THE CACHE-AND-FORWARD NETWORK ARCHITECTURE FOR EFFICIENT MOBILE CONTENT DELIVERY SERVICES IN THE FUTURE INTERNET

Sanjoy Paul[*], Roy Yates[*], Dipankar Raychaudhuri[*], Jim Kurose[+]

[*]WINLAB, Rutgers University, New Brunswick, NJ (sanjoy, ryates, ray@winlab.rutgers.edu)
[+]Dept. of Computer Science, University of Massachusetts at Amherst (kurose@cs.umass.edu)

ABSTRACT

This paper presents a novel "cache-and-forward" (CNF) protocol architecture for mobile content delivery services in the future Internet. The CNF architecture can be implemented as an overlay on top of the Internet Protocol (IP), or as a clean slate protocol for next-generation networks. CNF is based on the concept of store-and-forward routers with large storage, providing for opportunistic delivery to occasionally disconnected mobile users and for in-network caching of content. The proposed CNF protocol uses reliable hop-by-hop transfer of large data files between CNF routers in place of an end-to-end transport protocol like TCP. This approach makes it possible to serve mobile users with intermittent connectivity, while also mitigating self-interference problems which arise in multi-hop wireless scenarios. Hop-by-hop transport is similarly useful in wired networks where router storage can help to smooth out link congestion bottlenecks which arise in TCP/IP networks. A second key feature of the CNF protocol is the integration of address-based and content-based routing to support various content delivery modes that take advantage of in-network storage. An overview of the CNF architecture and major protocol components is given, and preliminary performance evaluation results are summarized to validate the main design principles.

Keywords— Future Internet architecture, network protocol, mobile data services, content delivery.

1. INTRODUCTION

Over the past 2-3 years, there has been a renewed interest in "clean-slate" Internet protocol design, supported by research programs such as the NSF FIND [1] and GENI [2] in the U.S. and FP7 Future Networks [3] and FIRE [4] in Europe. The objective of these programs is to explore new network architectures and protocols for efficient, high-performance and scalable support of future Internet service needs, without the constraint of backwards compatibility with IP. This paper presents the initial results of an NSF FIND project aimed at designing a novel, clean-slate network architecture for efficient delivery of media content to mobile users.

First, we address the need to support mobile content efficiently in future networks. It is well known that the ~2.5 billion cell phones in use worldwide significantly outnumber the ~500 million wired PC's on the Internet. As smart phones and PDA's with high-speed cellular and WiFi service proliferate, the number of Internet transactions from mobile devices may be expected to surpass those from wired network PC's over the next 5-10 years [5]. This

is clearly a historic shift of end-users from wired to mobile wireless devices, a trend that will inevitably drive major changes to the design and use of the future Internet. In addition, we note that Internet applications (both wired and wireless) are steadily migrating from communications to content services involving the delivery of large media files. This motivates a next-generation Internet protocol service optimized to support media content delivery to mobile users.

Existing Internet protocols (e.g., TCP/IP) are not well-suited for mobile content services because they were designed under very different assumptions, both in terms of service requirements and technology constraints. In particular, the TCP model assumes a contemporaneous source-to-destination path, and is based on the famous "end-to-end principle" [6] which argues for keeping in-network functions to a minimum and pushing service-specific complexity to the end-points at the edge of the network. While the TCP protocol [7] has served remarkably well for the first 25 years of the Internet's operation, the end-to-end principle has significant limitations when dealing with mobile users who experience intermittent and/or unreliable access over wireless channels [8-11]. Moreover, the connection oriented TCP/IP model was originally designed to support point-to-point data services rather than for multipoint content dissemination. We also note that many of the technology assumptions behind the end-to-end principle may no longer be applicable. In particular, the cost of semiconductor memory (currently ~$10/GB) has dropped by about 5-6 orders of magnitude since the Internet was first designed, link and CPU speeds have increased by 3-4 orders-of-magnitude to ~100 Mbps-1 Gbps and 1-10 GIPS respectively. These considerations argue for a back-to-basics reconsideration of the end-to-end networking model taking into account emerging requirements for large-scale mobile content services together with the increased capabilities of today's core technologies.

