# CATT: Potential Based Routing with Content Caching for ICN

Suyong Eum
National Institute of
Information and
Communications Technology
4-2-1 Nukui-Kitamachi,
Koganei, Tokyo, 184-8795
suyong@nict.go.jp

Kiyohide Nakauchi
National Institute of
Information and
Communications Technology
4-2-1 Nukui-Kitamachi,
Koganei, Tokyo, 184-8795
nakauchi@nict.go.jp

Masayuki Murata
Osaka University, Graduate
School of Information Science
and Technology
1-5 Yamadaoka, Suita, Osaka,
565-0871 Japan
murata@ist.osaka-u.ac.jp

Yozo Shoji
National Institute of
Information and
Communications Technology
4-2-1 Nukui-Kitamachi,
Koganei, Tokyo, 184-8795
shoji@nict.go.jp

Nozomu Nishinaga
National Institute of
Information and
Communications Technology
4-2-1 Nukui-Kitamachi,
Koganei, Tokyo, 184-8795
nisinaga@nict.go.jp

## ABSTRACT

Information Centric Networking (ICN) has shown possibilities to solve several problems of the Internet. At the same time, some problems need to be tackled in order to advance this promising architecture. In this paper we address two of the problems, namely routing and content caching. For the routing, we introduce the Potential Based Routing (PBR) to achieve several design goals such as availability, adaptability, diversity, and robustness. In addition, we examine the performance of a random caching policy which can be a promising candidate for ICN. The integrated system which combines the PBR and a caching policy is named the Cache Aware Target idenTification (CATT). Simulation results demonstrate that the PBR with replications located on less than 1% of total nodes can achieve a near optimal routing performance (close to the shortest path routing) even though a request message is randomly forwarded.

## Categories and Subject Descriptors

C.0 [**Computer System Organization**]: General—*System architectures*

## Keywords

ICN (Information Centric Networking), Routing, Caching.

*ICN'12,* August 17, 2012, Helsinki, Finland.

## 1. INTRODUCTION

Communications and network paradigm have been shifted from user-to-user to user-to-information since dominant use of networks has been largely changed to the retrieval of information rather than end-to-end voice conversation. This new networking paradigm requires a major transformation of current Internet architecture, which leads to the birth of Information Centric Networking (ICN) [1][2][3][4].

In ICN, a user request is routed based on the name of a desired content file rather than its location. This property known as route-by-name enables network elements to be aware of user requests as well as their counterpart responses. Due to the awareness, individual ICN elements have capability of responding to users' requests directly with support of storage or cache on them. In other words, ICN can be considered as a large scale distributed caching architecture whose elements function as caching points.

Several issues need to be taken into account to utilize the caching capability of ICN. In this paper we consider the following two questions; how to locate or select one of multiple identical contents which are distributed in the network? and how are these contents distributed or cached by taking advantage of in-network caching capability of ICN? While we answer to the questions, we integrate our solutions to design a new ICN architecture which is named as the Cache Aware Target idenTification (CATT).

The former question is known as a routing problem in ICN in a sense that a user request is routed to the location of the requested content. Since ICN can be considered as a largely distributed caching architecture, content files are assumed to be located not only in repository where an original content file is published but also in caches where copies are placed. For this reason, an ICN routing mechanism must be able to incorporate these multiple identical content files distributed in the network to provide a better choice for users. This is one of desired properties for ICN, namely availability.

In various ICN architectures, e.g., [1], availability is realized using a name resolution process which is a name-based anycast service where a user request is routed to the closest one among a group of copies identified by the same name. One problem of using a name resolution system to achieve availability is that maintaining the list of copies is not a trivial task, especially in ICN, due to the volatile behavior of copies in caches which appear and disappear at caches dynamically. In addition, a centralized name resolution system is vulnerable to a single point failure or attack scenario.

To tackle these problems above, we propose the Potential Based Routing (PBR) whose original idea was introduced by Chalermek et al under the name of the directed diffusion method[5] which has been adopted in various network applications (refer to Section. 2) since it provides abundant design spaces. Using the PBR approach, we intend to achieve following design goals;

- Availability which is to incorporate not only an original content file in repository but also its copies in caches into the retrieval process of the requested content file since copies are broadly distributed among in-network caches in ICN.
- Adaptability to handle volatile caching content files which appear and disappear from caches dynamically due to a cache management purpose.
- Diversity to provide abundant routing decision processes for users, e.g., based on not only proximity but also several conditions including the capability of provider or surrounding network conditions, etc.
- Robustness against a single failure based on a fully distributed mechanism.

