Postcards from the Edge:

A Cache-and-Forward Architecture for the Future Internet

PROJECT SUMMARY

The ubiquitous Internet architecture based on TCP/IP protocols has proved effective through a period of dramatic growth and technological change in the network, but now face a new set of challenges. Assumptions of stability and end-to-end connection have traditionally guided the design of these protocols, and have led to efficient information transfer and effective recovery strategies during periods of stress. Now, however, this end-to-end strategy is being threatened by a revolution in wireless access technology that alters dramatically the nature of internet traffic, and challenges the basic assumptions upon which its protocols were built. Where the end-points of Internet traffic were once stable and predictable, they are increasingly embodied in wireless devices, whose numbers and information rates are increasing dramatically, and which have left the stable environment of the home and office to wander in the mobile world. They have introduced instability to Internet connectivity and made the easy assumptions of end-to-end traffic flow increasingly untenable.

Because the changes caused by wireless mobility are fundamental and pervasive, their solution requires comparably fundamental changes in the architecture and protocols of the future Internet. In this proposal, we outline a cache-and-forward architecture that exploits the decreasing cost and increasing capacity of storage devices to provide unified and efficient transport services to end hosts that may be wired or wireless; static, mobile, and/or intermittently disconnected; and either resource rich or poor. Fundamental to this architecture is a "transport layer" service that operates in a hop-by-hop store-and-forward manner with large files.

To bring the possibilities of this architecture to realization, we propose an ambitious research program to design, implement and evaluate an Internet architecture that incorporates the following elements:

Reliable hop-by-hop transport of large files: Classical store-and-forward transport of large files, with in-network storage and reliable link layer

Push-Pull architecture: For mobile nodes, the architecture enables opportunistic push-pull delivery of files, both to and from the wired network.

Enhanced Naming: Routing to and from mobile terminals will exploit location information provided by an enhanced name service.

Distributed Caching: Distributed caching of popular content will occur throughout the network, thus making peer-to-peer file sharing a first-class service and enabling efficient reliable multicast.

The experimental evaluation phase of the project will make extensive use of PlanetLab, the ORBIT radio grid testbed, and real world environments supported by future experimental systems, and will consider network performance trade-offs for various possible protocol design choices and usage/traffic scenarios.

The *intellectual merit* of this program is in the complementary mix of architectural design (based on a qualitative analysis of competing approaches), the development of macroscopic models quantifying the performance benefits of architectural components, and a prototype implementation and experimental validation of key architectural innovations. The *broader impact* of this program is that it contributes towards selection of one or more protocol architectures for the future Internet, and could lead to new services and applications of value to both scientific and commercial end users.

Roy Yates
Dipankar Raychaudhuri
Sanjoy Paul
WINLAB, Rutgers University

Jim Kurose
Department of Computer Science
University of Massachusetts

1 Introduction

Network protocols were invented when storage at the router was expensive, and when bulky desktop computers with wired network interfaces were emerging as network endpoints. The ubiquitous TCP/IP protocol used in today's Internet was designed to maximize end-to-end information flow while avoiding buffer overflows. TCP achieved this result by verifying packet delivery on a near-real-time basis, and by applying flow control when endpoint packet delivery is interrupted [1]. The capacity of the network and the speed of access devices has increased by orders of magnitude since the introduction of TCP/IP, but the simple strategies that control its traffic have proved remarkably durable. However, this end-to-end strategy is now being threatened by a new revolution in access technology – a revolution that alters dramatically the nature of internet traffic and challenges the basic assumptions upon which its protocols were built. Though this threat has been recognized for some time, most proposed solutions [2–6] have involved only modest changes in existing protocols. We believe the changes caused by wireless mobility are sufficiently fundamental and pervasive that an effective solution may require equally fundamental changes in the architecture and protocols of the future Internet. The goal of this activity is to propose and evaluate such an approach.

Many aspects of this revolution in acess technology are well-known, though their potential impact is not always appreciated. For example, the throughput of low-cost communication devices has increased by several orders of magnitude while their prices have dropped dramatically. Fifteen years ago, a wired 9.6Kb/s modem cost $500, while now a 54 Mb/s wireless interface can be found for $25. Both these trends are continuing, and the traffic that such devices deliver to the Internet is increasing geometrically. Moreover, this type of device was typically used from stable locations in the home or business, and was virtually "wired" in its performance. This in turn supported the paradigm of end-to-end connectivity, in which interruptions in throughput are generally associated with congestion and associated buffer overflows, and are corrected with strategies appropriate to such situations. Today, these devices are often mobile, appearing at unexpected locations and even moving during access. In such situations, it is increasingly probable that interruptions in connectivity will be caused by with a failure of the radio path, or by contention for wireless access. They will not be corrected, and may in fact be worsened, by strategies that assume end-to-end connectivity.

Looking a bit further into the future, we see the emergence of ad hoc and sensor networks. In the former case, access to a wired portal can be achieved through a series of peer-to-peer hops, and it becomes increasing likely that the connection will be broken after a brief period. Once again, traditional Internet strategies will interpret these breaks as buffer overflows, and react in ways that are inefficient at best. In the latter case, sensors may spend the majority of their lives in a sleep mode, making end-to-end connectivity virtually impossible.

One approach to these problems is through specialized, environment-specific protocols (e.g., specific protocols for sensor networks), but there are dramatic advantages to common protocols that can be run across a variety of networks - the Internet itself is an example of the power of such approaches. We believe that a better answer to these emerging problems may be found in the innovative application of even more dramatic advances in a second area of technology-that of data storage. Where wireless access rates have increased 50-fold in the last decade, solid-state storage capacities have increased 100-fold, while dropping in cost to $50/GB, and magnetic storage devices have increased 100-fold, while dropping in cost to $0.50/GB.

In this proposal, we outline a cache-and-forward architecture that exploits the decreasing cost and increasing capacity of storage devices to provide unified and efficient transport services to end hosts that may be wired or wireless; static, mobile, and/or intermittently disconnected; and either resource rich or poor. Fundamental to this architecture is a "transport layer" service that operates in a hop-by-hop store-and-forward manner with large-sized data units (which we will refer to simply as "files"). For mobile nodes, the architecture enables opportunistic push-pull delivery of files, both to and from the wired network. Routing to and from mobile terminals will exploit location information provided by an enhanced name service. Distributed caching of popular content will occur throughout the network, thus making peer-to-peer file

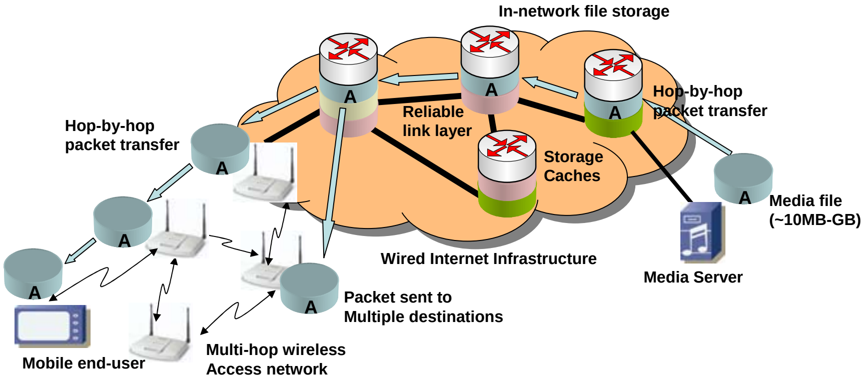

In-network file storage
Hop-by-hop packet transfer
A
A
A
Reliable link layer
Hop-by-hop packet transfer
A
A
Storage Caches
A
Media file (~10MB-GB)
A
Wired Internet Infrastructure
Media Server
A
Packet sent to Multiple destinations
A
Mobile end-user
Multi-hop wireless Access network

Figure 1: A Conceptual Overview of the Cache-and-Forward Network: A File *A* is cached in the network as it is forwarded to a mobile host.

sharing a first-class service. *Our research consists of a complementary mix of architectural design (based on a qualitative analysis of competing approaches), the development of macroscopic models quantifying the performance benefits of architectural components, and a prototype implementation and validation within the ORBIT grid of key architectural innovations.*

System Description

We envision a network in which storage will be exploited at every node. Caching will occur at routers in the wired backbone, at wireless access points by the edge of the wired network, and at mobile nodes, some of which can be assumed to be operating in a disconnected setting. Access points will act both as wireless gateways for ad hoc networks and as infostations [7–9] for drive-by file delivery for highly mobile hosts. In the simplest scenario, like that depicted in Figure 1, 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 “post office” nodes.
- The sender will forward the file to one or more post office nodes.
- These post office nodes will “hold” the file until contacted by the mobile host in order to arrange delivery.
- Delivery from the post office could be by a direct transmission if the mobile host is in range, or by a series of wireless hops.

