Content and Context Aware Networking Using Semantic Tagging

Sethuram Balaji Kodeswaran, Anupam Joshi
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000, Hilltop Circle, Baltimore, MD 21250
Email: (kodeswar, joshi)@cs.umbc.edu

Abstract

Today's model of networking primarily concentrates intelligence at the end hosts with the network itself offering a simple"best-effort","data agnostic" communication medium. However, this paradigm has proven to be insufficient to meet todays needs considering the diversity of applications and devices that are networked. To offer value added services to these end users and applications, more and more intelligence needs to be migrated away from the edges and into the network in a controlled and tractable manner. In this paper, we present our approach of utilizing semantic data tagging to provide content level information for data streams flowing through a network. A policy based management mechanism is utilized within the network fabric allowing routers to reason over the content and make intelligent decisions regarding the handling of data packets. Service differentiation, in-network content adaptation, traffic monitoring and control etc. are some of the new services that can now be offered by the network in a generic and flexible manner. By deploying our proposed architecture, a network need no longer be viewed as a simple data transport medium but rather as a policy-controlled intelligent packet/stream processor that can offer specialized handling based on application needs.

1 Introduction

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. In this paper, we present our vision of a generic and flexible framework that enables the incremental deployment of intelligent services into the network with the aim of optimizing the end-user experience for networked applications. One of the key enablers of this approach is to provide network routers with visibility into the type of data that they are carrying. This content-level information can then be used by the routers to make more intelligent routing and data handling decisions. Unlike other approaches such as active networks, our framework relies on providing metadata for the streams, leaving the actual decision making to the routers, which are controlled through rules expressed as policies. A policy enforcer is built into the architecture to enforce rules which can either be local or system-wide. These rules are specified by the network provider to control traffic flows in the network. Recent advances in network processors point to more and more processing power available at the network routers, and in many cases, allowing for limited packet processing at line speed. Further, industry initiatives such as [1] further motivate in-network processing for specialized handling of data streams.

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 provider's individual labeling scheme. For this reason, we have chosen to use RDF[3] as the mechanism to convey this information. 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 (or OWL for schema definition) as the mechanism to markup flows/packets, intermediary intelligent routing entities can use this metadata to reason over their existing knowledgebase 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 provide specialized handling for MPEG-4 streams can choose to handle a particular packet as though it were part of a mul-

timedia 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.
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. The network stack is implemented as different layers, where each layer offers a well defined functionality, is well encapsulated and can be developed independent of the others. Well defined Service Access Points (SAPs) are specified to clearly define 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. Proponents of this model argue that with such interactions, bandwidth usage optimization, more efficient routing, better QoS guarantees, better power utilization etc. can be achieved. Our framework provides the means to exploit such cross layer techniques to allow "hints" provided by higher layers to be used lower down in the stack and likewise, applications running on top of the framework can also register to be notified of changes in the network conditions that are meaningful to them. A local policy enforcement engine controls the interactions that are permissible. Through this, system wide policies can be distributed throughout the network that are enforced locally and across the network to coordinate the interactions and data transport.
The rest of this paper is organized as follows. Section 2 presents some of the relevant background. Section 3 presents our proposed approach. Section 4 presents results of our simulation studies highlighting benefits of content awareness in the network followed by our conclusion.

2 Related Work

2.1 Active Networks

Active Networks is one of the revolutionary approaches 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 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 seem less and on-demand manner. Several projects have taken this approach to imparting intelligence into the network. The Smart Packets[29] 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 SwitchWare project[4] takes a similar approach utilizing a specialized language called PLAN whose capabilities are restricted to only performing "safe" operations on any node. ANTS[34] 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 capable of expressing mainly forwarding routines and mechanisms are built into the model for on-demand code dissemination. Other similar approaches [19] have focused on ensuring quick just-in-time compilation of downloaded code to optimize their performance along with static language verification. The Active Signaling Protocol (ASP)[13] takes a different approach to active networks by going away from the capsule model and having data packets carry a reference to some portable code which can be downloaded on the fly and run on an ASP Execution Environment as necessary to enable a new active application. Active Services[5] takes an alternate model to introduction of intelligence into the network by restricting the intelligence mainly to the application layer thereby preserving the routing and forwarding semantics of the Internet architecture. A similar application level processing of user-data is considered in [10] to handle congestion control in the network. The architecture allows for applications to specify intra-network processing by using Active Processing Function Identifiers (APFI) so that bandwidth allocated can be intelligently reduced in a manner tailored to the application rather than generically.