In addition to the routing problem in ICN, we investigate the second question; how content files are distributed or cached in ICN? Here we demonstrate an interesting result that a single node random caching can function as good as or even better than all node caching which is the current default implementation of CCNx[6] in terms of the latency analysis.

The rest of this paper is organized as follows: Section 2 introduces related works with the Potential Based Routing (PBR). Section 3 presents a description of the CATT architecture including the PBR and content caching algorithms. This is followed by some numerical evaluations of the proposed architecture in Section 4. Finally, we finalize the paper with several discussions in Section 5.

## 2. RELATED WORK

Since Chalermek et al introduced the directed diffusion method[5], which is one of sub projects in SCADDS (Scalable Coordination Architectures for Deeply Distributed Systems), a family of field or potential based approaches have been used for a nearly decade in various network applications such as traffic or connectivity aware routing[7][8], service discovery[9], load balancing[10], and data aggregation[11].

In [7], they defined a scalar potential field on the network and route packets in the direction of maximum force (steepest positive gradient in the field), and demonstrated that the potential based routing significantly improves end-to-end delays and jitter without much control overhead. In [9], Lenders et al. applied this idea for dynamic networks such as wireless ad hoc networks for efficient and robust service discovery. The approach was based on soft-state principle where a system functions although a service instance is discarded without a notice to a network. In [10], Jung et al. extended the idea for autonomous load balancing as well as distributed routing for wireless mesh networks. In [11], Zhang et al. designed an effective data aggregation mechanism supported by dynamic routing. They made full use of local information to make the routing decisions rapidly and ensured that the decisions were efficiently adapted to both data gathering and event-based applications using the field based approach.

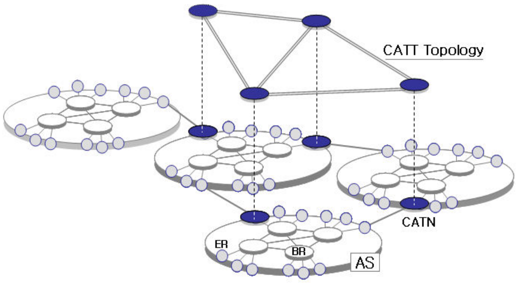

**Figure 1: Initial deployment scheme for CATT. ER (Edge router), BR (Backbone router), AS (Autonomous system, e.g., ISP), CATN (CATT node).**

The Potential Based Routing (PBR) we are proposing can be considered as the extended version of directed diffusion [5] and Service discovery [9]. PBR uses a similar idea with the directed diffusion in a sense that a data packet navigates a potential field defined in the network to facilitate "routing" towards a desired location. The difference is that the directed diffusion approach was introduced for "adv/pub" architecture (Users are responsible for creating routing information to subscribe a certain information). On the other hand, in PBR content files are published first and at the same time routing information to them are setup. In this manner, the proposed PBR is more close to the service discovery mechanism in [9] which has been applied to mobile ad hoc networks. However, that approach aims to find a fixed service provider located in ad hoc networks so that it is not well suited for our purpose.

As mentioned previously, a typical routing approach of ICN is to adopt anycast mechanism through a name resolution process which routes a user request to a nearby copy. We believe that maintaining lists of copies by tracking down all copies in in-network caches is not feasible. In addition, a centralized name resolution system is vulnerable to a single point failure or attack scenario.

## 3. CATT ARCHITECTURE

### 3.1 Overview of CATT

CATT (Cache Aware Target idenTification) is designed mainly considering two components, namely routing and content caching. Although there are several other important design issues such as naming or security, we focus on describing the two components only in this paper. For the routing, we propose the Potential Based Routing (PBR) which is elaborated in the next section. For the caching component, CATT supports selective caching mechanisms internally or externally. The internal caching scheme determines which content file is cached depending on its internal policy. The external process selects a caching point to improve the global performance in a sense that it reduces the workload of network by distributing content files efficiently among caching points over the wide area network. For the external process, a caching decision is preferably made while a content file is downloaded from the content provider to its requester since it reduces the control overhead. All nodes along the downloading path can be considered as caching points, which is the current default implementation of CCNx [6]. On the other hand, one of the nodes on the path can be selected based on several factors such as topological characteristic, traffic load, or even randomly.

