ranging from primitive sensor data to rich media content. We assume that DCN is used in a large scale network where there are large numbers of terminals, network nodes, and data IDs, and large traffic, frequent moving of terminals and data, and large communication delay for each physical network link are expected.

- **Requirement 1:** Massive number of low-cost, low-function M2M terminals
  More than 50 billion terminals will be connected to the network, in which most of the terminals will be extremely low-cost, low-function terminals such as wireless sensors. The terminals require very simple communication methods and protocols when communicating via the network.

- **Requirement 2:** Simple mobility
  There will be frequent movement, transfer, or replacement of M2M terminals when they are deployed on the field. Mobility will be the norm for M2M terminals, as most of them will be wireless and will be carried by humans or placed on mobile entities such as transportation vehicles. Even when M2M terminals are placed on a fixed location, the transfer or replacement of these
terminals will be conducted by non-experts of networking technology working on the field. This will require the network to perform terminal mobility with little or no reconfiguration to the network or the terminal.

- **Requirement 3: Tracking of dynamic data**

M2M terminals will generate dynamic data, which is data that is added, moved, updated, and deleted frequently by the terminals. It will be difficult to predict how and when the data will be used, and there will be some data that will be generated and stored for a certain amount of time, but will be deleted without any application accessing them. The network is required to support tracking and easy access to the ever changing dynamic data.

- **Requirement 4: Many-to-many communication**

It is expected that multiple services will make use of the data generated by massive number of M2M terminals, hence many-to-many communication, where many terminals disseminate data to many terminals without end-to-end communication between terminals [21], will be the predominant type of communication. The network is required to support different kinds of non-unicast communication such as broadcast, multicast, publish/subscribe, key-value-store (KVS) query.

### 2.2 Proposed Architecture

In order to accommodate the requirements presented above, we propose a network architecture called Data-centric Network (DCN). In conventional network architectures, such as public switched telephone network (PSTN) or the Internet, the main objective was to provide data communication between two terminals. On the other hand, current network is mainly used to access a particular data, irrespective of its location in the network. DCN embraces this change in paradigm and aims to provide optimal architecture for easy and efficient access to the requested data generated from massive number of terminals. For this reason, in DCN the terminal is not aware of the terminal it is communicating with, and just specifies the ID of the data that it wants to access.

Figure 1 shows our view and assumption of paradigm changes of networks in the past and the future. Conventional PSTN mainly handled human-to-human communication such as voice, and the current Internet mainly handles human-to-machine communication such as the web. DCN optimizes for M2M communication, in which terminals and applications communicate in many-to-many manner.

There is another important characteristic in DCN which is reduction of traffic using data cache inside the network. In this paper, we focus on the aspect of communication using the data ID which is strongly relevant with requirements of M2M service.

The items below show the comparison between the current internet protocol (IP) network and DCN based on the requirements presented in 2.1.

<table>
<thead>
<tr>
<th>Requirement 1: Massive number of low-cost, low-function, M2M terminals</th>
<th>IP network</th>
<th>DCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal needs to be equipped with functions (DHCP, DNS, TCP) to acquire IP address, do name lookup, and control end-to-end data transmission.</td>
<td></td>
<td>Terminal performs simple communication by using the data’s ID and registering or retrieving the data by communicating with the adjacent network node, without handling any kind of address configuration, name resolution, or end-to-end data transmission.</td>
</tr>
</tbody>
</table>

**Figure 1**  Paradigm changes of network architectures.

- **Requirement 1: Massive number of low-cost, low-function, M2M terminals**

  [IP network] Terminal needs to be equipped with functions (dynamic host configuration protocol (DHCP), domain name system (DNS), transmission control protocol (TCP)) to acquire internet protocol IP address, do name lookup, and control end-to-end data transmission.

  [DCN] Terminal performs simple communication by using the data’s ID and registering or retrieving the data by communicating with the adjacent network node, without handling any kind of address configuration, name resolution, or end-to-end data transmission.

- **Requirement 2: Simple mobility**

  [IP network] Terminal needs to reconfigure IP address when moving to a different subnet.

  [DCN] Terminal does not need to reconfigure when a terminal or a data moves or changes, as it uses data ID that is independent of its location.

- **Requirement 3: Tracking of dynamic data**

  [IP network] Server mediates many-to-many communication.

  [DCN] Network node supports many-to-many communication.

- **Requirement 4: Many-to-many communication**

  [IP network] Server mediates many-to-many communication.

  [DCN] Network node supports many-to-many communication.

