Journal of Network and Computer Applications 50 (2015) 159–167

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Design and implementation of ICN-enabled IEEE 802.11 access points as nano data centers

S. Eum [a,*], Y. Shoji [a], M. Murata [b], N. Nishinaga [a]

[a] National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
[b] Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

ARTICLE INFO

Article history:
Received 31 January 2014
Received in revised form
26 May 2014
Accepted 16 July 2014
Available online 1 August 2014

Keywords:
ICN
Nano data center

ABSTRACT

Network operators have suffered from explosive increase in mobile users, especially who are eager to watch all forms of video. At the same time, the mobile users tend to experience unexpected delay and disruption due to content retrieval from inappropriate location and frequent handover. To address the problems, this paper introduces an application scenario of Information Centric Networking (ICN) named CATT (Cache Aware Target idenTification), more specifically ICN-enabled IEEE 802.11 wireless access points as nano data centers. We describe its design tenet and implementation, and current deployment over the virtualization platform of JGN-X in Japan. The synergy between ICN supporting mobility as the norm and widely deployed broadband IEEE 802.11 wireless access points makes this proposal attractive as a potential application scenario for ICN.

1. Introduction

The number of Internet transactions from mobile devices is expected to surpass those from wired machines over the next 5–10 years (http://www.ipsos-na.com/news/pressrelease.cfm?id=3049/). Moreover, all forms of multimedia traffic (TV, video on demand (VoD), Internet, and P2P) would continue to be approximately 90% of the global consumer traffic by 2015 (Cisco Visual Netowrking Index, 2010). For example, YouTube itself is responsible for a tremendous 17% of mobile data bandwidth usage and continues as the single most significant web-site for video streaming, accounting for 45% of total video streaming traffic in 2010 (Allot MobileTrends Report, 2010). With these observations, the following problems are expected in the current network environment:

First, mobile and fixed-line operators suffer from the increase of operational cost, especially in the back-haul links. According to Anand et al. (2009), 15–60% of redundant traffic was observed in enterprize and university access links. This observation is likely to occur in the mobile environment as well due to the significant increase in the number of Internet transactions from mobile devices. Second, mobile users tend to experience longer delay since highly popular content files are downloaded repeatedly from a distant content provider rather than being served from nearby storage points. Third, mobile users experience a service disruption or disconnection during handover since the service is operated on the current end-to-end connection oriented network architecture. In addition, the perpetual connection prohibits mobile users from retrieving a content file from the best location after the mobile users move to another network.

Each problem defined above can be handled by a unique solution. For instance, Content Delivery Networks (CDN) can be a solution for the second problem but not for the first and the third problem. One candidate to deal with all these problems simultaneously from a network architecture perspective is Information Centric Networking (ICN) (Jacobson et al., 2009; Dannewitz, 2009; Ain et al., 2009; Eum et al., 2012a), which has attracted much attention of network research community[1] recently due to its paradigm shift in the network from “end host” to “information”. The essence of ICN is the name based routing, which enables individual ICN nodes to be aware of user requests as well as the corresponding responses. Due to this awareness feature, ICN nodes can function as an independent content provider. For this reason, by deploying ICN nodes in the edge of the network, users can access content files more efficiently and economically by eliminating latency and numerous hops. Also, operators can save network

* Corresponding author.
E-mail addresses: suyong@nict.go.jp (S. Eum), shoji@nict.go.jp (Y. Shoji), murata@osaka-u.ac.jp (M. Murata), nisinaga@nict.go.jp (N. Nishinaga).

[1] Recent establishment of the Information Centric Networking Research Group (ICNRG) (http://irtf.org/icnrg/) within the Internet Research Task Force (IRTF).

http://dx.doi.org/10.1016/j.jnca.2014.07.031
1084-8045/

resource by reducing traffic or localizing bursty traffic caused by proliferated mobile devices. Moreover, a perpetual connectivity is no longer required between end-to-end hosts, which enables ICN to support mobility as the norm. More specifically, ICN communicates using the name of data object, which eliminates the need for end terminals to resolve the location of the data based on its name. This simplifies mobility management for the end hosts, especially when the data or the host holding the data is relocated during the communication.

In the light of the observations, we extend an ICN architecture named CATT (Cache Aware Target idenTification) (Eum et al., 2012a) to mobile environment to deal with the problems discussed previously. We design IEEE 802.11 wireless access points with CATT so that network operators can use it as a nano data center which can be complementing existing data centers. Due to the extension, a mobile user can retrieve content files from the network with low latency as well as high reliability even when the network condition becomes unstable or degraded. At the same time, a network operator can utilize network resource in more efficient manner. After the elaboration of the design tenet and implementation, we demonstrate the performance of the prototype implementation as the proof of concept on a test bed.

This paper is organized as follows. In Section 2, we review related research works and justify the rationale why IEEE 802.11 access point is selected as a nano data center. In Section 3, we present simulation results to understand what benefits could be achieved from the proposal. In Section 4, we describe the design components of the proposal. This is followed by the description of the functional blocks and its implementation in Section 5. In Section 6, we demonstrate the operation of the prototype implementation based on experiments. Finally, we conclude this paper in Section 7.

2. Related works: pushing resources to the edge of the network

Centralization of computing and storage resources has been a major trend for the design of current Internet services since the cost per unit operation decreases as the size of the system increases – the economies of scale. However, the centralized design principle has also introduced problems such as high energy consumption due to heat dissipation[2] and difficulty of supporting delay-sensitive applications including the emerging Internet of Things (IoTs).

On the other side of the design principle, the decentralized approach has gained attention to mitigate the problems with a motto of pushing resources to the edge of the network. This design trend has become attractive since the manufacturing costs of processing units and memories have been dramatically reduced, which results in almost ubiquitous availability of processing and memory to the edge of the network.

In Laoutaris et al. (2008), the authors introduced a distributed storage platform named nano data centers, which creates a fully distributed service platform based on tiny managed "servers" located at the edge of the network. It reduces latency and improves the user experience by providing rich content faster and more economically. The full control over the distributed computing and storage entities makes the approach different from uncontrollable P2P cases. In addition, due to the fully distributed deployment and the use of existing edge devices, they become more energy efficient than centralized conventional approaches (Valancius et al., 2009). However, the nano data center solution mainly focuses on how to utilize the storage function of edge nodes to reduce the energy consumption, especially in the application level approach.

On the other hand, our proposal provides not only storage function but also processing function. Moreover, we select IEEE 802.11 wireless access points as the deployment location for the nano data center scenario. IEEE 802.11 generally known as WiFi is probably the most widely accepted broadband wireless networking technology so that many of mobile devices recently in production are equipped with IEEE 802.11 series. It supports up to massive gigabit throughput (http://www.qualcomm.com/media/documents/files/ieee802-11ac-the-next-evolution-of-wi-fi.pdf) that provides the highest transmission rate among standard wireless technologies. Also, the transmission range of a typical WiFi device is up to 100 m (http://focus.ti.com/pdfs/vf/bband/coexistence.pdf) but its exact transmission range can be extended to several kilometers depending on transmission power and surrounding environments. Also, IEEE 802.11 access points are widely deployed already near mobile users so that it does not require almost any capital expenditure for its deployment, e.g., real estate cost, which is the major cost associated with the deployment of data centers.

In Bonomi et al. (2012), the authors extended the clouding computing to the edge of the network and named it as fog computing, which provides storage as well as processing function, especially for the future IoT applications. This system is similar to our proposal in a sense that both storage and processing functions are pushed to nearby end users. However, the fog computing is based on a systematic approach, which requires additional system components to realize the functional blocks such as deep packet inspection for context awareness. On the other hand, ICN is a network architectural approach which provides context awareness as its mandatory function. Figure 1 illustrates the operational scenario of the overall system which supports not only a pull-based caching mechanism[3] but also a push-based caching mechanism.[4] Thus, the system can deliver a nearby copy at WiFi access point to users if available as shown in the left. In particular, when a mobile user switches to another WiFi access point while retrieving a content as shown in the right, the system supports the mobile user to find the same content nearby the new WiFi access point and retrieve the content from the point.

3. What benefits from the ICN-enabled WiFi access point?

Prior to the design and implementation of a system, the estimation of the benefits that the proposed idea delivers is a prerequisite condition. For this reason, we carried out simulation studies using a developed event driven simulator, with the goal of demonstrating the benefits of ICN-enabled WiFi access point from the following perspectives: First, a caching function is pushed to nearby end users and so the end users experience less delay due to the retrieval of content files from the nearby caching point. Second, the name based communication of ICN with the receiver-driven transmission enables mobile end users to retrieve content file with a cost effective manner even a complete disconnection of session caused by, e.g., handover.

The simulator has three components, namely content provider, access point, and mobile user. The link bandwidth between the content provider and the access point is set to *WD*, and the link bandwidth between the access point and the mobile terminal is

[2] At least 20–50% of the total power consumption (http://www.google.com/corporate/datacenters/measuring.html).

[3] Content caching while being downloaded from provider to a mobile user.

[4] Providers place popular contents at WiFi access points nearby users.

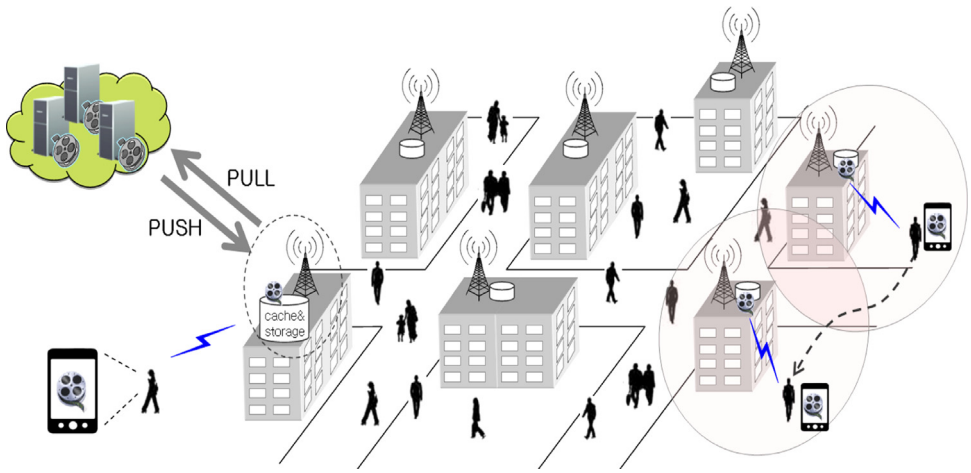

Fig. 1. IEEE 802.11 wireless access points with ICN as nano data centers.

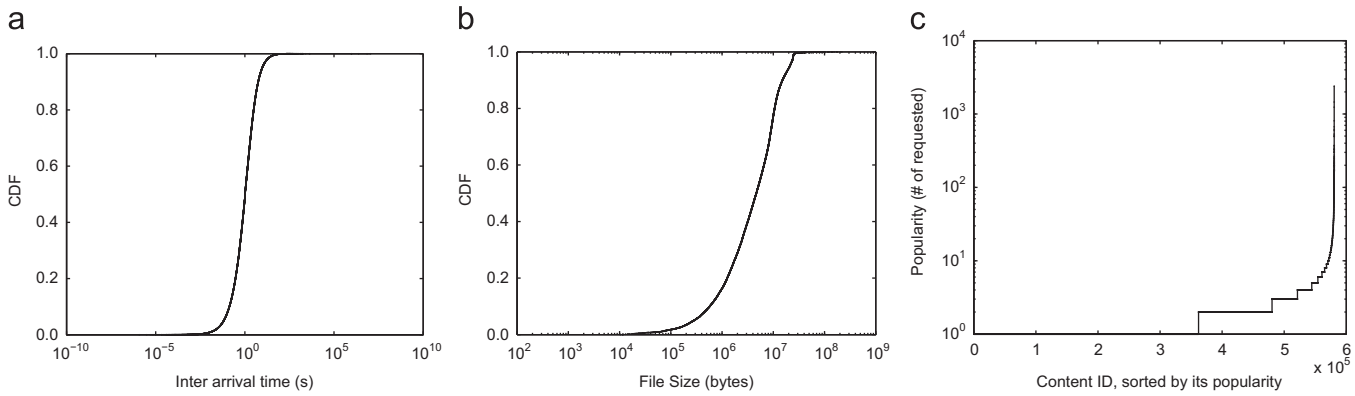

Fig. 2. Characteristics of the trace of YouTube video clips obtained from UMASS Trace Repository, Online: inter-arrival times, file sizes, and their popularity. The average file size is 6.67 Mbytes, and its max and min sizes of the files are 218 Mbytes and 10 Kbytes, respectively. The most popular video clip was requested 2396 times, e.g., total number of requests were 1 205 783 ≈1.2 millions. (a) CDF of inter-arrival times. (b) CDF of file sizes. (c) Popularity of each file.

set to *WL*. We assumed that the link bandwidths *WLs* between the access point and all mobile users are same. This simulation is based on YouTube trace files from the UMASS trace repository (http://traces.cs.umass.edu/index.php/Network) which were collected from the campus network between June 2007 and March 2008. The trace based simulation provides more realistic results comparing to a traffic model based simulation which was given in our previous work (Eum et al., 2012b). We extracted the arrival time of each video-clip request and the size of the video-clip from the trace file, especially the size of the video clip is used to estimate its service time with a given link capacity. For the ICN scenario, each video clip is segmented into chunks (4 Kbytes) which can be identified by its name in ICN domain. The access point has a cache which has a limited size and so Least Recently Used (LRU), Least Frequently Used (LFU), and Random replacement algorithms are incorporated into the cache. LRU is used in the simulation unless otherwise noted. Then, the mobile user requests a video-clip in the content provider (the arrival time and the size of the requested video-clip are from the trace files).

Since the UMASS trace repository does not include all traces which were analyzed in Zink et al. (2009), and the properties of the trace are not clearly described, firstly we analyze the properties of the trace which we use for this simulation. Figure 2 plots CDFs of inter-arrival times of YouTube video clips and their payload sizes. We transform the file size distribution into service time distribution. Moreover, the popularity of each file is plotted based on the number of requested times. For this particular set of YouTube traffic trace, we analyze the expected benefits of the proposal.

3.1. Cache hits

Figure 3 shows the cache hit probability when the size of cache increases from 1 Gbyte to 1 Tbyte. This simulation assumes that cache is located at the aggregation device which combines multiple network connections in parallel from individual access points. Depending on the cache replacement algorithm, even small cache size, e.g., 1 Gbyte can achieve around 0.15 cache hit rate which implies that 15% of total requests from mobile users for YouTube video clips could be served from wireless access points directly. This result can be interpreted in various manners, e.g., the reduction of energy consumption due to the installation of nano data center at WiFi access points since energy saving mainly comes from reducing hop counts by serving data object near by end users (Lee et al., 2011). From a user perspective, this result also shows that end users can download a content file with less delay and less disturbance when the performance of wired network is fluctuated or degraded. In this manner, the network operator can derive some revenues by delivering the content from their access points under the agreement with content provider.

3.2. Content transmission cost

In Lo et al. (2004), the cost of traffic transmission was defined as the product of the traffic volume and the hop distance. Since the YouTube traffic trace does not provide the distance information, we replace the hop distance with link bandwidth and define it as

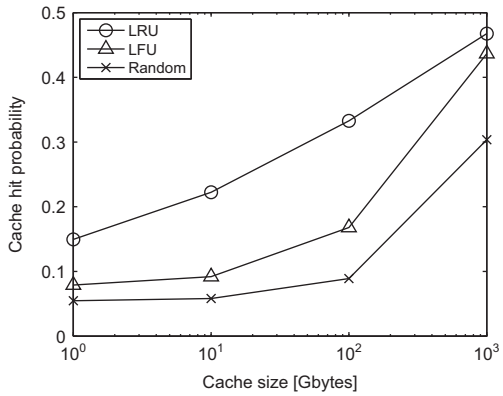

Fig. 3. Network resource saving as a function of cache size at the edge of the network. LRU: Least Recently Used, LFU: Least Frequently Used.

weighted transmission cost as follows:

$$\text{weighted transmission cost} = \text{traffic volume [bytes]}/(WD\ \text{[bps]} + WL\ \text{[bps]}) \quad (1)$$

where *WD* represents the link bandwidth between content provider and access point, and *WL* shows the link bandwidth between access point and mobile terminal.

Figure 4 plots the weighted transmission cost per request while the downloading YouTube video clips experience interruption, e.g., session disconnection at ICN-enabled and conventional wireless access points. The results demonstrate that this proposal reduces the weight transmission cost by more than 10% compared to the conventional case. There are two reasons that the proposal reduces the transmission cost. One is the cache at the access point and the other is the receiver driven downloading. Due to the caching function at the wireless access point, the transmission cost involved with wired section can be reduced when many of requested video clips are retrieved from the caching point directly. Moreover, the ICN architecture named CATT is operated based on the receiver driven transmission where the receiver continuously sends the requests for specific segments[5] of the requested video clip. In detail, there is a one-to-one correspondence between a request and a response message. Each request asks for a segment of the content. Since a fraction of the segment are requested by a sequence of requests, we can achieve a transport protocol that is similar to TCP by controlling the sending of the request messages. When a request for a segment is failed or lost, it firstly reduces the sending rate, similar to TCP congestion window, and then simply re-sends the request to retrieve the lost segment. This approach of realizing a transport protocol has been already proposed by CCNx (Jacobson et al., 2009) and demonstrated as a feasible approach. For this reason, although an interruption occurs while downloading a video clip, the receiver is able to retrieve the video clip from the segment which has not been retrieved. On the other hand, the conventional approach retrieves the video clip from the beginning whenever a session is lost.

4. Design components

This section describes our design choices for the ICN-enable IEEE 802.11 access points and elaborates on the rationales behind these choices.

[5] E.g. 4 Kbytes per segment.

4.1. Naming

Designing the name of a data object may be the most important design piece for this proposal. A large number of identical copies of a data object are distributed in different locations in ICN due to its ubiquitous caching capability. Thus, the name of a data object should be persistent and unique so that users can access a data object simply based on its unique name regardless of its location. The name of a data object mainly comprises the "*label* (L)" of the data object and its "*principal* (P)" information. The label of the data object (L) is given by the publisher which is unique under the publisher domain with which the data object is logically associated. It includes meta data which provides the attributes of the data object. The principal (P) information identifies the publisher of the data object, which is the cryptographic hash of the publisher's public key. Thus, the combination of (L:P) ensures the globally unique name of the data object. Here, the definition of publisher includes any host, e.g., nano data center, which is authorized by the publisher to server its data object. Thus, the authorized host holds the data object with its signature and the public key of the publisher. For example, assuming that a user requests a data object with its name (L:P) and receives the requested data object with its signature and the public key of the publisher. Then, the user can authenticate whether the data object is from the publisher by comparing the hash of the received public key with the principal (P) information. After the authentication, the user hashes the received data object into a message digest and compares it with the message digest from the decrypted signature (using the received public key). If the message digests are same, the user confirms that the data object has not been changed since it was signed.

Other than using the principal (P) information to verify the integrity between the data object and its name based on "self-certifying" approach, it is used in the following cases as well: First, it is used to verify the staleness of data objects at storage points. The nano data center should be able to verify the staleness of the data object before responding it to any user request. Since a common staleness verification approach of data object is to consult the publisher of the data object, the nano data center should be able to recognize the publisher of the data object by simply examining the name of the data object. Second, the principal information enables the network to make cluster of data objects according to their publisher information so that routing entries can be aggregated based on the information to achieve more scalable routing. In other words, the routing entries in the inter-publisher domain will be reduced by the average number of content files that each publisher can manage. For instance, if there are 10^{11} objects in the world wide[6] and each publisher can manage 10^4 on average, the maximum size of the routing entries in the inter-publisher domain will be 10^7, which is within the reach of current technology.

Moreover, the control overhead of the routing algorithm is closely related to the naming scheme which enables the routing algorithm to aggregate the routing entries. In this sense, the control overhead is proportional to the number of routing entries. Thus, we expect that the overhead due to publisher advertisement can be reduced by a factor of several orders comparing to the overhead due to the flooding to the entire network.

4.2. Data consistency and staleness verification

Another design issue is how to provide a consistency and coherence model for data objects at nano data center along with their revision handling and updating protocols. We incorporate

[6] The number of originally published content files that ICN is expected to support was estimated as 10^{11} back in 2007 (Koponen et al., 2007).

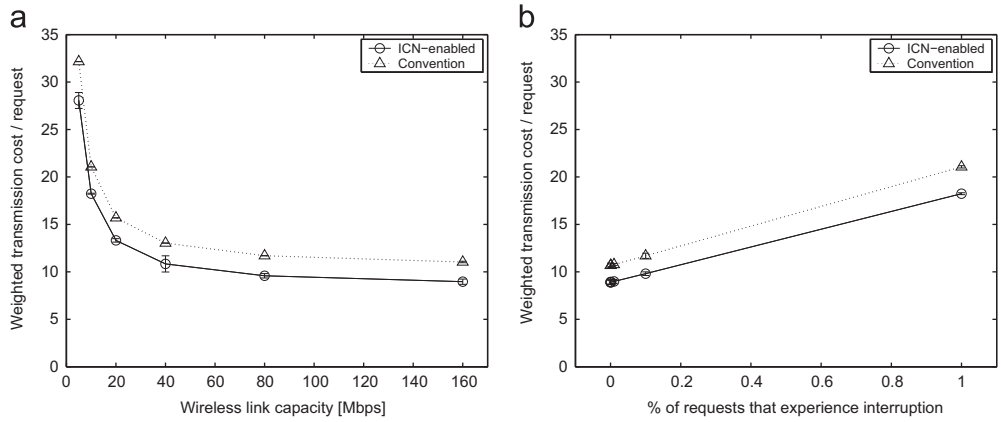

Fig. 4. Weighted transmission cost per request – (a) as *WL* increases, 1% of requests experience interruption (*WD*:10 Mbps), (b) as the network disturbance increases (*WD*:10 Mbps) with 95% confidential interval.

two staleness verification algorithms into the system, namely ICN_PER and ICN_CB which provide strong synchronization of data objects between their publishers and nano data centers located at access points (Eum et al., 2013b). In order to make the paper self-contained, here we briefly describe the operations.

The first approach, ICN_PER, is based on a pull based mechanism which consults the original data publisher before using any data object at nano data center. Since ICN follows a receiver-driven communication model where receivers can regulate if and when they wish to receive data (http://tools.ietf.org/html/draft-kutscher-icnrg-challenges-01; Duan et al., 2005), the pull based mechanism of ICN_PER is well matched to the receiver-driven model of ICN. However, it may trigger some complexity in the publisher side, e.g., scalability bottleneck since the provider may need to respond to many unnecessary requests for staleness verification. To enable this approach, the naming scheme should include "principal" information so that the nano data center can verify the staleness of data objects by asking to the publisher directly. The second approach, ICN_CB, is based on a push based mechanism that the publisher pushes an update message to nano data centers in order to invalidate staled data objects. In ICN literature, the push mechanism is realized based on mainly two different approaches. One is to use the location information, e.g., IP address and so an update message is pushed to the destination node using its locator. The other is based on a breadcrumb routing where the update message recipient creates a trail of breadcrumbs to the update message creator and so when an update message occurs, it follows the trail consuming the breadcrumbs to reach to the update message recipient. NetInf (Dannewitz, 2009) and PURSUIT (Ain et al., 2009) belong to the former while CCNx (Jacobson et al., 2009) belongs to the latter which CATT supports. Thus, a nano data center which holds a data object creates a trail of breadcrumbs to the publisher of the data object so that the publisher can push an update message into the nano data center using the trail.

4.3. Access control

Access control is one of the critical issues that this ICN application should overcome. ICN generally deals with the issue by encrypting each content to discourage unauthorized users from accessing the network. In other words, everyone is allowed to retrieve any content in ICN but only authorized user can decrypt the content in principle. Although, this encryption based access control mechanism corresponds to the design principle of ICN,[7] it introduces the concerns of security and network management. For instance, malicious mobile users may flood large number of fake requests to multiple access points with the intention of interrupting them. Unfortunately, such request flooding attack is hard to be detected since it is difficult to identify the source of the traffic due to non-existence perpetual connectivity between end-to-end hosts in ICN. More specifically, ICN benefits from multiple copies of a content file which are distributed in the network. Thus, fractions of a content file might be retrieved from multiple in-network caches simultaneously. Due to this content dispersion, it is required that individual ICN entities support a complicated access control mechanism as well as be allowed to access to the user management system, which is unrealistic considering the capability of each ICN entity. In addition, the operator who provides the network resource as well as the publisher/owner of data objects tend to demand the user statistics of accessing the network or the caching data objects in order to bill or make an invoice for ICN traffic. To deal with the security as well as the network management issues, user or device identifier better be incorporated into an access control mechanism for ICN. For this reason, we restrict user access to the network based on the network port authentication through the use of 802.1X, which is a common native authentication framework for IEEE 802.11 network.

4.4. Routing

Here we adopt Potential Based Routing (PBR) which was introduced in our previous work (Eum et al., 2012b). PBR defines a potential value per an object, e.g., a content file, or a publisher, in each node and creates a potential field connecting the values at individual nodes as shown in Fig. 5. Then, a user request is forwarded in a direction of reducing the potential value to find the object. To define the potential value at each node, an advertiser, e.g., content holder, issues an advertisement message which has mainly two fields: content quality Q and distance d. The content quality Q is defined by the advertiser prior to its issue. For instance, a node with high processing power or high capacity of out-going links can have large value of Q, which represents a better service provider. The field of distance d denotes the distance, e.g., hop counts between the advertiser and the node which receives the advertisement message. Thus, when the advertisement message is issued, the distance field is set to zero and increases by one whenever the message is forwarded. Based on the values of Q and d in the advertisement message, each node can calculate its own potential value (ψ_n) with the model of $\psi_n = -Q/d$. With a different interpretation of the parameters, the

[7] Security is built into the data object itself rather than a connection or a device.

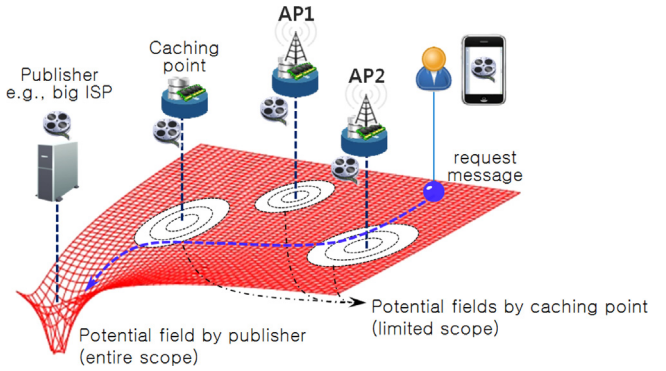

Fig. 5. Potential based routing: potential fields of publisher and caching data objects that belongs to the publisher. Mobile terminal takes advantages of the PBR to select an access point in order to download a content file in an optimal way.

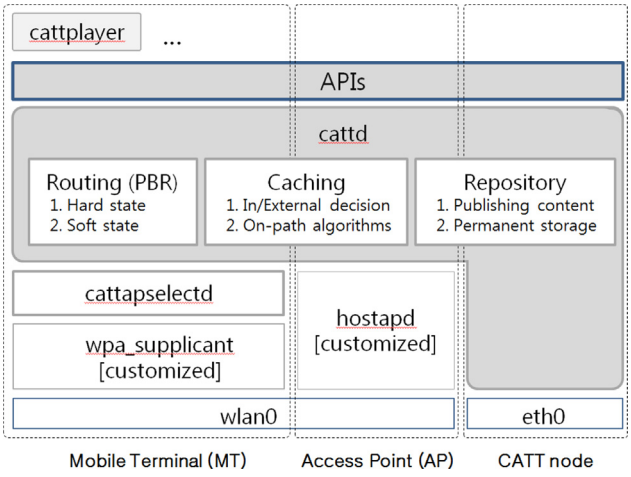

Fig. 6. System block diagram.

model to calculate the potential value can deal with various scenarios. For instance, when the forwarding capabilities of the intermediate nodes should be taken into account to calculate the potential value, the actual transmission delay between the advertiser and the receiver of the advertisement message can be a good candidate to define the parameter d rather than the hop counts between them.

Two different types of potential fields are considered in this system. One is constructed by each publisher using its publisher identifier so that any request for a data object is initially forwarded to the publisher of the data object. The other is constructed by each caching or storage point including nano data center at the access point using the name of the data object. Figure 5 illustrates the operation of the PBR. While the publisher floods its advertisement to the entire network (among publishers, e.g., ISP level), ICN nodes holding a copy of data object flood its advertisement within limited scope. Flooding an advertisement message to the entire network is an expensive operation so that it is only triggered by a special event, e.g., a topological change due to a failure (hard state). On the other hand, each ICN node with a copy of data object floods the advertisement message periodically (soft state) within limited area so that it adaptively changes the scope of advertisement depending on its capability or willingness to serve user requests, e.g., a high capacity node may create a potential field which has large advertisement scope to serve more user requests. In the current implementation, ICN node selects some popular data objects, e.g., top 5%, from its cache based on an integrated LFU or LRU algorithm, and advertises them within the scope which is initially set as a system parameter. The number of advertised caching data objects or their advertisement scopes can be dynamically changed during the operation of the system. Due to this self-organizing operation, the control overhead traffic can be minimized comparing to other cooperative caching mechanisms (Eum et al., 2013a).

5. Implementation

Figure 6 shows the system block diagram. For this implementation, three system modules, namely cattapselectd, wpa_supplicant, and hostapd, are newly developed and extended from existing free software packages.

5.1. CATT daemon – cattd

cattd consists of three main functions, namely caching, repository, and routing. The caching function provides CATT nodes with an application independent caching function. Its roles are to cache any content files which pass through associated nodes and to respond to any content request from a mobile terminal. Caching decision at wireless access point is determined internally or externally. Each access point determines which content should be cached/stored depending on its internal policy, e.g., size or type of data objects. At the same time, the external process determines whether a downloading data object is allowed to be cached/stored. We are considering two types of operations; pull and push. For the pull based approach, a caching/storaging decision is preferably made while a content file is downloaded from the provider to mobile users. On the other hand, the push based caching approach pushes some popular data objects into the nano data centers at wireless access points. The repository is responsible for managing storage space where users publish content files or register some popular caching content files to repository permanently. Lastly, the routing block routes a user request based on the name of the requested content file. For this routing mechanism, PBR is used as described in Section 4.4.

Current implementation of the CATT daemon aims for demonstrating a proof of concept and so the efficiency of, e.g., forwarding or caching functions was not in a high priority. However, the performance can be practically improved by optimizing the current implementation. For instance, additional hardware can be deployed to speed up the retrieval of data from the cache. According to Ko et al. (2012), multiple SSDs in single node can achieve 100 Gbps of aggregate throughput. In addition, an efficient data structure can be designed to support fast lookup for ICN packet and cache, e.g., using bloom filter (Varvello et al., 2012).

5.2. CATT access point selection daemon – cattapselectd

Due to the PBR, each wireless access point has a routing entry for a data object, which shows the accessability information (potential value) to any data object. In our system, the information is used for a mobile terminal to select a wireless access point in order to download the data object in an optimal manner.

To realize this function, the software module, “cattapselectd”, is newly developed. Its role is to carry out the access point selection process which includes initiating active or passive scanning and connection setup process, to obtain the potential values from access points. In the active scan, mobile terminal broadcasts a probe request frame with the name of the requesting data object. When access points receive the probe request, they send a response frame which includes the corresponding potential value. Then, the mobile terminal, which received the information, selects an access point which provides the best accessability to the data object.

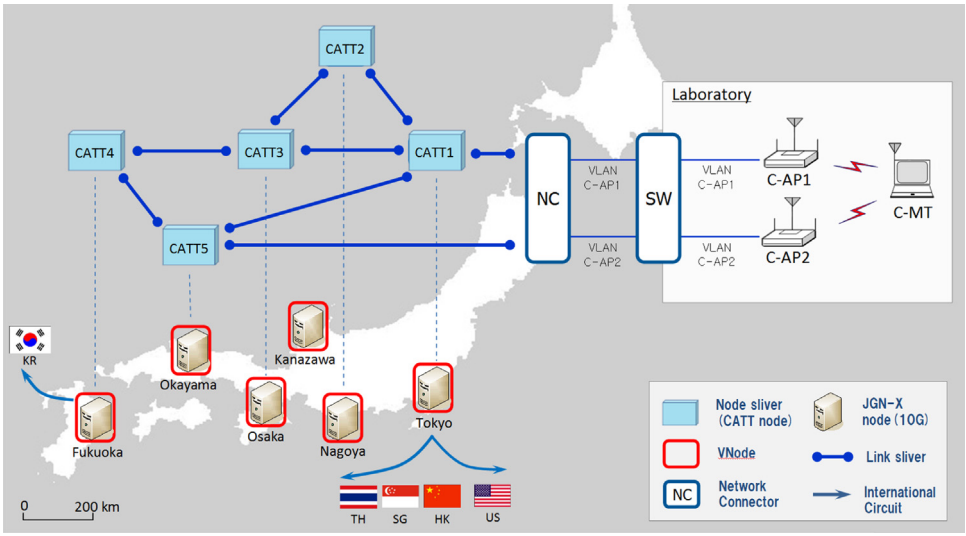

Fig. 7. Test-bed deployment for experiments on JGN-X.

In the passive scan, each access point periodically broadcasts the information using IEEE 802.11 beacons frame which is used to announce the presence of a WiFi network. Thus, when a mobile terminal receives the information in the beacon frame, it maintains the information in its cache, and refers to the values to select an access point. Due to the caching values, mobile terminal can instantly access to the value and select an access point without much delay.

5.3. *Customized modules – wpa_supplicant and hostapd*

Free software packages: *hostapd* and *wpa_supplicant* (http://hostap.epitest.fi/) are used to carry out the network port access control based on 801.1X authentication and association by exchanging probe frames (request/response/beacon) between mobile terminal and access points. The Host Access Point daemon, hostapd, provides strong WPA2 encryption and authentication on Linux-based wireless access points. In order to integrate them with CATT, the packages are customized. For instance, we use the vendor specific field of the probe frames to carry the information from access points to mobile terminal, where the IEEE 802.11 standard allows vendors to add 256 byte length of information. Although SSID (Service Set Identifier) field is an easy option,[8] there are many wireless devices which display the SSID information on its screen so that it may interfere with other applications which use SSID information.

6. Performance evaluation on testbed

We have deployed the prototype of CATT over the virtualization platform of JGN-X (Japan Gigabit Network eXtreme) which corresponds with GENI (Global Environment for Network Information) in US or FIRE (Future Internet Research and Experimentation) in EU project.

Five virtual nodes, CATT1 ...CATT5, across Japan are interconnected as a vnode overlay as shown in Fig. 7. For an initial deployment scheme of CATT architecture on the current Internet, we previously proposed to deploy CATT nodes at the edges of an autonomous system (AS) in Eum et al. (2012a). Thus, when we consider each city as an ISP, each CATT node in the figure can be regarded as a caching point which is located at the edge of the corresponding ISP. Through the caching function of CATT node, ISP caches transit traffic between them, which provides the cost effective approach for the ISP.

Since we propose to push the caching point located at the edge of the ISP to WiFi access point, two of the virtual nodes, CATT1 and CATT5, are respectively connected to wireless CATT access points, C-AP1 and C-AP2 through the network connector (NC), which connects between a virtual network on JGN-X and a real physical network. Due to the caching function at the WiFi access points, rapidly increasing content retrieval from mobile devices can be localized within an access network, which results in the reduction of network operational cost in the back-haul area (from an operator perspective) as well as the reduction of latency (from a mobile user perspective). The evaluation scenarios here aim for demonstrating the benefits of the proposal comparing with conventional content retrieval system based on HTTP. The first scenario uses downloading time as a performance metric which can be interpreted as the amount of time that network resource is occupied as well as the latency that mobile user experiences. The second evaluation presents the additional time required for the proposed scheme in order to select an optimized location for content retrieval from the mobile user.

For an experiment, we disable the links of CATT1-CATT3, CATT1-CATT5, and CATT4-CATT5 to create a line topology. Initially, a data object is published at C-AP1, CATT1, ..., CATT4, and they are downloaded from C-MT through C-AP1. For the purpose of comparison, we install an *apache* web server into the C-AP1, CATT1, ..., CATT4, and the same data objects are registered there. Then, C-MT downloads them using HTTP protocols (*wget*). We also consider two different scenarios; the wired link speed is faster than the wireless link speed and vice versa. The wireless links between C-APs and C-MT are connected through IEEE 802.11g which has the maximum speed of 54 Mbps. On the other hand, two different wired link speeds, 10 Mbps[9] and 100 Mbps, are set to the CATT wired network for this experiment.

Figure 8 shows the downloading time as a function of file size and the location that the files are published. Downloading time linearly increases in proportion to the file size for both CATT and HTTP. In terms of the location of file, HTTP downloading time is

[8] Most commercial access points provide a user interface to set the SSID.

[9] For the 10 Mbps network, we built a physical test bed in our lab. The line speed was intentionally degraded using a utility called "ethtool" (http://www.kernel.org/pub/software/network/ethtool/).

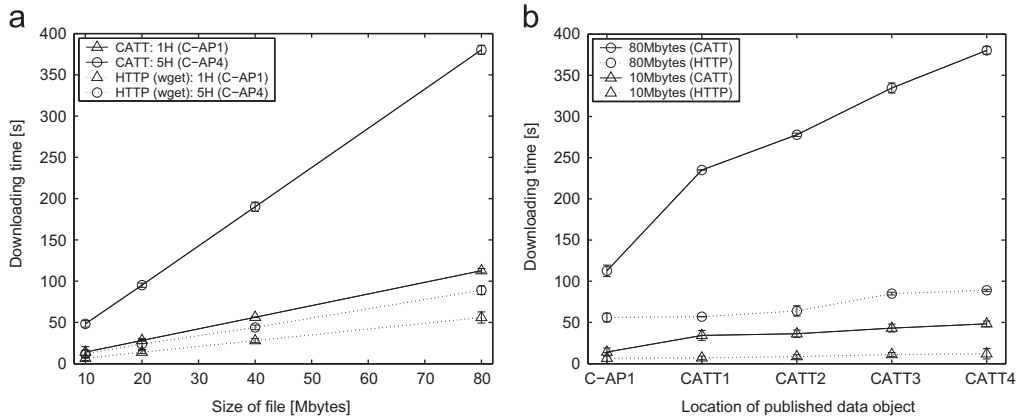

Fig. 8. Downloading time as a function of file size and published location using CATT and HTTP with 95% confidential interval.

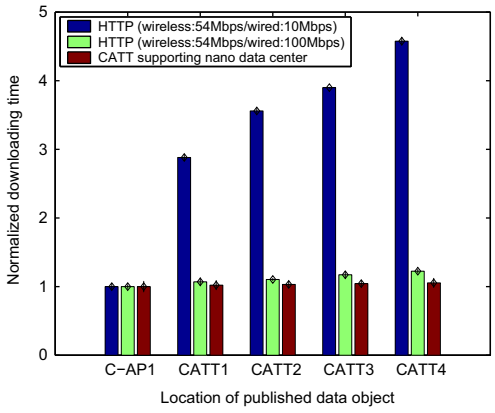

Fig. 9. Normalized downloading time for CATT and HTTP(wget) with 95% confidential interval.

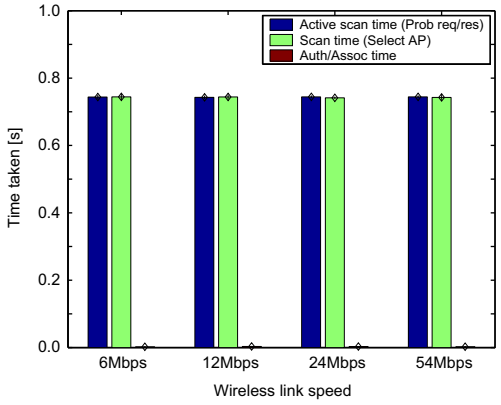

Fig. 10. Average scanning time with 95% confidence interval.

not directly influenced by the location while CATT downloading time has shown strong linear correlation with the location – hop counts. It is due to the *store and forward* function of CATT in which data object is sent to an intermediate node where the data object is kept (cached) first and then sent to the final destination or to another intermediate node. Since each intermediate node needs to wait until it receives all relevant packets to assemble them into an identifiable data unit,[10] it delays the actual downloading time. On the other hand, due to the cached data object in the intermediate nodes of CATT, when there are much redundant traffic, these traffic can be served from the intermediate nodes directly. For this reason, CATT can provide a higher performance than HTTP when much redundant traffic exists in the network.

Figure 9 shows the downloading times of the redundant files. The downloading times from different locations, e.g., C-AP1, CATT1, ..., CATT4, to C-MT are normalized by the downloading times from C-AP to C-MT for both CATT and HTTP cases. Since C-AP1 functions as a nano data center, which stores the data object during the first downloading attempt and uses the data object to respond C-MT requests directly from the second downloading attempt, the initial location of video object does not much affect the overall downloading time for CATT. On the other hand, when CATT is not enabled at C-AP, the downloading time gradually increases in proportional to the number of hop counts between C-MT and the content provider. This result also demonstrates that the ICN-enabled nano data center especially appeals to the case when the performance of wired network is fluctuated or degraded because it stabilizes the downloading time for mobile users regardless of the line speed of the network. Although, the downloading time is used as a performance metric for this comparison, it can be interpreted in various manners. For example, this result can be understood as the reduction of energy consumption due to the installation of nano data center at WiFi access points since energy saving mainly comes from reducing hop counts by serving data object near by end users (Lee et al., 2011).

Figure 10 shows the scanning measurements including scanning and auth/assoc times. For this experiment, we publish video objects at CATT1 and CATT5 with same Q value,[11] and vary wireless link speed between C-MT and C-APs dynamically, e.g., 54 Mbps, 24 Mbps, 12 Mbps, 6 Mbps so that C-MT selects an access point depending on the Q and the wireless link speed. The overhead time due to the additional scanning process (active scanning) takes around 0.7 s in the various range of wireless link speeds. Comparing to the downloading time of data object, especially large one, e.g., several hundred Mbytes, the overall scanning time can be negligible. For instance, the additional time for the active scanning (0.7 s) is far less than the time to download a large file from a remote content provider.

7. Conclusion

We have presented an ICN application scenario which takes advantage of both intrinsic features of ICN and the popularity of IEEE 802.11 wireless technology. The application scenario aims to follow current technical trend – pushing resources to the edge of the network.

[10] Each chunk of data object.

[11] Refer to Section 4.4.

We initially verified the expected benefits of the application scenario based on YouTube trace-based simulation. It showed that even a small size cache, e.g., 1 Gbyte near to mobile users can reduce overall traffic by 15%, which enables mobile operators to reduce its operational cost in the back-haul area. In addition, according to the performance metric called weighted transmission cost, the proposed scenario reduces the weight transmission cost by more than 10% compared to the conventional approaches. Encouraged by the simulation results, we prototyped the proposal and deployed over the virtualization platform of JGN-X (Japan Gigabit Network eXtreme) where we carried out some experiments to demonstrate its proof-of-concept.

Further research on this idea of ICN-enabled wireless access point as a nano data center is encouraged, especially in terms of energy issue. Moreover, special attention needs to be paid on an access control mechanism which protects not only data objects but also network resource abuse from malicious mobile users. Currently, we consider using the idea of Software Defined Networking (SDN) to provide an access control mechanism for this particular application scenario. For instance, SDN deals with the access control issue by reacting to external events by changing forwarding policies at the access points.

Acknowledgment

The authors would like to thank New Generation Network (NWGN) Project members for their constructive suggestions on this ICN research project. In addition, special thanks to Saito Yuki and Fukawa Masaki for helping us to build the virtualized test-bed on JGN-X.

References

Ain M, Trossen D, Nikander P, Tarkoma S, Visala K, Rimey K, Burbridge T, et al. D2.3 – architecture definition, component descriptions, and requirements. Deliverable, PSIRP 7th FP EU-funded project, 2009.

Allot MobileTrends Report. Global mobile data bandwidth usage in 2010 [Online]. Available: ⟨http://www.allot.com/⟩.

Anand A, Muthukrishnan C, Akella A, Ramjee R. Redundancy in network traffic: findings and implications. SIGMETRICS Perform Eval Rev 2009;20:37–48.

Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the Internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, Helsinki, Finland, August 2012.

Cisco Visual Networking Index. Forecast and methodology, 2010–2015 [Online]. Available: ⟨http://www.cisco.com/⟩.

Dannewitz C. NetInf: an information-centric design for the future Internet. In: Proceedings of 3rd GI/ITG KuVS workshop on the future Internet, Munich, Germany, May 2009.

Duan Z, Gopalan K, Dong Y. Push vs. pull: implications of protocol design on controlling unwanted traffic. In: Proceedings of SRUTI'05: the steps to reducing unwanted traffic on the Internet on steps to reducing unwanted traffic on the Internet workshop (USENIX Association), Cambridge, MA, USA, July 2005.

Eum S, Nakauchi K, Murata M, Shoji Y, Nishinaga N. CATT: cache aware target identification for ICN. IEEE Commun Mag 2012a;50:60–7.

Eum S, Nakauchi K, Murata M, Shoji Y, Nishinaga N. CATT: potential based routing with content caching for ICN. In: Proceedings of the SIGCOMM 2012 ICN workshop, Helsinki, Finland, August 2012b.

Eum S, Nakauchi K, Murata M, Shoji Y, Nishinaga N. Potential based routing as a secondary best-effort routing for Information Centric Networking. Comput Netw 2013a:3154–64.

Eum S, Nakauchi K, Murata M, Shoji Y, Nishinaga N. Staleness verification of caching data in ICN. In: Proceedings of the 4th international conference on ICT convergence 2013 (ICTC 2013), Jeju Island, Korea, October 2013b.

ethtool: utility for controlling network drivers and hardware [Online]. Available: ⟨http://www.kernel.org/pub/software/network/ethtool/⟩.

Google Data Center Power Usage Efficiency [Online]. Available: ⟨http://www.google.com/corporate/datacenters/measuring.html⟩.

hostapd and wpa_supplicant [Online]. Available: ⟨http://hostap.epitest.fi/⟩.

ICN Research Challenges [Online]. Available: ⟨http://tools.ietf.org/html/draft-kutscher-icnrg-challenges-01⟩.

ICNRG: Information Centric Networking Research Group [Online]. Available: ⟨http://irtf.org/icnrg/⟩.

IEEE802.11ac: the next evolution of Wi-Fi standards [Online]. Available: ⟨http://www.qualcomm.com/media/documents/files/ieee802-11ac-the-next-evolution-of-wi-fi.pdf⟩.

Jacobson V, Smetters D, Thornton J, Plass M, Briggs N, Braynard R. Networking named content. In: Proceedings of the 5th international conference on emerging networking experiments and technologies (ACM CoNEXT '09), Rome, Italy, December 2009.

Ko B, Pappas V, Raghavendra R, Song Y, Dilmaghani R, Lee K, et al. An information-centric architecture for data center networks. In: Proceedings of the second edition of the ICN workshop on information-centric networking, Helsinki, Finland, August 2012.

Koponen T, Ermolinskiy A, Chawla M, Kim K, Stoica I, Chun B, et al. A data-oriented (and beyond) network architecture. In: Proceedings of the 2007 conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM '07), Kyoto, Japan, August 2007.

Laoutaris N, Rodriguez P, Massoulie L. ECHOS: edge capacity hosting overlays of nano data centers. SIGCOMM Comput Commun Rev 2008:51–4.

Lee U, Rimac I, Kilper D, Hilt V. Toward energy-efficient content dissemination. IEEE Netw Mag 2011:14–9.

Lo S, Lee G, Chen W, Liu J. Architecture for mobility and QoS support in all-IP wireless networks. IEEE J Sel Areas Commun 2004:691–705.

Mobile phones could soon rival the PC as world's dominant Internet platform [Online]. Available: ⟨http://www.ipsos-na.com/news/pressrelease.cfm?id=3049/⟩.

Texas Instruments White Paper: WiFi and Bluetooth [Online]. Available: ⟨http://focus.ti.com/pdfs/vf/bband/coexistence.pdf⟩.

UMASS Trace Repository: YouTube traces from the campus network [Online]. Available: ⟨http://traces.cs.umass.edu/index.php/Network⟩.

Valancius V, Laoutaris N, Massoulié L, Diot C, Rodriguez P. Greening the Internet with nano data centers. In: Proceedings of the 5th international conference on emerging networking experiments and technologies, Rome, Italy, December 2009.

Varvello M, Perino D, Esteban J. Caesar: a content router for high speed forwarding. In: Proceedings of the SIGCOMM 2012 ICN workshop, Helsinki, Finland, August 2012.

Zink M, Suh K, Gu Y, Kurose J. Characteristics of YouTube network traffic at a campus network – measurements, models, and implications. Comput Netw 2009:501–14.