Fig. 1 shows the initial deployment scheme of the CATT on the current Internet as its transition phase before its move to "clean-slate" phase running on the pure CATT architecture. In this scheme, one or more CATT nodes (CATNs) are deployed on edges of autonomous system (AS) depending on the size of AS. Considering CATNs as caching points for transition traffic between ISPs, the ISPs with CATN(s) can enjoy the cost effective approach which is caching content files nearby end users instead of using global transit to download them. The deployment scheme is rather highly related to ISP routing policies. Users within an AS are assumed to know the location of its local CATN(s) so that they publish their content files directly on it. Thus, any request is forwarded to its local CATN first and the process of content resolution (routing) is initiated from the point. CATN which receives a user request first checks the existence of the requested content file in its local cache, if it exists, it responds to the request directly, otherwise it routes the request to one of its neighbor CATNs based on the PBR.

## 3.2 Potential Based Routing (PBR)

We model a network as an undirected graph. Each node $n$ has the set of neighbor nodes $B(n)$. Nodes in the network are divided into two groups, namely caching nodes $n_c$ and non-caching nodes $n_{nc}$. A caching node is defined as a node that has a content file in its cache. $T$ represents the number of caching nodes that have an identical content ($n_c^1$, $n_c^2$, …, $n_c^T$). Initially, we assume that there is only one caching node ($T$=1) in the network. Then, the potential value at node $n$ is defined as $\psi_n$ in Equ. (1).

$$\psi_n = \begin{cases} -Q_{n_c} & n = n_c \\ -\frac{Q_{n_c}}{d_{n_c \leftrightarrow n}+1} & n \neq n_c \end{cases} \qquad (1)$$

Let's define the potential value of the caching node $n_c$ as $-Q_{n_c}$. The absolute value $|Q_{n_c}|$ represents the expected quality of the content which has several interpretations. For instance, a caching node with high capacity outgoing links or high processing power can be assumed to provide high quality content. $d_{n_c \leftrightarrow n}$ denotes the distance between the caching node $n_c$ and a node $n$. Intuitively, we conjecture that the quality of the content file is degraded as the distance $d_{n_c \leftrightarrow n}$ increases. There are various physical interpretations of the distance such as geographic proximity, hop counts, transmission delay, link cost, etc. The negative sign in Equ. (1) makes sure that a request keeps moving down toward the lower part of the potential field.

Now, let's move to the case for the multiple caching nodes. The potential value at each node can be simply considered as the linear summation of the potential values which are influenced by individual caching nodes. Thus, Equ. (1) is turned into as follows;

$$\psi_n = \begin{cases} -Q_{n_c^i} - \sum_{(n_c \neq n_c^i)} \frac{Q_{n_c}}{d_{n_c \leftrightarrow n}+1} & n{=}n_c^i \in n_c \\ -\sum_{i=1}^{T} \frac{Q_{n_c^i}}{d_{n_c^i \leftrightarrow n}+1} & n \notin n_c \end{cases} \qquad (2)$$

When a request arrives at node $n$, it is forwarded to a neighbor of the node $n$ until it reaches to a content provider which can be either a caching point or a repository. The choice for the next hop $b_{next}$ is decided in terms of the potential difference $F_{n \to b}$ between the node $n$ and its neighbors $b \in B(n)$.

$$b_{next} = [\ \text{Max } F_{n \to b} = \psi_n \text{ - } \psi_b \mid \text{b} \in \text{B(n)}] \qquad (3)$$

Thus, a user request on node $n$ is forwarded to a neighbor of the node $n$ that maximizes the potential difference $F_{n \to b}$. To illustrate the construction process of a potential field, an example is prepared in Fig. 2.