The "cache-and-forward" network architecture proposed here exploits these dramatic reductions in storage and processing costs to design a network that directly addresses the mobile content delivery problem. We observe here that TCP/IP does not work well for mobile devices because wireless links tend to have variable error rates, and because these devices may occasionally become disconnected due to lack of coverage. These problems are further compounded by the emergence of multi-hop wireless access networks such as ad hoc peer-to-peer [12,13], metropolitan area mesh [14] and sensor networks [15], in which the probability of at least one bad radio link or temporary disconnection tends to be higher than in single hop networks. On the other hand, these emerging peer-to-peer and multi-hop wireless networks are valuable for opportunistically delivering high-speed services and improving the overall service economics, and should thus be supported by any new protocol for mobile content. Earlier work on the wireless "Infostations" concept [16-18] demonstrated the
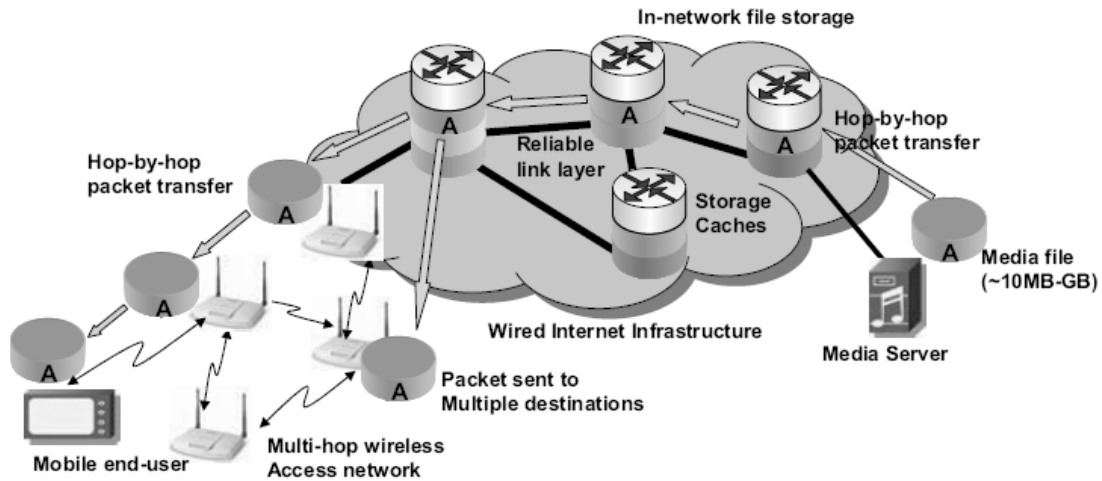
Figure 1: Conceptual View of the Cache-and-Forward Network

benefits of opportunistic transport in mobile service scenarios – at that time, this was envisioned as an overlay service typically over a single wireless hop without requiring any major changes to the networking protocol itself. Disconnected operation is also associated with delay-tolerant networks (DTN) originally intended for robust communication in tactical or vehicular environments [19-21], but we feel that this can be an important ingredient for mainstream mobile content delivery services as well. The key idea is to facilitate opportunistic transport on a hop-by-hop basis rather than end-to-end streaming of data as in TCP/IP. Such a hop-by-hop transport model implies large in-network storage of content files as they make their way through the network, made possible by remarkable recent reductions in the cost of semiconductor storage. In-network storage also enables the use of content caching [22-24] and content-aware routing [25-27] as a basic network capability rather than as an external overlay service [28-29] as currently implemented in the Internet.

Before proceeding to the details of our proposed solution, we briefly mention the concept of service virtualization [30], in future networks. The idea behind virtualization is that each major service in the future Internet may be designed and optimized separately because network computing and transmission platforms will be capable of simultaneously running multiple virtual networks with different protocols [31]. Thus, the mobile content service under discussion here could be kept distinct from conventional real-time services such as voice, video streaming and web browsing using virtualization techniques to multiplex multiple protocol stacks on the same networking platforms. Note that in scenarios where a clean-slate network with virtualization is not feasible, it is also possible to consider overlay implementations of CNF using IP tunnels between the routers, albeit at the cost of protocol efficiency.

2. SYSTEM OVERVIEW

The CNF architecture outlined in Figure 1 integrates the following functional capabilities:
- reliable hop-by-hop transport of large files;
- a configurable link-layer protocol;
- content-based routing and conventional address-based routing;
- opportunistic delivery of files to and from mobile hosts;
- location-independent naming of content;

- enhanced naming of devices to provide location information for mobiles;
- distributed caching of static and dynamic content.

