# On the Cache-and-Forward Network Architecture

Lijun Dong Hongbo Liu Yanyong Zhang Sanjoy Paul Dipankar Raychaudhuri
WINLAB, Rutgers University
Technology Center of New Jersey
671 Route 1 South
North Brunswick, NJ 08902-3390
{lijdong, hongbol, yyzhang, sanjoy, ray}@winlab.rutgers.edu

***Abstract*—In order to meet the increasing demands of content dissemination in Internet, we propose a novel architecture for the future Internet called Cache-and-Forward (CNF), which transports content as "packages" in a hop-by-hop manner towards the destination, instead of transporting a stream of fragmented packets along an established TCP/IP connection. In this paper, we discuss how the CNF network architecture can be designed for efficient content retrieval. We first introduce several specific services provided in CNF network which are centered around content handling and mobile access. We then give an overview of the CNF protocol stack, which is built on top of IP, and consists of a data plane and a control plane. We provide detailed descriptions of each protocol within both planes. Then we present two caching algorithms, where one involves each CNF router making independent decisions on content caching while the other coordinates node caching within an autonomous system (AS) through hashing. Finally, we gave the initial simulation results to show the performance benefits of hop-by-hop transport and content caching.**

## I. Introduction

In the past few decades, the Internet have enabled a large array of applications, which have profoundly changed the way we interact with the rest of the world. However, as applications become more demanding, and as new technology makes available larger storage, higher bandwidth, as well as diverse means of connecting to the Internet, the current design of the Internet may not be sufficient to address the future needs and opportunities. In response to this challenge, the research community recently initiated an effort aimed at design and evaluation of "clean slate" protocols for the future Internet [1], [2]. One of these clean-slate Internet projects is the "Cache-and-Forward" architecture which proposes the use of hop-by-hop transport along with in-network storage and caching to achieve efficient content delivery to both fixed and mobile end-points [3].

The overwhelming use of today's network is for an endpoint to acquire a named content. The named content can be a web page, a picture, a song, a movie/video or an email. For example, BitTorrent, which contributes more than 30% of traffic on the Internet, is centered around content dissemination. As a result, efficient content discovery and dissemination are becoming one of the key challenges for the design of future Internet protocols that go beyond the basic point-to-point transport service provided by TCP/IP.

At the same time, with the rapid growth of wireless networking technology, the number of mobile endpoints connected to the Internet will soon overtake the number of wired PC's. In comparison with wired links, wireless links are time-varying in many aspects such as bandwidth, error rate, and connectivity. Consequently, opportunistic transport which is capable of dealing with varying wireless link quality and disconnections is considered to be an important requirement for future networks. However, it is difficult and inefficient to handle mobility and opportunistic transport in today's Internet. For example, the widely used TCP/IP service does not allow for any kind of delayed delivery or in-network storage. In addition, connecting to the Internet requires a globally unique IP address that is topologically stable on routing timescale (minutes to hours).

Based on the above considerations, we identify the two important design changes as the need to support efficient content delivery services and the importance of opportunistic transport to mobile devices. The Cache-and-Forward (CNF) architecture has been proposed as a solution that leverages rapidly decreasing memory costs to provide in-network storage at routers. CNF network is an overlay on the Internet, just as the Internet is an overlay on the telephone network.

Fundamental to CNF architecture are two components: a transport layer service that operates in a hop-by-hop store-and-forward manner with large contents, and a caching scheme that integrates caching into each individual router to reduce network traffic and speed up content dissemination. In this paper, we provide a detailed description of the CNF architecture design. The overview of the architecture, including the key components of a CNF network, as well as the comparison between CNF and other network architectures such as Delay Tolerant Networks are presented in Section II. Next, a list of services provided by a CNF network, centered around content retrieval and mobile access, are given in Section III. Using content retrieval as an example, we then outline the protocol stack and the detailed protocols in Section IV. Following the protocol discussion, we next discuss two caching algorithms that are fundamental to a CNF network in Section V. Finally, we provide concluding remarks and future directions in Section VI.

## II. CNF Architecture

In this section, we first provide an overview of the CNF network architecture, and then compare CNF with other network architectures.