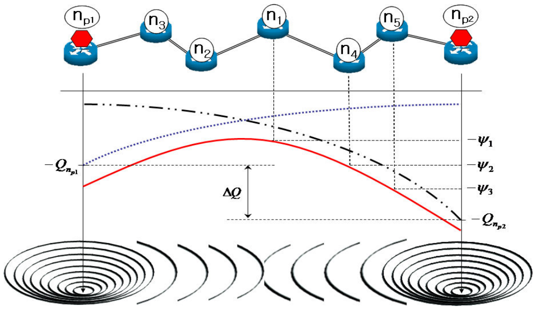

**Figure 2: Potential field constructed by two identical content files located at the end of the linear topology.**

Assume that there are two identical content files which are located on the two end points $n_{p1}$ and $n_{p2}$ of the linear topology. The expected quality of the content file on node $n_{p2}$ is also assumed better than the one on node $n_{p1}$. Thus, the absolute value $|Q_{n_{p2}}|$ is larger than $|Q_{n_{p1}}|$ (its difference is shown as $\Delta Q$), which makes the potential field (shown as black dash-dot line) created by $n_{p2}$ wider and larger than the one (blue dot line) by $n_{p1}$. The processing (CPU) power or the total outgoing link capacity of the node $n$ can be a candidate to define the value of $|Q_n|$. The red solid line represents the potential field which is the summation of the individual potential fields (black dash-dot line and blue dot line) defined by $n_{p1}$ and $n_{p2}$, respectively. Assume that a user request is launched on node $n_1$. Since its neighbor nodes are nodes $n_2$ and $n_4$, the node $n_1$ compares the potential values of its neighbors. The potential value of node $n_4$ is smaller than that of nodes $n_2$ so that the request is forwarded to the node $n_4$. Similarly, the request on the node $n_4$ is finally forwarded to the node $n_{p2}$ which has the requested content file. Although, the request is launched on node $n_1$ which is located in the middle of both providers $n_{p1}$ and $n_{p2}$, the request message is attracted to the provider $n_{p2}$ due to the large value of $|Q_{n_{p2}}|$.

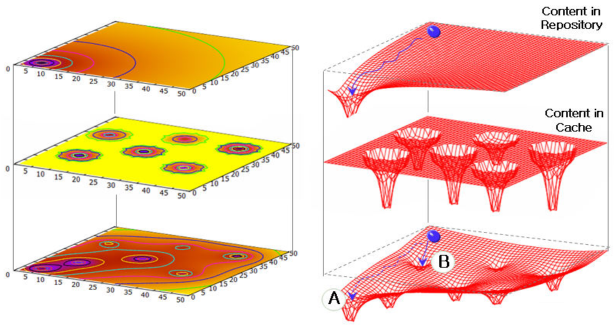

**Figure 3: There are three different types of potential fields for the PBR routing in CATT.**

## 3.3 Potential field use cases

### 3.3.1 *Permanent potential field (PPF) for a content file in repository*

A content file published in repository is generally maintained permanently unless there is an arbitrary modification or any mechanical accident on the repository. Thus, once routing information to the content file is defined, the information tends to be maintained for a long period. For such a consistent content file, a permanent potential field is constructed as shown in the top layer of Fig. 3. To produce this permanent potential field, a node floods an advertisement packet to inform the possess of the content file in its repository within an administrative domain. One example of the administrative domain can be a publisher which manages all content files produced within the publisher domain. This clustering structure based on the publisher domains localizes an advertisement packet within the domain without being flooded in the entire network. In addition, due to the consistency of the published content files in repository, the potential field is seldom necessary to be updated or modified. For this reason, the scalability issue caused by flooding overheads of advertisement packets can be minimized.

### 3.3.2 *Volatile potential field (VPF) for content files in caches*

The characteristic of a copy in cache is different from that of an originally published one in repository. Due to the capacity limitation of each cache, some copies in cache are regularly replaced by newly appeared ones so that they are rather volatile compared to one in repository. Moreover, the number of copies in caches is generally far more than the original content file in repository. Since all the identical copies contribute to the counterpart potential field, the volatile behavior of large number of copies forces the potential field to be re-defined frequently. To minimize this effect, potential fields for those in caches are constructed within limited scope as shown in the middle of Fig. 3. Due to the limited flooding scheme, volatile behavior of copies in caches can be handled without much disturbance to the network.

### 3.3.3 *Combined potential field (CPF)*

The bottom of Fig. 3 illustrates the linear combination between permanent potential field (PPF) and volatile potential field (VPF). Due to the former (PPF), all requests are forwarded to the repository where the original content file is published initially. While the request is on the way to the repository, it may be attracted to the node which is caching the content file. This process is analogically illustrated in the bottom of Fig. 3. While the ball representing a user request moves down to “A” which represents the location of the originally published content file, it is attracted to “B” which shows the location of the copied one in the cache.