The requirements cannot be met by current IP network alone, so it is usually the case that application specific servers and gateways are needed to be deployed and operated to fulfill these requirements. DCN aims to provide a versatile platform that can support diverse types of applications.
3. Prior Art

3.1 Overview of CCN/ICN

In this section, we explain prior arts related to DCN. In research area of Future Networks, there are many research projects that relates to communication using ID of content or data object [6], [7]. Some works have already been done comparing different architectures and approaches [10], [11]. In this paper, we observe and compare with two prior arts that are most relevant to DCN. Content-centric Network (CCN) [12] is a network architecture where communication and routing of the data is done using the ID (or name) of a content. Information-centric Network (ICN) [13], [14] is a more general term used for architecture that may include routing by location dependant addresses. One example of ICN project is NetINF [13]. The characteristics of CCN/ICN are as follows.

- The terminals communicate using ID (or name) of a content, not location dependant addresses such as IP addresses.
- The intermediate nodes store the content in its storage and relay the content hop-by-hop, eliminating the need for end-to-end communication between the terminals.
- The terminals do not authenticate and encrypt communication sessions using IP security (IPSec), secure socket layer (SSL), etc., but instead the contents itself are encrypted and authenticated at each intermediate node.

Comparison between CCN and ICN is shown in Table 1. The main difference between CCN and ICN is the routing. Both CCN and ICN needs to handle routing that can accommodate large number of content IDs and this has become one of the key challenges in this area.

3.2 Problem Regarding CCN/ICN

CCN and ICN both have problems when realizing requirements 2 and 3 of Sect. 2.1., which are relevant to routing.

In CCN, route information based on name (i.e. content ID) stored in intermediate nodes grow as number of contents grows. The route information of CCN is substantially larger than that of IP routing, since the number of the content ID is significantly larger than that of the terminals. CCN makes use of its hierarchical naming scheme and aggregates routing information based on name prefixes. However, when the contents with the same name prefix is widely scattered inside the network, the name cannot be aggregated based on its prefixes, and the routing information will become immense. This does not satisfy requirement 2, where contents can move around and be scattered widely inside the network. Moreover, CCN requires the route information to be propagated throughout the network by exchanging routing information between nodes. This may lead to difficulties when applying to applications with frequent mobility for the routing information may not be updated in sufficient time.

In ICN, the name resolution service manages the mapping between content ID and its address. Every time a content is added, moved, updated, or deleted, the name resolution service needs to be accessed and the mapping information needs to be modified. This will lead to enormous amount of access transactions to the name resolution service. Also communication delay will be caused by accessing the centralized name resolution service. This does not satisfy requirement 3, where frequent access to the name resolution service is expected.

4. Mechanism of Proposed Architecture

4.1 Basic Design of DCN

The proposed architecture is different from prior arts explained in previous section in following aspects.

In DCN, there is no inter-node exchange of route information or session management between terminals. Instead, route information is recorded at each network node when the message and the data are transferred through the network and transit a network node. This will eliminate necessity for inter-node protocol between network nodes or end-to-end session between terminals which will be enormous volume when the number of data objects and the frequency of mobility increases. In comparison, prior art such as CCN requires exchange of its routing information between nodes using routing protocol such as Border Gateway Protocol (BGP) in large scale networks. Also, Breadcrumbs [9] enables routing to cached data using a similar method but requires a name resolution service for the initial retrieval of a data.

DCN performs route attraction and aggregation, where routes are attracted and aggregated to each aggregation point in the network to reduce route information on each network nodes.

DCN also performs optimization where optimized shortcut routes are created to reduce concentration of transactions load and communication delay to the aggregation points.

4.2 Structure of DCN

Network nodes called DCN nodes are placed in a hierarchical structure. Each DCN nodes are interconnected in native (clean slate) or overlay style [22]. In native style, DCN nodes are interconnected directly to each other using physical link or layer 2 connections. In overlay style, DCN nodes
are interconnected using the routing protocol of underlying IP network and the communication between DCN nodes is done by specifying the IP address of the next-hop DCN node.

Figure 2 shows an example of the structure of DCN operating in overlay style. Node00 is the top-tier node, Node01, 02, 03 are tier 2 nodes, Node11, 12, 13 are tier 3 nodes, Node111, 112, 113, 121, 122, 131 are tier 4 nodes. Each network node (called DCN node) records IDs of DCN nodes (called node IDs) and its parent/child relations of all the neighbor nodes in the neighbor node table. Table 2 shows the neighbor node table of Node12. Node02 is marked as parent, Node11, 13 are marked as equal, and Node121, 122 are marked as child.