Caches in the network can create more complex scenarios. To receive a popular file, a host would query a file name service that would return the addresses of nodes that cache the file. The file could then be requested from a cache, or chunks of the file could be requested from a collection of caches. On the other hand, to send a popular file, a host might request that a cache near the destination send the file.

A number of networked applications have adopted a cache-and-forward architecture in the past. From the early UseNet news to todays commercial Content Distribution Networks such as Akamai, several large-scale networked applications have operated under the control of a single organizational entity, and focused primarily on a one-to-many push of data from an origin server. Our focus here is on providing network services to support numerous (many-to-many) individual, user-driven (both push and pull) cache-and-forward services. Disruption Tolerant Networks represent a new breed of cache-and-forward networks. Most proposed DTN systems are infrastructureless, i.e., do not consider constructs such as the access nodes (AN) and post offices (PO) that serve as indirection (rendezvous) points in our architecture, although research is currently underway [10] to provide drop-and-store services in the context of a mobile DTN bus network. We note that ANs and POs provide a level of indirection between sender, the architectural value of which has been elegantly argued in [11]. Rutgers own Infostation provided capabilities similar to POs, but did not operate in an end-end cache-and-forward manner.

Some Benefits of Cache-and-Forward

Before going into the details of the cache-and-forward architecture and protocols, we examine some example situations in which cache-and-forward affords improvements over the existing TCP/IP architecture.

Efficient Multihop Wireless Transmission: Consider an ad hoc wireless network with stationary nodes such that the PHY layer radio connectivity is adequate. Suppose that these nodes are supporting a TCP file transfer over a multihop radio path. In this case, data packets in the forward direction (from sender to receiver) contend for the channel with RTS/CTS at the PHY layer as well with TCP ACK messages in the reverse direction. These contending data packets cause self interference to the multihop route and can disrupt timely control message exchanges. This condition can be perceived as a lost link, triggering inappropriate route repair or route discovery mechanisms, ultimately resulting in transport layer timeouts and dramatic reductions in throughput. This deficiency is in addition to the problems caused by physical layer outages induced by fading on a single link, for which solutions [2, 4] have been developed. Alternatively, hop-by-hop transmission of the file by cache-and-forward nodes avoids self-interference, since the transmission on any hop doesn't start until the previous hop is completed. Although this forfeits the potential benefits of pipelining, preliminary experiments [12] indicate that the reduction in self-interference more than compensates.

The advantages of cache-and-forward are even more pronounced when the network nodes are mobile. In this case, the TCP connection over an n hop path would break as soon as any link in the path failed. The rate of mobility-induced disconnection on the n hop path would be roughly n times the disconnection rate of a single hop. Moreoever, compared to a single hop transmission, the holding time of the n hop connection is extended, by at least a factor of n because the channel bandwidth is shared by all n hops. Thus the probability of the n hop connection failing before the file transfer is completed is increased by a factor of n^2 over the probability of a single hop failure. This is an almost an order of magnitude increase for $n = 3$ hops and is two orders of magnitude for $n = 10$ hops. Consequently, a single hop file transfer is far more reliable. The cache-and-forward approach can combine reliable single hop forwarding with the invocation of route discovery after each hop as needed. We believe that this combination will offer significantly faster and more reliable transport for large files in multihop mobile networks.

Facilitating cache-and-carry to increase capacity in mobile scenarios: Finally, we observe that physical carriage of data using two-hop routing via "mobile infostations" yields a fundamental increase in wireless network capacity wireless networks [13] by using physical motion to greatly reduce the distances over which radio transmission must take place. This approach obviously presupposes the use of caches, but, formalized protocols to standardize the use of such caches have yet to emerge. The cache-and-forward approach naturally facilitates these file transfer mechanisms as well as similar situations that arise in sensor networks with mobile collector nodes. Specifically, cache-and-forward allows for a seamless unified routing solution for wired and wireless networks. In this case, a route could be a sequence of hosts capable of sustaining a real-time connection or a sequence of hosts physically carrying the data, or even some combination of these approaches. The potential diversity of routes is inreased because an end-to-end real-time connection is no longer mandated.

Making file sharing a first class service: Cache-and-forward can even provide benefits in the wired Internet where peer-to-peer (P2P) traffic has become widespread. Since P2P data transfer is not a "service" offered by the Internet, several independent applications with very different architecture and protocols

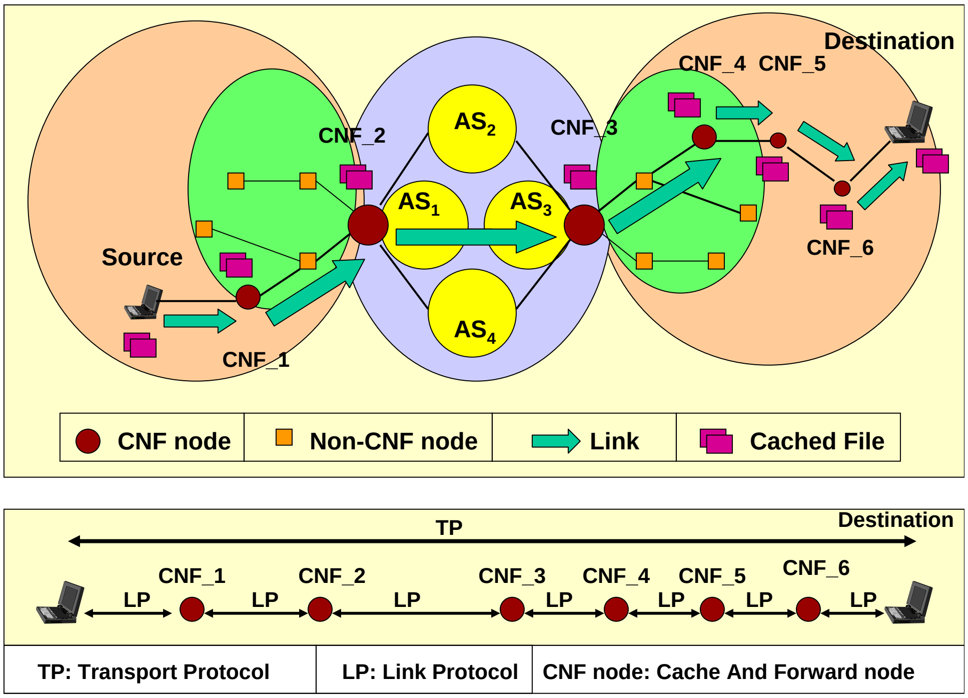


Figure 2: System Overview

(Gnutella, KaZaa, BitTorrent , Skype etc.) [14–16] have been developed to accomplish what is essentially the same result. Such peer-to-peer file transfers have beome so common that it is worth having a common service from the network to meet these needs. This is analogous to TCP, without which each application that requires reliable transport of packets would have had to develop its own reliable transport protocol. Just as TCP offered a "reliable byte stream" service to the hosts connected to the Internet, a cache-and-forward architecture can provide an "efficient file transport service" between hosts on the next generation network.

In such an architecture, caching of popular files becomes a natural component of the network layer. Multiple copies of any 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 files may be isolated files or may be chunks of very large files. These chunks could be simple segments, as in BitTorrent [16, 17], or could network coding could be employed with segments constructed by erasure codes or random linear block codes [18–20]. In either case, cache-and-forward provides network support.

Multicasting: Finally, we note that multicast support in a network with intermittently connected hosts is challenging. Existing multicast routing protocols such as PIM [21], DVMRP etc. operate assuming connectivity between multicast group members and the last-hop routers (IGMP messages are critical for IP multicast). However, with multicast members being on multi-hop wireless links and some with intermittent connectivity, the IGMP protocol messages become unreliable, and as a result, IP multicast routing becomes very inefficient and does not work in many cases. Once again, cache-and-forward represents a solution to this problem.