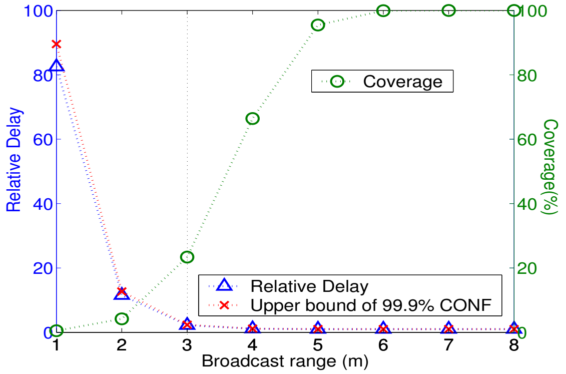

**Figure 4: Relative delay vs Coverage. For reference, when no advertisement packet is flooded ($m$=0), the relative delay of PBR over OSPF routing is around 480±19.22 (99.9% confidence interval).**

## 4. PERFORMANCE EVALUATION

For this simulation, an event driven simulator was developed. We look at an ICN network as a graph, and model the flow of data from content providers to users as flows on the network graph. A power law topology was constructed to reflect the Autonomous System (AS) level topology [12] (the number of nodes 1000 with the average degree 2). For the comparison between the PBR and a shortest path routing such as OSPF (Open Shortest Path First), the Dijkstra algorithm was implemented.

## 4.1 Delay analysis of the PBR

A content file is published on a randomly chosen node and its advertisement packet is flooded through the network to construct a potential field. The packet travels $m$ number of hops from the node. The percentage of total nodes where the packet is delivered is defined as coverage. Since the coverage has a positive correlation with the size of routing table, this metric is used to represent the size of expected routing table. Then, a randomly chosen user generates a request for the content file, and the query is routed to the node where the content file is published. The request is forwarded randomly until it moves into the potential field. Thus, it generally experiences more delay than the shortest path routing, e.g., OSPF. We call this delay as relative delay.

The averaged coverage and relative delay are plotted in Fig. 4 as the flooding range of the advertising packet increases. When the node holding the content file floods an advertisement packet to its neighbors only ($m$=1), its coverage is less than 1% of total nodes, which means the rest 99% of nodes do not need to maintain the advertised information

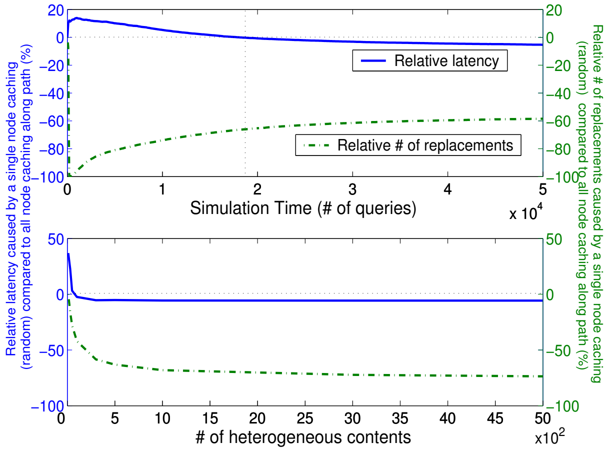

**Figure 5: Relative latency and relative # of replacements. This result is the average of 10 realizations.**

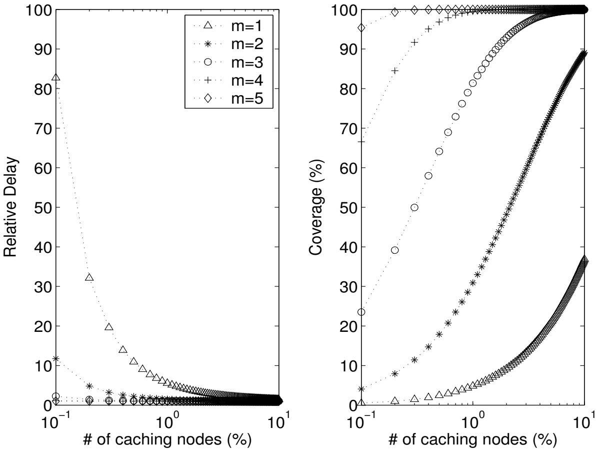

**Figure 6: Relative delay and coverage are plotted when multiple cached nodes as well as different broadcast range $m$ are used.**

in the routing table. However, users suffer from longer latency which is around 80 times longer than the OSPF routing. Interestingly, the relative delay dramatically decreases as the flooding range increases. When the advertisement range is equal to 3 ($m$=3), although the message is delivered to less than 30 % of total nodes, its relative delay is close to the optimal - around 2.2 times longer latency than the OSPF routing.