While active networks are a very powerful paradigm, there are several drawbacks that have hindered their acceptance and wide spread deployment. The primary concern is with security and protection against active routers running malicious code hidden in active packets. Approaches that limit the allowable operations to a safe subset limit the power of specialized handling that is possible. An equally important issue is that network operators prefer to have complete control over their network and allowing user defined code to be downloaded and executed that could potentially make routing decisions for data streams is not very amenable for acceptance. Many approaches utilized customized platforms and languages raising interoperability concerns. Also, the state of the technology in network processors at that time presented challenges for the true realization of this approach. With our framework, we are working towards the same goal of making networks smarter but doing so with a completely different approach than active networks. Our framework allows for the specification of "what type of data" is flowing through the network rather than specifying "how that data should be handled". Essentially, our framework provides the hints to the network fabric through content metadata but the actual decision of how best to handle

that type of stream is left to the router and the network operators. Also, the use of a standard mechanism to describe metadata provides a viable solution to the interoperability concerns. The policy mechanism used in our framework allows the network operator to specify handlers that can be triggered based on reasoning over the content metadata to invoke specialized handling, where the handlers are essentially value added services offered by the network provider themselves and thereby, trusted.

2.2 Content Adaptation

Content adaptation is a commonly used paradigm to address issues such as efficient transfer of bandwidth intensive data over a bandwidth limited network, making the content amenable to devices with diverse/limited capabilities, personalization, policy-based adaptations, security related adaptations etc. The Open Pluggable Edge Services (OPES)[7] is an IETF working group tasked with defining a common architecture for enabling edge based content adaptation. The idea is to specify a platform providing networked services at the application level for off loading origin servers and improving user experience. OPES is generally applicable for operating on request or response for data using HTTP/SMTP/RTSP with rules specified in Intermediary Rule Markup Language(IRML)[8] to determine what service should be used on what type of content. Content adaptation has also been studied extensively through proxy based systems. [14] 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. [16, 18] argues 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. Adaptation of content here is through data-type specific lossy compression and distillation. The Conductor[35] framework is a similar approach that can dynamically deploy multiple adapters to operate along an application's communication path. A planning algorithm is built-in to determine what adapters to use and where.
[26] presents a framework for adapting multimedia web documents to optimally match the needs of the requesting client devices using a multimodal, multiresolution representation hierarchy for the content and a Customizer that picks the best content representation that provides maximum value to the client. 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). [31] presents 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. 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 at close to line speed. [9] is a proxy based web content adaptation for supporting browsing by mobile devices over wireless links using user specified preferences. [32] uses a link level redirection infrastructure called SelNet that tags packets at link level with function identifiers to enable a proxy based content adaptation. [6] propose a server centric content adaptation framework creating service specific overlay networks through the use of dynamic proxies along the data path. [30] 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 primary business focus such as VoiceAge Networks, Volantis, Adamind, LightSurf, SenseStream, Mobixell etc.
One of the goals that we have identified for our framework is to support content adaptation for data streams. Our approach of using semantic tags embedded in the stream will allow content providers to "signal" their adaptation policies and requirements. As discussed in Section 3, this can be both in-band or out-of-band. An intelligent stream processor can reason over this information and using its existing knowledge base, the appropriate content adaptation services can be applied.

2.3 Cross Layer Techniques

Strict Layering has been the primary design technique for networking stacks which has resulted in simple, scalable and interoperable solutions. Proponents of the layered model such as [21] argue 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 counter argument has been that restricting the interactions between layers to data and well defined primitives imposes a rather uniform handling semantic for data streams which is not always the most efficient. Much of the work on cross layer optimizations has focused on optimizations at the network edges primarily for wireless technologies. In [17], a system that combines seamless handoff with adaptive video streaming is presented. Using an Eligible Rate Estimate, the quality of the video is dynamically adjusted to offer the user, uninterrupted video at the best quality that can be sustained. MobiWeb[25] is a framework for supporting adaptive ap-