Each node in the CNF network shown in Fig. 1 is assumed to have a large storage cache (~TB) that can be used to store packages (files/file segments) in transit, as well as to offer in-network caching of popular content. CNF routers may either be wired or wireless, and some wireless routers may also be mobile. The basic service provided by the network is that of file delivery either in "push" or "pull" mode, i.e., a mobile end-user may request a specific piece of content, or the content provider may push the content to one (unicast) or more (multicast) end-users. Each query and content file transported on the CNF network is carried as a CNF packet data unit or *package* in a strictly hop-by-hop fashion. The package is transported reliably between data stores at each CNF router before being prepared for the next hop towards its destination. The CNF network assumes the existence of a reliable link-layer protocol between any pair of CNF routers, and this protocol can be customized to the requirements of each wireless or wired link in the network. Packages are forwarded from node to node using opportunistic, short-horizon routing and scheduling policies at CNF nodes that take into consideration factors such as package size, link quality and buffer occupancy. Alternative routing techniques may also be used opportunistically to deal with congestion or link failure.

For delivery of content to and from mobiles, we further introduce the concept of a "Post Office (PO)" at the edge of the wired core network. The Post Office serves as a holding and forwarding point for content to mobiles which may be disconnected at times. In the simplest scenario, sending a file to a mobile host would involve these steps:

- The sender contacts a name resolution service that resolves the name of the mobile host to a set of PO nodes.
- The sender will forward the file, or portions of the file, to one or more PO's using conventional point-to-point routing.
- These PO's will "hold" the file until contacted by the mobile host to arrange delivery.
- Delivery from the PO could be by direct transmission if the mobile is in range, or by a series of wireless hops as determined by the routing protocol.

Caches in the network can create more complex scenarios. To retrieve any content, a host would send a query to the network with the location-independent content ID (CID), and the query would then be routed to the "nearest" CNF router using a *content routing* procedure, and the content would then be routed back to the host using the conventional routing capability mentioned earlier.

Before going into the details of the cache-and-forward protocols, we examine some example situations in which this type of network architecture provides improvements over existing TCP/IP:

Efficient wireless access: When TCP/IP is used over wireless links, performance is often degraded due to transport layer timeouts, and in-network solutions such as indirect TCP have been proposed in earlier work [8-10]. In addition, when TCP is used over multiple wireless hops (an increasingly common scenario), the so-called "self-interference" effect in which packets from the same flow contend for the same radio resources can further degrade end-to-end performance [32]. For multi-hop, another basic advantage of cache-and-forward relates to the fact that the probability of impairment or disconnection in at least one radio link can be quite high as the number of hops, n, increases. It can be shown that the probability of failure before the file transfer is completed is increased by a factor of n^2 over the probability of a single hop failure. This is almost an order-of-magnitude increase for $n = 3$ hops and is two orders of magnitude increase for $n = 10$ hops.

Exploiting mobility: The concept of ferrying content via "mobile Infostations" has been shown to yield a fundamental increase in wireless network capacity [33]. This approach obviously presupposes the use of caches, but, formalized routing protocols to standardize the use of such in-network caches have yet to emerge. The cache-and-forward approach naturally facilitates these file transfer mechanisms and can be used in similar scenarios that arise in sensor networks with mobile collector nodes.

Content delivery as a first class service: Cache-and-forward can also provide benefits in the wired Internet where content delivery and peer-to-peer (P2P) file sharing are widespread. Since named content file transfer is not a "service" currently offered by the Internet, several "overlay" Internet applications with very different architecture and protocols (Gnutella, KaZaa, BitTorrent) [34-36] have been developed in response to the need. Just as TCP offers a "reliable byte stream" service on today's Internet, the cache-and-forward architecture can provide an efficient content delivery service on the next generation network. In such an architecture, content addressing and caching of popular files becomes a natural component of the network layer. Multiple copies of a large file may be stored in caches to maximize the

probability of timely delivery when the location of the recipient is not certain. Moreover, these may be complete files or chunks of very large files. These chunks could be simple segments, as in BitTorrent [36], or network coding could be employed using erasure or random linear block codes [37,38].

Efficient Multicasting: Multicast support in a network with intermittently connected hosts can be quite challenging. Existing multicast routing protocols such as PIM [39], and DVMRP operate assuming connectivity between multicast group members and the last-hop routers (IGMP [40]). However, with multicast members being on multi-hop wireless links and some with intermittent connectivity, the IGMP/PIM protocol messages and multicast data messages themselves can be lost. Consequently, reliable IP multicast [41] becomes very difficult. CNF represents a solution to this problem because the multicast group members need not be connected to the network when content is being multicast by the sender. Group members can pick up content from the persistent storage within the network at a later time, when they are connected to the network.

Sensor Applications: Finally, we observe here that Internet applications involving sensors are expected to grow rapidly in the next 10 years. Sensor scenarios have unique networking requirements [42,43] including the ability to deal with disconnections due to wireless channel impairments as well as sensor hardware sleep modes. In addition, sensor applications tend to be data-centric and are thus more interested in content-aware services (e.g., querying data) rather than in connecting to a specific IP address. Efficient multicasting and location awareness are additional requirements for sensor networks. The CNF approach, with its focus on content, storage, and multicasting can help address these needs.