## 4.2 Caching at a node / all nodes on path

In ICN, a caching decision is preferably made while a content file is downloaded from the content provider to its requester since all nodes along the downloading path are aware of user request as well as its corresponding response. One interesting question is whether a single node is selected or all nodes along the path are considered as caching points for the downloading content file.

For the comparison, all node caching($AL$), and a single node random caching($RD$) are implemented. $AL$ is the current default implementation of CCNx[6] which caches a content file on every node along the downloading path. $RD$ selects one of nodes on the trajectory randomly. Fig. 5 shows that $RD$ performs even better than $AL$ as the caching system approaches to the stable state. $AL$ triggers more replacements on individual caches than $RD$. In other words, some popular content files can be replaced by newly appeared ones more frequently in $AL$. Due to the frequent replacements of some popular content files, users tend to experience more delay with $AL$ than $RD$.

In the bottom figure of Fig. 5, we vary the number of heterogeneous content files in the network to investigate the effect of unlimited & limited cache size. Each cache has 300 units of space, and the size of each content file is drawn from the ZIPF distribution within a range between [1-50]. When the number of heterogeneous content files is equal to 10, individual caches can accommodate all contents (The case of unlimited cache size). It implies that users can directly retrieve any content file from the node to which the users belong. Thus, no replacement occurs in both cases of $AL$ and $RD$ so that it begins from zero. However, $RD$ has around 40% large relative delay than $AL$ because $AL$ distributes copies faster than $RD$ at the beginning of the simulation so that users experience less delay with $AL$. As the number of heterogeneous content files increases, individual caches become saturated and need to remove caching content file(s) to accommodate a newly arriving content file. One interesting observation here is that $RD$ also shows smaller latency than AL as the number of contents increases. It is because $AL$ causes frequent content replacements in caches, which increases the chance to remove some popular content files. The result demonstrates that $RD$ (a single node random caching) can do the same job or even better than $AL$ (all node caching) in terms of the latency analysis.

## 4.3 Multiple caching nodes

Previously we showed that a single random point caching performs as good as or even better than all node caching in terms of the latency analysis. Then, the next question becomes how many copies need to be distributed in order to achieve a certain routing performance.

Fig. 6 plots the relative delay and the coverage as the number of copies in the network increases with different advertisement range $m$. We can achieve a near optimal routing performance (close to the shortest path routing) when copies are distributed randomly less then 1% of total nodes and the nodes advertise 2 hops at most ($m$=2) which defines the potential field less than 30% of the network. In other words, the rest 70 % of the network does not need to maintain the routing information for the content file.

# 5. DISCUSSIONS

This section discusses several design issues that we encounter to utilize the PBR approach.

## 5.1 Control overhead

Potential fields provide routing information for users to find content files which are available in the network. However, due to dynamic variation of the network, e.g., a network failure or a content deletion, a process that updates the potential fields according to the network state is necessary. We define control overhead as the amount of required control messages to update the potential fields to reflect the network condition in real time. There are two major design approaches for the control overhead issue, namely hard-state and soft-state approaches.

In the hard-state, each node possessing a content file is responsible for creating and deleting its potential field, and reacting to any network change such as a failure or a content deletion. In contrast, for the soft-state approach, a content holder generates an advertisement message periodically to update potential fields regardless of the network state. Default implementation for CATT adopts the hard-state approach to minimize control overhead since current version of CATT is assumed to run on a stable wired network.

## 5.2 Scalability & Feasibility

This issue has been addressed in DONA[1] that currently available technology (back in 2007) could easily support routing entries upto $10^{11}$ which yields a total storage requirement about 4TB under the assumption of 42 bytes per entry (40 bytes for the name and 2 bytes for a next-hop). In CATT, each entry includes the name of content and its potential value (a real number with one or two decimal points) so that the storage requirement is pretty much similar to the case of DONA. In this sense, CATT can be easily realized also within currently available technology.