2 Key Concepts

We have seen that cache-and-forward can mitigate certain shortcomings of the current Internet. In this section, we provide an overview of the architecture and protocols. It is noted here that cache-and-forward , as proposed in this work, represents an important new service category that would facilitate emerging applications and end-user categories that need transport mechanisms for big files. However, we recognize that there will still be a need for conventional best effort datagram and streaming services for today's applications such as email, VOIP, internet radio and streaming video. In our view, it is possible and even desirable to focus on file transport as a separate service because of advances in network virtualization technology which permit each router or network element to build customized protocols suited for a particular application category. In this view of future networks, multiple services will be optimized separately without being integrated into a single transport framework. Thus we believe it is reasonable an appropriate to limit the scope of this proposal to an individual major class of service.

Our cache-and-forward protocol stack assumes IP as the control plane protocol and proposes a family of new protocols (depicted in Figure 3) for the data transfer plane. Use of IP will be assumed for addressing and routing of control messages. Although IP is not essential to the design, the data transfer protocols require the basic functions of IP. Assuming IP underneath avoids reinventing basic network functions and thus simplifies the subsequent discussion. The architecture makes the following assumptions:

- **The network is hierarchical.** A very high-bandwidth static core has edge nodes (EN) that connect via a medium-high bandwidth static access network to access nodes (AN) that act as wireless gateways. At the mobile fringe are mobile nodes that connect to the AN via low-medium rate multi-hop wireless links as well as mobile nodes that exploit disconnected high-speed file exchanges. The AN is the aggregation point for the mobile nodes and ad hoc mobile networks, and the EN is the aggregation point for the Access Nodes. Edge nodes are potentially the border gateways of the autonomous systems (AS) and are connected to edge nodes of other autonomous systems as shown in Figure 2.

- **Transport is provided by cache-and-forward (CNF) nodes.** These cache-and-forward nodes may appear throughout the network hierarchy, as caching routers or edge nodes in the core, as caching access nodes, or even as caching mobile hosts in the mobile fringe.

- **Every mobile node has a Home Autonomous System (HAS)** that is used only for authenticating the mobile should the mobile node move to a different part of the network. Note that this is different from Mobile IP Home Agent that not only authenticates the mobile but also tunnels data packets to the Foreign Agent of the mobile.

- **Every mobile node is associated with a set of Post Offices (PO)**. Typically, access nodes on the wired network will serve as post offices. However, our design allows any CNF node, even a mobile CNF node, to be a PO. Each mobile has a list of post office descriptors (POD) that that would characterize both the mobile's time-varying network connection as well as the properties, such as mobility, of the associated POs. Note that a PO is different from Mobile IP Foreign Agent because the PO is not required to forward data to the mobile; rather the mobile is expected to arrange to pick up any data destined for it from the PO. In addition, unlike Mobile IP, there may be multiple POs corresponding to a mobile.

- **Each mobile node is responsible for updating its post office descriptors.** A mobile node can submit a modified POD to its Authoritative Name Resolution Server (NRS) only after being authenticated by its corresponding Home Autonomous System. This is also quite different from Mobile IP where DNS entries are not touched.

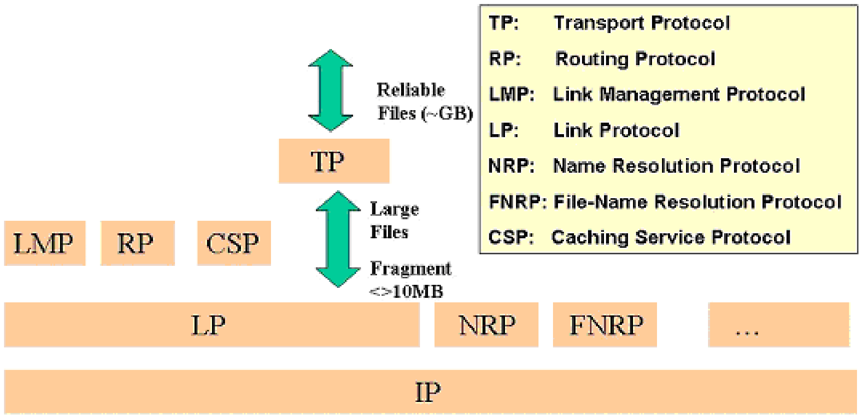


Figure 3: New data and control plane protocols.

The network will have both cache-and-forward (CNF) nodes as well as traditional stream-based routers, called non-CNF nodes, that do not cache files. A *link* between CNF nodes may consist of path that includes multiple non-CNF nodes. For example, in a wired network, a "link" might be a multi-hop path and the CNF nodes could be the endpoints of a TCP session. More generally, the link is any communication pathway between CNF nodes such that a link is up if it can provide reliable transmission of a large file. For a pair of CNF nodes in the wired network, a link will be up almost all the time. However, for a mobile CNF node, links with other CNF nodes will go up and down as the node moves about.

Just as every publicly addressable host in the Internet today has a DNS entry in its Authoritative Name Server and there is a hierarchical DNS service, we propose to use a unique identifier (UFID.FQDN) of every file in the network and have an entry indicating the potential cached locations of the file in the Authoritative File Name Resolution System (FNRS) server. Note that every file is assumed to have a *home* location identified by the Fully Qualified Domain Name (FQDN) and the Authoritative FNRS server corresponding to a file is identified by fnrs.FQDN. FNRS is assumed to have a DNS-like hierarchical architecture where intermediate FNRS servers cache the IP addresses of the Authoritative FNRS servers corresponding to the oft-visited FQDNs.

The Routing Protocol (RP) leverages caches within the network to optimize routing and forwarding of files. We introduce a Caching Service Protocol that is similar in spirit to Inter Cache Protocol (ICP) [RFC 2186] but with subtle differences. The primary function of CSP is to enable a CNF node to retrieve a requested file and return it to the requestor. The requestor may be a host (after employing FNRS to identify which CNF nodes have a specific UFID.FQDN) that sends a request for UFID.FQDN to a specific CNF node. Alternatively, a sending host can direct a specific CNF node to send the requested UFID.FQDN to a specific destination on its behalf. A third possibility is that a receiving host can request a specific CNF node to retrieve the file on its behalf and deliver the file. The CSP will exchange "summary cache" information among the CNF nodes within a domain very much like the exchange of routing vectors and/or link status information in the routing protocols among routers in the Internet today. This enables a CNF node to use the CSP to enquire about the availability of a file in another CNF node.

Keeping in mind that the final destinations of a file may not always be connected to the network or their exact locations may be unknown due to mobility, we introduce the notion of Post Offices where files destined for the end hosts are dumped and the end hosts (mobile/intermittently connected hosts) are notified of the availability of files for them to pick up. However, if the end hosts are connected and the Post Office knows how to reach them and/or the file is urgent, the Post Office will deliver the file directly to the end

Unique File ID (UFID.FQDN)	Original Source Address (OSA)
Intermediate Source Address (ISA)	Final Destination Address (FDA)
Intermediate Destination Address (IDA)	Post Office Descriptor (POD)
Type of Service (TOS)	Payload

Table 1: Package header fields.

host. In order to distinguish between files' content types, we use a Type of Service (TOS) byte. The TOS byte may indicate, for example, that a file is (1) *popular* and needs caching with high TTL, (2) *transient* and needs caching with small TTL, (3) *short message* that needs quick forwarding, or (4) *real-time* and requires forwarding within bounded time.

Routing based on the post office descriptor (POD) and the TOS byte is introduced to account for the fact that the exact location of a mobile host may be unknown and that the POD may contain a list of possible post offices where the end host might be at a given point of time. In that case, depending on the POD and the TOS byte, a file may be forwarded to one or more post offices in the POD.

Reliable Multicast routing and forwarding is achieved by leveraging caches within the network. The end hosts subscribe to a multicast group by using an IGMP-like protocol with their corresponding Post Offices. The Post Offices then leverage the Caching Service Protocol and/or the FNRS to pull the requested files from the network and deliver them to the end hosts.

3 Architecture and Protocol Details

To describe the network protocols for caching and forwarding files, we follow common practice and tag a transmitted file, or chunk of a file, with a header. We call the combination of a header and payload file as a *package*, so as to distinguish it from a packet, a term we employ for the transmission units of lower (typically PHY) layer protocols. In Table 1, we enumerate the header fields needed for the family of cache-and-forward protocols that we now describe.

3.1 Link Protocol (LP)

The *Link Protocol* operates between two CNF nodes. Non-CNF nodes between two CNF nodes simply look at the Intermediate Destination Address (IDA) of a package and forward towards the IDA using the IP control layer. The LP has two components: the *Link Session Protocol (LSP)* and the *Link Transport Protocol (LTP)*. LSP is used to negotiate the type of link transport and the corresponding parameters. LTP most likely will be influenced by the characteristic of the link and the TOS in the package. For example, for a multi-hop wired link and TOS set to *popular* or *transient,* a TCP-like LTP may be used, whereas for a wireless link with TOS set to *real-time* may require a cooperative relay LTP that enlists, at the PHY layer, a relay node that provides a second signal path to improve the wireless diversity mode, as in [22, 23].

3.2 Link Management Protocol (LMP)

Link Management Protocol (LMP) provides a "ping" like functionality for any CNF node (including end hosts) to reach out to any other CNF node for diagnostic purposes. The side effect of LMP is the availability of a "traceroute" like functionality for the sender to monitor progress of file delivery towards a given destination. LMP also helps provide a generic congestion control mechanism for the hop-by-hop transport of files without overwhelming either the network or the CNF nodes within the network.

The Link Management Protocol works closely with Link Protocol (LP) and Transport Protocol (TP) which is covered in a later section. A CNF node can have multiple incoming links from and multiple outgoing links to other CNF nodes. LMP provides link status monitoring, buffer management, transmission scheduling, and FNRS updating.

LMP monitors error conditions and reception rates from multiple senders, and decides, for every file, what chunks need to be fetched from which senders. On incoming links, LMP checks the available buffer space at the CNF node and decides whether LTP should send a throttle message to a sender. On outgoing links, the LMP is responsible for link scheduling. Based on the TOS bytes of the files waiting in the queue to be sent and any other information it might have, the LMP decides in what order queued files should be sent.

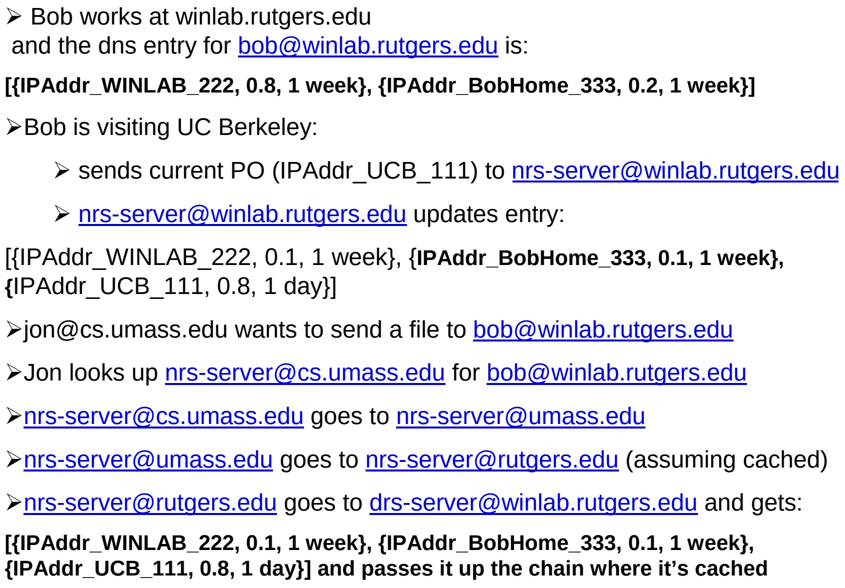
The LMP periodically sends out ACK messages to the Original Source Address (OSA) of a file ID (UFID) contained in a package when a CNF node (identified by Intermediate Destination Address: IDA) completely receives a File. The ACK message contains the Intermediate Source Address (ISA) from which the CNF node received the file. The LMP at an IDA stops sending an ACK to an OSA of file UFID when it receives a STOP_ACK [UFID, IDA] message from the OSA of file UFID. Note that this means a file has moved on to the next hop towards the destination, and ACK messages need not be sent from an earlier hop any more. Moreover, if the TOS byte of the file UFID is *transient*, the LMP removes the file UFID from its cache since as the next hop towards the destination now has the file and the file is not *popular*, there is no need to cache the file.

When a CNF node does cache the file UFID.FQDN, the LMP signals fnrs.FQDN that the IP address of the CNF node should be added to the entry for UFID.FQDN. Ideally this update should be done only for popular files and not for every file in order to reduce load on FNRS. The LMP updates fnrs.FQDN corresponding to UFID.FQDN by removing the IP address of a CNF node when it flushes the file UFID from its cache.

3.3 Routing Protocol (RP)

As a result of the diversity of routes available in a cache-and-forward network, the Routing Protocol RP will come in several flavors. In all cases, finding a route to a previously unknown CNF node starts with query to the name resolution service. The NRS returns a post office descriptor (POD) that describes post offices used by that CNF node. A CNF node in the (wired) edge or core network will serve as its own post office and its POD will hold just its own IP address. Thus routing to a wired CNF will be similar to existing methods that distinguish between Intra-AS and Inter-AS routing. In Intra-AS routing, the CNF nodes belonging to an AS exchange reachability information (similar to Path Vectors in BGP) and for each link in the path between two CNF nodes, information about link speed, link latency, and preferred modes for data reception are exchanged. Inter-AS routing will be similar to BGP where Edge Nodes will exchange "reachability/path vectors" just as in BGP.

More complex possibilities exist when one or more nodes is mobile. An NRS query for a mobile node would return a POD that includes several post offices. Based on the POD, a file would be sent to one or more post offices listed in the POD. The appropriate method is likely to be application specific, may or may not be deterministic, and may depend on properties of the individual post offices. A simple delivery from a post office to a mobile host incorporates the fundamental subproblem of discovering and maintaining routes in an ad hoc wireless subnet and thus any new methods would start with time-tested solutions [24–27]. The design of specific PODS and routing mechanisms is a proposed research topic in Section 4.



- ➢ Bob works at winlab.rutgers.edu
  and the dns entry for bob@winlab.rutgers.edu is:

[{IPAddr_WINLAB_222, 0.8, 1 week}, {IPAddr_BobHome_333, 0.2, 1 week}]

- ➢Bob is visiting UC Berkeley:
  - ➢ sends current PO (IPAddr_UCB_111) to nrs-server@winlab.rutgers.edu
  - ➢ nrs-server@winlab.rutgers.edu updates entry:

[{IPAddr_WINLAB_222, 0.1, 1 week}, **{IPAddr_BobHome_333, 0.1, 1 week},**
{IPAddr_UCB_111, 0.8, 1 day}]

- ➢jon@cs.umass.edu wants to send a file to bob@winlab.rutgers.edu
- ➢Jon looks up nrs-server@cs.umass.edu for bob@winlab.rutgers.edu
- ➢nrs-server@cs.umass.edu goes to nrs-server@umass.edu
- ➢nrs-server@umass.edu goes to nrs-server@rutgers.edu (assuming cached)
- ➢nrs-server@rutgers.edu goes to drs-server@winlab.rutgers.edu and gets:

[{IPAddr_WINLAB_222, 0.1, 1 week}, {IPAddr_BobHome_333, 0.1, 1 week},
{IPAddr_UCB_111, 0.8, 1 day}] and passes it up the chain where it's cached

Figure 4: The Name Resolution Service (NRS): The tuple {IPAddr,p,TTL}is an example of a post office descriptor (POD). IPAddr is a post office address, p is the probability that the mobile is found at that post office, and TTL indicates how long the PO is valid.

3.4 Name Resolution Protocol (NRP)

Name Resolution Protocol is essentially similar to DNS except that it is closely tied with Routing Protocol (RP). The role of NRP is to maintain the post office descriptor (POD) for each mobile terminal. When a mobile terminal informs its corresponding Authoritative NRS server of a new Post Office, the NRS server will update the POD using some algorithm. The exact algorithm may vary from one NRS domain to another.

Note that the POD corresponding to a mobile host remains confined to its Authoritative NRS server, and does not propagate through the network. Just as in case of DNS, intermediate NRS servers may cache the IP address of a specific Authoritative NRS server (NRS_1@FQDN) should there be a query to retrieve the Post Offices corresponding to a mobile host in NRS_1@FQDN domain. In case there are a few very popular names that need to be frequently resolved, the idea of using collaborative distributed DNS [28] can be used for fault tolerance, redundancy and load balancing.

3.5 File Name Resolution Protocol (FNRP)

Since our architecture is about caching and forwarding *files*, enabling hosts to retrieve files from the network and not necessarily from the origin server, we need to uniquely identify the files. Thus if there are a hundred copies of a file in the network, any CNF node and/or host should be able to recognize them as identical. To that effect, we propose to identify a file using the unique file indentifier UFID.FQDN where UFID is obtained by a one-way hashing (MD5) algorithm on the original URL of the file and FQDN is the fully qualified domain name corresponding to the Origin Server of a file.

Just as every publicly addressable host has an NRS entry (a POD) in its corresponding authoritative NRS server, every file has an entry in its corresponding authoritative FNRS server. Analogous to the list of post offices in a POD, the FNRS entry contains a list of CNF nodes that have a copy of the file. As mentioned earlier, when a file UFID.FQDN is cached or flushed from a cache of a CNF node, the LMP sends an update to the Authoritative FNRS server of a file UFID.FQDN For scalability reasons, it probably makes sense to update the entries only for popular files. Popularity of a file may be determined statically by the content provider or dynamically by the network based on the observed demand for the file.

Just as in the case of NRS, we assume an exactly similar hierarchical structure for FNRS. FNRP a is a query-response protocol like DNS and each host and CNF node will be running a filename resolver (client) component of FNRP while each FNRS server will be running the server component of FNRP.

Scalability of FNRS can be improved significantly by the presence of Caching Service Protocol (CSP) which could aid CNF nodes with the potential availability of a desired file in a neighboring CNF node. Details of CSP are given below. For improving the scalability of FNRS even further, a collaborative distributed DNS-like system [28] can be used in which popular entries would be replicated in multiple FNRS servers for load balancing purposes.

3.6 Caching Service Protocol (CSP)

The functionality of the Caching Service Protocol was summarized on page 6. As noted, the primary function of CSP is to enable a CNF node to retrieve a requested file and return it to the requestor. The efficiency of the CSP depends on peer CNF nodes to exchange “summary cache” information where the summary cache is a digest of the files cached at a CNF node. This helps FNRS scale because FNRS query messages will originate only from hosts and not from CNF nodes (who will use CSP for checking availability of files). Moreover, this permits a CNF node to quickly find out if a peer CNF node is likely to have a requested file that is not cached locally. Note that there is a possibility that a CNF node thinks that its peer CNF node has particular file based on the summary cache information provided by the peer; however the peer CNF node may not actually have the file cached. The likelihood of such a “false positive” can be reduced significantly by increasing the number of bits used to represent the footprint of a UFID.FQDN.

3.7 Transport Protocol (TP)

In the cache-and-forward architecture, the role of the transport protocol TP is greatly reduced. As depicted in Figure 2, the transport protocol runs at the original source and final destination. The primary functions of TP are fragementation and reassembly of large files. TP at the original source segments large files into chunks if necessary and invokes LP to start sending the files. TP at the final destination reassembles the chunks into files at the Final Destination. TP also monitors the progress of file delivery through the network towards the Final Destination and confirms final delivery of each files at their final destinations

4 Proposed Research

The proposed architecture will support big file transfer (bulk transfer/P2P/e-mail etc.) as well as short message transfer (instant messaging.) We remind the reader that we assume advances in network virtualization will enable us to consider these services separately from other important services such as VoIP or streaming real-time video delivery. Even within the restricted scope of this project, our cache-and-forward arhcitecture dictates a thorough reexamination of all aspects of the network and transport layers. Some key issues include

- *How should routing be integrated with caching?*
- *How do we implement congestion control?*
- *What are the storage needs of a CNF nodes?*
- *What are the necessary post office descriptors*
- *How do we specify routing tables?*
- *How should cache-based multicast be implemented?*

We note that it would be quite possible to write an entire proposal that only addresses just one of these questions. Given space constraints, we highlight a subset of issues related to storage, routing and multicast for cache-and-forward file delivery.

4.1 Storage Issues

With a cache-and-forward architecture, the amount of storage needed (both within the CNF nodes, as well as in PO nodes) is an important consideration. This is particularly the case when multiple copies are stored (either due to delivery of a single-user object to multiple POs, or an object cached at multiple locations for multiple users, e.g., as in a P2P file sharing system). As part of our proposed research, we will use macroscopic simulation fluid models (both via simulation and analysis [29–31]) of our cache-and-forward architecture; such fluid models will allow us to abstract out the fine-grain microscopic details of individual requests, and the in/out flow of data as smooth fluids. We will use these model to investigate questions that include the following: (i) For POs with disconnected users, how much storage is required to ensure with high probability that files can be stored long enough to be successfully retrieved when a user reconnects? What is the tradeoff between storage capacity and successful delivery probability, both for traditional files, and chunk-oriented RLC-encoded files (ii) When multiple copies of a file are stored at different POs, what is the tradeoff between number of stored copies, and the time needed for an intermittently connected user to successfully obtain that file? We note that our goal here is less to advance the state-of-the-art in fluid modeling of cache-based systems than to develop a coarse understanding of the tradeoff between available storage capacity and system-level performance metrics such as delivery probability and delivery delay.

4.2 Routing Issues

In a cache-and-forward architecture, the types of files and their latency requirements will vary greatly. For example, instant messages are short with a data field of perhaps a hundred bytes, but delivery must occur in a few seconds. A webpage might several thousand bytes and delivery in seconds is again required. On the other hand, an itunes transaction may result in a several megabyte file, but delivery in hours may be acceptable. As an extreme case, a TV show or movie may result in a transfer of several hundred megabytes.

The routing algorithm must take into account the traffic types. Consider the IM example where a small delay sensitive file (an instant message) is to be sent to a recipient. First, an NRS query returns the recipient's POD. Because the file is small and timely delivery is required, the file may be transmitted to all POs listed in the POD. On the other hand, if the file is large, the cost of delivery may be high and an optimized post office forwarding strategy could be employed. For example, the file may be delivered to the first post office. If the recipient does not take delivery in a reasonable time, the file may be forwarded to subsequent post offices. A greater tolerance for latency allows optimization over a larger set of possibilities. A post office may simply let a delay-insensitive file sit until the recipient contacts the post office to retrieve the file. We note that for a mobile host seeking to retrieve a file by multihop cache-and-forward, time-tested methods of efficient route discovery [24–27] must be re-examined.

4.3 Post Office Descriptors

We also note that it is a research question to determine the appropriate forms for the POD. Figure 4 introduced one example of a POD: tuples of the form {IPAddr,p,TTL} indicating that for time TTL that a node can be found with probability p near the post office at address IPAddr. Here we propose an alternate form of a POD: the list $\{(a_1, t_1), (a_2, t_2), \ldots, (a_n, t_n)\}$ where a_i is an IP address and t_i is an integer time delay. Associated with a particular POD structure would be one or more routing algorithms. For example, for the (a, t) POD, here is a plausible routing algorithm:

- After a file spends time t_i at post office i, the file is forwarded to post office $i + 1$. If $t_i = 0$, the file is sent simultaneously to post offices i and $i + 1$. The time unit associated with the integer delay t_i would be a time unit D. One possibility is that the sender specifies D depending on the urgency of the message. The time unit D would be embedded in the file header by the sender. Alternatively, a receiver requesting a file may request a value of D to reflect his urgency. In addition, in making a request, the recipient could embed a temporary POD for a single transaction.

Although it would be possible for the network to use adaptive learning methods to fine tune a user's PO, personal privacy issues suggest this may be undesirable. The alternative is to rely on the user to update its POD, thus allowing the user to strike a balance between exposing detailed location information against timely data delivery.

4.4 Multicast

The problems of "multicast routing" [32–35] and "reliable multicast" [36–38] have been traditionally dealt with separately in the context of the Internet. Due to the uniqueness of our architecture with a hierarchy of CNF nodes within the network, we are in a unique situation to deal with both problems at the same time.

Multicast routing has been dealt with at length in the context of the Internet, and several protocols have been standardized [PIM-SM, PIM-DM, DVMRP, SSM, etc.]. In addition, the basic functionality of protocols, such as PIM-SM has been further enhanced to span multiple Autonomous Systems using protocols such as MSDP. However, these approaches do not explicitly address following considerations:

- The source of the multicast is mobile.
- The members (destinations) of the multicast group are mobile.
- Some members of a multicast group may not be connected when the source starts to send packets. This is particularly the case for mobile nodes that are temporarily out of power or out of range.
- Reliable transport.

In the following, we describe the main ideas of our solutions.

Architecturally we decouple the wired part of a multicast tree from the wireless mobile part. This is made possible by the concept of Post Offices which are "representatives" of the mobile hosts in the wired network. Thus, regardless of where the original source(s) and/or final destination(s) are, we have a wired part of the multicast tree with end nodes (post offices) that, on one hand, would hold packages for delivery on behalf of the senders and on the other hand, would hold "delivered" packages for the receivers of the multicast group to pick up at the opportune moment.

Conceptually, both the senders and the receivers inform their corresponding Post Offices about their intention to participate in multicast delivery. Assuming that the original source is mobile, it will connect with different post offices during the transport of a given file to a given multicast group. Some chunks of a given file would be deposited in a PO while some other chunks of the same file might be deposited in a different PO for delivery to the same set of final destinations. Similarly, a given multicast group member may pick up chunks of the same file from different POs in the network.

A multicast tree in the wired network would span the POs corresponding to the senders and receivers. CNF nodes that belong to the wired part of the multicast tree may need to combine chunks from different CNF nodes to assemble complete files for caching. Note that the design becomes simpler if we assume caching of "complete" files at CNF nodes instead of caching only chunks of files. However, the granularity of link-level transport would still be chunks.

Setting up of the multicast tree within an autonomous system would be very similar to the way it is done for PIM except that the Rendezvous Point (RP) for a multicast group would be the post office corresponding to the original source. However, note that there might be more than one RP for the same multicast tree as the original source *attaches* to the wired network at various post offices. For mobile multicast receivers, the change of PO can be thought of as a combination of *leave* from the old PO and a *join* to the new PO. Thus mobility is handled in the same way as dynamic membership of a multicast group is handled. For inter-AS multicast routing, a protocol like MSDP would be used to announce the AS with the source to all other ASs and to determine which ASs have multicast group members.

Disconnected group members need not receive the package (file) in real-time. The package would be cached at the PO corresponding to a disconnected member and the PO would periodically send reminders to the member about the availability of a package. When the member gets connected (battery recharged or wired network in range), it can pick up the package from its PO. In addition, to avoid frequent announcements, a member whose status changed from *disconnected* to *connected,* could also enquire with its PO if it has any package waiting to be delivered to it.

Reliability is handled during CNF node-by-node transport. Specifically, the fact that CNF nodes in the multicast tree would also cache the file makes it easy for distant CNF nodes or final destinations to pick up the missing chunks of a file from a nearby CNF node instead of going back all the way to the original source which in the first place may not even be connected to the network.

This solution raises a large collection of issues and potential research questions. For a mobile source connecting at multiple post offices, should there be the concept of an “anchor” PO which will assemble all the chunks corresponding to a file or should each PO independently forward the chunks towards the destination POs along the multicast tree? More generally, should a CNF node within the network during multicast, cache chunks or complete files? If it’s the former, how should Caching Service Protocol (CSP) be extended to support delivery of chunks as opposed to files?

Should a CNF node in the multicast tree dynamically select the downstream CNF nodes for delivery of chunks based on the change of POs of the multicast group members? Or should the file be transported to the destination POs fixed at the beginning of transport and then forwarded to the new POs?

Should the setting up of a multicast tree change depending on the Type of Service? For example, if the goal of the multicast tree is to deliver big files, we might select high capacity links, whereas if the goal is to stream real-time traffic, low latency links might be selected between CNF nodes in the multicast tree. Is there a unified multicast routing protocol to cover the entire gamut of services?

How should multicast be done in the mobile fringe? Should it be an extension of the epidemic routing for multicast [39]? For energy constrained nodes, efficient use of transmit power will be essential [40]. With highly mobile nodes, we would like to leverage the mobile CNF nodes for delivery of packages to multicast group members.

5 Experimental Validation

Experimental validation is essential for a network architecture research project. In this project, the cache-and-forward design is driven by the file transfer to and from mobile nodes. In the past, simulators such as NS2, OPNET and GloMoSim were the only available platforms for protocol evaluation for mobile network terminals. However, with PlanetLab overlays providing a network backbone, mobility emulation in the ORBIT grid [41] will enable real-world testing of proposed protocols. protocols. Experimentation can be divided into the following phases:

Simulator Implementation/Validation of the link protocols LP and LMP, naming protocols NRP and FNRP, the network protocols RP and CSP, and the transport protocol TP. Validation of the simulator implementation would examine a sequence of traffic scenarios including a single flow to ensure correctness, multiple simultaneous reliable homogeneous (and later heterogeneous) flows to examine fairness issues. Topologies to examine include completely wired networks, wired networks with a single wireless hop at the edge, and wireless multihop network with static nodes, and then with mobile nodes.

Testbed Implementation of the cache-and-forward protocols will be validated with the above types of traffic and topologies. From past experience with ORBIT and PlanetLab, we note that generating simultaneous traffic flows is a non-trivial task. The availability of a central experiment-controller is essential. Repeatable small-scale wired/wireless experiments are possible with the 400-node ORBIT radio grid testbed. Limited node mobility emulation also can be achieved [41]. Experimentation with a wide area Internet scenario and wireless node mobility may be achieved with the ORBIT outdoor testbed grid in combination with PlanetLab. However, we have observed that it's quite challenging to conduct these combination experiments due to implementation differences between PlanetLab and ORBIT. Several new capabilities are required, including

- intelligent gateways between the two testbeds that can have higher access privileges in PlanetLab virtual slices to achieve wired/wireless routing
- translation of multicast-based ORBIT control packets into unicast tunnels to reach PlanetLab end nodes,
- extension to PlanetLab of the central experiment-controller functionality available in ORBIT to provide seamless control to the experimenter.

Testbed Experiments to be performed include

- *Qualitative and quantitative evaluation of reliable link layer protocols between wireless nodes and wired/wireless nodes.*
- *Evaluation of multicast routing and reliable transport protocols in the presence of node mobility* We note that caching nodes can be collocated at key ORBIT nodes, and wireless multicasting experiments can be performed.
- *Combined PlanetLab and ORBIT experiments to examine proposed cache-and-forward protocols over large-delay wired/wireless networks* Here we seek to understand the pros and cons of big-file hop-by-hop transport vs. end-to-end packet-by-packet transfer with TCP/IP. This evaluation would indicate the typical buffer requirement at intermediate hops, and average queuing delays. The effects of mobility induced link failures should be examined here.
- *The evaluation of network layer protocols, including routing and naming services.* This should also include node mobility.
- *End-to-end file transport experiments to determine overall network throughput vs. delay under various system assumptions.* We would experiment with multiple simultaneous flows, with single wireless hop on the edge. What queuing delays are experienced? What transport protocol(s) should be used in the mobile fringe?

6 Impact Statement

The *intellectual merit* of this program is in the complementary mix of architectural design (based on a qualitative analysis of competing approaches), the development of macroscopic models quantifying the performance benefits of architectural components, and a prototype implementation and experimental validation of key architectural innovations. The *broader impact* of this program is that it contributes towards selection of one or more protocol architectures for the future Internet, and could lead to new services and applications of value to both scientific and commercial end users.

7 Education and Knowledge Transfer

This project is expected to provide valuable training to Rutgers ECE Dept. and WINLAB students on Internet architecture and protocols. In particular, the ECE curriculum currently has three graduate courses (Communication Networks I, Communication Networks II and Wireless Systems) with coverage of network protocols and their implementation. The main project assignment in Comm Nets II and Wireless Systems involves groups of students in designing, building and performing interoperability tests on specified routing or wireless network protocols. The research proposed here involves several key components (probabilistic routing, multicast, and caching) which are suitable topics for future assignments, and we believe that students will benefit from the exposure to larger-scale evaluations being conducted on ORBIT and PlanetLab.

WINLAB also has an active industrial outreach program involving technical seminar, collaborative research and visits from technical staff from companies. The proposed project may be expected to create new opportunities for joint work with industry sponsors working on wireless access and Internet technologies or services.

8 Results from Prior NSF Support

Roy D. Yates has been co-PI or PI on previous NSF grants [42–47]. Work on current NSF support [48–50] has been on transmitter optimization for energy-efficient operation in time-varying channels [51–54], low-power broadcast and multicast protocols for sensor networks [55–59], information theoretic channels [60, 61], spectrum servers for link scheduling [62–64] and mapping wireless network topologies to indoor wireless testbeds [65, 66].

Dipankar Raychaudhuri has 25 years research experience in networking and communications, and has led several large industrial research projects including prototyping and field trials of an early 5 GHz WLAN (WATMnet) system. After moving to Rutgers in 2001, Prof. Raychaudhuri began NSF-funded work [47] on etiquette protocols and coordination algorithms for spectrum sharing in unlicensed bands. His current research is also directed towards protocol design and prototyping of next-generation wireless systems and sensor networks [12, 67–72], and he is currently serving as PI on a NSF NRT collaborative project entitled "ORBIT: Open Access Research Testbed for Next Generation Wireless Networks" [49]. He is also co-PI of a collaborative Lucent/WINLAB/GA Tech NSF project [73] aimed at development of a network-centric cognitive radio hardware platform.

Sanjoy Paul is currently a co-PI on the NSF funded ORBIT testbed project [49], leading to publications [71, 72, 74–76]. Dr. Paul has extensive network research experience at Bell Labs and venture companies, and is credited with several protocol innovations including RMTP (reliable multicast), content caching and hybrid 3G/WLAN mobile networks.

Jim Kurose is a Co-PI on the NSF Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere (Award number EEC-0313747 001), which is substantially independent of the proposed effort. He is a participant on two NSF-CNS research grants broadly in the area of network measurement, ITR 0325868 *Hyperion - next generation measurement infrastructure and application use* and ANI 0240487 *Measurement-in-the-Middle* . His publications from these two efforts include [77–85]. He has an ending award, ITR 0085848 *Scalable Quality-of-Service Control for the Next Generation Internet: Fundamental Challenges and Effective Solutions*. Recent publications include [86–90].

Professor Kurose received funding in Fall 2005 for the effort, *Network X-ities – Foundations and Applications*, CNS-0519998 which is investigating the theoretical foundations, quantitative frameworks, and well-defined metrics for "robust" network operations, including properties of non-fragility, manageability, diagnosability, optimizability, scalability, and evolvability. A publication includes [91].

References

[1] Van Jacobson. Congestion avoidance and control. In *ACM SIGCOMM '88*, pages 314–329, Stanford, CA, August 1988.

[2] Ajay Bakre and B. R. Badrinath. I-TCP: Indirect TCP for mobile hosts. *15th International Conference on Distributed Computing Systems*, 1995.

[3] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP performance over wireless networks.

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

[5] Tom Goff, James Moronski, Dhananjay S. Phatak, and Vipul Gupta. Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In *INFOCOM (3)*, pages 1537–1545, 2000.

[6] 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.

[7] Tao Ye, Hans-Arno Jacobsen, and Randy H. Katz. Mobile awareness in a wide area wireless network of info-stations. In *Mobile Computing and Networking*, pages 109–120, 1998.

[8] 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*, 7(2):66–71, April 2000. .

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

[10] John Burgess, Brian Gallagher, and Brian Levine David Jensen. Maxprop: Routing for vehicle-based disruption-tolerant. In *Proc. 2006 IEEE INFOCOM*, 2006.

[11] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana. Internet indirection infrastructure. In *Proceedings of ACM SIGCOMM*, 2006.

[12] S. Gopal and D. Raychaudhuri. Experimental evaluation of the TCP simultaneous-send problem in 802.11 wireless local area networks. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis EWIND-05*, Aug 2005.

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

[14] Gnutella. `http://www.gnutella.com`.

[15] KaZaA. `http://www.kazaa.com`.

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

[17] B. Cohen. Incentives build robustness in bittorrent. In *P2P Economics Workshop*, 2003. Berkeley, CA.

[18] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67, 1998.

[19] C. Gkantsidis and P.R. Rodriguez. Network coding for large scale content distribution. In *Proc. INFOCOM 2005*, volume 4, pages 2235–2245, March 2005.

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

[21] Stephen E. Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. The pim architecture for wide-area multicast routing. *IEEE/ACM Trans. Netw.*, 4(2):153–162, 1996.

[22] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: efficient protocols and outage behavior. *IEEE Trans. Information Theory*, 50(12):3062 – 3080, Dec. 2004.

[23] T.E. Hunter, S. Sanayei, and A. Nosratinia. Outage analysis of coded cooperation. *IEEE Trans. Information Theory*, 52(2), February.

[24] C. Perkins. Ad-hoc on-demand distance vector routing, 1997.

[25] Elizabeth M. Belding-Royer and Charles E. Perkins. Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. *Ad Hoc Networks*, 1(1):125–150, 2003.

[26] 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. In *Mobile Computing and Networking*, pages 85–97, 1998.

[27] D. Johnson, D. Maltz, and J. Broch. *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*.

[28] Venugopalan Ramasubramanian and Emin Gün Sirer. The design and implementation of a next generation name service for the internet. In *SIGCOMM*, pages 331–342, 2004.

[29] F. Clevenot, P. Nain, and K.W. Ross. Stochastic fluid models for cache clusters. *Performance Evaluation*, 59:1–18, 2005.

[30] Syam Gadde, Jeff Chase, and Amin Vahdat. Coarse-grained network simulation for wide-area distributed systems. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation*, 2002.

[31] Daniel Figueiredo, Benyuan Liu, Yang Guo, Jim Kurose, and Don Towsley. On the efficiency of fluid simulation of networks. *Computer Networks (to appear)*, 2006.

[32] K. Almeroth. The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment. *IEEE Network*, 14:10–20, January/February 2000.

[33] Stephen E. Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. An architecture for wide-area multicast routing. In *SIGCOMM*, pages 126–135, London, UK, August 1994. ACM.

[34] Stephen Deering, Deborah L. Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153–162, 1996.

[35] Hugh W. Holbrook and David R. Cheriton. IP multicast channels: EXPRESS support for large-scale single-source applications. In *SIGCOMM*, pages 65–78, Cambridge, MA, September 1999. ACM.

[36] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, December 1997.

[37] Sanjoy Paul, Krishan K. Sabnani, John C.-H. Lin, and Supratik Bhattacharyya. Reliable multicast transport protocol (RMTP). *IEEE Journal of Selected Areas in Communications*, 15(3):407–421, 1997.

[38] Christos Papadopoulos, Guru M. Parulkar, and George Varghese. An error control scheme for large-scale multicast applications. In *Symposium on Principles of Distributed Computing*, page 310, 1998.

[39] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks, 2002.

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

[41] K. Ramachandran, S. Kaul, S. Mathur, M. Gruteser, and I. Seskar. Towards mobility emulation through spatial switching on a wireless grid. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis EWIND-05*, Aug 2005.

[42] C. Rose and R. Yates. Searching for Good Call Admission Policies in Communications Networks. NSF grant NCRI 92-06148.

[43] R. Yates and C. Rose. Power Control for Packet Radio Networks. NSF grant NCRI 95-06505, $469,897.

[44] D. J. Goodman, N.B. Mandayam, A. T. Ogielski, C. Rose, and R. D. Yates. Parallel Computing for Wireless Networking Research. NSF NCR 97-29863, $170,000.

[45] C. Rose and R. Yates. Interference Avoidance in Wireless Systems. NSF grant CCR 99-73012, $430,000.

[46] R. D. Yates, N. B. Mandayam, and C. Rose. Free Bits: The Real Challenge of the Wireless Internet. NSF grant ITR 00-85986, $860,000.

[47] R. Yates, C. Rose, N. Mandayam, P. Spasojevic, and D. Raychaudhuri. ITR: Collaborative Research: Achieving Innovative and Reliable Services in Unlicensed Spectrum. NSF grant CCR-0205362, in collaboration with Michigan State (CCR-0205362) and Cornell (CCR-0205431).

[48] R. D. Yates, L. J. Greenstein, and P. Spasojevic. Collaborative Research:MAMA (Multiple Antennas Multiple Appliances) Wideband Wireless Networks: A Pervasive Technology for the Home and Workplace. National Science Foundation Grant SPN-0338805, 1/1/2004–12/31/2006, $676,595.

[49] D. Raychaudhuri, R. Yates, W. Trappe, Y. Zhang, M. Parashar, H. Kobayashi, and H. Schulzrinne. ORBIT: Open-Access Research Testbed for Next-Generation Wireless Networks. National Science Foundation CNS-0335244, 9/1/2003 – 8/31/2007, $5,453,115.

[50] N. Mandayam, C. Rose, P. Spasojevic, and R. Yates. NeTs Pro-Win: Cognitive Radios for Open Access to Spectrum. National Science Foundation Grant CNS-0434854, 9/15/2004 – 8/31/2007, $670,000.

[51] J. Luo, L. Lin, R. Yates, and P. Spasojević. Service outage based power and rate allocation. *IEEE Trans. Info Theory*, 49(1):323–330, Jan 2003. .

[52] L. Lin, R. Yates, and P. Spasojevic. Adaptive transmission with discrete code rates and power levels. *IEEE Trans. Commun.*, 51(12):2115 – 2125, Dec. 2003. .

[53] Jianghong Luo, Roy Yates, and Predrag Spasojevic. Service outage based power and rate allocation for parallel fading channels. *IEEE Trans. Info Theory*, 51(7):2594–2611, 2005. .

[54] L. Lin, R. Yates, and P. Spasojevic. Adaptive transmission with finite code rates. *IEEE Trans. Info Theory*, 2006. Accepted. .

[55] I. Maric and R. Yates. Cooperative multihop broadcast for wireless networks. *IEEE J. Sel. Areas Commun.*, 22(6):1080 – 1088, Aug. 2004. Special issue on fundamental performance limits of wireless sensor networks. .

[56] I. Maric and R. Yates. Cooperative multicast for maximum network lifetime. *IEEE J. Sel. Areas Commun.*, 23(1):127 – 135, Jan. 2005. Special issue on wireless ad hoc networks. .

[57] I. Maric and R. Yates. Power and bandwidth allocation for cooperative strategies in gaussian relay networks. In *Proceedings of Asilomar*, November 2004. Monterey, CA. (invited) .

[58] I. Maric and R. Yates. Forwarding strategies for parallel-relay networks. In *International Symposium On Information Theory (ISIT'04)*, July 2004. Chicago, IL.

[59] I. Maric and R. Yates. Cooperative multicast for network lifetime maximization. In *Proceedings of the 42nd Allerton Conference on Communications, Control, and Computing*, Oct. 2004. (invited) .

[60] I. Maric, R. Yates, and G. Kramer. The strong interference channel with common information. In *Proceedings of Asilomar*, November 2005. Monterey, CA. (invited) .

[61] I. Maric, R. Yates, and G. Kramer. The discrete memoryless compound multiple access channel with conferencing encoders. In *Proc. IEEE International Symposium On Information Theory (ISIT'05)*, Sept. 2005.

[62] C. Raman, R. D. Yates, and N. Mandayam. Scheduling variable rate links via a spectrum server. In *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks DySPAN'05*, Nov 2005. Baltimore MD. .

[63] R. D. Yates, C. Raman, and N. Mandayam. Fair and efficient scheduling variable rate links via a spectrum server. In *IEEE International Communications Conference ICC'2006*, June 2006. To appear.

[64] R. D. Yates, C. Raman, and N. Mandayam. A spectrum server for fair and efficient scheduling of variable rate links. *IEEE J. Sel. Areas Commun.*, June 2006. Submitted for the special issue on adaptive, spectrum agile and cognitive wireless networks.

[65] J. Lei, R. Yates, L. Greenstein, and H. Liu. Wireless link SNR mapping onto an indoor testbed. In *Proceedings of The First International Conference of Testbeds and Research Infrastructures for the Development of Networks and Communities*, February 2005. Trento Italy. .

[66] J. Lei, R. Yates, L. Greenstein, and H. Liu. Mapping link SNRs of wireless mesh networks onto an indoor testbed. In *Proceedings of The Second International Conference of Testbeds and Research Infrastructures for the Development of Networks and Communities*, March 2006. Barcelona Spain.

[67] S. Zhao, K. Tepe, I. Seskar, and D. Raychaudhuri. Routing protocols for self-organizing hierarchical ad hoc wireless networks. In *Proceedings of the IEEE Sarnoff Symposium*, Mar 2003. Trenton, NJ.

[68] L. Raju, S. Ganu, B. Anepu, I. Seskar, and D. Raychaudhuri. BOOST: A BOOtSTrapping mechanism for self-organizing hierarchical wireless adhoc networks. In *IEEE Sarnoff Symposium*, April 2004. Princeton, NJ.

[69] S. Ganu, L. Raju, B. Anepu, I. Seskar, and D. Raychaudhuri. Architecture and prototyping of an 802.11-based self-organizing hierarchical ad-hoc wireless network (SOHAN). In *Proc. PIMRC*, Sep 2004. Barcelona, Spain.

[70] S. Zhao, I. Seskar, and D. Raychaudhuri. Performance and scalability of self-organizing hierarchical ad hoc wireless networks. In *Proc. IEEE WCNC 2004*, March 2004.

[71] S. Gopal, S. Paul, and D. Raychaudhuri. Investigation of the TCP simultaneous-send problem in 802.11 wireless local area networks. In *International Conference on Communication ICC*, May 2005.

[72] S. Ganu, I. Seskar, M. Ott, D. Raychaudhuri, and S. Paul. Architecture and framework for supporting open-access multi-user wireless experimentation. In *First International Conference on Communication Systems Software and Middleware*, 2006.

[73] B. Ackland, D. Raychaudhuri, M. Bushnell, and C. Rose. High performance cognitive radio platform with integrated physical and network layer capabilities. NSF CNS-0435370.

[74] S. Rangarajan, J. Lin, and S. Paul. A campus-wide hybrid 802.11/3G wireless network and its implementation. *Bell Labs Technical Journal*, 10(2), June 2005. Special Issue on Wireless Networks.

[75] R. Bhatia, L. Li, H. Luo, R. Ramjee, and S. Paul. ICAM: Integrated cellular and ad-hoc multicast. *IEEE Trans. on Mobile Computing (TMC)*, 2006. To appear.

[76] Sarit Mukherjee, Sampath Rangarajan, John Lin, and Sanjoy Paul. User identity based session redirection in CDMA2000 networks. In *MobiQuitous*, pages 105–110, 2004.

[77] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and classification of out-of-sequence packets in a tier-1 IP backbone. In *Proc. 2003 IEEE INFOCOM*, 2003.

[78] W. Chen, Y. Huang, B. F. Ribeiro, K. Suh, H. Zhang, E. de Souza e Silva, J. Kurose, and D. Towsley. Exploiting the ipid field to infer network path and end-system characteristics. In *Passive and Active Measurement*, 2005.

[79] K. Suh, Y. Guo, J. Kurose, and D. Towsley. Locating network monitors: Complexity, heuristics, and coverage. In *Proc. 2005 IEEE INFOCOM*, March 2005.

[80] S.Vasudevan, K.Papagiannaki, C.Diot, and J. Kurose amd D. Towsley. Facilitating access point selection in ieee 802.11 wireless networks. In *ACM Internet Measurement Conference*, 2005.

[81] W. Wei, B. Wang, C. Zhang, J. Kurose, and D. Towsley. Classification of access network types: Ethernet, wireless LAN, ADSL, cable modem or dialup? In *Proc. 2005 IEEE INFOCOM*, March 2005.

[82] K. Suh, D. Figueiredo, J. Kurose, and D. Towsley. Characterizing and detecting relayed traffic: A case study using skype. In *to appear in 2006 IEEE Infocom conference*, 2006.

[83] S. Jaiswal, G. Iannaccone, J. Kurose, and D. Towlsey. Formal analysis of passive measurement inference techniques. In *to appear in 2006 IEEE Infocom conference*, 2006.

[84] W. Wei, S. Jaiswal, J. Kurose, and D. Towsley. Identifying 802.11 traffic from passive measurements using iterative bayesian inference. In *to appear in 2006 IEEE Infocom conference*, 2006.

[85] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley. Inference and performance evaluation of cellular data networks through end-to-end measurements. In *Proc. Passive and Active Measurement Workshop*, 2006.

[86] S. Vasudevan, J. Kurose, and D. Towsley. Design and analysis of a leader election algorithm for mobile ad hoc networks. In *Proc. ICNP 2004*, 2004.

[87] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley. On optimal routing with multiple traffic matrices. In *IEEE Infocom Conference*, 2005.

[88] C. Zhang, Z. Ge, J. Kurose, Y. Liu, and D. Towsley. Optimal routing with multiple traffic matrices: Tradeoff between average case and worst case performance. In *IEEE Int Conference on Network Protocols*, 2005.

[89] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. In *IFIP Networking Conference*, 2006.

[90] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *Network Coding Workshop*, 2006.

[91] H. Zhang, J. Kurose, and D. Towsley. Can an overlay compensate for a careless underlay? In *to appear in 2006 IEEE Infocom conference*, 2006.