3. CACHE AND FORWARD (CNF) ARCHITECTURE

The cache and forward (CNF) architecture is based on a network infrastructure for hop-by-hop store-and-forward of large objects. The key components are CNF routers that have persistent storage to hold objects in transit for potentially long periods of time. Figure 2 shows the end-to-end view of a CNF network.

3.1. Key Concepts

The cache-and-forward architecture represents a set of new protocols that can be implemented either as a "clean-slate" implementation or on top of IP. The main concepts of the architecture are listed below:

Post Office (PO): The CNF architecture is based on the model of a postal network designed to transport large objects and provide a
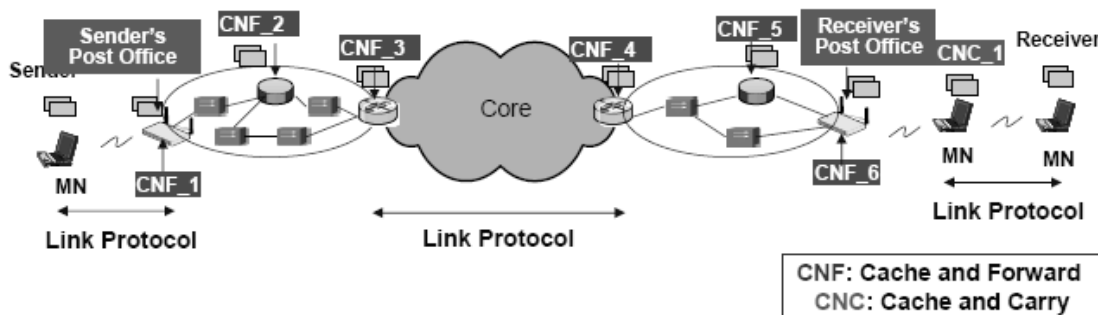

Figure 2: Cache and Forward (CNF) Network Architecture

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 "Post Office" (PO) which serves as an indirection (rendezvous) point for senders and receivers. A sender deposits the object to be delivered in its PO and the network routes it to the receiver's PO, which holds the object until it is delivered to the final destination. 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 endpoint (sender/receiver).

Cache and Forward (CNF) Router: The CNF Router is a network element with persistent storage and is responsible for routing packages within the CNF network. Packages are forwarded hop-by-hop (where a hop refers to a CNF hop and not an IP hop) from the sender's PO towards the receiver's PO using forwarding tables updated by a routing protocol running either in the background (proactive) or on demand (reactive).

Cache and Carry (CNC) Router: The 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 desired receiver.

Content Identifier (CID): To make content a first class entity in the network, we introduce the notion of persistent and globally unique content identifiers. Thus if a content is stored in multiple locations within the CNF network, it will be referred to by the *same* content identifier. The notion of a CID is in contrast to identifiers in the Internet, where content is identified by a url whose prefix consists of a string identifying the *location* of the content. CNF endpoints will request content from the network using content identifiers.

Content Discovery: Since copies of the same content can be cached in multiple CNF routers in the network, discovering the CNF router with the desired content that is "closest" to the requesting endpoint must be designed into the architecture. We discuss this in more detail in the next section.

Type of Service: In order to differentiate between packages with different service delivery requirements (high priority, medium priority, low priority), a Type of Service (ToS) byte will be used in the package header. The ToS byte can be used in the cache replacement policy and the delivery schedule of packages at the CNF routers.

Multiple delivery mechanisms: A package destined for a receiver would be first delivered to, and stored in, the receiver's PO. There are several ways in which the package can be delivered from the PO to the receiver:

- A PO can inform the receiver that there is a package waiting for it at the PO and it (the receiver) should arrange to pick it up. The receiver can pick up the package when in range of that PO. Otherwise, it may ask its new PO and/or a CNC router to pick up the package on its behalf.
- A receiver can poll the PO to find out if there is a package waiting for pick up. If it is and the receiver is within range of the PO, it can pick up the package itself. Otherwise, it may ask its new PO and/or a CNC router to pick up the package on its behalf.
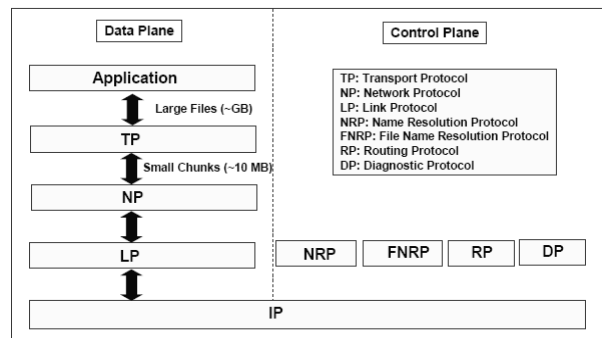- A PO can proactively *push* the package to the receiver either directly or via CNC routers.