However, it may be arguable that the number $10^{11}$ is not large enough to represent all content files in current Internet. For this argument, we propose one extended use case of the PBR. In this use case, potential fields are defined not in the entire network but only in the part of the network similar to the one in the middle of Fig. 3. Each node possessing content files either in repository or in cache creates their potential fields within limited range. Then, a user request is naturally attracted to one of the potential fields while the request is forwarded based on some mechanisms. One of the mechanisms can be a random forwarding. The results in Fig. 4 and Fig. 6 have already demonstrated that the random forwarding on a partially defined potential field can provide a near optimal routing solution. Moreover, each node does not need to create potential fields for all content files that the node possesses. It selects content files as many as the node can support based on a criteria such as popularity or access time, etc. On the other hand, we may simply assume similar to [13] that a user request is always forwarded to the repository where the original content file exists. In this sense, the PBR plays a subsidiary role for any main routing mechanism which boosts the caching capacity of ICN.

# 6. CONCLUSION

An ICN architecture named CATT was introduced based on the PBR which was proposed to achieve several design goals in the context of ICN such as availability, adaptability, diversity, and robustness.

PBR achieves availability by incorporating not only an original content file published in repository but also all copies in caches into the routing process. Especially, it adaptably takes into account copies in caches which tend to have a high volatile behavior due to a replacement scheme. In addition, PBR provides a mechanism to select a content file based on not only proximity but also its quality which makes the selection process rather diverse. Most importantly, due to its implementation based on a fully distributed algorithm, it is robust against a single point failure scenario.

The simulation results showed that the PBR with a random forwarding can reduce the size of routing table by a factor of 1/3 with reasonable degradation of routing performance. In addition, the PBR can take advantage of only a few copies that are randomly distributed in the network. For instance, making random replications on 1% of total nodes which advertise 2 hops at most ($m$=2) can achieve a near optimal routing performance which demonstrates how the PBR can achieve availability.

# 7. REFERENCES

[1] T. Koponen, A. Ermolinskiy, M. Chawla, K. H. Kim, I. Stoica, B. gon Chun, and S. Shenker, "A data oriented (and beyond) network architecture," *Proc. of SIGCOMM*, Kyoto, Japan, Aug 2007.

[2] C. Dannewitz, "NetInf: An Information-Centric Design for the Future Internet," *Proc. 3rd GI/ITG KuVS Workshop on The Future Internet*, Munich, Germany, May 2009.

[3] K. Visala, D. Lagutin, and S. Tarkoma, "LANES: An Inter-Domain Data-Oriented Routing Architecture," *Proc. of the 2009 workshop on Re-architecting the Internet*, Rome, Itary, Dec 2009.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," *Proc. of the 5th international conference on Emerging networking experiments and technologies*, Rome, Italy, Dec 2009.

[5] R. G. C. Intanagonwiwat and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proc. of Mobile computing and networking*, Boston, USA, Aug 2000.

[6] Project CCNx. [Online]: http://www.ccnx.org/

[7] A. Basu, A. Lin, and S. Ramanathan, "Routing using potentials: a dynamic traffic-aware routing algorithm," *Proc. of SIGCOMM*, Karlsruhe, Germany, Aug 2003.

[8] P. Kumar, J. Kuri, P. Nuggehalli, M. Strasser, M. May, and B. Plattner, "Connectivity-Aware Routing in Sensor Networks," *Proc. of SensorComm 2007*, Valencia, Spain, Oct 2007.

[9] V. Lenders, M. May, and B. Plattner, "Service discovery in mobile ad hoc networks: A field theoretic approach," *Pervasive and Mobile Computing*, vol. 3, pp. 343–370, 2005.

[10] S. Jung, M. Kserawi, D. Lee, and J.-K. K. Rhee, "Distributed potential field based routing and autonomous load balancing for wireless mesh networks," *Comm. Letters.*, vol. 13, pp. 429–431, 2009.

[11] J. Zhang, Q. Wu, F. Ren, T. He, and C. Lin, "Effective Data Aggregation Supported by Dynamic Routing in Wireless Sensor Networks," *Proc. of the IEEE ICC*, Cape Town, South Africa, May 2010.

[12] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," *Proc. of Applications, technologies, architectures, and protocols for computer communication*, Cambridge, Massachusetts, USA, Sep 1999.

[13] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, Best-Effort Content Location in Cache Networks," *Proc. of the IEEE INFOCOM 2009*, Rio de Janeiro, Brazil, Apr 2009.