Data ID used in DCN is hierarchical (e.g. www.example1.com/vehicle01/speed). Each DCN node records route information (node ID of the next-hop DCN node) for a data ID or a prefix of data ID in route information table. The prefix route information with a name domain of an aggregator node is set from the top-tier node to the aggregator node and the aggregator node can be reached using this prefix route information. The allocation of aggregator nodes inside DCN network and setting of the prefix route information from top-tier node to the aggregator nodes are done by the network operator. In this example, the prefix route information recorded in Node00 toward Node01 is www.example1.com → Node01.

DCN is intended to be used in global network where distance between the top-tier node and the aggregator node is extremely long and may involve many hops. In this example, the aggregator node Node01 is placed as tier 2 node, but an aggregator node can be placed anywhere inside the network, for instance in a more distributed manner in lower tiers, as long as the prefix route from the top-tier node to the aggregator node is established.

To construct the network, the network operator needs to specify an aggregator node for each name domain and set prefix route information to DCN nodes that are located between the top-tier node and the aggregator node. The number of prefix route information will only increase based on the number of name domains. The number of name domains and the frequency of addition and deletion of name domains in DCN can be greatly reduced by allocating large number of data IDs to a single name domain, thus reducing the network operator’s burden for setting the prefix route information. Even as the network grows to global scale, the number of name domains and the frequency of addition and deletion of name domains are assumed to be similar to that of domain names in global DNS (Domain Name Services) because of their similar structures.

To obtain further scalability, there can be multiple DCN nodes from the top-tier node to the aggregator node, and the prefix route information from top-tier node to

---

**Table 2** Neighbor node table of Node12.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node02</td>
<td>parent</td>
</tr>
<tr>
<td>Node11</td>
<td>equal</td>
</tr>
<tr>
<td>Node13</td>
<td>equal</td>
</tr>
<tr>
<td>Node121</td>
<td>child</td>
</tr>
<tr>
<td>Node122</td>
<td>child</td>
</tr>
</tbody>
</table>

**Table 3** Route information table of Node11.

<table>
<thead>
<tr>
<th>Data ID</th>
<th>Nexthop</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.example1.com/vehicle01/speed">www.example1.com/vehicle01/speed</a></td>
<td>Node111</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.example1.com/vehicle02/speed">www.example1.com/vehicle02/speed</a></td>
<td>Node112</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.example2.com/vehicle03/speed">www.example2.com/vehicle03/speed</a></td>
<td>Node113</td>
<td>optimized</td>
</tr>
</tbody>
</table>
the aggregator node can be set based on the hierarchical structure of the name domain (e.g., top-tier node will be “com”, second tier node “example1.com”, third tier node “www.example1.com”). This will also reduce the number of prefix route information in the higher tiers.

4.3 Procedure of DCN

The DCN node has 2 basic procedures, data registration and data retrieval.

When a DCN node receives a registration message, it searches the route information table for the next hop node. If there is no route information, it updates the route information to the originating node (node that the registration message was sent from) and transfers the message to the parent node. If there is route information, it updates the route information to the originating node and sends delete route message to the next hop node of the old route information. The delete route message deletes the old route information of each DCN nodes as it is transferred along the old route. The registration message is transferred until a route information is updated or until the message reaches the aggregator node of its name domain.

When a DCN node receives a retrieval message, it transfers the message based on the route information recorded in the route information table. When there is no route information for the data ID, the message is transferred to its parent node. When the message is transferred, the node ID of each transit node is recorded inside the message as route history. When the message reaches the DCN node that stores the data, retrieval message along with the data itself is transferred back using the same route based on the route history recorded inside the message. When a DCN node finds that there is a neighbor node in the route information inside the message and that optimized route (e.g., route that has less node hops) can be created, it updates the route information to the optimized route by recording the neighbor node as the next hop.

The "optimized route" is defined here as a route that has smaller hops counts than the route that the retrieval message has traversed from the requesting DCN node to the destination DCN node. The DCN node creates the optimized route based on the following procedure. The DCN node reads the route history recorded inside the retrieval message. The route history is a list of node IDs that the request message has traversed, in the order that it has traversed. The DCN node refers the node IDs in the route history in the order it is listed and also refers to the neighbor node table. DCN updates the next hop to the first node ID that matches the node ID in the neighbor node table, thus creating the optimized route.

The data in DCN is categorized into original data which is a data that is originally sent from the terminal to the first hop DCN node, and cached data which is a copy of the original data which is stored in other transit DCN nodes. The cache function in DCN is “on-path” caching [10], in which cached data is stored in each DCN node based on the node’s caching algorithm when the cache function is enabled at the DCN nodes. As mentioned in requirement 3 of Sect. 2.1, DCN allows tracking of dynamic data. DCN realizes this by creating route towards the most recently updated original data. The host can specify in the retrieval message if it requests for the most recently updated original data, or if it allows data cache. DCN route the retrieval message based on the route information if the most recently updated original data is requested, or returns the data cache if data cache is allowed. To explain the routing mechanism to the most recently updated original data, the following explanation and experiments in this paper regards to scenarios where cache function is disabled and the host requests for the original data.

An example of procedure for registration and retrieval of data is shown in Fig. 2 and Fig. 3. The routes for registration message and retrieval message are shown in solid lines and routes for delete message are shown in dotted lines.

1. Registration of data (sequence 1 of Fig. 2)

A mobile terminal Host_A registers Data_A (data ID is “www.example1.com/vehicle01/speed” and its aggregator node is Node01) to Node111. Node111 stores Data_A and sends a registration message to its parent nodes until it reaches aggregator node Node01 (Node111 → Node11 → Node01). The route information for Data_A is recorded in each transit node and route Node01 → Node11 → Node111 is created.

2. Registration of data (sequence 2 of Fig. 2)

Host_A moves and connects to Node112 by exchanging discovery message, using for example Ethernet broadcast, and registers updated Data_A to Node112. Node112 stores Data_A and sends registration message to the parent node Node11. Node11 updates the route information of Data_A and route Node 11 → Node112 is created. Node11 sends delete message to Node111 and Node111 deletes Data_A from its storage. (With this procedure, the most recent data of Data_A is stored in Node112, thus Data_A has moved from Node111 to Node112.) Note that local mobility is achieved, in which the registration message does not need to go up to aggregator node Node01 or to top-tier node Node00.

3. Retrieval of data (sequence 3 of Fig. 2)

Host_B requests retrieval of Data_A to Node121. Node121 transfers the retrieval message to its parent nodes (Node121 → Node12 → Node02 → Node00). Node00 has prefix route information to aggregator node Node01 and the message is transferred to Node01, Node11, and eventually to Node112 based on the route information (Node 00 → Node01 → Node11 → Node112). Node112 returns the data via the same route using the route history recorded in the message. When the data is returned, each transit nodes seek optimized routes and records the route on its routing table. In this case, Node12 refers the routing history recorded inside the message and updates its route information to an optimized route (Data_A → Node11).
4. Retrieval of data (sequence 4 of Fig. 2)
Host_C requests retrieval of Data_A to Node122. Node122 transfers the retrieval message to Node12. Node12 refers to its route information (Data_A → Node11) and transfers the request to Node11. The same procedure as the Host_B is carried out for the rest of the sequence.

5. Registration of data (sequence 5 of Fig. 3)
Host_A requests registration of Data_A to Node131. Node131 transfers the registration message to its parent nodes until it reaches Node01 (Node131 → Node13 → Node01). Node01 updates route information to Data_A → Node13 and sends delete message to old route (Node01 → Node11 → Node12).

6. Retrieval of data (sequence 6 of Fig. 3)
Host_B requests retrieval of Data_A and the message is transferred along the old route to Node11 (Node121 → Node12 → Node11). Node11 finds that it does not have the route for Data_A and it transfers the message to its parent node Node01 and to Node131 (Node01 → Node13 → Node131).

4.4 Route Attraction and Aggregation
In DCN, the route information will be smaller than that of CCN, because the routes will be attracted and aggregated at the aggregator nodes. For example, Fig. 4 shows the case where routes are not aggregated at the aggregator node Node01, and Node01, 11, 12, need to record route information of Data_A and Data_B. On the other hand, Fig. 5 shows the case where routes are aggregated at the aggregator node Node01, and only Node01 needs to record route information of Data_A and Data_B.

4.5 Route Optimization
Many networks inside an ISP or network operator today consists of hierarchical redundant topology [23] with a node connecting to two or more parent nodes or equal nodes to support load balancing and redundancy. In this type of network topology, multiple routes can be established by selecting the next hop from multiple parent or equal nodes. DCN is assumed to be constructed using this hierarchical redundant topology as shown as an example in Fig. 6. When no route optimization is performed, Node 21 will access data which its aggregator node is Node12, via parent node Node11, Node01, Node12, and to Node24, shown in dotted line. When route optimization is performed, Node12 can establish an optimized route and access data via parent node Node12 and to Node24, shown in solid line.

The amount of hop count and communication delay that can be reduced by the optimized route depends on the difference of hop count and communication delay between non-optimized and optimized routes and the probability of route optimization. To evaluate this, we will use a simple topology model where the number of tiers is T, the number of DCN nodes at each tier t is n(t), and the number of parent nodes of each DCN node is p. Note that there can be equal nodes or parent nodes that are more than one tier higher, but we will use this model for simplicity. We assume that a retrieval message is sent from a DCN node in tier T to another DCN node in tier T, in non-optimal mode the retrieval message turns down in tier t1, and in optimal mode the retrieval message turns down in tier t2 by successful establishment of optimized route between tier t2 and tier t2+1.

The number of hops of the retrieval message that turns down at tier t node will be 2(T − t + 1) − 1 one way, if we assume for simplicity that the aggregator node is placed along the route of the retrieval message and it can be reached without extra hops. The reduction of the hops which is the difference between non-optimal mode and optimal mode will be H_{reduce}(t2, t1) = 2(t2 − t1), which is larger when the difference between t2 and t1 is larger. If we assume that the probability of successful establishment of optimized route at tier t2 is P_{optimized}(t2), the expected value of the effectiveness of the reduction of hop count is E(t2, t1) = H_{reduce}(t2, t1) * P_{optimized}(t2). The probability of P_{optimized}(t) is dependant to the network topology, but for example, if the selection of parent nodes by each child node is independent to each
other, the probability of two child nodes in tier $t+1$ sharing parent nodes (excluding the case where they share all parent nodes) in tier $t$ is Eq. (1) which will be larger when the nodes are in higher tier and $n(t)$ is smaller.

$$1 - \left( \frac{(n(t)-p)C_p + 1}{n(t)C_p} \right)$$

(1)

The effectiveness of the optimized method is expected to be larger and reduces the hop count more compared to non-optimized route when the network is larger and there are more DCN nodes and tiers. For example, the difference between expected value of the effectiveness for network with $T+1$ tiers and $T$ tiers and will be Eq. (2), if we assume that $P_{\text{optimized}}(t)$ of both networks is roughly the same.

$$\sum_{i=1}^{T} H_{\text{reduce}}(T, t) \cdot \frac{P_{\text{optimized}}(T)}{n(t)C_p}$$

(2)

Additionally, if the communication delay of each link between DCN nodes is larger (e.g. international links), reduction of the communication delay for each hop due to the optimized route will be larger.

In this paper, we have evaluated method for establishing an optimized route using neighboring DCN nodes which is one hop away. The probability of establishing an optimized route will increase if more than two hops are allowed, and the method for achieving this is for future study.

To prevent the route information from increasing infinitely, DCN node can be preset with a maximum number for route information and it can delete the old optimized route information so that the number of route information will not exceed the limit. Even when the optimized route information is deleted, the retrieval message can always be routed to the aggregator node by being transferred to parent node or along the prefix route which will ensure that the request will eventually reach the final designated data.

4.6 Effect of Route Optimization

We evaluate the effect of route optimization for the proposed architecture using the conditions shown below. First we calculate number of route information that needs to be managed by an aggregator node. We assume 50 billion terminals [16], 16 million aggregator nodes (which is similar to number of DNS servers in 2009 [17]), 1 trillion data objects (the number of indexed web pages is 50 billion in 2012 [18]). By this assumption, each aggregator node needs to manage average of 3,125 terminals and 62,500 data objects.

Next we calculate the effect of the route information. If we assume that memory usage for one route information entry is 16 bytes (8 bytes for hash value of a data ID, 6 bytes for ID of next hop node, 2 bytes for other attributes), 500 million entries can be recorded on a 8 Gbyte dynamic random access memory. This number will be 8,000 times the amount of average data objects managed by an aggregator node.

According to [19], internet access from an enterprise with 40,000 employees will be 100 million accesses in 150 hours, which equals to 10,000 accesses in one minute. If we assume that 1 entry is recorded for every access, it would take approximately 1 month to record 500 million entries. According to an experiment regarding web cache [20], they observed 90% cache hit ratio when 70 to 80% of all the web access were cached.

These statistics show that large number of entries can be recorded in each node as optimized routes, and we can expect these optimized routes to reduce the traffic to the aggregator node.

4.7 Consideration Regarding the Scalability of the Top-tier Node

Because the top-tier node is logically a single node in a DCN, considerations should be taken regarding its scalability against data registration. Top-tier node needs to record prefix routes for all name domains in the network. In addition, top-tier node needs to record route information for registration messages that transits the top-tier node.

To provide scalability, different name domains for each data is allocated based on location of each data in the network topology and the aggregator nodes are placed near each location. For example, name domains are allocated for each physical area where M2M terminals are distributed, or for each network operator the M2M terminals are connected to, and place the aggregator node in each physical area or network operator. In this case, most of the registration message sent from M2M will reach its aggregator node via the prefix route before it reaches the top-tier node.

Future study is needed to provide scalability in the case where it is difficult to presume the location of the data, or in the case where massive number of data moves to extremely remote area away from the aggregator node. One solution is to have DCN nodes in the higher tiers that have direct links that connect each other, and the registration messages are routed using prefix route information between these DCN nodes and to aggregator nodes without using the route information of the top-tier node.

4.8 Comparison with Prior Arts

Comparison of the proposed architecture with CCN and ICN is shown in Table 4. In CCN, route information grows exponentially when the data is distributed throughout the network but DCN reduces the number of route information by placing aggregator node and aggregating the routes at the aggregator node. In ICN, transaction load and communication delay to name resolution service will be large but DCN reduces these by using local mobility and route optimization. Also, DCN does not need any inter-node information exchange as seen in IP routing or CCN, since route information is recorded when the request traverses a node during registration or retrieval of the data.

Comparison with other prior arts is explained below. Data Oriented Architecture (DONA) [7] uses hierarchical network structure and the nodes route the data based on ID
of the data, which is similar to DCN. However, DONA uses flat ID and the route information at the high level nodes will be enormous, while DCN can reduce the number of route information by attracting and aggregating the routes to the aggregator node.

Publish-Subscribe Internet Routing Paradigm (PSIRP) [8] has a rendezvous point that handles registration of the data and matching of publish and subscribe requests, which resembles the aggregation point of DCN. However, because PSIRP manages the matching of publish and subscribe requests at the rendezvous point, it requires all requests to transit through the rendezvous point, which increases the transaction load and communication delay. DCN can reduce transaction load and communication delay using local mobility and route optimization.

Breadcrumbs [9] records route information while the data transits a node, which is similar to DCN. However, Breadcrumbs only uses the route information for routing to cached data and requires additional functions such as name resolution services to access newly registered data.

Domain Name Systems (DNS) uses group of servers that are structured in hierarchical manner based on name domains, which is similar to DCN. However, in DNS all requests needs to access DNS servers, which increases the transactional load and communication delay. The transactional load and communication delay may be reduced using information cache at local DNS servers, but when the location of a data changes it requires time for the information cache to be updated.

DCN makes use of many of the approaches of prior arts mentioned above, but focuses on scalable architecture for frequent mobility and data changes in widely distributed environment which is required for M2M services.

### 5. Experiment Results

#### 5.1 Experiment Environment

We developed an experiment environment using seven virtual servers with a prototype of DCN node running on each server (3.0 GHz CPU, 8.0 GB memory). Each DCN node is interlinked using Ethernet connection with topology shown in Fig. 7, operating in native style. The topology is hierarchical, with a top-tier node Node01 and two nodes for second tier (Node12, 22), third tier (Node13, 23), and fourth tier (Node14, 24). To demonstrate the effect of the optimized route, the second tier node has bypass links to third and fourth tier nodes.

The experiment is designed to demonstrate the changes in the routes of the data retrieval. We emulated one mobile terminal that moves from one node to another (Node13 → 23 → 24 → 14 → 13) on 10 second interval to emulate terminal mobility from one access point to another. The terminal registers a data with the same data ID on 15 second interval. The aggregator node for the registered data is Node24. Four terminals (T01, T02, T03, T04) that are statically connected to Node14, 13, 24, 23 each requests and retrieves the registered data in 100 msec interval. The total request caused by this is 40 requests per second. The size of each data is 100KB which is a typical size for M2M data, so the estimated total traffic caused by the retrieval of data is 32 Mbps with additional overhead for header information such as the data ID, etc.

In this experiment, hop count from terminal to the destination node, round trip communication delay, traffic, and number of route information were measured. Round trip delay was measured from the time the retrieval message was sent from originating host to the time it was received by the same host, and the delay mostly consists of transmitting delay between the DCN nodes and transfer delay within the DCN nodes. The experiment was done using two different modes, non-optimal and optimal mode. Non-optimal mode does not record optimized routes at the relative nodes, so the traffic always goes to the aggregator node, which is similar to concentration of transaction load to the name resolution service in ICN. Also in the non-optimal mode, the traffic is also likely to traverse top-tier node Node01, as the bypass links will not be used. Optimal mode records optimized routes at the relative nodes and demonstrates the effect of establishing optimized routes. In this experiment we did not set limit to number of route information, so the transaction load, communication delay, and number of route information will be similar to CCN.
5.2 Measurement of Hop Counts and Communication Delay

Figure 7 shows case I where T01 requests a data which is registered to Node24. In non-optimal mode, the request from T01 is routed from Node14 to the top-tier node Node01 and then to the destination Node24 (shown in dotted line). In optimal mode, the first request from T01 is routed through the same route as non-optimal mode, but at the second request the route is optimized and routed directly from Node14 to Node22 and then to Node24 (shown in solid line). This shows that at the first request route information of optimized route (Node14 → Node22) is recorded in Node14 and reduced the traffic to Node01 and the hop count.

Figure 8 shows case II where T01 requests a data which is registered to Node23. In non-optimal mode, the request is routed to Node01 and to aggregator node Node24 and back to destination Node23. In optimal mode, the request is directly routed to Node22 and to Node23. This reduces the traffic to Node01, hop count, and also traffic to Node24.

Table 5 shows the comparison of hop counts and delay for the two cases. In case I, the optimal mode reduced approximately 43% of both hop count and average delay. In case II, the hop count was reduced by 63% and delay by 58%.

5.3 Measurement of Traffic

We measured the traffic at each link between the DCN nodes to evaluate reduction of traffic at the aggregator node and top-tier node. Figure 9 and Fig. 10 show the outbound traffic of aggregator node Node24 to other DCN nodes. Figure 9 shows the non-optimal mode, where the traffic is mostly around 40 Mbps, which indicates that all traffic traverses through Node24. There are drops in traffic from 30 to 40 sec and 70 to 80 sec, which indicates that the data object is located on Node24 and the data object can be retrieved back to the adjacent terminal T03 without using outbound link to another DCN node. Figure 10 shows the optimal mode, where the traffic is mostly around 8 Mbps. The traffic is reduced compared to non-optimal mode, because most of the traffic does not need to traverse Node24. There is an increase in traffic from 42 to 52 sec, which indicates that the data object is located on Node24, and it is retrieved via outbound link to another DCN.

Figure 11 and Fig. 12 show the outbound traffic of top-tier node Node01. Figure 11 shows the non-optimal mode, where peak traffic is around 33 Mbps. There are drops in traffic which indicates that the data object was located on the same side as the aggregator node Node24, which means that the data object was located on Node23 or 24, and some requests were routed to Node24 and to the data object without traversing Node01. Figure 12 shows the optimal mode, where there is almost no traffic except for 39 and 70 sec. This indicates that most of the requests do not traverse Node01, except for cases where optimized route is not recorded at the relevant nodes and the request needs to...
traverse Node01.

5.4 Measurement of Number of Route Information

We conducted another experiment by increasing the number of data IDs to measure the route information recorded at each DCN nodes. The network topology is same as the previous experiment and 4 terminals, each with different name domains, registers the data randomly with 10 different data IDs (e.g. 4 name domains and 40 data IDs) in 15 second interval. The aggregator nodes that manage each name domains are Node 14, 13, 23, 24. The terminal was moved in the same manner as the previous experiment, in 10 second interval. The number and types of route information records was retrieved from each DCN node after the experiment was conducted for two minutes and the number of route information records has become stable.

Table 6 shows the number of route information records on a node when measured in optimal mode. The number of route information for aggregation routes (routes that belong to the name domain that the aggregator node manages) was 10, which is same as the number of data IDs an aggregator node manages. The number of route information for optimized routes was 20, which is same as the number of routes that traverse bypass links. Also there were other route information that are neither aggregation routes nor optimized routes that was recorded when the data traversed the node but was not optimized.

In this experiment, we did not set a limit to number of route information a DCN node can record, so the number of route information was 40 for all nodes. The minimum route information required for DCN to route a message or data is aggregation routes and other routes. This means that the minimum number of route information is 20 for Node 13, 23 and 10 for Node 14, 24. In the case of CCN, when data with different data IDs is distributed within the network as it is the case in this experiment, it cannot make use of prefix aggregation and the number of route information needs to be 40. If we compare DCN and CCN, DCN is expected to reduce the number of route information by 50 to 75% at the maximum compared to CCN.

5.5 Evaluation

In this experiment using prototype of DCN nodes, we confirmed that all the requested data was successfully retrieved even while the terminal is moving and registering dynamic data both for non-optimal and optimal mode. We observed that the hop counts and average delay can be significantly reduced in optimal mode compared to non-optimal mode, up to 63% and 58% respectively. We saw decrease in the peak, minimum, and average traffic of both aggregator node Node24 and top-tier node Node01 in optimal mode compared to non-optimal mode. From these results, we found that optimal mode reduces communication delay and also reduces traffic to aggregator nodes and top-tier nodes.

In this experiment, we used a small scale local network with 7 DCN nodes and 40 terminals. We estimate that the reduction of transaction load and communication delay will be more significant in a large scale wide area network, where the communication delay and the hop counts are large, if the aggregator nodes and the bypass links are placed in lower tiers in a distributed manner and the delay and the number of hops from the top-tier to the aggregator node is large.

The disadvantage DCN may have is that DCN needs to process data or message and conduct routing decisions based on its data ID and routing history inside the message, which requires larger computational load at each node compared to conventional method such as IP routing. Another disadvantage may be that DCN requires deletion of route information which may cause additional transaction load and communication delay. Finally, the old optimized routes can cause a detour for data retrieval. We think that these points need to be evaluated further in the future using a large scale network.
There is a tradeoff between route aggregation and route optimization, as route aggregation reduces the route information but increases transaction load and communication delay to aggregator node, and establishment of optimized routes is vice versa. For this reason, the balancing of these two characteristics is important and it needs to be evaluated in future studies.

6. Conclusion

We proposed a network architecture called DCN which tackles the problems regarding conventional architectures. DCN reduces the routing table size by attracting and aggregating routes to aggregation point and reduces both access transactions to aggregation point and communication delay by establishing optimized routes that do not traverse the aggregation point. We conducted an experiment using prototype of DCN nodes and measured hop counts, delay, and traffic traversing aggregator node and top-tier node. The results showed that the communication delay and the traffic were reduced significantly in optimal mode.

Although DCN does not require a name resolution service, it can be interact with a name resolution service to provide an alternate solution for scalability. For instance, DCN nodes at high-level tiers such as top-tier node and tier 2 nodes can be replaced by name resolution service, thus lower tier nodes can refer to the name resolution service and transfer the data and the message directly with each other without using routing information recorded in DCN nodes at high-level tiers. We will also investigate this type of architecture in the future.

Testing DCN in a large-scale wide area network is an important next step to evaluate the scalability and feasibility of the architecture. The deployment of such environment is cost and time consuming, as each DCN node needs to be equipped with considerable amount of computation and storage resource, and DCN prototype software needs to be deployed in each network nodes. We plan to construct experiment environment of DCN using JGN-X network [5], where we can make use of computation and storage resource that is already installed in the network, and conduct various experiments regarding mobility performance and scalability.

Acknowledgments

Part of this project is funded by National Institute of Information and Communications Technology (NICT).

References


Daisuke Matsubara received the B.S. and M.S. degrees in Electrical Engineering from Kyoto University, Japan in 1996 and 1998, respectively. In 1998, he joined Hitachi, Ltd. where he was involved in researches such as ATM/STM exchange unit, QoS path control using MPLS/DiffServ, P2P network systems, resource control for NGN, network virtualization, and data-oriented networks.
Hitoshi Yabusaki received the B.S. degrees in Mechanical Engineering from Yokohama National University in 2006, and M.S degrees in Engineering Synthesis from University of Tokyo in 2008. In 2008, he joined the Central Research Laboratory of Hitachi, Ltd. He is involved in researches on path control using MPLS-TP and network virtualization.

Satoru Okamoto received B.E., M.E. and Ph.D. degrees in electronics engineering from Hokkaido University, Hokkaido, Japan, in 1986, 1988 and 1994. He is currently a project Professor of Keio University, Japan. In 1988, he joined Nippon Telegraph and Telephone Corporation (NTT), Japan, where he conducted research on ATM cross-connect system architectures, photonic switching systems, and optical path network architectures and participated in the development of GMPLS- controlled HIKARI router (“photonic MPLS router”) systems. He is an IEEE Senior Member.

Naoaki Yamanaka graduated from Keio University, Japan where he received B.E., M.E. and Ph.D. degrees in engineering in 1981, 1983 and 1991, respectively. In 1983 he joined Nippon Telegraph and Telephone Corporation’s (NTT’s) Communication Switching Laboratories, Tokyo Japan, where he was engaged in research and development of a high-speed switching system and high-speed switching technologies for Broadband ISDN services and ATM base backbone network as NTT’s Distinguished Technical Member. He moved to Keio University, Department of Information and Computer Science in 2004. He is now researching future optical IP network and optical MPLS router system. Dr. Yamanaka is an IEEE Fellow.

Tatsuro Takahashi received the B.E., M.E. in Electrical Engineering from Kyoto University, Kyoto, Japan, in 1973 and 1975 respectively, and Dr. of Engineering in Information Science from Kyoto University in 1975 to 2000, making R&D on high speed networks and switching systems for circuit switching, packet switching, frame relaying, and ATM. Since July 1, 2000, he is a Professor, Communications and Computer Engineering, Graduate School of Informatics, Kyoto University. He received the Achievement Award from IEICE in 1996, the Minister of Science and Technology Award in 1998, and the Contribution Award from IEICE in 2011. He was a Vice President of the ATM Forum from 1996 to 1997, and the Chairman of the Network Systems(NS) Technical Group in the Communications Society of IEICE from 2001 to 2002. Prof. Takahashi is an IEEE Fellow.