plications 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. [33] presents 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 MAC layer retransmission according to a target loss rate used as parameter to describe the desired QoS for a TCP connection. [28] present an application level ARQ mechanism for 802.11 MAC. Here the application decides which packets require retransmission and just those are retransmitted. [24] presents a transport layer solution to enhancing network QoS using application level information. Here applications can split data into different streams which can be delivered across diverse routes so that more critical data is transmitted over a more reliable connection while the less critical data can follow a less reliable route. [23] presents an approach to support congestion controlled multicast real-time communication using self-organized transcoding to handle local repair in wired networks. The transcoding parameters are constantly updated to reflect the real time state of the link being used to serve the group to provide a network-friendly congestion control.

Our proposed framework provides the necessary support for implementing cross layer optimizations at the local node level. One of the primary concerns against cross layer techniques is that exposing too much information between layers makes the system overly complicated. To this end, our framework takes the approach of parameterizing the interactions similar to [15]. The goal is to expose meaningful parameters and their associated costs at each layer while keeping the actual mechanism of achieving them hidden within that layer. In our framework, the local policy enforcer can specify parameters to be used across the different layers of the stack to meet application needs. The objective is to keep the interactions tractable while still achieving the efficiencies of cross layer techniques. An additional and necessary goal is to coordinate interactions across a network since some techniques such as link level reliability need to be applied at a system level to really see the benefits. Our approach to cross layer optimizations goes one step further by allowing for the specification of system level optimizations that can span multiple nodes. System level information can be exchanged between nodes to inform each node of the dynamic system variations that can then drive the local optimizations that are applied.

2.4 Service Differentiation

Going beyond the "best effort" paradigm of the current Internet has been the goal of several initiatives. Integrated Services (IntServ)[11] proposes an architecture for 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)[12] is the signaling protocol used to convey a "flow-spec" and "filter spec" to intermediary routers describing the traffic characteristics and the resource reservations that need to be made. Closely coupled to IntServ is the policy admission control framework [36] allowing a network operator to specify policy based admission control rules that can be looked up and enforced at the routing elements. Two key components are the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). The PEP 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. One of the main drawbacks of IntServ is the maintenance of soft state representing per flow resource reservations on routers preventing it from scaling for large network sizes. Also, the filters are based on packet headers and have limited expressivity.

Differentiated Services (DiffServ)[27] 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. 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. There is also limited expressivity as the number of DSCPs is limited.

For our framework, we use a combination of approaches. We use a PEP/PDP like approach to allow for policies to be specified that are locally enforced and can specified at a system level by a network operator. Similar to the DiffServ approach, we rely on packet tagging to offer different levels of service. However, unlike the DSCP tags, we rely on custom tags that point to semantic metadata about the content. This metadata can be retrieved and reasoned over to determine what policies need to enforced for that stream. For example, a policy specified by an enterprise that drops all music MP3 streams flowing into their office network should still allow an official presentation containing MP3 audio to en-

ter the network. Approaches based on packet filtering and header checking cannot offer this level of flexibility since they lack inference capabilities. Likewise, applications running on top of our framework can specify their resource needs, with this input varying in detail from being highly specific to just communicating high level semantic information like the "flow of type JPEG with frame rate x and quality y". Our framework running on top of an Intserv or DiffServ network will be able to convert the application request to the corresponding realization mechanism such as signalling an RSVP reservation or mapping the data packets to the corresponding DSCP as the case may be.

2.5 Semantic Markup

Resource Description Framework (RDF)[3] 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 objects 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 Schema[2] is the mechanism for specifying vocabularies that can be used to form RDF statements. The vocabulary gives the actual meaning to the statement. Through this, objects and their properties and what they refer to can be specified. More powerful schema definition can be provided using OWL.
Providing markup for content is prevalent in many other areas albeit not always using a standard machine processable means such as RDF. Session Description Protocol (SDP) is a textual description for multimedia sessions. MPEG-7 and MPEG-21 are other multimedia markup standards. TV-Anytime is a specification for audio-video content markup. ID3 tags applied to MP3 files can convey metadata such as titles and composers. Many of these approaches use unstructured XML which is difficult to use for conveying knowledge. We have chosen to use RDF for our framework mainly to provide a standard and simple mechanism through which data streams can be described. The main objective is to keep our framework flexible to allow for the introduction of new concepts as needed rather than enforce a rigid schema to be followed by all participants.

3 Proposed Approach

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.

3.1 Node Level

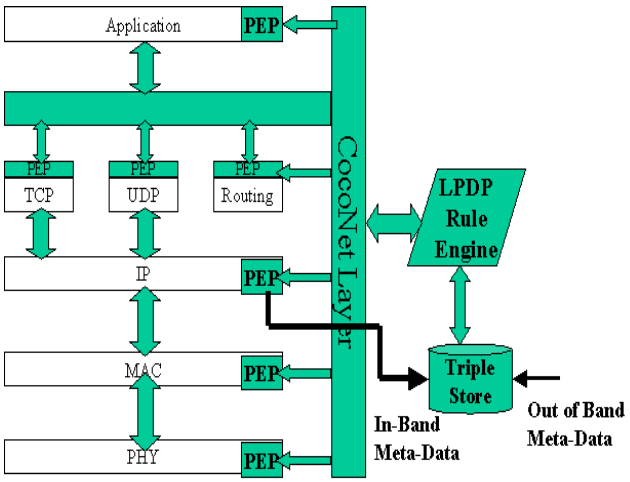

Figure 1. CoCoNet Node Framework

At the node level, the architecture we propose introduces our Node 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 level information for messages transmitted over this interface. A Local Policy Decision Point (LPDP) 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 [36] 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. The PEP exposes the interlayer optimization points that any particular layer supports. The 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 the mechanism of 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. The policies can be specified as production rules (*if (condition) then (action)*) or event-condition-action rules (*on event if (condition) then (action)*). In essence, the Node Framework provides a rich, extensible option for realizing policy controlled cross-layer interactions within a node's network stack. By parameterizing the possible set of interactions

that are permissible, the cross layer interactions are kept tractable without making the implementation overly convoluted [21].

3.2 Network Level

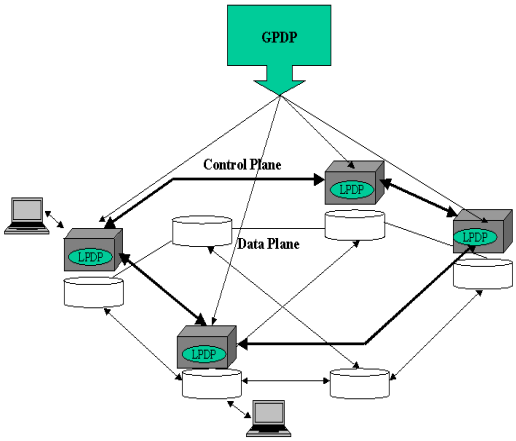

Figure 2. Overlay Network

At the network level, we envision that there will be an overlay network comprised of routers that run the CoCoNet Router Framework. 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 Router Layers at the routing elements.
- A data plane component through which the data packets are flowing.

Over the CoCoNet Router 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 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 data plane can be implemented as either:

- A UDP connection between two routers.
- A TCP connection between two routers.
- An IP-in-IP tunnel between two routers.
- A layer 2 LSP.
- A DiffServ aware network.
- An IntServ aware network.

A CoCoNet Router Framework will perform the necessary mapping based on policy, content and context. For instance, 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). Where a CoCoNet Router Framework runs is very implementation dependent. For example, in case of a wireless adhoc network, every host is a router and hence can potentially run a (albeit simplified) CoCoNet Router Framework. Likewise, in an enterprise setting, the host machines within the enterprise will likely run only the Coconet Node Framework with only the exterior gateway routers running the Coconet Router Framework. A network service provider will most likely have only edge routers run the Coconet Router Framework leaving the core optimized for fast data flow handling. The role of the Global Policy Distribution Point (GPDP) is to disseminate any network wide policies that need to be enforced. This can include 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”. Using a structured

Peer-to-Peer overlay to allow for key lookups will be needed for this option. 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.

The information conveyed in the metadata is really left up to the application. For example, an MP3 stream may have the following description which can be used by the example enterprise described in the earlier section to determine whether to drop or not (in this case, the stream is allowed to pass since it is part of official business).

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf=
 "http://www.w3.org/1999/
          02/22-rdf-syntax-ns#"
 xmlns:mmschema=
  "http://www.mySchema.org/mms#">
<rdf:Description
 rdf:about=
 "http://www.myContent.com/
  SalesReport.mp3">
 <rdf:type rdf:resource=
  "http://www.mySchema.org/mms#audio"/>
 <mms:LengthInMin>5</mms:LengthInMin>
 <mms:LengthInMB>4</<mms:LenghtInMB>
 <mms:technicalType>
  http://www.mySchema.org/mms#MP3
 </mms:technicalType>
 <mms:semanticType>
  http://www.mySchema.org/mms#Lecture
 </mms:semanticType>
</rdf:Description>
```

A similar framework is proposed in [22] specifying a metadata architecture for a policy based content delivery network. Here, the authors envision using a metadata gateway capable of extracting network delivery related metadata from content metadata. This delivery related metadata is used to implement network services such as QoS and content caching control. Our approach is different mainly in that we encode the content metadata onto the streams themselves (in-band or out-of-band). The intermediary routers (and not just the edge as in [22]) can then infer based on local knowledgebase and context, what policies are applicable and take local action. This allows for on-demand content adaptation based on transient conditions as opposed to end-to-end adaptations. Furthermore, providing content information so that a router can differentiate between, for example, video streaming from a surveillance camera and a streaming movie allows the network to make smart decisions on routing data streams across links with different reliability characteristics. Also, for our architecture, we are using RDF which provides a generic mechanism to convey metadata which can be reasoned over.

4 Preliminary Results

To investigate some of the potential benefits that content awareness within the network can bring, we simulated a cross layer approach where the routing layer proactively maintains alternate routes that can be used immediately upon failure of the primary route. We considered a wireless network running the DSR[20] routing protocol over which a 2000 frame MPEG-4 video is streamed. DSR is a reactive protocol ie. routes are discovered when needed by an application or data packet. Data is then transmitted through source routing and when the route breaks, the node upstream from the break signals a route error to the source and can optionally try to salvage the current packet through re-routing. Routes are discovered to a destination on demand by the source by broadcasting a route request message. 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. DSR then relies on source routing to transfer data. In our work, we exploit the inherent multipath capability of DSR by maintaining alternate routes. We assume that the application marks packets as Intracoded Frames (I Frames), Predictive-coded Frames (P Frames) or Bidirectionally-interpolated Frames (B Frames). The "marking" in our simulation is simple; the application interface to the transport layer has been extended to carry along with each segment of data, a globally known identifier representing the frame type. Our extension to DSR utilizes the "less important data" (B Frames in this case) as tracers to check for the viability of an alternate route. The rational here is that even if we lose this frame, the user is not going to see a big drop in the quality of the received signal since these are the most compressed frames. We used NS2 for our simulations and considered an area of 1000m x 1000m with varying number of nodes ranging from 25 to 100 with speeds between 0 and 20m/s using the 802.11 MAC. The two approaches compared were regular DSR and our content aware enhancement. For the regular DSR, the only modification made was to turn off flow state and disabled route requests from intermediary nodes for the purpose of salvaging. With our enhancement, the source node continues to use the primary path for bulk of the data transfer. Periodically, the source will pick a B Frame (the application specified marking is used to detect at the routing layer, which frames are B Frames) to be routed along alternate routes that were obtained through the initial (and snooped)

route discovery process. The aim here is to use a less important frame to validate alternate routes so that when the primary route fails, a cached alternate route can be used that has a high probability of being valid. Fig.[3,4,5,6] show

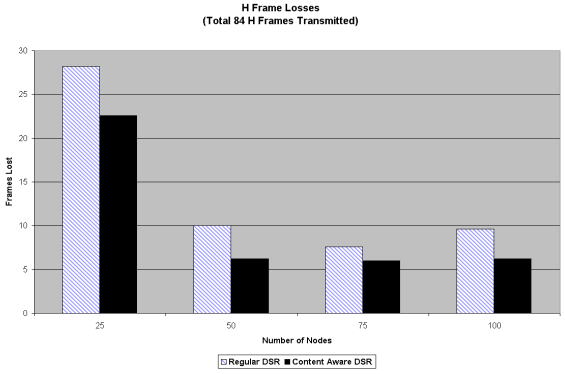

Figure 3. H Frame Losses

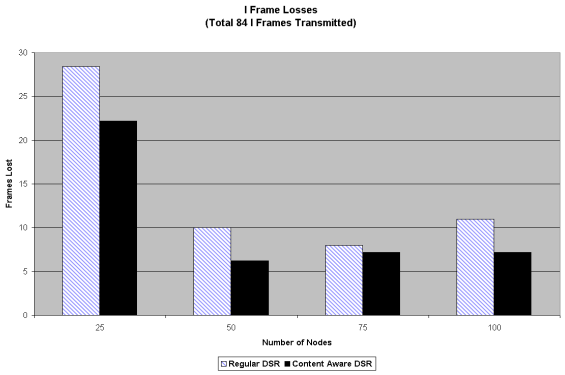

Figure 4. I Frame Losses

the improvements in the losses observed for H, I, P and B Frames when regular DSR is compared against our enhanced version. In this experiment, we modified the number of nodes in the network while keeping the speed of nodes fixed at 15m/s and 802.11 range to 250m. For each node count, the simulation was run 5 times with different random mobility scenarios and the results averaged. In all cases, our enhanced version out-performs regular DSR with very little overhead (salvaging if the alternate route chosen for validation is broken). When a route break occurs in regular DSR, a cached alternate route is chosen. In many cases, this route also ends up being broken but the source does not know this until it receives a route break error. If a valuable frame such as a I Frame had been transmitted over this broken route, there is less likelihood that the frame will arrive at the destination thereby preventing the GOP from being decoded. Using content information, our enhancement is able to identify low impact candidate frames to use to

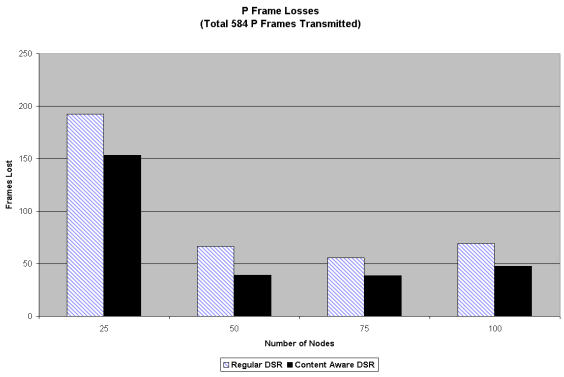

Figure 5. P Frame Losses

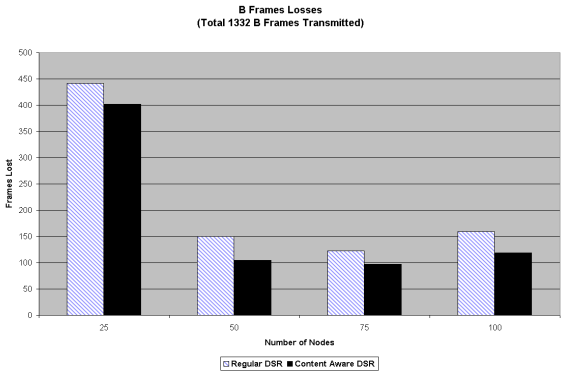

Figure 6. B Frame Losses

test alternate routes and remove them from cache if broken in a proactive manner. Fig.[7] shows the impact of our enhancement as the transmission range of 802.11 is increased keeping the number of nodes fixed at 100 and speed at 15m/s. At low ranges, our enhancement behaves similar to regular DSR mainly due to the lack of alternate routes. As the range increases, our enhancement starts to show improvements over the regular scheme. Fig.[8] shows the impact of speed of the mobile nodes. At low speeds, both schemes are comparable since route breakages are not too frequent. With increasing speeds, routes break more frequently with neighbors going out of range where our enhancement shows marked improvement over regular DSR.

5 Conclusion

In this paper, we present an architecture to enable intelligent processing of data streams within the network. Our architecture relies on semantic tags providing content metadata that can be reasoned over on the routing elements to enable specialized handling for flows. Our aim is to facilitate applications and users to communicate their preferences

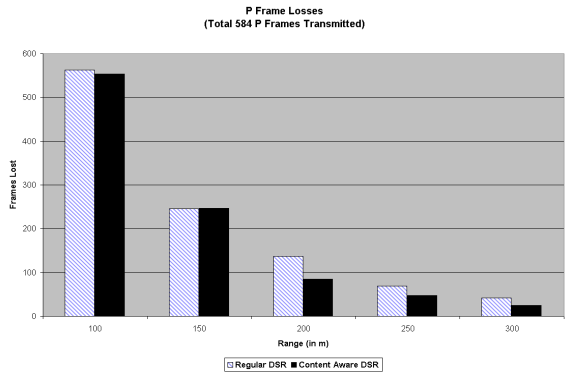

Figure 7. Frame Losses vs. Range

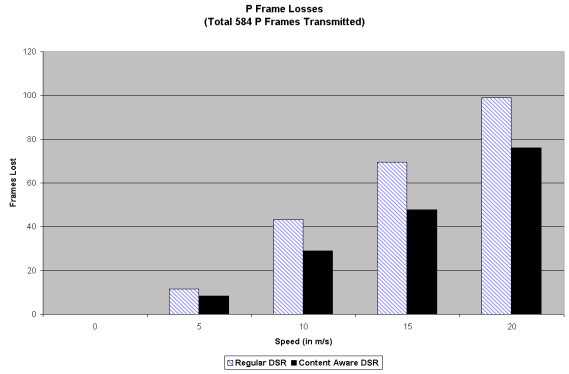

Figure 8. Frame Losses vs. Speed

to the network and let the network try to provide the needed level of service. Policies are used to drive adaptations both locally at a node and coordinated across a network. Controlled cross layer interactions are used to provide more efficient processing while still retaining tractability. We have also presented the results of our simulation studies showing that content awareness at the routing layer offers benefits over traditional content agnostic approaches to data distribution over a wireless network.

References

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

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

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

[4] D. S. Alexander, W. A. Arbaugh, M. Hicks, P. Kakkar, A. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith. The SwitchWare active network architecture. *IEEE Network Magazine*, 12(3):29–36, 1998. Special issue on Active and Controllable Networks.

[5] E. Amir, S. McCanne, and R. H. Katz. An active service framework and its application to real-time multimedia transcoding. In *SIGCOMM*, pages 178–189, 1998.

[6] S. Ardon, P. Gunningberg, B. LandFeldt, M. P. 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(1), 2003.

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

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

[9] H. Bharadvaj, A. Joshi, and S. Auephanwiriyakul. An Active Transcoding Proxy to Support Mobile Web Access. In *17th IEEE Symposium on Reliable Distributed Systems*, October 1998.

[10] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura. On active networking and congestion. Technical Report GIT-CC-96-02.

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

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

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

[14] S. Chandra, C. S. Ellis, and A. Vahdat. Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding. In *IEEE Journal on Selected Areas in Communications*, volume 18, December 2000.

[15] L.-U. Choi, M. T. Ivrlac, E. Steinbach, and J. A. Nossek. Bottom-up approach to cross-layer design for video transmission over wireless channels. In *62nd IEEE Vehicular Technology Conference (to be published)*, September 2005.

[16] A. Fox, S. D. Gribble, Y. Chawathe, and E. A. Brewer. Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives. In *IEEE Personal Communications*, September 1998.

[17] M. Gerla, L.-J. Chen, T. Sun, and G. Yang. Ubiquitous video streaming: A system perspective. In *Advances in Pervasive Computing and Networking*, 2004.

[18] S. D. Gribble, M. Welsh, J. R. von Behren, E. A. Brewer, D. E. Culler, N. Borisov, S. E. Czerwinski, R. Gummadi, J. R. Hill, A. D. Joseph, R. H. Katz, Z. M. Mao, S. Ross, and B. Y. Zhao. The ninja architecture for robust internet-scale systems and services. *Computer Networks*, 35(4):473–497, 2001.

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

[20] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) draft-ietf-manet-dsr-10.txt, July 2004.

[21] V. Kawadia and P. Kumar. A Cautionary Perspective on Cross-layer Design, Febuary 2005.

[22] M. Kawarasaki and R. S. Atarashi. Policy based content delivery management using metadata. In *SAINT Workshops*, pages 272–279, 2004.

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

[24] V. K. Madisetti and A. D. Argyriou. Transport Layer QoS Management for Wireless Multimedia Services.

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

[26] R. Mohan, J. Smith, and C.-S. Li. