COCONET: CONTENT AND CONTEXT AWARE NETWORKING

A Dissertation Proposal Presented to the
Department of Computer Science and Electrical Engineering
by
Sethuram Balaji Kodeswaran

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
THE UNIVERSITY OF MARYLAND, BALTIMORE COUNTY
BALTIMORE, MARYLAND
29TH SEPTEMBER, 2005

Abstract

The current Internet was originally designed to provide "best-effort" data transport over a wired infrastructure with end hosts utilizing a layered network stack to provide reliability, quality of service, security etc. for user applications. However, the proliferation of inelastic applications, coupled with wide spread migration towards hybrid networks utilizing wired and wireless links and the plethora of end host variants ranging from cell phones to enterprise servers necessitates the migration of more and more services away from the edges and into the network. The aim of this thesis is to provide a generic and flexible framework and associated algorithms that will enable the incremental deployment of intelligent services into the network with the aim of optimizing the end-user experience for networked applications. We focus our research on two key facets; cross layer optimization algorithms to enable efficient transport of data across a hybrid network coupled with the inclusion of semantic information in the data packets that can be intelligently processed within the network. The aim is to show that if routers in a network have visibility into the type of data that they are currently handling (either at a packet level or a flow level), the routers can then perform optimizations and content adaptations relevant to that specific type of data based on local policies. We intend to use RDF/RDFS as the medium to convey this semantic information thereby allowing interim routers to reason over their existing knowledge base on how to specifically handle a given data packet or stream in a flexible and generic manner. Mechanisms to incrementally deploy our framework into a large scale network along with the implementation of new value added services that can now be offered will be demonstrated using the PlanetLab testbed.

Contents

Chapter 1

Introduction

One of the greatest revolutions in recent times has been the explosive growth of the Internet. The new capabilities and services that are now offered starting from the World Wide Web to emails and instant messaging have become integral parts of our society. At the same time, networking technologies have evolved from traditional Ethernet, ATM, Frame Relay, SONET/SDH, Satellite broadband to more recent advances in wireless networking such as 802.11a/b/g/e/n, Infra Red, BlueTooth, GPRS, Ultra Wide Band(UWB) etc. More and more consumer devices ranging from cell phones to kitchen appliances are becoming network enabled leading to the development and deployment of diverse network applications, each with its own unique networking requirements. All this leads to the motivation of this thesis;

"Is the current model of the Internet where the network just provides data transport with the intelligent services residing at the edges still the most efficient way considering the diversity of networking technologies, application needs, device capabilities and overall user expectations?"

It can be argued that one of the most important reasons why the current Internet is so widely accepted and deployed is primarily due to its layered architecture. A

network stack that is implemented as different layers where each layer offers a well defined functionality is a clean abstraction. Each layer is well encapsulated and can be developed independent of the others. Well defined Service Access Points (SAPs) are specified that clearly defines the data and primitives exchanged between adjacent layers in the stack. While this is the currently accepted model of a network stack, recent years have seen increased interest in the area of cross layer optimizations. The idea here is to try to provide a layer in the networking stack with additional contextual information so that a more informed decision can be made on the data handled by that layer. Layers are no longer shielded from each other and a layer can communicate with any other layer (not just the adjacent ones) to perform its functions. Proponents of this model argue that with such interactions, more optimized networks in terms of bandwidth usage, more efficient routing, better QoS guarantees, better power utilization etc. can be achieved. In this research, we are proposing to follow a similar model. Allow layers to interact across boundaries such that any extra contextual information that can be utilized by a specific layer should be made available to that layer.

Existing work in this area is confined mainly to cross layer optimizations at the edges of the network (either at the source or near the sink of the data). We are proposing to expand on this notion and allow such optimizations to occur within the network. The aim is to make the network smarter and not have intelligence confined only to the periphery. Intermediary routers are envisioned to be more than simple flow/packet forwarding entities but rather intelligent processors that can perform special handling on data streams and packets based on local policies. At the same time, the core function of a router ie. transport of data must not be comprised. Our proposed

approach is to take an incremental deployment strategy where intelligence is introduced into the network initially as an overlay. The expectation is that our framework and algorithms can eventually be absorbed into the network fabric once the benefits of intelligent flow handling have been adequately demonstrated along with the new classes of value added services that can now be deployed and enabled with minimal change to the existing infrastructure. The key focus of the cross layer optimizations that we will be studying are those that can be realized when a routing entity is provided content level information regarding the data streams that it is handling. A goal of this thesis is to validate that by coupling this content information along with external context information, end user experience with networked applications can be significantly enhanced over todays existing models. We will also demonstrate new services such as content adaptation that can now be handled within the network efficiently without needing dedicated proxies to be deployed at the edges of the network. Context information can include current network state (congestion, link failures etc.), application profiles (security requirements, delay, jitter etc.), network technology (wired, hybrid, MANET, cellular etc.) and user profiles (customer paying more for service, end device capabilities etc.).

For any content labeling solution to be viable in a large scale network, it must be both flexible and generic. Having a proprietary content labeling scheme does not scale well and forces every routing entity to know how to handle each content providers individual labeling scheme. A novel approach that we are proposing in this research is to employ RDF/RDFS [21, 20] as the medium to describe the content that is being handled. As demonstrated in its application in the semantic web, RDF/RDFS is very flexible, generic and has seen widespread acceptance as a defacto metadata markup

for web content. By utilizing RDF/RDFS as the mechanism to markup flows/packets, intermediary intelligent routing entities can use this metadata to reason over their existing knowledge base to determine how best to handle a given flow. In addition, inferences can be made to further "generalize" or "specialize" a given flow. For example, a router that can handle MPEG-4 [14] streams can choose to handle a particular packet as though it were part of a multimedia stream (ie. generalize) or a part of a P frame (ie. specialize) depending upon the granularity of the description provided and the knowledge base of the router. For our research, we will specify an ontology to use for content description. We will investigate techniques to perform this content tagging both in-band and out-of-band. We aim to show that coupling content level information in this manner will allow intelligent routers to offer new services in an incremental fashion and provide a more efficient data transport network from an end-user perspective by handling flows in a smarter manner.

Our answer to the question that we posed at the beginning of this chapter:

We contend that an alternate model where the network is no longer viewed as a simplistic data forwarding mechanism but rather as an intelligent content delivery medium will be able to better support future networking needs. To realize this model, a holistic end-to-end cross layer approach coupled with metadata describing the various data flows presents a generic, flexible and incrementally deployable solution.

Chapter 2

Background and Related Work

This chapter outlines some of the existing and ongoing research mainly in the areas of overlay and peer to peer networks, cross layer optimizations, intelligent networking, Internet content adaption mechanisms and semantic metadata concepts.

2.1 OverLay Networks

The current Internet is highly resistant to change. Also, it spans across a multitude of autonomous domains where each domain is independently managed and owned. For data to be successfully routed from one point to another across a wide area network, well established collaboration is needed between the routing entities across autonomous domains. While this is absolutely critical for data transport, it does present a limitation for the deployment of new services into the Internet. Also, many networks are built on legacy platforms whose limitations can prevent the eventual deployment of the feature into the network. Before a new service can be supported inherently in the network, all organizations with vested interests (network vendors, service providers, content providers, standards organization etc.) need to get involved in a standardization process that is both resource and time consuming. As witnessed

with the delays in the role out of IPv6[47] and Multicast routing into the Internet, this can span many years before it eventually gets deployed. A viable solution that can be followed in the interim is to use network overlays as the initial deployment mechanism. Overlays serve primarily as testbeds that can be incrementally deployed over an existing network to demonstrate the benefits of proposed new services.

Several different types of overlays have been proposed each with specific goals in mind. MBone[78] is an overlay network built on top of the existing Internet to provide multicasting capabilities. Similarly, 6Bone[1] is an overlay on top of the Internet enabling IPv6 demonstrations and application deployments. ABone[2] is an overlay for supporting active networks research. Resilient Overlay Networks [29] is an effort to improve the resiliency and availability of paths over the existing Internet. RON is an application level overlay where the RON nodes monitor the quality and availability of paths between themselves and decide whether to route packets directly over the Internet to the destination or intentionally route packets through some other RON node to optimize an application specific routing metric. OverQos[84] is an overlay network that utilizes the notion of *controlled loss virtual links* to bound the loss rates observed by traffic flows. The idea is provide statistical loss and bandwidth guarantees on traffic aggregates to bound the QoS offered by the network. Application specific preferential treatment for packets can also be performed such as dropping less important ones to accommodate the more important packets. However, having policies that are application specific tends to have scalability issues as the domain of applications grows. Also, the main focus of OverQoS is to bound QoS guarantees not necessarily to improve them. Our proposed research will focus on both these areas by using the lessons learned from this work but going further to generalize and further

improve on available capabilities of the network.

Semantic Overlay Network(SON)[46] is a peer-to-peer overlay that utilizes semantic information about the participating peers to determine how best to route a given query. Nodes are clustered based on their semantic likeness so that a query can be efficiently routed only to nodes that have a high probability of being able to handle that query. This work is interesting to our research mainly in that it uses semantic information for clustering in a peer-to-peer. One of the areas that we will be investigating is this identification of particular overlay networks to use for a given type of data assuming the existence of multiple service specific network overlays. Handurukande et al.[5] exploit similar semantic clustering to connect peers based on their locality of interest. Loser et al.[65] present a super peer based semantic overlay built on top of a distributed hashtable providing providing a catalog service. Nejdl et al.[70] present a similar system utilizing formal RDF[21] metadata and supporting queries expressed in RDF-QEL[19].

The Internet Indirection Infrastructure(i3)[82] is routing overlay that aims to simplify the deployment of new Internet services by decoupling the networking binding between a source and sink. Here identifiers from the sender are used to map to interested receiver without direct binding to a receivers network address. This approach supports issues such as mobility, multicasting, anycasting etc. This is beneficial for our research mainly as an overlay on top of which we can build our framework. We will focus on enhancing the simple rendezvous mechanism used in i3 to one that is based on semantic inference (similar to SON) providing a platform that is capable of intelligently routing data over a large scale network that our framework can exploit.

Network virtualization[30] is an idea to move the overlay right into the network itself

allowing the simultaneous coexistence of multiple overlay networks on top of a single physical substrate. The idea here is to utilize a new breed of routers that will allow multiple networks each with potentially varying technologies to coexist. Virtual nodes and links can be abstracted on the same physical hardware. The authors claim that one of the limitations of current overlay approaches is that they still rely on the existing Internet to enable the data communication. This limits the scope of architectural innovation to incremental changes only. Through virtual networks, the authors claim that radical networking changes are now possible while ensuring complete isolation between the various virtual networks so that one does not interfere with an other. The framework that we are proposing to build in this research can definitely make use of the processing power that is offered by these new routers to form our content aware networks. While advances in network processors and FPGA technologies do point the way towards such routers being widely available in the near future, how quickly this will be adopted by ISPs is debatable. Till that time, traditional wide area overlay networks will have to serve the need for the research projects such as ours that are pushing for a change in the way we do networking today.

2.2 Peer-to-Peer Protocols

Many of the present overlay networks are built using a peer-to-peer protocol as the underlying mechanism. Based on type of peering protocol used, the overlay that is formed can either be structured or unstructured. Unstructured overlays utilize simple construction mechanisms resulting in random overlay graphs that make searches inefficient. Gnutella[6] is one of the more popular P2P protocols for building file sharing overlays. It relies on a simple but inefficient controlled flooding to perform searches

that are both time consuming and bandwidth wasting. Freenet[45] is another unstructured overlay which utilizes resources of the participating nodes to store data such that the data originator and the origin and destination of data transmissions are anonymous. The general model is to route requests towards nodes that are likely to handle that request (based on key proximity) along with data caching to support future. Napster[16] utilizes a central directory that clients can connect to publish and lookup resources. Kazaa[9] and Morpheus[12] utilize a more hierarchical graph with the concept of nodes and supernodes. JXTA[8] is a peering protocol based on XML message passing. Basic mechanisms are specified using which peer-to-peer applications can be built. In this model, the physical connections between the peers which can utilize any underlying protocol such as HTTP, FTP etc. is abstracted into higher level pipes with messages routed at the application level.

Structured peer-to-peer protocols result in overlays that conform to a specific graph structure. These protocols tend to incur more overhead for the construction and maintenance of the overlay. The benefit, however, is that lookup and routing on these overlays is highly reliable, efficient and scalable. Most of these structured overlays use a key based routing interface that routes messages to a node responsible for a given key in a deterministic manner. CHORD[83] is a Distributed HashTable (DHT) providing a scalable lookup service for large networks. A consistent hashing function such as [49] is used to map a key to a node in the network such that look ups take O(log N) messages and the routing information needed at each node is O(log N). Each node is only required to know the neighbor that exists clockwise in the identifier space. In general, each node also maintains a set of neighbors in a *finger* table. The i^{th} entry in the finger table for a node N is the closest clockwise node such

that $(N + 2^{i-1}) mod ID_{space}$. To route a message to an object identifier D, node N sends the message to its $(log_2(D - N))^{th}$ finger entry. CHORD does not utilize any spatial locality information when routing as neighbors in the object ID_{space} could in reality be separated by many hops in the underlying physical network.

Pastry[76] provides a similar function by mapping a 128 bit object identifier to a node (or k nodes where k is the degree of replication of the object) whose object id is the closest match based on Plaxton Mesh. Leaf sets are maintained at each node containing nodes that are close to each other in the id space (not necessarily spatially close). Before a message is routed, it is checked to see if it falls within the leaf set in which case, it can be directly routed to the closest matching node in the set. Otherwise, Messages are routed incrementally towards the destination through prefix matching using routes stored at the node. In case a node with a longer prefix match is not in the routing table (or is down), a node in the leaf set that is closer to the object id is chosen as the next hop. Pastry attempts to route messages more efficiently by utilizing proximity information of nodes in the underlying Internet (based on round trip delays) such that each routing table entry refers to a node close to the local node among all nodes with the appropriate node id prefix.

Tapestry[93] is a similar p2p system aiming to provide decentralized object location and routing using an incremental suffix based routing using 160 bit identifier space. Tapestry relies on modified surrogate routing[94] by matching an object to a node whose id closest matches in the greatest number of trailing bits. Multilevel routing table is maintained at each node where the level is the length of the matched suffix between the current node's id and the identifier that is being routed (actually plus 1). At each node, an attempt is made to route to a neighbor at a higher level till

we eventually arrive at the object (short circuiting is possible if along the route, a node that has a direct pointer to the object being searched is found). If there is no match at a higher level, the same level is checked to see if we can route to a neighbor that is close enough to the next digit that is being matched. Unlike Pastry where replicas are randomly located, Tapestry replication tries to replicate objects close to the locality of the request.

Content Addressable Networks (CAN)[74] also implement a DHT. Each key is mapped to a point in d-dimensional cartesian space. Each node in the network is responsible for a d-dimensional cube in space. To locate an object, the request is routed to the node responsible for the space that that key maps to as this is where the key-value mapping will be stored. Each node maintains information about all nodes whose d-dimensional cubes are adjoining the current nodes cube. For routing a message over the network, a node chooses a neighbor which has the smallest cartesian distance to the destination. This cartesian distance will monotonically decrease till the message arrives at the node responsible for the cartesian space that the search key matches.

2.3 Active Networking

Active networks is an approach to inject intelligence into the network. At an extreme case, this can be viewed as an effort to augment data packets with code fragments containing specialized processing logic for handling that packet. "Active" routers execute the code carried in the packet allowing for highly customized handling of flows or packets. This architecture permits massive increase in the computation performed within a network allowing for the deployment of new services into the network in a totally seemless and on demand manner. This is the exact model followed by

Softnet[92], a user programmable packet radio network. Packets transmitted over the network contained code written in the FORTH programming language that was interpreted at every node to control how that packet should be handled. Problems with system security and stability prevented widespread acceptance of Softnet for Ham radio community.

Smart Packets[79] approach utilizes a specialized programming language called sprocket and an associated assembly language called spanner to encode a complete program into a single IPv4 or IPv6 datagram. The program was executed by a virtual machine instantiated at each intermediary node. Typical applications include network management applications, detecting network anomalies, host and interface configuration etc. Security was enforced through language design techniques.

The SwitchWare project[27] took a similar approach of utilizing a specialized language called PLAN[58] (Packet Language for Active Networks) whose capabilities are restricted to only performing "safe" operations on any node. PLAN code is embedded in "active" packets for executing on the intermediary routers. Downloadable extensions written in Caml provide the service primitives (called active extensions) that the active packets can call upon. All of this is supported by the core framework written in OCaml. Security is handled both from a traditional sense (cryptographic verification of code, trust relationships etc.) and from a programming language perspective (based on formally verifiable lambda calculus, only simple data and control structures supported, strongly typed, resource bound, no persistent state retained for any active packet and very restricted state change allowed at router).

ANTS[88] (Active Network Transport System) utilizes a combination of mobile code, demand loading and caching. Similar to the other models, an ANTS network consists

of nodes running an ANTS platform, packets are replaced with smart capsules and mechanisms are built into the model for on-demand code dissemination. Every capsule identifies the processing routine to use for handling that packet. If the routine is not available at a node, it is dynamically loaded to handle that packet. For security considerations and unwanted interactions between the various co-existing protocols that may be running on an active node, only a limited set of special primitives are exposed that can be used for expressing mainly forwarding routines. A Java based toolkit is available for enabling an ANTS active network.

Liquid Software[55] follows a similar model based on mobile code. The mobile code here is intended for more efficient data transport and not for generic computation. To accomplish this, a specialized operating system called Scout[69] is utilized. Scout is a light weight, communication oriented and configurable environment offering efficient communication primitives. The prototype system [56] built using the ANTS toolkit to demonstrate the benefits of this scheme utilizes a reimplemented Java Virtual Machine that is customized for quick compilation. One of the key goals of this project is to ensure quick just-in-time compilation of downloaded code to optimize their performance along with static language verification. Security is enforced through user control, implicit trust and verified access.

Active Signaling Protocol (ASP)[39] takes a different approach to active networks by going away from the capsule model. In this work, the packet carries a reference to some portable code which can be downloaded and run on an ASP Execution Environment (EE) as necessary to enable a new active application. Each active node runs a node operating system that can support multiple ASP EEs simultaneously by resource restricting each EE and shielding one from the other. The ASP EE provides

a rich operating system like platform on which Active Applications (AA) can be run. Each ASP EE can support multiple AAs concurrently and is responsible for shielding them from each other and the node's operating system. Each packet contains an AASpec that specifies a AAName (a globally unique name), location where the code can be downloaded from and a AABase which is the entry class (Java is the language used for this) for bootstrapping the application. Unlike other models, the AAs can be persistent and have access to system resources like the file system (the amount of access is restricted though). The programming model used is a restricted version of Java.

Netscript[91] is another programming language and execution environment for active networking. Netscript code creates a mobile agent that can be dispatched into the network and executed dynamically at runtime. Netscript is a dataflow programming language where higher level constructs provide abstractions for units of resource allocation, flow management and security. Through this abstraction, the heterogeneity of the underlying network is hidden simplifying the deployment of new applications into the network. The Netscript runtime interprets the netscript program bundled as a mobile agent to perform special handling of data streams flowing through the smart router.

CANES[77] (Composable Active Network Elements) is an active networks framework focused on service composition. A composition based language, LIANE, is used to construct composite network services from components ensuring high performance, scalability, security and ease of management. The idea here is that higher level services are composed of lower level programs that contain processing slots. These slots

can be customized by the users to insert specialized logic. The current implementation of CANES execution environment runs on top of BOWMAN which is a stripped down version of the NodeOS specification.

Active Services[28] takes an alternate model to introduction of intelligence into the network. The claim here is to restrict the intelligence mainly to the application layer thereby preserving the routing and forwarding semantics of the Internet architecture. Clients can request new active services called servents to be instantiated on active nodes to support their specific application needs. The service environment used is an extended Tcl interpreter (the example implementation provides multimedia extensions). The servents are implemented as OTcl scripts that are interpreted by the runtime. Service platforms are located through DHCP or through advertisements. A control protocol (ASCP) is specified for launching, configuring and terminating servents on a platform. The reference implementation supports a media gateway providing on-the-fly transcoding services and supporting non-multicast clients accessing multicast media streams.

A similar application level processing of user-data is considered by Bhattacharjee et al.[36] to handle congestion control in the network. The architecture allows for applications to specify intra-network processing so that bandwidth allocated can be intelligently reduced in a manner tailored to the application rather than generically. Each data packet is tagged with a Active Processing Function Identifier (APFI) specifying the function to be computed by an active node on the packet and an associated set of labels called Association Descriptor. Through this framework, the challenges of end-to-end congestion management can be moved into the network. Some of the example functions that have been demonstrated include buffering and rate control,

unit-level packet dropping, media transformation and multistream interactions.

2.4 Going Beyond "Best-Effort" Networks

The current Internet is viewed as a "best effort" network - an attempt is made to deliver a transmitted packet from source to destination without any guarantees on reliability, security, or any other quality of service characteristics. While this model is quite simple and works (as witnessed by the Internet), it does pose several challenges to application developers that so far, has been handled at the edges of the network (such as reliable transport protocols, secure sockets etc). There have been several attempts to move the network to offering "better than best-effort" service.

Integrated Services(IntServ)[37] is a architecture for enabling QoS guarantees to be supported by networks. IntServ proposes a fine grained QoS system that is based on a per-flow resource reservation scheme. The basic model utilizes a packet classifier, scheduler, admission control and a reservation set up mechanism. Resource Reservation Protocol (RSVP)[38] is the signaling protocol used to convey a "flow-spec" and "filter spec" to intermediary routers describing the resource reservations that needs to be made. A "flow spec" contains a TSPEC describing the traffic characteristics and an RSPEC describing the needed guarantees. The packet classifier is responsible for mapping the input flow to its corresponding resource reservation using the "filter spec". The packet scheduler is responsible for appropriately scheduling a packet on the output queue such that the QoS needs of the flow are handled. The call admission control module is responsible for handling flow reservation requests. It uses local state and policy to determine if a given request can be accepted or not. The resource reservations are receiver initiated - the sender sends PATH messages towards

the receiver carrying flow characteristics and the receiver sends back RESV messages that follow the exact reverse route making the reservations along the way. All of the state maintained at the routers are soft and need to be refreshed periodically. One of the drawbacks of IntServ is the maintenance of this soft state on routers preventing it from scaling for large network sizes.

Closely coupled to IntServ is the policy admission control framework [90]. Here, a framework is provided for a network operator to specify policy based admission control rules that can be looked and enforced at the routing elements. Two key components are the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). The PDP receives request for resource (through RSVP). The PEP contacts the PDP to determine what policy to use. The PDP uses a policy database to convey a particular policy decision to the PEP. Each PEP also has a local PDP that can used in case the PDP is not reachable. Through this mechanism, admission control policies such as time of day, SLAs, prioritization, prepaid transactions, sender specified restrictions etc can be enforced. COPS[48] is a simple query-response protocol that can be used between a PDP and a PEP for handling general administration, configuration and enforcement of policies. [57] specifies the customization of COPS for RSVP.

Differentiated Services(DiffServ)[71] takes a different approach to enabling QoS guarantees. The idea here is to move away from a per-flow, fine grain QoS model to a more coarse grained QoS model to allow for scaling to large networks. The approach followed is to use the IPv4 Type of Service (TOS) or the IPv6 Traffic class byte to convey aggregate QoS requirements. 6 bits of this field are used to specify 64 Differentiated Services Code Points (DSCPs). Each DSCP maps to a specific Per Hop

Behavior(PHB) at every intermediary router. Attempts have been made to standardize on some DCSPs to allow interoperability across multiple domains and vendors. An expedited forwarding (EF) PHB is defined to provide "virtual leased lines" offering low loss, low latency, low jitter and assured bandwidth. Assured Forwarding (AF) PHB is defined to provide various levels of forwarding assurances to data packets. AF is comprised of 4 classes and within each class, 3 precedence levels are specified to control which packets should be dropped should there be a need for this due to congestion (higher drop precedence values are dropped to protect the lower ones). Packets are marked to corresponding DSCPs at the network ingress based on Service Level Agreements. One of the main problems with DiffServ is that since it is coarse grain, it is really usable only by ISPs and not by end users. Also, marking is static and does not ensure that adequate resources are actually available to handle the QoS requirements.

IPSEC[32] is an attempt to move security from the application layer directly into the network layer. Internet Security Association and Key Management Protocol (ISAKMP) is used to set up the keys to use for encryption/decryption. Authentication Header (AH) protocol provides a mechanism to validating that the packet has not been tampered with while in transit. The Encryption Header (EH) protocol provides mechanisms to encrypt the whole packet. Two modes of operation are specified, the transport mode for encrypting the data portion of the IP packet being transmitted and the tunnel model for encrypting and encapsulating the original IP packet into an external IP packet for transmission.

Darwin[40] is an alternate effort for resource management in a new breed of networks termed application-aware networks. The idea here it to view a network as different

layers, the lowest level offering bit transport, the next level are network services that can be composed by the service provider and the upper most level being the end user applications that interact with the network services. The focus is on intelligent resource allocation within the network for value added services. Applications at the network end points submit their requests to a global resource broker called Xena. Xena uses a signaling protocol, Beagle to signal the resource requirements to the network. Local resource managers are responsible for setting up the packet classifiers and schedulers to offer the desired level of service. Applications convey their resource needs in the form of an application input graph. This input can vary in details from being highly specific to just communicating high level semantic information like the "flow of type JPEG with frame rate x and quality y". Xena can now insert semantic preserving transformations into the network to optimize the network transfer (static mappings are used to determine the resource needed for different types of flows and also all semantic preserving transformations). Darwin utilizes customizable local resource management modules (similar to the active networks paradigm) called delegates that can be flow specific. A Java runtime is the execution environment supporting a delegate.

2.5 Cross Layer Issues

It has been argued that one of the most important reasons why the current model of the Internet is so widely accepted and deployed is primarily due to its layered architecture. Layering provides a nice abstraction where functionality offered by each layer is well encapsulated and each layer can be developed independent of the other

layers. Well-defined Service Access Points (SAPs) are specified in a layered architecture that clearly defines the data and primitives exchanged between the different layers. Traditionally models such as the OSI model and the present data Internet model are examples of such an architecture. Recently, there has been a lot of in-

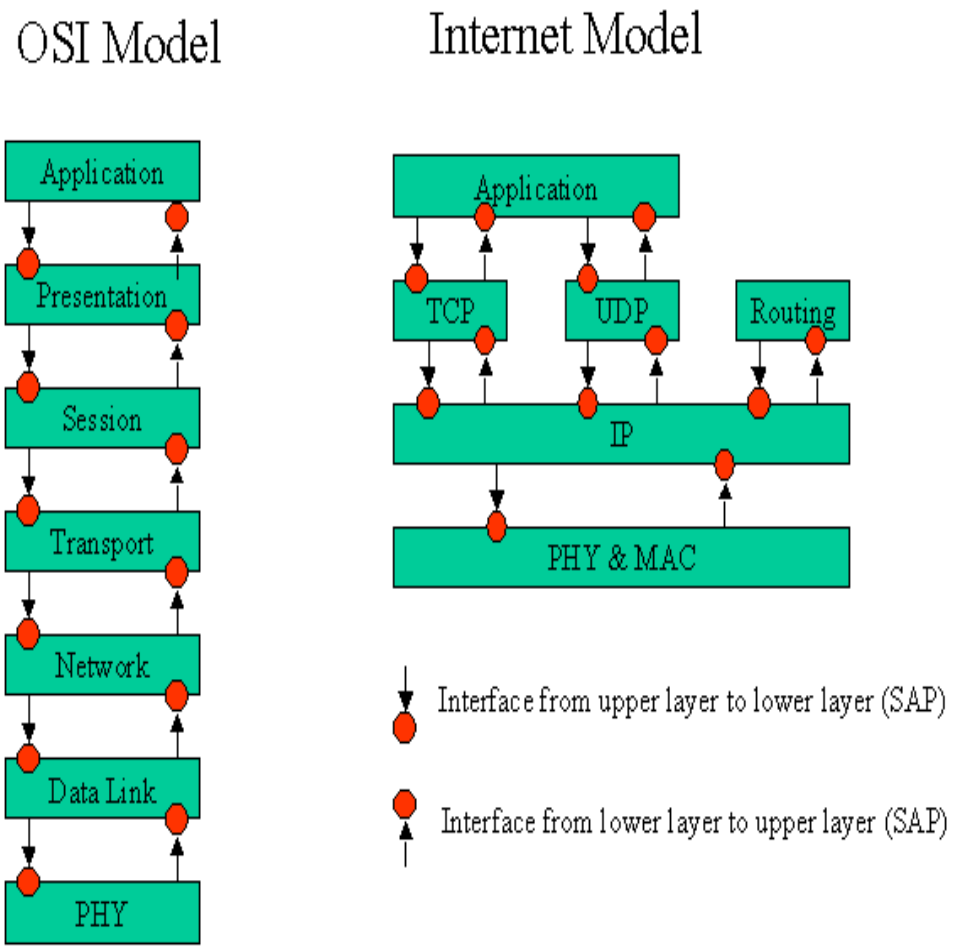

Figure 2.1: Layered Network Stacks

terest in cross layer optimizations. The idea here is to try to provide a layer in the networking stack with additional information so that the layer can make a more informed decision with the data that it needs to process hence the name cross layer

optimization. Proponents of this model argue that with such optimizations, the networks resources in terms of bandwidth, more efficient routing, better QoS guarantees, better power utilization etc. can be achieved. The counter argument [60] has been that cross layer optimization breaks the architectural simplicity of layering and is too short term in vision and highly specific (to topology, technology, application etc.) leading to spaghetti implementations and will not see wide spread deployment. The argument is that techniques such as rate adaptive 802.11 MAC and its issues with DSDV and end-to-end topology control and its issues with TCP focus on short term performance gains without due consideration of long term goals and without taking a holistic system view to ensure that their optimizations don't cause breakages. They argue that a good architecture is sometimes better than a specialized optimal solution that may not work for all cases.

While some of the arguments against cross layer optimizations are valid, it has not deterred the progress in this area. Much of the work on cross layer optimizations has focused on optimizations at the network edges primarily for wireless technologies. In [52], the authors describe a system that combines seamless handoff with adaptive video streaming. The model that authors are considering is a moving user who migrates from a high speed network (802.11b) to lower speed network (GPRS) and vice versa and all the while is running a video application on his mobile device. The authors combine USHA (Universal Seamless Handoff Architecture) with VTP (Video Transfer Protocol) to achieve this. The USHA model basically employs IP tunnels between the MH and a handoff server (HS). Applications on the MH bind to a virtual interface on the MH. All traffic is sent through the tunnel encapsulated in UDP to the HS (not sure why extra encapsulation with UDP is needed). When MH moves and

a handoff occurs (either vertical or horizontal), the tunnel endpoint IPs may change but the applications on the MH are unaffected as they are still bound to the virtual interface. The HS effectively becomes the GW for this MH. The VTP component uses a Eligible Rate Estimate to figure out what is the bandwidth available to connect to this mobile MH. Using this, the VTP source basically adjusts the data transmission rate (when in a 802.11b network, a higher quality video is used >500kbps while on GPRS, a low quality video is used <100kbps). The claim here is that using this combination, users experience uninterrupted video on their MH with the best quality that can be sustained.

MobiWeb[67] is a framework for supporting adaptive applications that have real-time characteristics over wireless links. An inter-stream priority scheme is used to address the short term fluctuations of wireless streams leaving long-term adaptation to the application. Packets are classified according to the importance of the stream that they belong to in the current context. This priority changes dynamically as new streams are added or user preferences change and are used my the scheduler. QoS information provided by the lower level network is used to adjust the stream characteristics based on application needs. In this architecture, multiple transport connections are piggy-backed over a single transport connection with appropriate scheduling between streams.

Choi et al.[43] present a cross layer model of transmitting compressed video over a lossy wireless channel. They view optimization as an inter-layer function and as an intra-layer function. To implement intra-layer functions, private parameters called operating modes are defined. To implement inter-layer functions, interfacing parameters called operating points are defined. A bottom-up approach is presented where the

lower layers expose the available system parameters and associated costs and the application chooses the most suitable system from this set. As an example, transmission rate and associated packet error probability are operating points while the associated transmitted power level is an operating mode. From an application perspective, the idea is to choose an operating point from the ones offered so as to maximize the quality of the service for the user. In [44], the authors extend this model to the concept of parameter abstraction. This maps layer specific parameters into parameters that are comprehensible to a cross layer optimizer. A joint optimization is performed on the radio link and application layers to arrive at the optimal layer parameters for satisfying an applications QoS needs.

Vacirca et al.[87] present a cross layer algorithm that adapts the link level ARQ to the end-to-end packet loss observed by TCP. The idea here is to adapt the retransmission according to a target loss rate used as parameter to describe the desired QoS for a TCP connection (in their work, TCP over a UMTS link is considered). In this work, the TCP layer passes the end-to-end TCP loss rate to the link layer entity. Errors are detected on receivers based on receiving 3 packets with higher sequence numbers than the lost packet and on senders based on fast retransmit, both measures are used to compute average packets transmitted between losses. This information is used to update the ARQ mechanism in the MAC layer.

Bucciol et al.[73] present an application level ARQ mechanism for 802.11 MAC. Here the application decides which packets require retransmission and just those are retransmitted. Video streaming is the example scenario that has been considered. The scheme computes a priority function for each packet to determine the best scheduling and retransmission instants to retransmit packets. Comparing this to content

transparent MAC level ARQ shows noticeable performance gains.

Chen et al.[42] present a data accessibility service for mobile users with high success rate. They employ a cross layer approach for advanced data advertising, lookup and replication services, and a predictive location based QoS routing protocol in an integrated fashion. The framework is divided into an application layer, middleware layer and a routing layer. The routing layer uses node location information and movement parameters to compute and maintain a set of active routes with their respective QoS characteristics. The middleware layer implements a data accessibility service for advertising and sharing data. Users select the data that they are interested in. The middleware layer then retrieves the data from the remote location. The middleware maps the application need to QoS parameters. The routing layer can use the current route or recompute new routes to meet these needs. The routing layer can also ask the middleware layer to change some of these needs if the network is unable to handle the requirements (such as request compression at source). The routing and middleware layers also collaborate to detect impending network partitioning and replicate any necessary data needed from nodes that are estimated to get disconnected based on their location.

Madisetti et al.[66] present a transport layer solution to enhancing network QoS using application level information. In this work, Stream Control Transmission Protocol (SCTP)[81] is the transport protocol used. SCTP is an emerging transport layer standard for stream based applications such as multimedia flow. SCTP inherently supports multi-homed hosts. An association between two end points can run over n physical interfaces (IP addresses). Reliability and data delivery are separated so that applications can customize based on their needs. In this work, they present *VoMo*, an

enhancement to SCTP to allow MPEG streaming over multiple paths to the receiver. Here applications can split data into different streams which can be delivered across diverse routes so that more critical data (the base layer encoding) is transmitted over a more reliable connection while the less critical data (enhancement layers) can follow a less reliable route.

Kouvelas et al.[63] present an approach to support congestion controlled multicast real-time communication using self-organized transcoding to handle local repair. The idea here is to form groups of receivers that are experiencing bad reception. A group representative is then responsible for finding some other receiver that is receiving better service and is willing to transcode on behalf of these receivers. The transcoding provides local repair to the congestion experienced by this group of receivers. The transcoding parameters are constantly updated to reflect the real time state of the link being used to serve the group. Using this approach, a network friendly congestion control of the realtime multicast stream can be achieved for large scale networks such as the MBONE.

Krishnaswamy et al.[64] propose the use of a system-wide, cross-node, cross layer optimization architecture to scale the number of multimedia users and streams with varying bandwidth requirements. A distributed Network Information Base is proposed with service agents providing adaptive routing and end-to-end QoS management. For a wireless network, they present optimizations at the different layers in the stack. At the PHY layer, varying the modulation techniques and the use of MIMO antennae can dramatically affect the throughput offered by a network. At the MAC layer, issues such as the bandwidth offered by PHY with its associated PER, protocol timing overheads such as interframe spacing, random backoff algorithms, exclusion regions,

idle network time etc. are parameters that need to be considered. At higher layers, issues such as the transport protocol to use, error concealment strategies, application level FEC etc are parameters to consider. While these are parameters that can be controlled at each node, this paper suggests going one step further and specifying a system level optimization that can span multiple nodes. System level information is exchanged between nodes to inform each node of the dynamic system variations. Additionally service agents are used on each node to provide information for neighbor queries relating to the information based maintained at each node.

2.6 Content Adaptation

Content adaptation is the process of transforming data from one format to another for a variety of reasons. It is often applied for more efficient transfer of bandwidth intensive data over a bandwidth limited network. Content adaptation is also prevalent in scenarios where there target devices have diverse/limited capabilities and making the content amenable to this is required. Personalization, policy-based adaptations, security related adaptations etc. are other common scenarios where adaptation is useful.

Open Pluggable Edge Services (OPES)[33] is an IETF working group tasked with defining a common architecture of enabling edge based content adaptation. The idea is specify a platform providing networked services at the application level for offloading origin servers and improving user experience. The general scenarios where OPES is useful is in handling services that operate on request or response for data where data is in HTTP or SMTP format (RTSP is in the works). Services that operate on an incoming request intending to modify the request include features such as URL

filtering, URL redirecting, hiding requester identity, adding additional preference information to requests etc. Other services operating on the request that do not modify the request itself are those such as user profiling, billing etc. Services that operate on responses for client requests intending to modify the response include features such as content adaptation to fit client devices (scaling down images for example) and language translation. Other services that deal with responses transparently include logging and monitoring services. Services can also be specified in OPES to directly handle client requests by assembling the needed web pages together for localization and personalization. Additionally, useful features such as virus scanning of downloaded content can be performed on behalf of the users. Two types of overlays are specified, a surrogate overlay that is controlled by the content provider to generate the desired content (eg. localization, advertisement insertion etc.) and a delegate overlay that works on behalf of the client (content filtering, content adaptation etc). A call-out mechanism is specified allowing for multiple OPES processors to be chained to provide a higher level service. Internet Content Adaptation Protocol (ICAP)[50] employs a nearly identical model for content adaptation for HTTP. OPES group is looking at ICAP to see if the ICAP call out mechanism is appropriate for the OPES model. Intermediary Rule Markup Language(IRML)[34] specifies the rules to determine what service should be used on what type of content.

Content adaptation has been studied extensively in the past through proxy based systems. [41] presents an application level content adaptation for multimedia data as an efficient solution for handling dissemination of rich content to a wide variety of devices connected over links with diverse characteristics. In this work, the available network resources, type of client, device characteristics, type of data etc. are used

to perform transcoding of data to meet the needs of the client. Compression metrics of JPEG are altered to enable a quality versus size decision at a proxy or server to enable the most suitable version of content to be delivered to a client.

Fox et al.[51] argue that with the diversity of end user devices, on-the-fly adaptation by translational proxies at the application level is both a necessary and a cost effective, flexible solution. They argue that placing these proxies within the network infrastructure is more efficient than inserting into end servers leading to more incremental deployment strategies and amortizing costs by moving complexity away from clients and servers. Adaptation of content here is through data-type specific lossy compression and distillation. They employ a TACC (transformation, aggregation, caching and customizing) model where high level services are built by chaining to together lower level atomic services on a cluster-based server architecture. The Ninja[53] framework provides an implementation of this architecture by utilizing the notion of a path that allows services to be composed starting from a service provider through an active content adapter through to the end device.

Mohan et al.[68] present a framework for adapting multimedia web documents to optimally match the needs of the requesting client devices. Their model employs two key components, an InfoPyramid that provides a multimodal, multiresolution representation hierarchy for the content and a Customizer that picks the best content representation that provides maximum value to the client. They address one of the problems with proxy based transcoding - the content provider does not control the actual displayed to the user. In this model, the content author provides the transcoding policies and controls the adaptation and this is done in an offline manner (as opposed to on request in most proxy based transcoding systems). The argument

is made that this model is better than a transcoding proxy model since the entire control is at the server and can take semantic information about the content into the adaptation process. They also argue that by transcoding at the server, customized, smaller content is delivered over the net.

Conductor[89] is another content adaptation framework that aims to move the network complexity out of the application and into the network. Conductor is an application level framework that can dynamically deploy multiple adapters to operate along an application's communication path. A planning algorithm is builtin to determine what adapters to use and where. Conductor works by intercepting the socket call made by an end application and from this deduces the type of data. Conductor probes the current routing path to identify Conductor nodes along the path. Local information about each node (link capacity, CPU resource etc) is collected and all this is fed into the planner. The planner then specifies the series of adaptation that is required along the data path. End-to-end reliability is replaced with a link-by-link reliability model and the notion of semantic segmentation is used to identify how to handle retransmissions and failures.

Subramanian et al[85] present a Content-aware Active Gateway (CAG) architecture allowing for on the fly content adaptation. They rely on the ability to filter specific types of traffic by identifying patterns in the header or payload of packets. In addition to network QoS, a compute QoS (CQoS) can be specified which determines if the application on a CAG will be supported on a general purpose CPU, a higher performance network processor, or at the FPGA or ASIC level. They demonstrate two types of services enabled by a CAG; multicast media streaming over unicast links and JPEG transcoding for HTML content. The general idea is to run specialized

programs (statically installed or downloaded) on enterprise/residential gateways that can register filters (port number, IP address etc) on the packet routing fabric. These filters are activated when packets matching those criteria are encountered and passed to the CAG computation layer for processing. Here application specific decisions can be made (content duplication for supporting multiple unicast or JPEG transcoding) and all this has been demonstrated at close to line speed.

Several other attempts at content adaptation have been undertaken. [35] is a proxy based web content adaptation for supporting browsing by mobile devices over wireless links using user specified preferences. [86] uses a link level redirection infrastructure called SelNet that tags packets at link level with function identifiers to enable a proxy based content adaptation. Ardon et al.[31] propose a server centric content adaptation framework creating service specific overlay networks through the use of dynamic proxies along the data path. Steinberg et al. [80] propose a client centric content adaptation using Web Stream Customizers (WSC) allowing for system based and content based customization. In addition to research projects, several commercial establishments have been launched with content adaptation as their business focus. VoiceAge Networks[24], Volantis[25], Adamind[3], LightSurf[10], SenseStream[22], Mobixell[11] just to name a few.

2.7 Semantic Markup

Resource Description Framework (RDF)[21] is a framework for representing metadata regarding web content using XML as the encoding mechanism. RDF provides a standard framework for interchanging information across applications. RDF is based

on the idea of identifying things using Uniform Resource Identifiers (URI) and describing resources in terms of simple properties and property values. A statement is a triple specifying a subject (using URI), a predicate (representing a property) and an object (representing the value). This results in structured graphs with nodes for subjects and objects with arcs representing the predicates. RDF provides a number of additional capabilities, such as built-in types and properties for representing groups of resources and RDF statements, and capabilities for representing XML fragments as property values. RDF Schema[20] is the mechanism for specifying vocabularies that can be used to form RDF statements. The vocabulary gives the actual meaning to the statement. The RDF Schema facilities are themselves provided in the form of a specialized RDF vocabulary. Through this, objects and their properties and what they refer to can be specified.

Providing markup for content is also prevalent in many other areas albeit not always using a standard machine processable means such as RDF. Session Description Protocol (SDP)[54] is a textual description for multimedia sessions. MPEG-7[15] and MPEG-21[13] are other multimedia markup standards. TV-Anytime[23] is a specification for audio-video content markup. ID3[72] tags applied to MP3 files can convey metadata such as titles and composers.

Chapter 3

Motivation and Research Focus

The approach to move intelligence into the network is not new. Several approaches (as shown in chapter 2) have attempted to achieve exactly this objective. However, in reality, the degree of success for these approaches has been limited. Part of the reason has been the variety of solutions that have been proposed. While the benefits are real, the downsides such as being too radical, not scaling, requiring deployment of specialized hardware etc. make these solutions viable only in specific scenarios. What is needed is a generic and extensible framework that can be incrementally deployed to seemlessly offer new value-added services in the network without specifying an absolute need for infrastructural changes. As illustrations, let us consider some scenarios where intra-network intelligence is a necessary enabler for value added services.

Scenario 1: Adam is watching a streaming video on his home computer for which he has paid $5.00 for a video-on-demand service offered by Disney through his Comcast cable modem. The movie is currently stored at a remote location from which it is being streamed. At the same time, Adam's neighbor, Bob, through his cable connection is attempting to stream a free trailer from Pixar (through layered encoding) which is located at a different remote site. Since the cable is shared between Adam and Bob,

their respective data access is going to affect each other due to network bandwidth limitations. Comcast has an established business agreement with Disney but not with Pixar. For this reason, the Comcast router detecting congestion on the outgoing link towards Adam and Bob starts to preferentially schedule Adam's traffic. Some of the enhancement layers for Bob's traffic are dropped. Around the same time, Chris decides to watch a free trailer for an upcoming Disney movie. The Comcast router instead of further downgrading Bob's traffic realizes that this is a free trailer request and informs the content provider of the available bandwidth. The Disney router streams the content using multidescriptor encoding. Based on the available bandwidth, the Comcast router drops some of these descriptors. Zack in the meantime starts a FTP download. The Comcast router rations out a part of the available bandwidth to Zack knowing that reliability is more important for Zack than delay or jitter and scheduling Zack's packets into bursts that can be interposed between the different streams. Using this model, Adam is getting the best service since he is paying extra for a particular flow, Bob gets acceptable service that is not too low since he is still a valued Comcast customer, Chris gets to view the desired content over a congested link with acceptable quality and Zack's download is supported at the same time.

Scenario 2: Sergeant David is viewing surveillance information streaming from wireless cameras located overseeing a secure facility. Data is being routed through the public Internet to his desktop. Around this time, a line break occurs in the network and traffic is being rerouted through a lower speed link causing congestion on this link. The network reacts by first realizing that data for Sergeant David is mission critical and cannot be lost. For this reason, it attempts one of two approaches, it

tries to reroute this data by asking the routing layer to find an alternate route that is not necessarily shortest path. Also, since this data is sensitive, IPSEC is used to traverse untrusted parts of the network that may run over foreign soil. When such a route is located, if it still congested, on the fly content adaptation is done in a step wise fashion - first drop to a lower fidelity, then remove color, then reduce frame rate etc. Sacrifice other traffic to ensure that a minimum level of quality is maintained for this flow provided the other traffic is not so critical. Sgt. David notices movement and sends a command to camera 1 to pivot to the right 30 degrees. This needs to be delivered reliably to the wireless camera. At the last hop to the camera, the ARQ mechanism increases its retransmission counter to ensure that reliability for this packet is higher and quicker (compared to reissuing the command). The data coming from this camera is now more important than the others and accordingly, available bandwidth is partitioned to reflect this.

Scenario 3: A video presentation is being multicast to receivers some of whom are connected through desktop PCs, others through PDAs and some just listening in through WiFi capable cell phones over the web. The content carried in the data flow is adequately marked as video and audio. The MPEG-4 encoded video has its various AVOs properly classified as in [26]. Along the multicast tree, at the forking points, decision is made to route only relevant data to the downlink such that all video objects are removed if there are no downlink video receivers. Additionally, a desktop PC with the viewer minimized does not require the video frames (since the client is not viewing it) streamed at its current fidelity. The network reduces the frame rate automatically. When the user maximizes the viewer, the network reacts by reverting the frame rate to the original value. Similarly, a fork point automatically drops unimportant AVOs

when downstream receivers are handhelds with low resolution.

These scenarios highlight two key features used by the network to offer higher levels of service. In scenario 1, the resource management issue for providing QoS is content aware. This awareness goes beyond just knowing the sampling rate and content type but to higher level constructs such as business relationships and client profiles (such as one who is paying more than the other). These can be highly dynamic in nature and can be quite complex. Similarly, generic rules such as drop B frames before P or I frames for MPEG transmission don't take the priorities between 2 MPEG streams into consideration. It is highly possible that no frames for a specific stream should be dropped (Adam's in this case) while both B and P should be dropped for the other (Bob in this case). Scenario 2 highlights a case where intra-network content adaptation is a key feature of the network. A network should degrade gracefully. Efforts should be made to "keep things going" for as long as possible till they are meaningful. Reducing video quality is a suitable content adaptation only upto a certain point beyond which it may be unacceptable to one end user but not to anther. For this reason, context information is essential in making content adaptation decisions. As in scenario 3, the network needs to know the user service expectation and strive to achieve them taking any optimizations that are possible.

The active networks philosophy [79, 27, 88, 55, 91, 77] of highly programmable network elements promised to yield networks where user/application specific computation can be dynamically handled by network elements. However, even though it has been around for a while, it has seen limited deployment and has remained mainly an academic initiative. One of the primary concerns has been the issue of trust. It is difficult to expect an ISP to allow some user specified code to be downloaded and executed on

their revenue generating hardware. Also, from a security perspective, it opens up the possibility of malicious users deploying code that could be potentially harmful both for the platform as well as the other traffic flowing through it. While approaches have been specified to limit the type of functionality that can be invoked on these execution environments, it comes at a cost of limiting the level of customization possible which directly impacts the underlying goal of the philosophy. Recent advances in network processors and FPGA technology do allow for significant amounts of computation to be performed, often at line speed, at these routers but it remains unlikely that ISPs and router vendors will be willing to let any untrusted (not from the vendor or associated third party) to run on a commercially deployed network. Both scenarios depicted above can, in theory, be achieved by injecting content and context specific capsules. However, it is unlikely that Comcast will allow Adam to inject code into their routers and also Adam is unlikely to be as kind to Bob which directly conflicts the goals of Comcast where Bob is a valued customer.

Traditionally, at the level of network routers, there is little visibility into the type of data that is being carried in the IP packets that are being routed. This raises the issue of how to implement service differentiation. How does a network router know that it is dealing with mission critical data that is being used to control an unmanned space vehicle versus an HTTP transfer of an advertisement tied to an HTML page that a user is viewing. IntServ[37] approaches this by explicitly making reservations for flows. However, this guarantees buffer space and bandwidth but not necessarily reliability. Also, per flow reservations have problems of scalability. Also, in environments that are dynamic, such as in MANETs, reservations don't work due to network dynamics. DiffServ[71] approach relies on packet tagging to enforce differentiation. While this

allows an ISP to specify that this packet is more important than an another based on SLAs, the end user has little control over it. In addition, it is a static mapping that is more or less invariant to the packets in the flows due to its coarse grained nature. Also, using just 6 bits, it is difficult to specify complex policies to be applied to a given flow.

Recent commercial initiatives are pointing to definite moves towards application awareness within the network. Cisco's Application Oriented Networking (AON)[4] platform is an initiative to build an Intelligent Information Network (IIN). The idea is to move application level functions such as message transformation, security or message routing into the network itself. The vision is for an application aware network that can provide end devices with the needed resources and service to meet the application needs. Alternately, it is envisioned that applications will be network-aware requesting specific configurations to be established in the network. The current product descriptions highlight adapters to perform specific adaptations but are limited in scope. Packet level inspection is the current model followed for content classification. Such industry initiatives are further validation of the move to view a network as something more than bit transport.

3.1 Research Focus

The focus of this research is to propose a new model for networking that is both content and context aware. Policies are used to drive the adaptation and differentiated services offered by the network. Supporting this model and the emerging class of complex services requires innovation in a number of areas, including support for

application-specific handling of traffic, sharing of resources between cooperating traffic streams, adapting quickly to changes in the network conditions and application requirements, systematic methods for balancing the constraints and priorities of services competing for network resources, and reliability and security. To do this, we propose to follow four key design concepts.

- Content awareness within the network.
- Content adaptation within the network.
- Policy Based Management.
- Cross Layer Optimization where applicable.

To provide content awareness, we propose to use RDF as the metadata markup language. This provides a standards based mechanism for expressing rich descriptions. Ontologies defined using RDFS can be used to build primitive vocabularies for describing semantic information about content. Content providers can use published ontologies to appropriately map flows/packets and based on this, routers can infer what type of packet it is and invoke appropriate actions. Our proposal relies on a policy based management approach. ISPs can now specify what type of action to apply based on content and context of that flow. These actions can be as simple as dropping a packet to invoking a new service to transcode a data stream from one format to another. The idea of cross layer interactions is key to achieving this objective. By exposing limited amount of information across the layers, better decisions can be made on how to handle a particular packet or dataflow. The novelty of this approach is threefold.

- Our approach is conceptually very different from the active networks technique since instead of specifying what to do with a particular type of packet, we are actually specifying to the network, what a particular packet is. This leaves the intelligence with the network operators as to how to handle a particular packet instead of relying on customer provided logic.

- The second novelty is the application of semantic techniques borrowed from the world of the Semantic Web to the realm of networking. We believe this has the potential of revolutionizing the management of networks and resources just as it did with web content.

- Thirdly, policy based management has often been publicized as a much needed feature of networks. In this research, we extend policies to use content and context information to enable powerful cross layer interactions to occur at a node and system wide across the network.

3.2 Proposed Framework

This section presents an overview of our proposed framework. We break it down into two components; at a node level and at a system level that spans the network.

3.2.1 Node Level

At the node level, the architecture we propose introduces our framework as an additional layer called the CoCoNet layer between the application and the transport layer. This layer is responsible for intercepting socket calls made by applications to the transport layer. The API is enhanced to allow the application to provide semantic

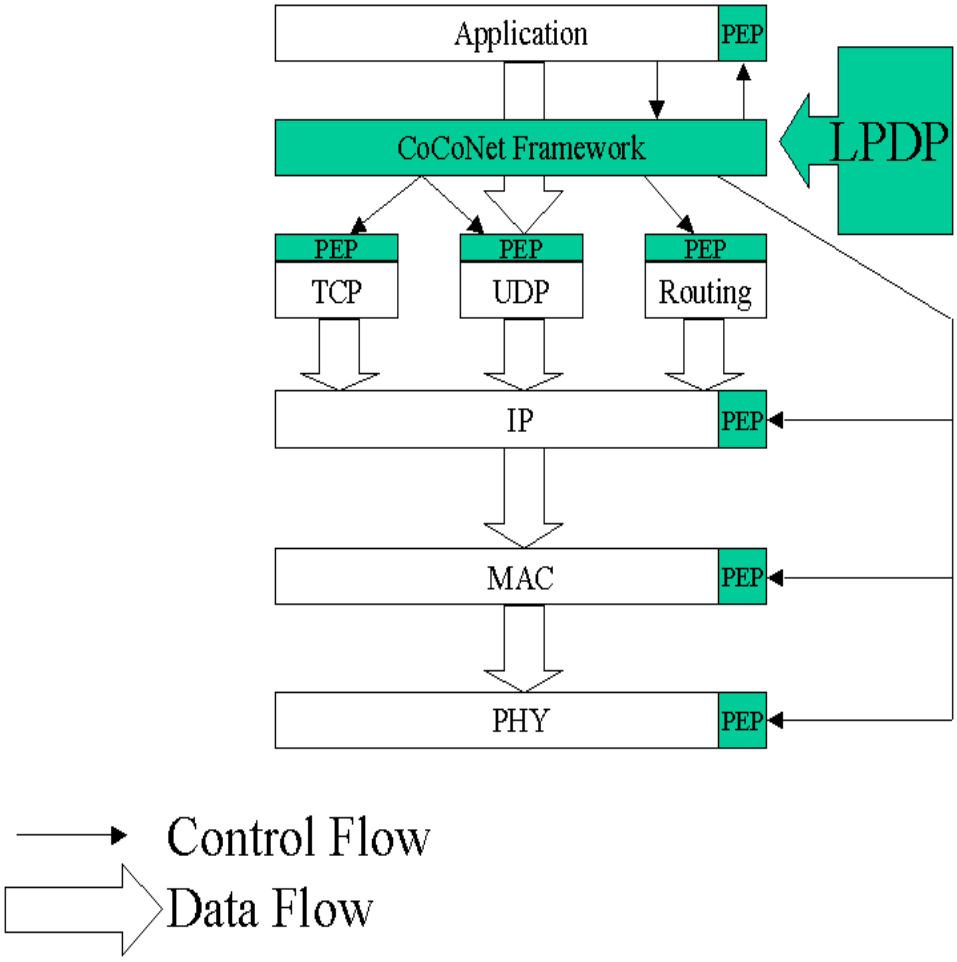

Figure 3.1: CoCoNet Node Framework

level information for messages transmitted over this interface. A local policy decision point (LPDP) (using the terminology from [90]) is used to determine what policies to enforce based on the content. In our framework, each policy enforcement point (PEP) is at every layer in the networking stack while [90] treats the PEP at a node level. Placement of the PEP at every level of the stack allows us to implement coordinated cross layer interactions initiated and controlled by our framework. Similar to [43], we

propose the notion of inter and intra layer optimizations. The PEP exposes the inter-layer optimization points that any particular layer supports. Our framework utilizes the policies stored in the LPDP to drive the settings to be applied to each of the PEPs in the stack. Essentially, we are proposing to expose a network stack as a collection of switches and dials and allow an external policy to determine the exact settings of each of these dials (based on content and context). We want to expose functionality, not necessarily how it is achieved (this falls under intra-layer optimization). For example, a MAC can advertise two different data rates and their associated packet error probabilities without exposing the FEC scheme used to achieve these rates.

3.2.2 Network Level

At the network level, we envision that there will be an overlay network comprised of routers that run the CoCoNet Framework. While the hope is that all routers will run this framework, the architecture should support incremental deployment. Client machines running our node framework communicate over this overlay. The overlay comprises of two components;

- A control plane component that involves interactions between the CoCoNet Layers at the routing elements
- A data plane component through which the data packets are flowing.

Over the CoCoNet control plane, routers can exchange traditional management information such as link states, buffer lengths etc. In addition, information such as content types currently being handled, adaptations currently available can be advertised. An additional key piece of information exchanged is the local policies that are currently

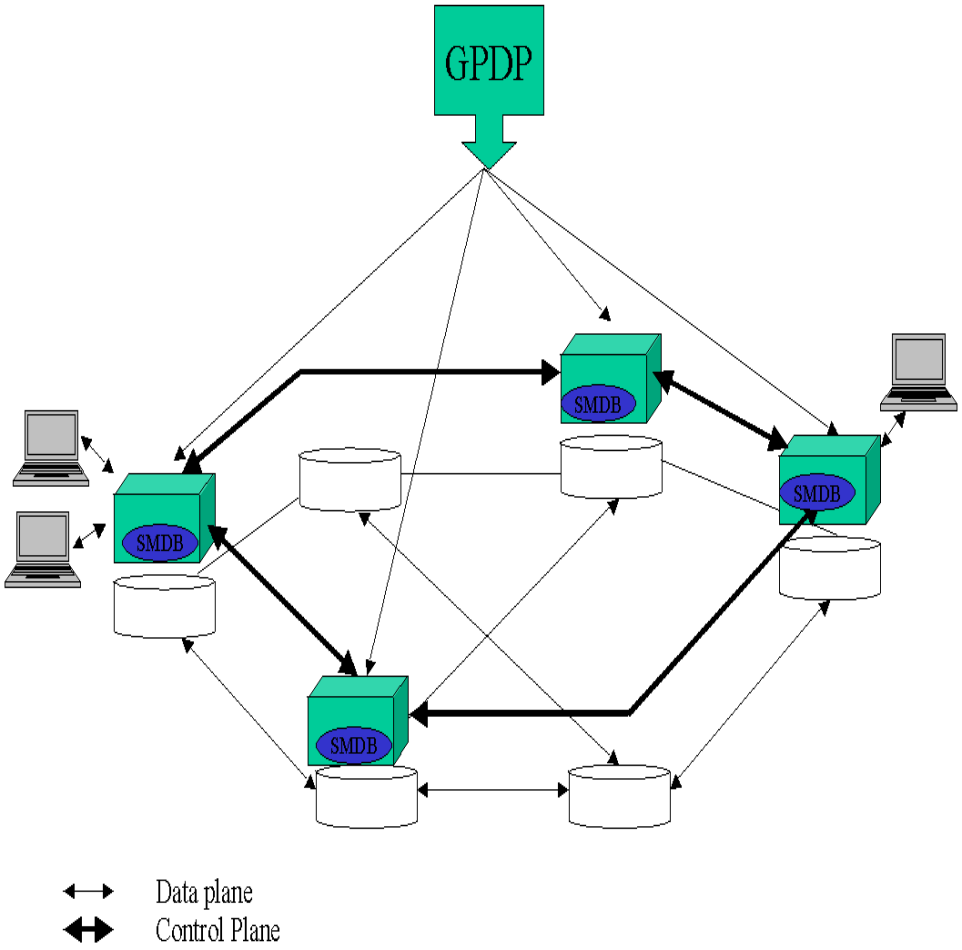

Figure 3.2: Overlay Network

being applied to a data stream that is being routed. Local PEP settings for a given stream or flow have global implications. For example, unless every hop is reliable, a data packet cannot be reliably routed through a network. The dataplane can be implemented as either

- A UDP connection between two routers
- A TCP connection between two routers
- A IP in IP tunnel between two routers

- A layer 2 LSP
- A DiffServ aware network
- An IntServ aware network

A CoCoNet framework will perform the necessary mapping based on policy, content and context. As an example, suppose a packet arrives at a router indicating that it requires reliable transfer semantics. The data plane chosen to the next hop in this case could be over a TCP connection. Likewise, a data packet indicating that it is sensitive information (telnet logins for example) but currently not encrypted can be routed to the next hop over an IPSEC tunnel or dropped if none is available (if that is the policy). The role of the Global Policy Distribution Point (GPDP) is to disseminate any network wide policies that need to be enforced. This can include, as shown in scenario 1, items such as preferential treatment that needs to be given to content originating from a particular domain, preferential treatment for a particular type of content, any content based adaptation techniques that need to be employed in the network etc. It is envisioned that the GPDP is controlled by the ISP to set forth global rules while the LPDP hosted at an enterprise location is possibly shared between the ISP and the enterprise. This can further be extended to say that the LDPD is under local user control (based on user policies and preferences) and can additionally, host user preferences. In order to propagate content level information for packets and flows, we propose to take one of the following approaches.

- The meta data can be directly encoded into the IP options field of an IP packet (size is an issue). We refer to this as “in-band tagging”.
- IP packets use the IP options field to carry a special 32 bit identifier. This

identifier is used to indicate a well known (global) content description meta data. Useful for non-flow based one-time packets such as telnet login or HTTP URL request. We refer to this as "out-of-band tagging".

- IP packets use the IP options field to carry a special key. This key is looked up in a directory service to identify the meta data describing that packet or flow. This is also "out-of-band tagging".

The use of a structured Peer-to-Peer overlay is to enable the third alternative. In this approach, before a client starts a flow through a network, it registers its content metadata with the first hop router and generates a key. This key is carried in the IP packets and is available to any intermediary router. At any point along the data flow, this key can be used by intermediary routers to fetch the metadata through an out of band mechanism.

3.3 Areas Beyond The Scope

We make the following assumptions as basis for our research

- The application specifies the content that is being transmitted. There is no classification mechanism that we will develop to infer the content based on traffic pattern, port numbers etc.

- Assumption is made that this is truthful data that is being presented. We are not looking into mechanism to validate the data that is provided.

- The content is marked using a standardized ontology. Mapping between ontologies is beyond the scope of this work.

- We are purely an application for semantic web technology in the field of networking. We are not contributing to the semantic web field. We will use the concepts and tools available and thereby are bound to their limitations. For instance, we are not intending to develop any new reasoner that has a small foot print and speed optimized.

- We are going to use the existing bit efficient XML encoding techniques and are bound to their limitations.

Chapter 4

Research Plan

The aim of this thesis is to enable networks that can adapt themselves based on content and context of the data that is being routed thereby offering key benefits to realize intra-network value added services. Novel algorithms and protocols will be developed to allow intra network cross layer optimization to be enabled. To this end, the following is our proposed research plan.

4.1 Research Tasks

4.1.1 Application-Transport Layer API Specification

The standard socket API will be extended to allow applications to pass in content metadata. Two end applications will be developed on top of this. An inelastic application that has strict requirements on delay, jitter and loss. Another will be an elastic application that has specific security constraints. These applications will be used to demonstrate the benefits of content labeling. We will also develop a simple content adaptation application that can scale down images from a web page to demonstrate intra-network content adaptation driven by a policy. The IP options

field will be used to convey this information. Initially, we will start with globally unique IDs that are known to all our routers. In the next phase, the ID carried in the options field is a key that needs to be looked up by the routers for obtaining the content description.

4.1.2 Specification of Content MetaData

In the initial phase, we will use a simple RDF[21] markup using an ontology specified using RDFS[20]. We will use the standard ontology available (DublinCore, CC/PP and searches through Swoogle) and extend for specific examples. If we encounter that RDFS constructs are not adequate, OWL-lite will be used.

4.1.3 Specification of Intra-Node Policy Enforcement Points

Here we are providing the configuration parameters that can be tuned by an external policy. For the PHY layer, this includes modulation schemes and power thresholds for wireless links. For the MAC layer, this includes ARQ and FEC adaptation including MAC specific adaptations. End-to-end resiliency and retransmission will be compared against link wise retransmission for a variety of application scenarios. For the network layer, we plan on using selective multipath and selective replication to see if we can improve resiliency. At the transport, we will expose congestion window and rate adaptation parameters. At the application level, application specific parameters will be exposed based on the example that we build (such as MPEG rate adaptation). NS2 and Glomosim will be used as simulators to study the behavior of the different algorithms that we propose.

4.1.4 Choosing a Policy Language

We will evaluate rule languages that can be used for specifying our policies. For the preliminary phase of this work, the rules used will be simplistic reactive rules. We will employ a model similar to *if (condition) then (action)* (production rules) or *on event if (condition) then (action)* (event-condition-action rules). The actions will be from a known set of functions that we can fire. In the next phase of our work, we will extend this to perform more complex inferences (our selection will primarily depend on the rule engine ie. JRULES, JENA or JESS and secondly on the ease of specifying the rules). For demonstration purposes, we will build a user interface to allow an ISP and an enterprise to specify rules that can be uploaded into the LPDP and the GPDP while flows are in progress to see the impacts of their rules.

4.1.5 Specification of Overlay Formation and Metadata Lookup

The various peering protocols will be evaluated to pick one for our testbed. For the initial phase, we will rely on a simple overlay based on configuration information. Content metadata lookup will be implemented through a well known directory. For our final testbed, we will pick a structured overlay protocol and use the key contained in the IP packets to dynamically lookup meta data on a as-needed basis.

4.1.6 Large Scale Testbed

A proof-of-concept prototype will be developed to highlight the application benefits of our scheme. A large scale testbed (similar to PlanetLab[18]) will be used to demonstrate the operation of our framework on a large scale network. (The PlanetLab nodes will play the role of the routing elements in our overlay network).

4.2 Evaluation Methodology

We will evaluate our proposal against existing partial approaches both analytically and heuristically as there is no complete existing system that we can directly compare ourselves against. Results drawn from NS2/Glomosim can be used to validate cross layer optimizations across a network of simulated elements. The successful optimizations can then implemented on top of our framework on our testbed for heuristic validation. Some of our cross layer optimization can be analytically modeled as constraints problems and benefits analyzed in this manner.

4.3 Time Line

Date	Task Completed
Today	Proposal
Dec 2005	Identification of PEP specification Phase 1
May 2006	Cross layer optimization across network, finalize PEP specification and identify a policy language to use and associated tools
August 2006	Evaluate structured overlay networks and specify ontology phase 1, Develop socket API to take content metadata, IP option header carries metadata key that is globally known
December 2006	Setup selected overlay network, start building framework to expose PEP as per specification on testbed, Demonstrate initial application developed on new API
March 2007	Complete demo setup and allow for specification of complex policies
August 2007	Testing and Refine Architecture
December 2007	Writing and Defense

Chapter 5

Preliminary Work

5.1 Using Peer-to-Peer Data Routing for Infrastructure-based Wireless Networks (Percom '03)

A mobile ad-hoc network is an autonomous system of mobile routers that are self-organizing and completely decentralized with no requirements for dedicated infrastructure support. Wireless Infrastructure in terms of base stations is often available in many popular areas offering high-speed data connectivity to a wired network. In this research, we describe an approach where infrastructure components utilize passing by mobile nodes to route data to other devices that are out of range. In our scheme, base stations track user mobility and determine data usage patterns of users as they pass by. Based on this, base stations predict the future data needs for a passing mobile device. These base stations then collaborate (over the wired network) to identify other mobile devices with spare capacity whose routes intersect that of a needy device and use these carriers to transport the needed data. When such a carrier meets a needy device, they form ad hoc peer-to-peer communities to transfer this data.

We designed and implemented *Numi*, our framework for supporting collaborative infrastructure and ad hoc computing along with a sample application built on top of this highlighting the benefits of our proposed approach.

The network model that we are considering are islands of high-speed wireless connectivity (*landing zones*) surrounded by regions of low or no network access (*transit zones*). We envision that devices that are within these islands have access to an infrastructure component (access point) while in surrounding areas, only ad-hoc communication is possible between neighboring peer devices. The key components of our network include *Service Portals* (*SPs*) which are infostations offering high-speed network connectivity and services, *Mobile Hosts* (*MHs*) that comprise user devices such as laptops, PDAs and cell phones, and *Services* such as MP3 downloads, electronic newspapers etc. *MHs* move in and out of range of *SPs* with the user expectation that the *Services* running on them are uninterrupted. Built on top of the *Numi* framework, our *SPs* are more intelligent than conventional infostation systems and can predict a users future data needs. Through collaboration with other *SPs* in the network, data can be scheduled to be piggy backed on other devices to support this device in a completely distributed manner.

We built a simulation model of our approach using Glomosim. *MHs* were assumed to move randomly between *SPs* that were uniformly placed throughout a geographic region of ten square kilometers. Nodes have access to a finite set of service data. 802.11 was used as the MAC protocol. We compared our approach against a conventional data hoarding scheme (*MH* downloads as much of data as can/needed till next *Portal*, *MHs* cannot communicate with each other) and a conventional ad-hoc querying scheme (*MHs* in a *transit zone* can communicate with peers to request data as well

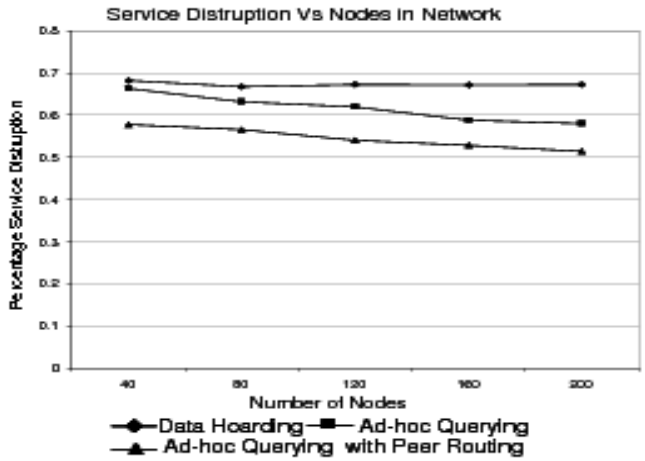

Figure 5.1: Service Disruption vs Nodes in Network

as download data from *SPs* that they visit). Our work mainly focused on modeling percentage of simulation time a node spends without data. We consider this to be a measure of expected service disruption in a network. Fig.5.1 shows the improvement in user perceived interruptions in service as they move through this network.

As shown in Fig.5.2, we also conducted tests by changing the speed of nodes in the network (100 nodes, data packet size <0.2 MB). We found that as the speed increases, the level of service disruption appears to decrease for all schemes. For data hoarding schemes, this is due to the reduction in the time that a node spends out of range of a *SP*. For ad-hoc querying schemes, this is due to the increased number of peers that a given *MH* can query. Again, our scheme outperforms the other two approaches. Increasing the node speeds allows carriers selected by *Portals* to reach a node in need quicker thereby reducing service disruption time. We also conducted tests by varying the memory capacities of mobile devices (node speed 20mt/sec). As expected, network comprised of devices with higher memory capacity suffered much less service disruption than the limited capacity network. Again, our approach results

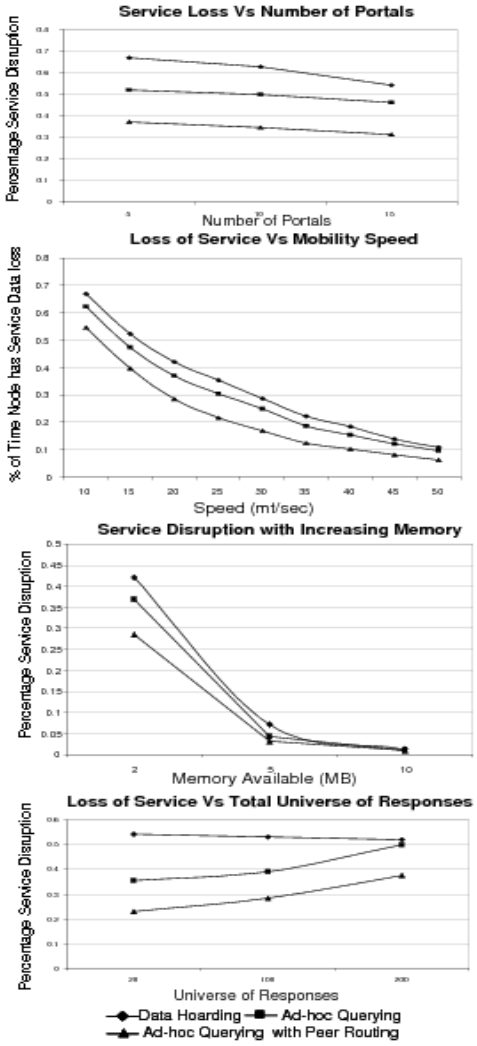

Figure 5.2: Simulation Results

in less disruption in service than both the data hoarding and ad-hoc query schemes. We varied the total universe of request/response pairs in our model (100 nodes, 5 *Portals*, 20 mt/sec, data packet size <0.2 MB). Conventional ad-hoc querying relies completely on chance that a *MHs* queries are heard by a passing peer that happens to have the desired data. We found that by increasing the universe of data in the network, there was less likelihood that a passing peers had the desired data. The ad-hoc schemes still performs better than simple data hoarding but as the universe of data grows, the difference between schemes becomes less significant. Our scheme performs better than the other two (the growth is slower then simple ad-hoc querying) as the *Portals* try to ensure that a peers passing by an *MH* in need, does in fact carry data that this *MH* would be needing.

We demonstrated the viability of this approach by building a demo MP3 application that runs on PDAs where data is intelligently routed to the device (with limited data store) such that the user does not see a service disruption as he listens to his playlist. Only 3 songs are stored at the device at any point with the *Numi* framework handling the removal of previously heard songs and replacing them with the next on the playlist. Requests and responses are routed through nodes whose movement patterns indicate that they can be used to carry requests to portals or responses from the portal to the device in need. In this research, application awareness of the network is used to perform data routing which is an network layer function. The application signals to the network layer of impending data needs. The network layer utilizes current mobility patterns (and future expected locations) to determine the best candidate node to route the data requests. [62] and [75] contain detailed information about this work.

5.2 Using Location Information for Scheduling in 802.15.3 MAC (Broadnets '05)

In recent years, UWB has received much attention as a suitable Physical Layer (PHY) for Wireless Personal Area Networks (WPANS). UWB allows for low cost, low power, high bandwidth, short reach communication well suited for personal operating spaces. One of the key features offered by UWB is very accurate ranging between a transmitter/receiver pair. The IEEE 802.15.3[17, 7] is a MAC protocol that has been proposed for WPANs. In this MAC, a combination of CSMA/CA and TDMA is used to achieve channel scheduling. The TDMA component ensures only one transmitter/receiver pair within a piconet is active at any given time thereby ensuring an exclusion region that covers the whole piconet. In this work, we propose a less stringent scheduling mechanism that allows for concurrent communication between UWB transmitter/receiver pairs within a piconet. Exclusion is necessary only when the communicating entities are close enough such that interference between them would adversely affect successful reception of data at the receivers. The Piconet Coordinator (PNC) uses the ranging information provided by UWB to accurately position transmitters and receivers. The PNC schedules parallel transmissions between distinct transmit/receive pairs as long as they do not interfere. The results of our simulations of our proposed modifications show that the network throughput can be significantly increased with very little change to the 802.15.3 MAC.

The network model considered in this paper comprises of an indoor environment

with devices participating in a WPAN similar to application scenarios such as conference rooms, a home entertainment room etc. We assume the existence of strategically placed UWB reference nodes within this environment. The locations of these nodes are well known apriori and are with respect to some predefined coordinate axis. Through this, the relative location of devices within a piconet with respect to the coordinate system followed by the reference nodes can be determined.The PNC uses the range information collected from multiple reference nodes to triangulate the location of the device within the piconet. This location information is fed into the scheduler to allow concurrent, power limited transmissions to occur within the piconet with the constraint that no two transmissions will interfere with each other. Two flows are permissible at the same time only if there is complete exclusion between the two flows, i.e.

- Transmission from $Source_{flowA}$ cannot be detected at $Receiver_{flowB}$
- Transmission from $Source_{flowB}$ cannot be detected at $Receiver_{flowA}$
- Transmission from $Source_{flowA}$ cannot be detected at $Source_{flowB}$ and vice versa
- Transmission from $Receiver_{flowA}$ cannot be detected at $Receiver_{flowB}$ and vice versa

We simulated the behavior of our enhanced scheduler using *NS2*. We considered an area of 40m X 40m and 26 randomly positioned nodes. 6 of these nodes were designated as reference nodes and one of these nodes served as the PNC for the piconet (PNC position was set to (20,20)). The line rate used for our simulations was set to 80Mbps and the all flows in the simulation were 10Mbps MPEG streams.

Fig.5.3 shows the performance comparison between the regular 802.15.3 MAC and

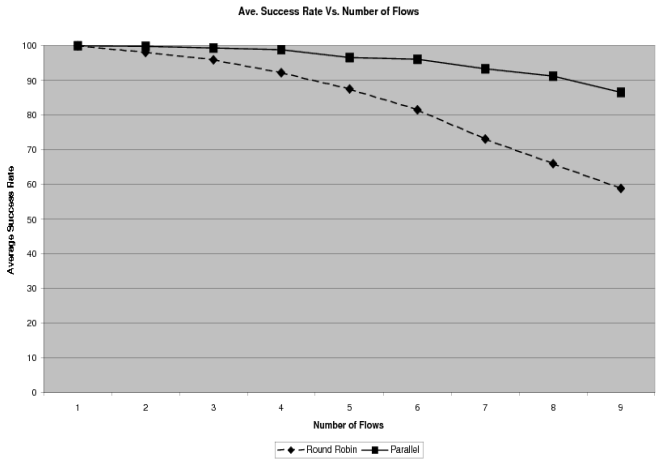

Figure 5.3: Success Rate vs. Number of Flows

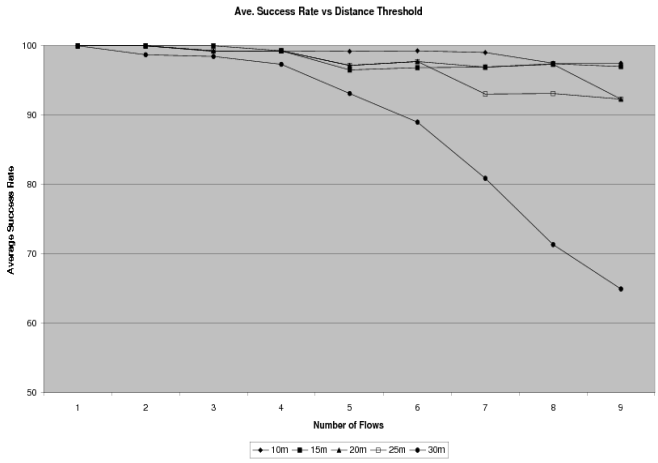

Figure 5.4: Success Rate vs. Distance Threshold

our enhancement when the number of flows in the network is increased. The GTS requests per flow (3ms), the maximum acceptable delay per packet (50ms) and the number of nodes (6 reference nodes and 20 communicating nodes) in the network were kept constant. Fig.5.4 shows the performance comparison of the two schemes when

we vary the distance between the transmitter and receiver for a flow (called distance threshold). With lower distance thresholds, we were seeing much better performance of our scheme while as the distance threshold increased, we had instances where devices at opposite ends of the area were randomly chosen for a flow. In these cases, the power levels of the transmissions were high enough that only one transmission could take place at any given time within the piconet. In general, the more localized the communication within a piconet, the better our scheme performs in comparison to the traditional round-robin approach. Fig.5.5 shows the performance comparison

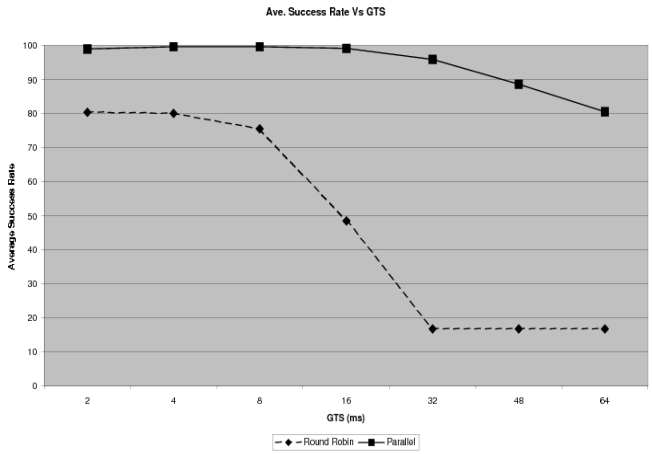

Figure 5.5: Success Rate vs. GTS

when we keep the number of flows in the network (6 flows), number of nodes and the maximum acceptable delay per packet constant and increase the GTS. The lower the GTS, the more beacon packets there are in the network due to the shorter superframe intervals. However, this reduces the average queuing at the nodes since each node gets frequent opportunities to transmit. As the GTS increases, the success rate stays more or less unchanged with corresponding decrease in the number of beacons. It can be argued that even with low GTS allocations, the number of beacon packets can be

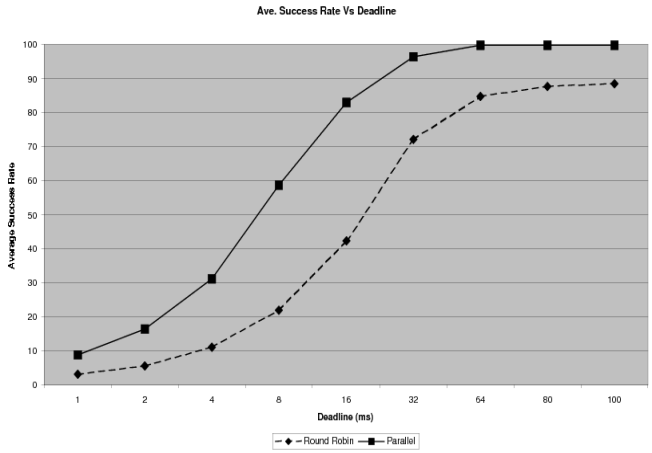

Figure 5.6: Success Rate vs. Deadline

minimized by having multiple GTS allocations within a single superframe for the same flow. For very high GTS requests, the success rate can fall significantly (in some cases to 100% failure) since packet deadlines cannot be met by a flow because a previous flow took up all the time. Fig.5.6 shows the effects of changing the maximum acceptable packet delays while keeping the number of flows and the GTS per flow constant. For low values, both schemes suffer from excessive packet losses due to queuing delays at the nodes. As the acceptable delay increases, the performance of both schemes improves. Even for relatively low delay thresholds, parallel scheduling outperforms a traditional round robin scheduling mechanism. By exploiting spatial location and power control, our scheme enables multiple flows to be scheduled in parallel thereby decreasing the time between any two GTS allocations for a given flow. This translates to lower delays experienced by the data packets at the nodes and hence the better performance.

In this work, the concept of cross layer interactions is exploited to achieve an enhanced MAC scheduler. Instead of the MAC simply using the information available to it

(requests for timeslots), it utilizes "out-of-band" information to make an informed MAC decision. A combination of power control and location information is used to efficiently schedule concurrent transmissions to achieve higher network throughput. This goes to show that going away from a rigid layered model does offer significant benefits with very minimal changes.

5.3 Reliability through redundancy in DSR

This is an ongoing research exploring content aware networking. Here the idea is to try to improve end user quality of MPEG-4 streamed through a MANET using the Dynamic Source Routing(DSR)[59] routing protocol. DSR is a reactive protocol ie. routes are discovered when needed by an application or data packet. As the name indicates, DSR implements strict source routing. Routes are discovered to a destination by the source broadcasting a route request message. The message is received by neighbors who, if they know a route to the destination, send back a route response containing this route. Alternately, the neighbors forward the broadcast message after appending their address. This mechanism will eventually result in the request arriving at the destination (limited by a TTL). The destination then reverses the collected route in the packet and unicasts the route back to the source. Mechanisms such as ring discovery, route shortening etc. are built into the protocol as enhancements to make this process more efficient. One key benefit with DSR is the fact that a source can, in one request, potentially receive multiple route responses indicating different paths that can be used for routing data from source to destination. We exploit this inherent multipath capability in DSR for our application. Unlike other work that uses multipath for load balancing, we are using multipath as an

enabler for redundancy. The application that we are considering is a MPEG-4 stream from source to destination. The application has the ability to mark the frames as intra-coded frames (I frames), predictively coded frames (P frames) or bidirectionally predictively coded frames (B frames). I frames are coded independently of other frames using transform coding and provide an access point to the data. P frames use motion-compensated prediction based on a previous I or P frame. B frames are encoded based on a previous and future I or P frame. Due to this relationship, a P frame cannot be decoded at the receiver if its previous I frame is lost and likewise a B frame cannot be decoded at the receiver if the previous (or future) I or P frame is lost. By using the application provided marking of which frames are I,P and B, we use multipath routing for selective redundancy. Here I frames are duplicated and transmitted along multiple paths to the destination. The destination uses a buffering technique to identify and drop duplicate frames. The multipaths used can be unconstrained multipath, node diverse multipath and link diverse multipath.
We implemented our scheme using the *NS2* simulator. We use Evalvid[61] as the MPEG-4 quality evaluation toolkit. The steps that we followed for our simulation and evaluation are as follows;

- Take a YUV sequence in QCIF format and compress it into 30fps MPEG-4.
- "Transmit" the MPEG-4 over the wire. In this stage, we log the packets as if we are sending it over the wire. This gives us a tracefile of the exact times that packets are sent over the wire. This file is used as input to generate an NS2 trace file to use for our simulator.
- The source node in NS2 runs an application that transmits the packets (as per

the tracefile) over UDP. The packets are marked as carrying I, P or B frame data.

- The packets are routed over a MANET running our modified DSR. Our extension to DSR looks at the packet type and for packets carrying I frame data, alternate redundant copies are created and routed using an available alternate route to destination. If no alternate route exists, a new route request process is initiated.

- Packets arriving at the destination are presented to our receiver application that filters out duplicates and timestamps the received data packets.

- Using the original YUV sequence, sender trace file and received trace file, lost packets, delays and jitter can be computed and from this PSNR for the received video can be obtained. Evalvid offers the tools for this computation.

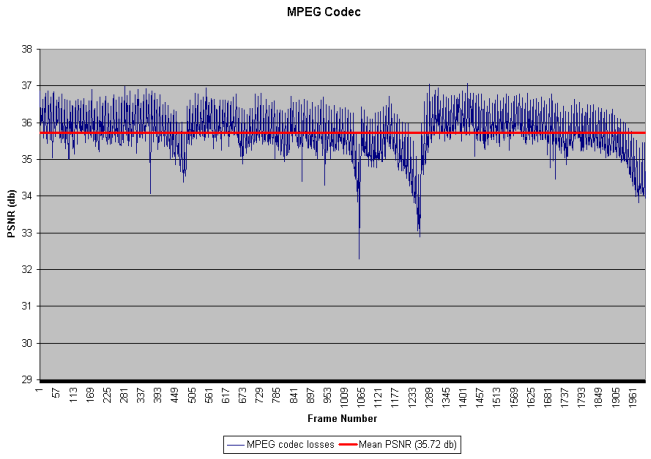

Figure 5.7: MPEG Codec Losses

Fig.5.7 shows the PSNR of the lossy encoding/decoding process for a 2000 frame

QCIF (Highway Drive). We take the output and run it through our NS2 simulator to come up with the PSNR under the varying scenarios.

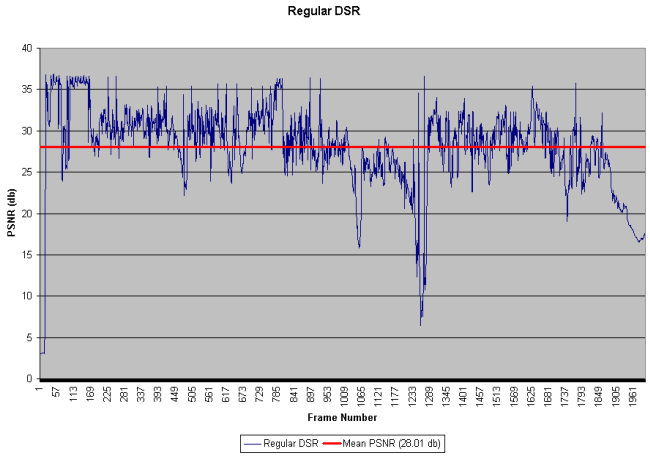

Figure 5.8: PSNR for Regular DSR

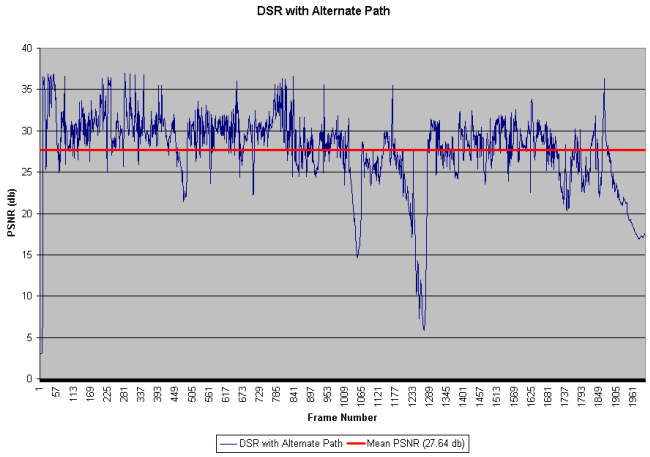

Figure 5.9: PSNR for Alternate Path

Fig.5.8-Fig.5.11 show the observed PSNR for the various schemes. Fig.5.12 shows the comparison for the different schemes. Fig.5.13 and Fig.5.14 show the corresponding frame and packet losses. As seem from these figures, through our modification, we are able to save about 5 times more I frames than regular DSR. These experiments

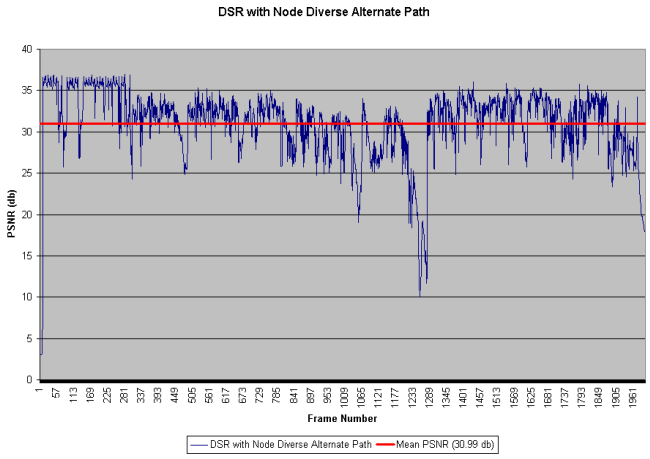

Figure 5.10: PSNR for Node Diverse Alternate Path

were done with an area of 670m x 670m. The number of nodes was set to 50 and speed was 5m/s. Maximum packet size was set to 1000 bytes.

Fig.5.15-Fig.5.19 show the performance of our scheme when the speed of movement is changed along with the number of nodes in the network. As expected our scheme out performs DSR as mobility rises since with greater mobility, higher the breakage on the primary route. With low number of nodes in the network, our scheme is not significantly better since there is a shortage of alternate routes. However, with more number of nodes, alternate routes are available and our scheme shows upto 2 db improvement over regular DSR.For these tests, we used an area of 1000m x 1000m with the source and destination separated by 600m. The 802.11 MAC transmission range is set to 250m.

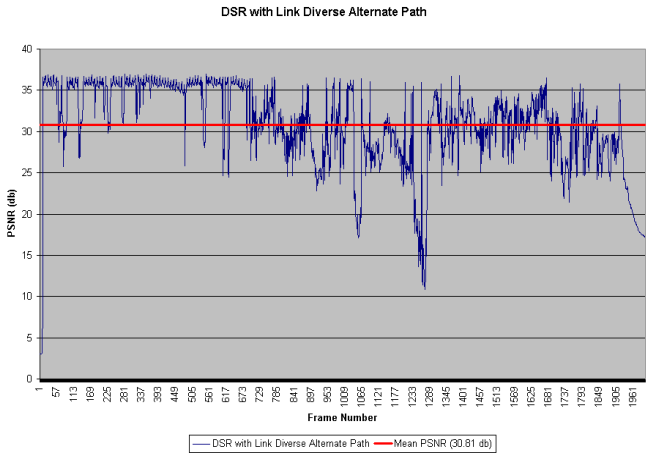

Figure 5.11: PSNR for Link Diverse Alternate Path

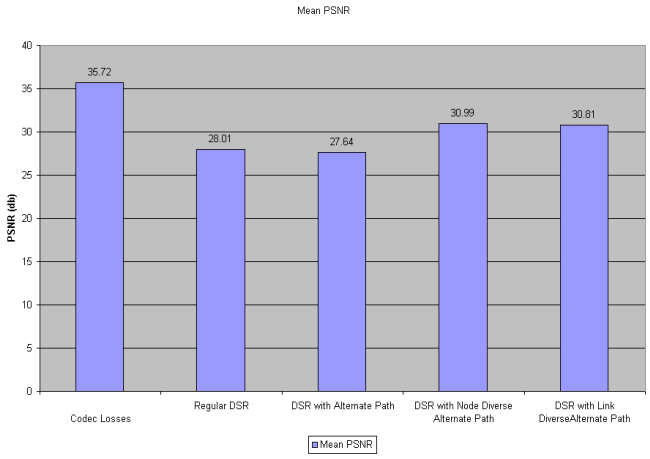

Figure 5.12: PSNR Comparison

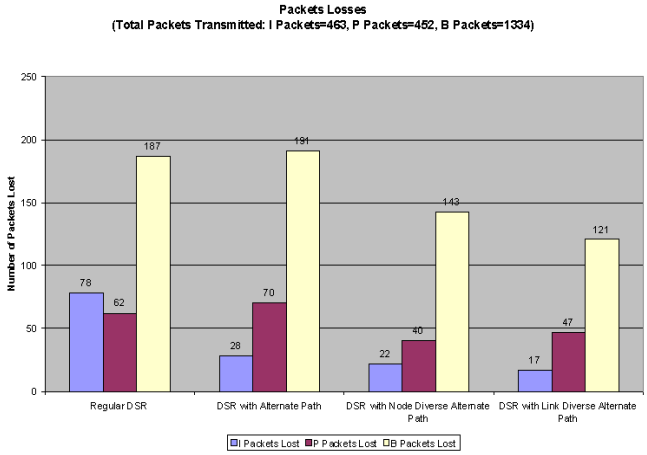

Figure 5.13: Packet Losses

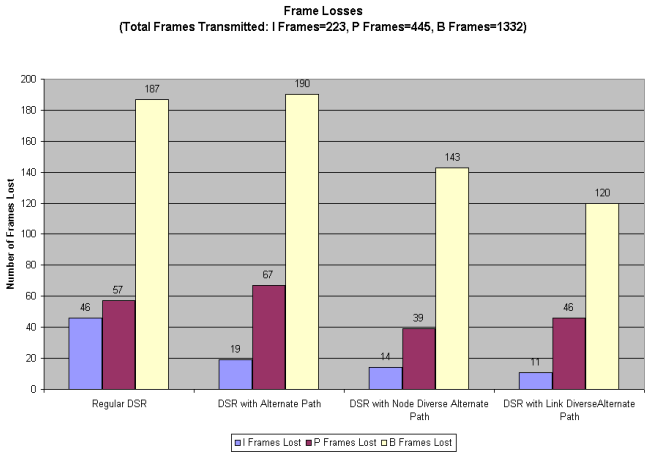

Figure 5.14: Frame Losses

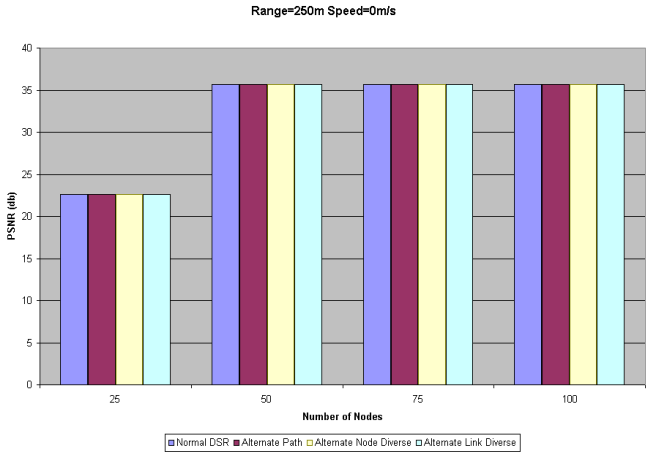

Figure 5.15: Speed 0 m/s

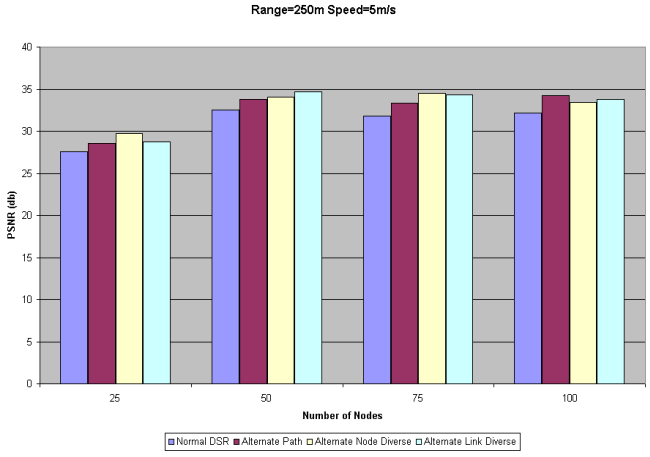

Figure 5.16: Speed 5m/s

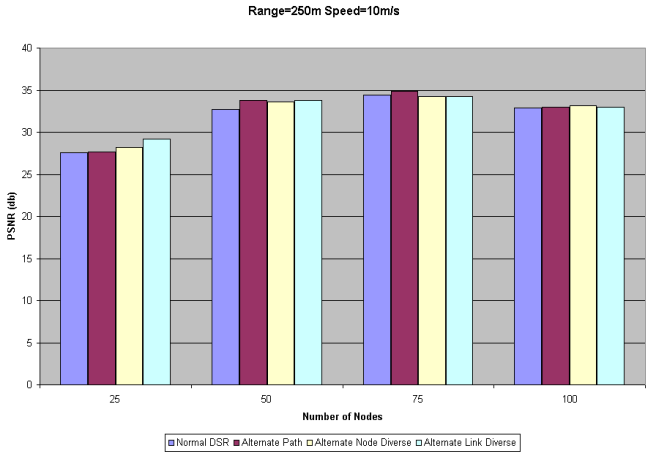

Figure 5.17: Speed 10m/s

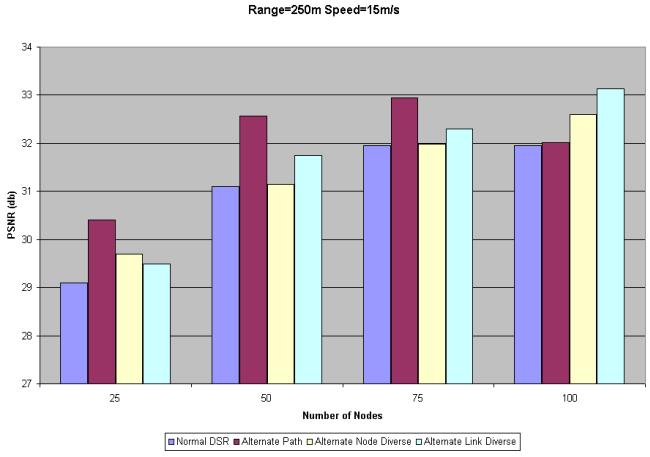

Figure 5.18: Speed 15m/s

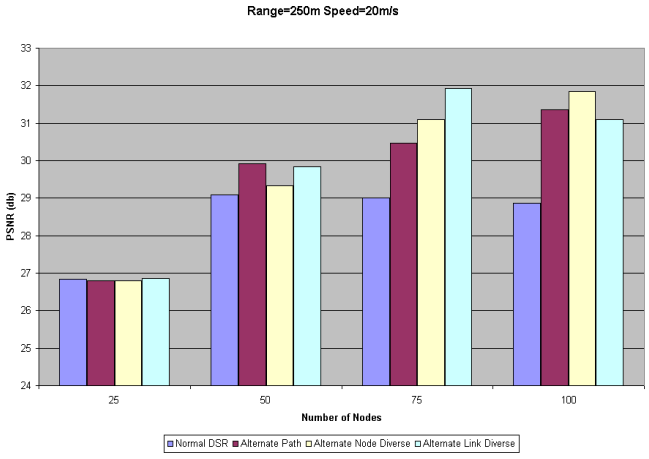

Figure 5.19: Speed 20m/s

Bibliography

[1] *6Bone Testbed for Deployment of IPv6 http://www.6bone.net.*

[2] *ABone: Active Network Backbone http://www.isi.edu/abone.*

[3] *Adamind http://www.adamind.com.*

[4] *Application-Oriented Networking (AON) http://www.cisco.com.*

[5] *eDonkey http://www.edonkey2000.com/.*

[6] *The gnutella protocol specification v0.4. clip2 distributed search services. available from http://www9.limewire.com/developer/gnutella protocol 0.4.pdf.*

[7] *IEEE 802.15 WPAN High Rate Alternative PHY Task Group 3a (TG3a) http://www.ieee802.org/15/pub/TG3a.html.*

[8] *JXTA v2.0 Protocols Specification http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.htn*

[9] *Kazaa http://www.kazaa.com.*

[10] *LightSurf http://www.lightsurf.com.*

[11] *Mobixell Networks http://www.mobixell.com.*

[12] *Morpheus http://www.morpheus.com.*

[13] *MPEG-21 Digital Item Declaration WD (v2.0) http://xml.coverpages.org/MPEG21-WG-11-N3971-200103.pdf.*

[14] *MPEG-4:ISO/IEC 14496 http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.*

[15] *MPEG-7:Multimedia Content Description Interface http://archive.dstc.edu.au/mpeg7-ddl/.*

[16] *Napster http://www.napster.com.*

[17] *Part 15.3: Wireless medium access control (mac) and physical layer (phy) specifications for high rate wireless perosnal area networks (wpan), ieee 802.15.3 working group, draft p802.15.3/d16, 2003.*

[18] *PLANETLAB: An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services http://www.planet-lab.org/.*

[19] *RDF Query Exchange Language (RDF-QEL) http://edutella.jxta.org/spec/qel.html.*

[20] *RDF Vocabulary Description Language 1.0: (RDF Schema) http://www.w3.org/TR/rdf-schema/.*

[21] *Resource Description Framework (RDF) http://www.w3.org/RDF/.*

[22] *SenseStream http://www.sensestream.com.*

[23] *TV-Anytime Forum http://www.tv-anytime.org/.*

[24] *VoiceAge Networks http://www.voiceagenetworks.com/.*

[25] *Volantis http://www.volantis.com/.*

[26] Toufik Ahmed, Ahmed Mehaoua, Raouf Boutaba, and Youssef Iraqi, *Adaptive Packet Video Streaming Over IP Networks: A Cross-Layer Approach*, vol. 23, Febuary 2005, pp. 385–401.

[27] D. Scott Alexander, William A. Arbaugh, Michael Hicks, Pankaj Kakkar, Angelos Keromytis, Jonathan T. Moore, Carl A. Gunter, Scott M. Nettles, and Jonathan M. Smith, *The SwitchWare active network architecture*, IEEE Network Magazine **12** (1998), no. 3, 29–36, Special issue on Active and Controllable Networks.

[28] Elan Amir, Steven McCanne, and Randy H. Katz, *An active service framework and its application to real-time multimedia transcoding.*, SIGCOMM, 1998, pp. 178–189.

[29] David G. Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris, *Resilient Overlay Networks*, 18th ACM SOSP (Banff, Canada), October 2001.

[30] Tom Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner, *Overcoming the internet impass through virtualization*, IEEE Computer Magazine, April 2005.

[31] S. Ardon, P. Gunningberg, B. LandFeldt, M. Portmann Y. Ismailov, and A. Seneviratne, *March: a distributed content adaptation architecture*, International Journal of Communication Systems, Special Issue: Wireless Access to the Global Internet: Mobile Radio Networks and Satellite Systems. **16** (2003), no. 1.

[32] R. Atkinson, *RFC 1825: Security Architecture for the Internet Protocol*, August 1995.

[33] A. Barbir, R. Penno, R. Chen, M. Hofmann, and H. Orman, *RFC 3835: An Architecture for Open Pluggable Edge Services (OPES)*, August 2004.

[34] Andre Beck and Markus Hofmann, *IRML: A Rule Specification Language for Intermediary Services:draft-beck-opes-irml-00.txt*, Febuary 2001.

[35] Harini Bharadvaj, Anupam Joshi, and Sansanee Auephanwiriyakul, *An Active Transcoding Proxy to Support Mobile Web Access*, 17th IEEE Symposium on Reliable Distributed Systems, October 1998.

[36] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen Witte Zegura, *On active networking and congestion*, Tech. Report GIT-CC-96-02.

[37] R. Braden, D. Clark, and S. Shenker, *RFC 1633: Integrated Services in the Internet Architecture: an Overview*, June 1994.

[38] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, *RFC 2205: Resource ReSerVation Protocol (RSVP) — version 1 functional specification*, September 1997.

[39] B. Branden, B. Lindell, S. Berson, and T. Faber, *The ASP EE: An Active Network Execution Environment*, DARPA Active Networks Conference and Exposition (DANCE 2002), June 2002, pp. 238–254.

[40] Prashant Chandra, Yang-Hua Chu, Allan Fisher, Jun Gao, Corey Kosak, T.S. Eugene Ng, Peter Steenkiste, Eduardo Takahashi, and Hui Zhang, *Darwin: Customizable resource management for value-added network services*, IEEE Network **15** (2001), no. 1.

[41] Surendar Chandra, Carla Schlatter Ellis, and Amin Vahdat, *Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding*, IEEE Journal on Selected Areas in Communications, vol. 18, December 2000.

[42] Kai Chen, Samarth H. Shah, and Klara Nahrstedt, *Cross-layer design for data accessibility in mobile ad hoc networks*, Wirel. Pers. Commun. **21** (2002), no. 1, 49–76.

[43] Lai-U Choi, Michel T. Ivrlac, Eckehard Steinbach, and Josef A. Nossek, *Bottom-up approach to cross-layer design for video transmission over wireless channels*, 62nd IEEE Vehicular Technology Conference (to be published), September 2005.

[44] Lai-U Choi, Wolfgang Kellerer, and Eckehard Steinbach, *Cross Layer Optimization for Wireless Multi-user Video Streaming*, IEEE International Conference on Image Processing (ICIP '04), October 2004.

[45] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, Lecture Notes in Computer Science **2009** (2001), 46+.

[46] Arturo Crespo and Hector Garcia-Molina, *Semantic overlay networks for p2p systems*.

[47] S. Deering and R. Hinden, *RFC 2460: Internet Protocol, Version 6 (IPv6) specification*, December 1998.

[48] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, *RFC 2748: The COPS (Common Open Policy Service) Protocol*, January 2000.

[49] D. Eastlake and P. Jones, *RFC 3174: US Secure Hash Algorithm 1 (SHA1)*, September 2001.

[50] J. Elson and A. Cerpa, *RFC 3507: Internet Content Adaptation Protocol (ICAP)*, April 2003.

[51] Armando Fox, Steven D. Gribble, Yatin Chawathe, and Eric A. Brewer, *Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives*, IEEE Personal Communications, September 1998.

[52] Mario Gerla, Ling-Jyh Chen, Tony Sun, and Guang Yang, *Ubiquitous video streaming: A system perspective*, Advances in Pervasive Computing and Networking, 2004.

[53] Steven D. Gribble, Matt Welsh, J. Robert von Behren, Eric A. Brewer, David E. Culler, N. Borisov, Steven E. Czerwinski, Ramakrishna Gummadi, Jon R. Hill, Anthony D. Joseph, Randy H. Katz, Z. M. Mao, S. Ross, and Ben Y. Zhao, *The ninja architecture for robust internet-scale systems and services*, Computer Networks **35** (2001), no. 4, 473–497.

[54] M. Handley and V. Jacobson, *RFC 2327: SDP: Session Description Protocol*, April 1998.

[55] J. Hartman, U. Manber, L. Peterson, and T. Proebsting, *Liquid software: A new paradigm for networked systems. technical report 96-11*, Tech. report, University of Arizona, 1996.

[56] John H. Hartman, Larry L. Peterson, Andy Bavier, Peter A. Bigot, Patrick Bridges, Brady Montz, Rob Piltz, Todd A. Proebsting, and Oliver Spatscheck, *Joust: A platform for liquid software - (tr97-16)*, Tech. report, University of Arizona, 1997.

[57] S . Herzog, J. Boyle, R. Cohen, D. Durham, R. Rajan, and A. Sastry, *RFC 2749: COPS usage for RSVP*, January 2000.

[58] Michael Hicks, Pankaj Kakkar, Jonathan T. Moore, Carl A. Gunter, and Scott Nettles, *Plan: a packet language for active networks*, ICFP '98: Proceedings of the third ACM SIGPLAN international conference on Functional programming (New York, NY, USA), ACM Press, 1998, pp. 86–93.

[59] David B. Johnson, David A. Maltz, and Yih-Chun Hu, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) draft-ietf-manet-dsr-10.txt*, July 2004.

[60] Vikas Kawadia and P.R. Kumar, *A Cautionary Perspective on Cross-layer Design*, Febuary 2005, pp. 3–11.

[61] Jirka Klaue, Berthold Rathke, and Adam Wolisz, *Evalvid - a framework for video transmission and quality evaluation.*, Computer Performance Evaluation / TOOLS, 2003, pp. 255–272.

[62] S. B. Kodeswaran, O. Ratsimor, A. Joshi, T. Finin, and Y. Yesha, *Using peer-to-peer data routing for infrastructure-based wireless networks*, IEEE International Conference on Pervasive Computing and Communications (PerCom 2003) (Fort Worth, TX, USA), March 2003.

[63] I. KOUVELAS, V. HARDMAN, and J. CROWCROFT, *Network adaptive continuous-media applications through self organised transcoding*, 1998.

[64] Dilip Krishnaswamy and John Vicente, *Scalable adaptive wireless networks for multimedia in the proactive enterprise*, j-INTEL-TECH-J **8** (2004), no. 4, 291–301.

[65] Alexander Loser, Kai Schubert, and Frederik Zimmer, *The semantic music store: Managing distributed semantic overlay networks.*

[66] Vijay K. Madisetti and Antonios D. Argyriou, *Transport Layer QoS Management for Wireless Multimedia Services.*

[67] Margaritis Margaritidis and George C. Polyzos, *Wireless Network Support for Adaptive Real-Time Applications*, Advanced Simulation Technologies Conference (ASTC'99), April 1999.

[68] R. Mohan, J.R. Smith, and Chung-Sheng Li, *Adapting multimedia Internet content for universal access*, IEEE Transactions on Multimedia, vol. 1, March 1999.

[69] Allen Brady Montz, David Mosberger, Sean W. O'Malley, Larry L. Peterson, Todd A. Proebsting, and John H. Hartman, *Scout: A communications-oriented operating system (abstract)*, Operating Systems Design and Implementation, 1994, p. 200.

[70] Wolfgang Nejdl, Martin Wolpers, Wolf Siberski, Christoph Schmitz, Mario Schlosser, Ingo Brunkhorst, and Alexander Löser, *Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks*, WWW '03: Proceedings of the 12th international conference on World Wide Web (New York, NY, USA), ACM Press, 2003, pp. 536–543.

[71] K. Nichols, S. Blake, F. Baker, and D. Black, *RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers*, December 1998.

[72] M. Nilsson, *ID3 tag version 2.3.0*, Febuary 1999.

[73] P.Bucciol and G. Davini and E. Masala and E. Filippi and J.C. De Martin, *Application-level Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless LAN's*, The Seventh International Symposium on WIRELESS PERSONAL MULTIMEDIA COMMUNICATIONS (WPMC 2004), September 2004.

[74] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker, *A scalable content addressable network*, Tech. Report TR-00-010, Berkeley, CA, 2000.

[75] O. Ratsimor, S. B. Kodeswaran, A. Joshi, T. Finin, and Y. Yesha, *Combining infrastructure and ad-hoc collaboration for data management in mobile wireless networks*, Workshop on Ad-hoc Communications and Collaboration in Ubiquitous Computing Environments (New Orleans, Louisiana, USA), November 2002.

[76] Antony Rowstron and Peter Druschel, *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), November 2001, pp. 329–350.

[77] S. Bhattacharjee, K. Calvert, Y. Chae, S. Merugu, M. Sanders, E. Zegura, *CANEs: An Execution Environment for Composable Services*, DARPA Active Networks Conference and Exposition (DANCE 2002), June 2002, pp. 255–273.

[78] Kevin Savetz, Neil Randall, and Yves Lepage, *MBONE: Multicasting Tomorrow's Internet http://www.savetz.com/mbone/.*

[79] ABeverly Schwartz, Alden W. Jackson, W. Timothy Strayer, Wenyi Zhou, Dennis Rockwell, and Craig Partridge, *Smart packets for active networks*, OpenArch '99, 1999.

[80] Jesse Steinberg and Joseph Pasquale, *A web middleware architecture for dynamic customization of content for wireless clients*, WWW '02: Proceedings of the 11th international conference on World Wide Web (New York, NY, USA), ACM Press, 2002, pp. 639–650.

[81] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, *RFC 2960: Stream Control Transmission Protocol (SCTP)*, October 2000.

[82] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, and Sonesh Surana, *Internet indirection infrastructure*, SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications (New York, NY, USA), ACM Press, 2002, pp. 73–86.

[83] Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, Frans F. Kaashoek, Frank Dabek, and Hari Balakrishnan, *Chord: A scalable peer-to-peer lookup protocol for internet applications*, IEEE/ACM Trans. Netw. **11** (2003), no. 1, 17–32.

[84] Lakshminarayanan Subramanian, Ion Stoica, Hari Balakrishnan, and Randy Katz, *OverQoS: An Overlay Based Architecture for Enhancing Internet QoS*,

1st Symposium on Networked Systems Design and Implementation (NSDI) (San Francisco, CA), March 2004.

[85] Siva Subramanian, Phil Wang, Ramesh Durairaj, Jennifer Rasimas, Franco Travostino, Tal Lavian, and Doan Hoang, *Practical Active Network Services within Content-Aware Gateways*, DARPA Active Networks Conference and Exposition (DANCE 2002), June 2002, pp. 344–355.

[86] Christian Tschudin and Richard Gold, *Selnet: A translating underlay network - tr2003-020*, Tech. report, Uppsala University, 2001.

[87] Francesco Vacirca, Andrea De Vendictis, and Andrea Baiocchi, *Investigating interactions between arq mechanisms and tcp over wireless links*, European Wireless (EW '04), Febuary 2004.

[88] D. Wetherall, J. Guttag, and D. Tennenhouse, *Ants: A toolkit for building and dynamically deploying network protocols*, 1998.

[89] Mark Yarvis, Peter L. Reiher, and Gerald J. Popek, *Conductor: A framework for distributed adaptation*, Workshop on Hot Topics in Operating Systems, 1999, pp. 44–.

[90] R. Yavatkar, D. Pendarakis, and R. Guerin, *RFC 2753: A Framework for Policy-based Admission Control*, January 2000.

[91] Y. Yemini and S. da Silva, *Towards Programmable Networks*, IFIP/IEEE International Workshop on Distributed Systems: Operations and Management '96, October 1996.

[92] J. Zander and R. Forchheimer, *Softnet – an approach to high level packet communication*, Second ARRL Amateur Radio Computer Networking Conference (AMRAD '83), March 1983.

[93] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz, *Tapestry: A global-scale overlay for rapid service deployment*, IEEE Journal on Selected Areas in Communications **22** (2004), no. 1, 41–53.

[94] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph., *Tapestry: An infrastructure for fault-tolerant wide-area location and routing - ucb/csd-01-1139*, Tech. report, U. C. Berkeley, March 2001.