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

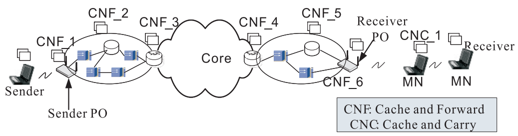


Fig. 1. Cache-and-Forward Architecture

### A. Overview of CNF Architecture

A cache-and-forward network is illustrated in Fig.1. We envision the future Internet adopts a tiered structure. In the core of the Internet are high-bandwidth static routers. Immediately outside the core are access networks, which are attached to a subset of core nodes (i.e. *edge nodes*). In addition to nodes with wired connections, the future internet also includes an increasing number of wireless endpoints. Wireless endpoints connect to the internet through *access nodes*. That is, access nodes serve as the aggregation points for the wireless nodes within access networks, while edge nodes serve as the aggregation points for the access networks within the core. Finally, we note that edge nodes are potentially the border gateways of the autonomous systems (ASes), and are connected to edge nodes (or, gateways) of other autonomous systems.

The concept of the CNF network is built upon *Cache and Forward (CNF) Routers*, which are routers with persistent storage. Compared to a router in today's Internet, a CNF router differs in the following two aspects: (1) it forwards packets hop by hop, with a hop being a CNF hop instead of an IP hop, and (2) a CNF router can choose to cache contents that are routed through the node. Caching in a CNF network is what we refer to as Integrated Caching as caching is tightly integrated into network routers, unlike traditional network architectures where caching is usually implemented on separate caching nodes. In addition to CNF routers, the network still has traditional routers that do not cache contents.

In order to facilitate efficient content handling for mobile endpoints, the CNF architecture is based on the model of a postal network designed to transport large objects through a range of delivery services. Keeping in mind that the sender and/or receiver of an object may be mobile and may not be connected to the network, we introduce the concept of a "Post Office" (PO) which serves as an indirection (rendezvous) point for senders and receivers. A mobile sender deposits the object to be delivered in its PO (suppose there is enough hold on the PO, otherwise a chunk of the object at a time) and the network routes it to the receiver's PO, which holds the object until it is picked up by the mobile receiver. Each sender and receiver may have multiple POs, where each PO is associated with a point of attachment in the wired network for a mobile sender/receiver.

The *Cache and Carry (CNC) Router* is a mobile network element that has persistent storage exactly as in a CNF Router, but is additionally mobile. Thus a CNC router can pick up a package from a CNF router, another CNC router or from a PO and carry it along. The CNC router may deliver the package to the intended receiver or to another CNC router that might have a better chance of delivering the package to the receiver.

### B. Related Work

**Disruption/Delay Tolerant Networking (DTN):** There are major differences between CNF architecture and DTN architecture [4]. DTN network is driven by *disruption* which implies potentially long periods of disconnection while CNF is driven by a combination of *wireless*, *intermittent connectivity* and *content*.

DTN network is an extension of the TCP/IP network for disconnected environment. As a result, applications interface with DTN network in a manner similar to how they interface with TCP/IP networks. In CNF network, applications interface with the network in a distinctly different way. Specifically, the interface abstraction is that of *content retrieval* as opposed to *conversation*. DTN routing [5] is again driven by *disconnection*, with the goal of delivering content to a destination which may not be connected. CNF routing has two phases, the first being a *content discovery* phase whereby the network locates/discovers the content requested by the end-user and the second phase is similar to DTN routing.

**Caching:** Caching has received much attention in the research community. There exists a rich literature body about the benefits of caching in various flavors(e.g. [6]). A significant amount of work has been done in placement of caches [7], cache replacement policies [8], etc.

Most of the work in the literature assumes an overlay of caches on the network. Caches have not been considered as an integral part of the underlying network in the same way routers have been. Thus there has been no need to extend the existing routing protocols with content related information. In this paper, we show how to integrate cache into content routing, which can in turn improve future deliveries of the same content.

**Content Routing:** Several content routing protocols have been proposed in the literature [9]. In these protocols, routing is conducted based on content location, instead of node IP. Fundamental to the design of content routing is content addressing and content discovery. These protocols provide efficient content access within a small network, but scaling them to a large network with a large number of contents is difficult, if at all possible.

Compared to content routing, CNF architecture provides a much more scalable solution, where contents are still accessed based on their addresses, but caching on the routing path can effectively speed up the access latencies. Finally, we note that our next design of CNF architecture will incorporate extensions in content routing, but in a very lightweight and distributed fashion.

*In summary, CNF network considers a clean-slate strict hop-by-hop design for both wired and wireless parts of the network, and also provides a generalized content delivery* *service that can range from unicast to multicast delivery or unicast to cached retrieval of a content.*

## III. CNF Network Services

CNF network provides a range of services to end users. Besides the default Internet services such as naming and routing, CNF network hosts several specific services which are centered around content handling and mobile access.

To make content the first class entity in the network, we introduce the notion of persistent, globally unique content identifiers, referred to as *CID*. CID must be location-independent: A content stored in multiple locations within the CNF network will be referred to by the same CID. One possible candidate for CID is the notion of a handle as in the Handle System [10].