Figure 3. Cache-and-Forward Protocol Stack

3.2 CNF Protocol Details

Figure 3 shows the overall CNF protocol stack (with IP being used as the base layer in this realization). Applications send down large files of arbitrary size to the transport layer which segments into moderately sized chunks ~10-100 MB. The network attaches a header to each chunk, and the combination is called a *package*. A package is the basic unit of transport through the CNF network layer.

Link Layer: A *link* in the CNF architecture is a *logical* link between two adjacent CNF nodes where a CNF node could be a CNF router, a CNC router, or a CNF endpoint. For example, if there are two CNF routers across an optical core network, the *link* between them would span the entire core network. On the other hand, if a CNF endpoint is connected to a CNF router (Access Point with persistent storage) using WiFi, the *link* would span just the wireless hop. The *Link Protocol* has two components: *Link Session Protocol (LSP)* and *Link Transport Protocol (LTP)*. LSP is used to negotiate the type of LTP, and the corresponding parameters. The choice of LTP will depend on the characteristics of the link.

Network Layer: The network layer is responsible for content discovery and for routing content towards the destination after it has been located in the network. The first part is addressed by (1) content-aware routing based on a content identifier (CID), while the second part is addressed by conventional (IP) address-based routing. In the latter mode, CNF routers exchange information about how to reach a given content file (=CID) rather than how to reach an "address" as in traditional routing protocols. Based on these exchanges, CNF routers set up query forwarding tables with CIDs as destinations. A CNF router, on receiving a content query for a given CID, checks if it has the requested content, and if it does, returns the content using conventional (IP) address-based routing. If it does not, it consults its Query Forwarding Table to determine the Next Hop, and forwards the request towards the CNF router that has the content.

As the query is routed through the CNF network, the content will be found either at an intermediate CNF router that has a cached copy, or in the worst case, would be found at the original source of the content. When the content is found, the next hop for forwarding the content is determined in two steps. First, on a slow timescale, a routing protocol updates the Content Forwarding Table at each CNF router, and then, at the time of forwarding a package, the CNF router will query the next-hop CNF router to see if it is prepared to accept the package. If the next-hop CNF router declines (due to bandwidth or storage limits), the forwarding CNF router will choose a different next hop on the fly.

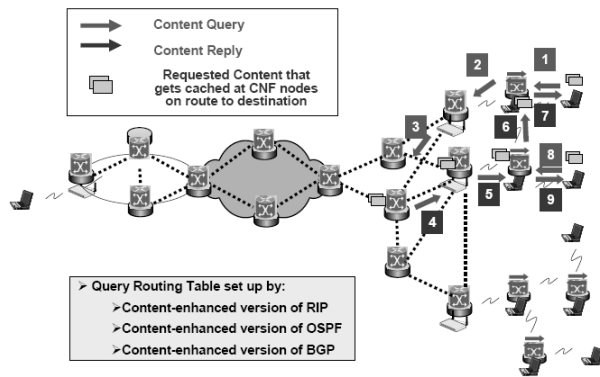Query and content routing is shown schematically in Figure 4 below.

Figure 4: Routing of Queries and Content in the CNF network

Figure 5: End-to-End Timing Diagram for an Example CNF Network Delivery

Transport Layer: The Transport Protocol (TP) runs at the endpoints, but is 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, in view of possible disconnection, the end-to-end message exchange in TP can happen over a long period of time (e.g., hours) - a much longer time than the sub-second end-to-end round-trip time in TCP. One function of CNF transport is to fragment very large files (10's of GB) into smaller chunks (~100 MB-1GB) at the original source before transporting them through the CNF network. Fragments are represented by a tuple [CID, Offset], where CID identifies the content the fragment is part of and the Offset represents the location of the fragment with respect to the beginning of the file. The TP at the final destination reassembles the fragments into the original large file. If it detects gaps, it can request retransmission of the missing fragment(s) from the network (as opposed to from the end host as in TCP) and any CNF router with the desired fragment(s) may provide the retransmission. Depending on the type of service requested by the application, there may also be an end-to-end file delivery acknowledgement.

Support Services:
Name Resolution Service (NRS): The main purpose of the Name Resolution Service (NRS) is to map the name of an endpoint to its corresponding POs. The CNF architecture is independent of the style of naming an endpoint in that an endpoint might be identified by using a handle [44], url (sanjoy@winlab.rutgers.edu), a role (fireman, police officer etc.), or by using names of local relevance (Jim's laptop, Sue's cellphone). Late binding is used to resolve the name of an endpoint to the address of its PO. Keeping in mind the address-format agnostic principle of CNF, the address of a PO could be as simple as an IP address, a DTN address which has a global and a local component, or some other type of address. POs could periodically send out advertisements or an endpoint could send out solicitations for PO whenever it moves to a new area or whenever it becomes active after a long period of inactivity.

File Name Resolution Service (FNRS): The main purpose of the File Name Resolution Service (FNRS) is to map a CID to corresponding attributes of the content. A possible implementation of FNRS would be the handle system [44]. Attributes corresponding to a CID would consist of a variety of information pertinent to the content, such as, Content Hash, Content Creator, Content Access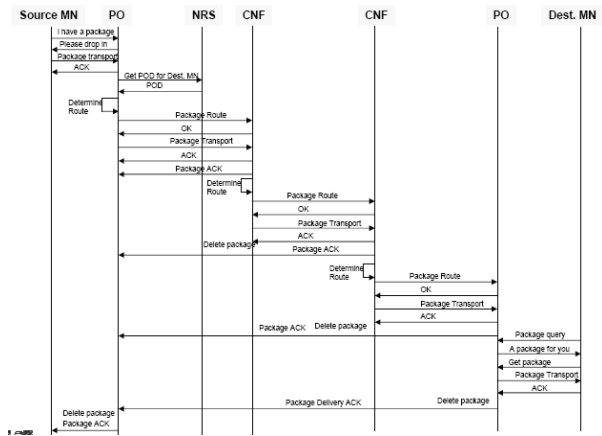 Rights, etc. It is conceivable that for popular content, an attribute may also consist of a list of CNF routers with cached copy of the content. Content Hash would be digitally signed by the Content Creator to establish authenticity of the content. Content Access Rights would implement DRM policies.

Conceptual Flow Diagram: In order to provide an intuitive feel of how the CNF architecture works, we present an end-to-end protocol timing diagram in Figure 5. In the diagram, MN = Mobile Node, PO = Post Office, NRS = Name Resolution Server, CNF = Cache and Forward Router. First, the source/sender (which may be a mobile node) drops a package at the sender's PO. The Sender's PO uses the Name Resolution Service (NRS) to retrieve the Post Office Descriptors (PODs) for the final destination (which may be a mobile node as well). Once the destination's PO is known, the next hop is determined at the PO and the package is forwarded towards the next-hop CNF router. Each CNF router independently determines the next hop and forwards the package towards the destination's PO. Note that each CNF router and the destination's PO along the route to the destination generate two acknowledgement messages: (1) ACK: a notification to the previous CNF router that it has received the package and (2) Package ACK: a notification to the Sender's PO that it has received the package. Thus the "Package ACK" tracks the progress of the package along the route to the destination while the "ACK" can be used to flush the buffer at the previous CNF router if needed. Although we show "Delete Package" at each CNF router after the reception of an ACK, this operation is "optional" in that a node can cache a package for future use. Once the package reaches the destination's PO, it is cached there until the destination MN checks with the PO and retrieves it from there.
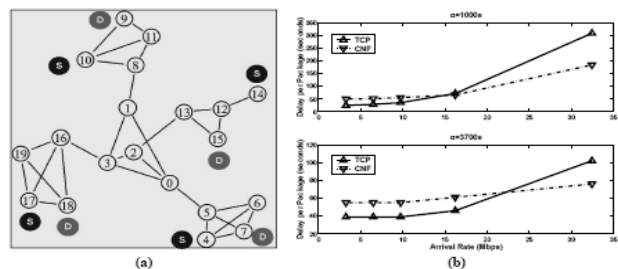

Figure 6: Network Model and delay vs. offered traffic results for α=1000s (bursty traffic) and α=3700s (less

4. PERFORMANCE SUMMARY

The longer-term objective of the CNF project is to develop a complete protocol implementation and validate it using large-scale wired and wireless testbeds such as VINI [45] and ORBIT [46]. This prototyping work is still ongoing, but we have completed an initial performance study of CNF aimed at confirming some of the basic design assumptions of the proposed architecture. Due to space limitations, we defer a detailed presentation of results for a future paper, and provide just a brief summary below.

Hop-by-hop vs. TCP performance: A fundamental question is whether CNF's pure hop-by-hop approach with intermediate storage is competitive with TCP/IP even when the network is a wired one with relatively few errors or disconnections. Clearly, CNF will do better for wireless access with occasional disconnections and/or when content caching gains are introduced. However, we wanted to first verify whether performance would be comparable to TCP for a wired network baseline. An ns2 network simulator was built for CNF and performance studies were conducted to determine measures such as end-to-end delay, network capacity, packet loss rate, etc. for various traffic parameters and topologies. An example network topology and simulation result is shown in Fig. 6 above.

The results show that TCP does offer lower delay at low network load, but CNF crosses over after that to offer better end-to-end delay in spite of the loss of pipelining gain. The cross-over point of offered traffic at which CNF is better is found to be smaller as the traffic becomes more bursty. These performance trade-offs can be attributed to the fact that CNF has large buffers which tend to smooth traffic overloads in the network.

Wireless multi-hop performance: A second set of ns2 simulations was conducted on the wireless access network to determine the benefit of CNF over TCP in multi-hop ad hoc and mesh networks. Our results show that significant throughput gains are possible with customized link layer protocols like CLAP [47] operating between CNF routers instead of TCP. Further work is ongoing to determine overall network capacity gains for various wireless access networks with realistic channel error and mobility models – large gains (perhaps ~2-5x in total capacity) are anticipated for typical mobility scenarios before taking into account content caching gains.

Content routing gains: A third simulation study has focused on quantifying the gains from in-network content caching of audio/video files, and from using a content routing protocol that enables CNF routers to exchange information about the content stored in their local cache leading them to compute the shortest path to a given content rather than to a given address as in the traditional IP network. As expected, the results confirm that in-network caching and content routing can provide significant performance gains over traditional routing in terms of reduction of content retrieval time and reduction of traffic load on the network. Details will be reported in a forthcoming paper. Overall, these initial results lead us to believe that the CNF architecture is viable for mobile content delivery and can provide significant performance capacity gains relative to TCP/IP.

5. CONCLUDING REMARKS

The cache-and-forward architecture presented here is intended as a fundamentally new approach to network design in response to growing needs for improved support for both mobility and content in the future internet. The design presented here is clearly preliminary and will be further refined over the course of this research project via discussions with other clean-slate projects, detailed performance studies and prototype implementation. Our next step is to complete simulation studies to benchmark the end-to-end performance of CNF for a variety of possible topologies, wireless channel models, and source traffic models. At the same time, we have started a prototype implementation on the orbit radio grid testbed and will aim for comprehensive end-to-end evaluations via integration with VINI and other wired network testbeds as they become available.

REFERENCES

[1]   "Networking Technology and Systems: Future Internet Design (FIND)", NSF Program Solicitation 07-507, 2007.

[2]   "Global Environment for Networking Innovations (GENI)", NSF Program Solicitation 06-601, 2006.

[3]   "FP7 Information and Communication Technologies: Pervasive and Trusted Network and Service Infrastructures", European Commission, FP7-ICT-2007-2.

[4]   M. Lemke, "Position Statement: The European FIRE Initiative", NSF/OECD Workshop on Social and Economic Factors Shaping the Future of the Internet, Washington D.C, Jan 2007.

[5]   A. Wright, "Mobile phones could soon rival the PC as world's dominant Internet platform", http://www.ipsos-na.com/news/pressrelease.cfm?id=3049

[6]   Jerome H. Saltzer, David P. Reed, and David D. Clark. "End-to-end arguments in system design", ACM Trans. Computer Systems, November 1984, pp. 277-288.

[7]   Van Jacobson, "Congestion avoidance and control", ACM SIGCOMM '88, pp. 314–329, Stanford, CA, August 1988.

[8]   Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. 15th Intl Conf. on Distributed Computing Systems, 1995.

[9]   Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Randy H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links", IEEE/ACM Trans Networking, 5(6):756–769, 1997.

[10]   Tom Goff, James Moronski, Dhananjay S. Phatak, and Vipul Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments", Proc. INFOCOM, pp. 1537–1545, 2000.

[11]   Mun Choon Chan and Ramachandran Ramjee, "TCP/IP performance over 3G wireless links with rate and delay variation", Wireless Networks, 11(1-2):81–97, 2005.

[12]   I. Chlamtac, M. Conti, and J. Liu, "Mobile ad hoc networking: Imperatives and challenges," Ad Hoc Network Journal, Jan-Mar 2003.

[13]   Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva, "A performance comparison of

multi-hop wireless ad hoc network routing protocols", Mobile Computing & Networking, pp. 85–97, 1998.

[14] Ian F. Akyildiz, Xudong Wang, and Weilin Wang, "Wireless mesh networks: a survey", Computer Networks, 47(4):445–487, March 2005.

[15] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks", IEEE Computer, August 2004.

[16] Tao Ye, Hans-Arno Jacobsen, and Randy H. Katz, "Mobile awareness in a wide area wireless network of info-stations", Mobile Computing & Networking, pp. 109–120, 1998.

[17] R. H. Frenkiel, B. R. Badrinath, J. Borras, and R. Yates, "The infostations challenge: Balancing cost and ubiquity in delivering wireless data", IEEE Personal Communications, April 2000, pp. 66-71.

[18] A. Iacono and C. Rose, "Infostations: A new perspective on wireless data networks", Kluwer Academic Publishers, 2000.

[19] K. Fall K. Fall, "A Delay Tolerant Network Architecture for Challenged Internets", Proc ACM SigComm, 2003.

[20] M. Chuah, L. Cheng, B. Davison, "Enhanced Disruption and Fault Tolerant Network Architecture for Bundle Delivery", Proc. IEEE Globecom, November 2005.

[21] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, and P. McDonald, "When TCP Breaks: Delay and Disruption Tolerant Networking", IEEE Internet Computing, 2006, pp. 72-78.

[22] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell, "A Hierarchical Internet Object Cache", Proc. USENIX Technical Conference, Jan. 1996.

[23] Li Fan and Pei Cao and Jussara Almeida and Andrei Z. Broder, "Summary cache: a scalable wide-area {Web} cache sharing protocol", IEEE/ACM Trans on Networking, June 2000.

[24] C. Greg Plaxton, Rajmohan Rajaraman, Andrea W. Richa "Accessing Nearby Copies of Replicated Objects in a Distributed Environment", ACM Symp Parallel Algorithms and Architectures, 1997.

[25] Mark A. Sheldon "Content Routing: A Scalable Architecture for Network-Based Information Discovery", MIT, PhD Thesis, 1995.

[26] J. Zhang, M. Ott, "ANSWER: Information Routing Based on Active Networks", Proc. APCC, Beijing, Oct. 1999.

[27] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker, "A scalable content-addressable network", Proc. ACM SIGCOMM, August 2001.

[28] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", Proc ACM SIGCOMM, Aug 2001.

[29] J. Jannotti, et al, "Overcast: Reliable multicasting with an overlay network", Proc. OSDI, October 2000.

[30] Scott Shenker, Larry Peterson, Jon Turner, "Overcoming the Internet Impasse Through Virtualization", ACM Hot Nets III, 2004.

[31] Nick Feamster, et al "The case for separating routing from routers", Proc. ACM SIGCOMM, August 2004.

[32] S. Gopal and D. Raychaudhuri, "Experimental evaluation of the TCP simultaneous-send problem in 802.11 wireless local area networks", ACM SIGCOMM EWIND, Aug 2005.

[33] Matthias Grossglauser and David N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks", IEEE/ACM Trans. Networking, 10(4):477–486, 2002.

[34] Gnutella. http://www.gnutella.com

[35] KaZaA. http://www.kazaa.com

[36] Bittorrent file sharing protocol. http://www.bittorrent.com

[37] C. Gkantsidis and P.R. Rodriguez, "Network coding for large scale content distribution", Proc. INFOCOM 2005, March 2005, pp. 2235-2245.

[38] S. Acedanski, S. Deb, M. Medard, and Ralf Koetter, "How good is random linear coding based distributed networked storage?" Proc. NetCod 2005, April 2005.

[39] Stephen E. Deering, at al, "The pim architecture for wide-area multicast routing", IEEE/ACM Trans. Networking, 1996, pp. 153–162.

[40] S. Deering, "Internet Group Management Protocol (IGMP)", IETF RFC 1112.

[41] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol," Proc. IEEE INFOCOM, March 1996, pp. 1414--1425.

[42] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks", ACM MONET, 7(6):481–492, 2002.

[43] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next century challenges: Scalable coordination in sensor networks", Proc. MOBICOM, 1999, Seattle, 263-270.

[44] S. Sun, S. Reilly, L. Lannom, and J. Petron, "Handle system protocol (ver2.1) specification", IETF RFC 3652.

[45] Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, and Jennifer Rexford, "In VINI Veritas: Realistic and controlled network experimentation", Proc. ACM SIGCOMM, Sept. 2006.

[46] D. Raychaudhuri, at al "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols", Proc. IEEE WCNC, 2005.

[47] S. Gopal, S. Paul, and D. Raychaudhuri. CLAP: A cross layer aware transport protocol for time-varying wireless links. WINLAB-TR 293, Rutgers University, Jan 2007.