**PO Registration Service:** The service is specially for mobile endpoints. For a fixed endpoint, its PO can be itself. Every mobile endpoint is associated with one or more postal offices (POs). This service enables the mobile node to discover POs and to associate with them.

In this service, a mobile node first contacts a well known PO discovery server in its home domain. The home domain PODS redirects the request to the mobile node's current domain PODS, which then returns a list of POs to the mobile node. The mobile node may also query one of POs in the list to obtain additional POs that are cached by that PO. From the list, the mobile node picks one or more POs to register.

This service is initiated by mobile nodes looking for corresponding POs, and the outcome is a list of POs for the requesting mobile node.

**Content Location Service:** We assume content requestors (i.e. end nodes) are equipped with the CIDs of the requested contents, but not their locations. Thus, given a CID, this service looks up its location(s).

This service is initiated by the content requestor with a CID, and the outcome of the service is a list of content sources that own a copy of that content. The requestor can then choose a content source based on its need, e.g. the closest content source.

**Content Retrieval Service:** This service helps the content requestor to retrieve the content after the content source's IP address has been obtained. Due to caching, the content may be found on an intermediate caching node before the request reaches the intended destination. Note that if the content requestor is a mobile node, the content will be deposited at its PO(s). The PO will then notify the mobile node to schedule a pick up.

This service is initiated by the content requestor with a CID and a content source IP address, and the outcome of the service is the requested content delivered to the requestor.

**Usage Example:** To illustrate the usage of the above services, let us next look at an example. Suppose a mobile node $M$ that is newly connected to the CNF network. $M$ first calls the `PO Registration service` to associate itself with one or more POs. CNF network supports two main content access methods. The first one is through content requests. To request a specific content with content ID $CID$, $M$ initiates the `Content Location Service` to obtain a list of content sources, and chooses the closest source among them. Finally, $M$ can call the `Content Retrieval Service` to retrieve the content from the select content source.

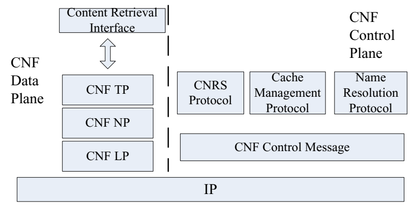


Fig. 2. CNF Protocol Stack

## IV. CNF Protocol Stack

Like an overlay network, the CNF network is built on top of IP. As a result, the Internet routing protocols such as OSPF and BGP are still used as the underlying routing protocols. The protocol stack for the CNF network is shown in Fig.2. The protocol stack has two main parts: CNF data plane and control plane. In this paper, we focus on the basic protocols involved in building a CNF network, while leaving out detailed protocols such as those between a wireless node and its PO.

### *A. CNF Data Plane*

**CNF Transport Protocol (CNF TP):** CNF TP runs at the endpoints, but is much simpler than TCP because most of the complexity, including congestion control and error control are embedded in the Link and Network layer protocols in the CNF architecture. Moreover, because of possible disconnection, the end-to-end message exchange in TP can occur over a long period of time (such as hours), a much longer time than the short (e.g., subsecond) end-to-end round-trip time in TCP. Depending on the type of service requested by the application, there may also be an end-to-end file delivery acknowledgement.

**CNF Network Protocols (CNF NP):** CNF NP is responsible for routing a content towards the destination. In our current design, for a content request/delivery, both the source and the destination are known, and we rely on conventional (IP) address-based routing, with extensions in the request phase to search for the cached copy of the content on the routing path.

As a `Content Request` packet is routed through the CNF network towards the content source, the routers on the route will first check whether the content is already cached. If the file is already cached, the router returns the requested copy; otherwise, the request is routed further along the path.

Routing the content back to the requestor involves fragmenting large contents (10's of GB) into smaller chunks (100 MB-1GB) at the original source before transporting them through the CNF network. Each fragmented chunk will contain an

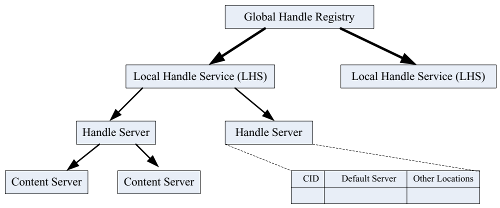


Fig. 3. CNRP Implementation Using Handle System

$MID$ field, represented by the tuple [CID, Offset, Src ID, Dst ID, Seq], which indicates the message is transporting portion of the same content. The NP at the final destination reassembles the fragments into the original large content. If it detects gaps, it can request retransmission of the missing fragment(s) from the network and any CNF router with the desired fragment(s) may provide the retransmission.

**CNF Link Protocols (CNP LP):** CNF LP operates between two CNF nodes. Non-CNF nodes between two CNF nodes simply rely on IP control layer to forward packages towards the destination. The LP has two components: the Link Session Protocol (LSP) and the Link Management Protocol (LMP). LMP is used to negotiate the type of link transport and the corresponding parameters. It is in charge of receiving buffer management and transmission scheduling. Receiving LSP acknowledges the completion of a package and delivers the received package to CNF NP. Transmitting LSP removes the package after its transmission finishes.

### B. CNF Control Plane

**Content Name Resolution Protocols (CNRP):** The main purpose of CNRP is for the Content Name Resolution Server (CNRS) to map a CID to a list of hosts that have a copy of the content. A possible implementation of CNRP would be the handle system, which consists of a global handle registry, local handle services, and handle servers that form a hierarchical structure, as shown in Fig.3.

A `Content Name Resolution` request is first routed to the requestor's local handle service node. If the requested CID cannot be found locally, the request will be routed to the destination local handle service through the global handle registry. The destination local handle service will consult its handle server, which stores a list of servers/hosts of the content. The list consists of two parts: the default server, and the other servers. The list will be routed back to the requesting node using IP protocols.

**Name Resolution Protocols (NRP):** The main purpose of NRP is to record the mappings between a mobile node and its PO(s). Every mobile node should have a unique identifier, and we can again use the handle system to assign a unique handle to each mobile.

Similar to CNRP, NRP also adopts a hierarchical structure, where a `Name Resolution` request may have to be routed through the global name resolution server to the correct destination. We note that in CNF, a mobile can either choose a PO on its own, or can be assigned to a PO by the name resolution server.

**Cache Management Protocol (CMP):** The main purpose of CMP is to keep track of what contents are cached in the network, which is an enhancement to CNRS service. If each CNF router makes individual caching decisions, CMP may announce each router's cached contents periodically to its neighbors. If a group of CNF routers makes a coordinated caching effort in a centralized fashion, CMP may rely on data structures like summary cache to record all the cached contents. The summary cache needs to be updated periodically.

## V. Implementation Algorithms and Results

Caching is an important aspect of the CNF architecture, which bears a significant impact on the content access performance.

### A. Caching Algorithms

**En Route Autonomous Caching.** In this scheme, as a content is routed towards the destination (the requestor), the en-route CNF router can choose to cache the content based on some criteria, such as content popularity (indicated by the package header) or interest level perceived by this router.

Another issue that is worth noting is the cache replacement policy. When the cache of a CNF router becomes full, it can evict a victim content to accommodate the new content. The victim can be selected based on a range of policies, including First In First Out, Least Recently Accessed First, Least Accessed Content First. Due to the nature of content access, we prefer the usage of Least Accessed Content First, i.e. the content that is accessed the least.

This caching scheme is rather straightforward, and autonomous, as every CNF router makes its own decision whether to cache a content. Upon reception of a content request, the CNF node first checks its local cache before forwarding the request to the intended content source node.

**Coordinated Caching.** The En Route Autonomous Caching scheme is easy to implement, but it may lead to unoptimized caching results. For example, popular files may be cached by every CNF router on the path, and thus the overall caching size is decreased due to the redundancy. The primary cause of this inefficiency stems from the lack of coordination in caching decisions among nodes. Our second caching scheme takes this into consideration. In this scheme, for a content that is routed through a CNF node, it will first check whether the content is already cached in the AS by consulting the summary cache. Since the first CNF router within the AS that receives a content is usually the gateway node itself, checking the summary cache is rather convenient. If the content is new, the node will assign a node in the AS to cache the content according to a hashing function. By hashing the contents, Coordinated Caching can evenly distribute the load among all the nodes and efficiently utilize the overall caching capacity.

TABLE I
NETWORK CAPACITY WITH HOP-BY-HOP AND END-TO-END TRANSPORT

| Scenarios | CNF Capacity (Mbps) | TCP Capacity (Mbps) |
|---|---|---|
| No packet error | 11.3 | 5.8 |
| Markovian Noise 1% | 9.3 | 3.7 |
| Markovian Noise 5% | 7.3 | 3.2 |
| Markovian Noise 10% | 6.7 | 2.7 |

### B. Initial Simulation Results

We are still in the design and initial evaluation phase of the CNF project. In this paper, we present a set of basic results, and the more in-depth evaluation results will be reported in the subsequent publications.

First, we demonstrate the benefit of hop-by-hop transport. The performance evaluation was based on NS2 simulation. The content server delivered the content to the clients based on a Poisson traffic model.

We adopted the Georgia Tech Internetwork Topology Model (GT-ITM) [11] to generate the wired part with base-station nodes at the edge of stub networks. A few wireless nodes were connected to the wired network through the base-station nodes. Both the clients and the servers were wireless nodes. A Markovian noise model was used for wireless channel with 5% PER (Packet Error Rate) in good state and 90% PER in bad state. We considered three scenarios with stationary probability for bad state as 1%, 5% and 10% and one scenario without Markovian noise or packet error.

The network capacity was measured as the throughput of the network with different traffic loads. The simulation results are summarized in Table I. In each scenario, hop-by-hop transport of CNF leads to a much higher network capacity than end-to-end transport of TCP.

Next, we compared two content retrieval methods: Server-Only (without caching) and Cache-and-Capture (en-route caching at each CNF router) to demonstrate the benefit of content caching. We developed a discrete event-driven simulator to model the wired part of CNF networks. We had a large network with 900 nodes, which in total hosted 900 contents. Each node requests 30 content. Fig. 4 shows the histogram of the content retrieval distances (the number of hops the request travels before a content is reached). CC can satisfy many more requests by nodes that are within a small number of hops from the requester than SO. The results show that even a simple caching scheme can significantly improve the access latency. We re-emphasize that this plot mainly severs to show the concept. Conducing more in-depth studies is part of our ongoing work.

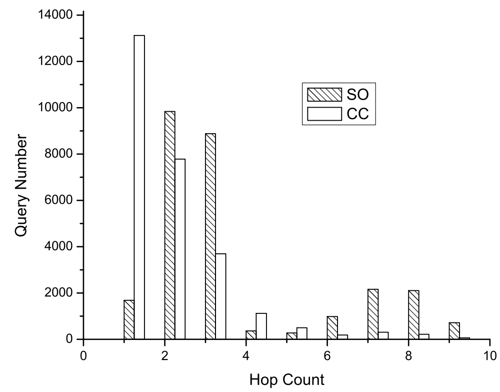


Fig. 4. The histogram of content retrieval latency under SO and CC.

## VI. CONCLUDING REMARKS AND FUTURE DIRECTIONS

Cache-and-Forward (CNF) Internet architecture is a significant departure from TCP/IP based Internet architecture in that it opportunistically transports named contents in "packages" in a hop-by-hop manner. CNF is designed to solve emerging problem of content dissemination and content retrieval in future Internet with a significant number of intermittently connected mobile endpoints. Such an architecture is made feasible by the increasing capacity of storage and processing at significantly lower price points.

Caching in CNF network is distinctively different from caching in today's Internet because caching is built into the very fabric of the CNF network by allowing each individual CNF router to cache rather than building caching as an overlay infrastructure on top of the core TCP/IP network. This is instrumental in keeping content close to the requester, no matter whether the content is globally or locally popular.

An initial design of the network architecture, protocols, and algorithms is discussed in this paper. The architecture is carefully designed to facilitate easy content handling and mobile access. To follow up, we plan to conduct detailed evaluations to validate the efficiency of the architectural design.

## REFERENCES

[1] "NSF NeTS FIND Initiative," http://www.nets-find.net/.
[2] I. Baldine, M. Vellala, and A. Wang, "A unified software architecture to enable cross-layer design in the future internet," in *Proceedings of 16th International Conference on Computer Communications and Networks*, 2007, pp. 26–32.
[3] D. Raychaudhuri, R. Yates, S. Paul, and J. Kurose, "The cache-and-forward network architecture for efficient mobile content delivery services in the future internet," in *ITU-NGN conference*, 2008.
[4] K. Fall, "A delay tolerant network architecture for challenged internets," in *Proceedings of SIGCOMM*, 2003.
[5] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2003.
[6] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, 2002.
[7] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *In Proc. of the ACM SIGCOMM*, 2002.
[8] A. Wierzbicki, N. Leibowitz, M. Ripeanu, and R. Wozniak, "Cache replacement policies revisited: the case of p2p traffic," in *IEEE International Symposium on Cluster Computing and the Grid*, 2004.
[9] D. S. Rosenblum A. Carzaniga and A. L. Wolf, "Content-based addressing and routing: A general model and its application," Tech. Rep. CU-CS-902-00, Department of Computer Science, University of Colorado, January 2000.
[10] S. Sun, L. Lannom, and B. Boesch, "RFC 3650:handle system overview," November 2003.
[11] K. Calvert, M. Doar, and E.W. Zegura, "Modeling internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, 1997.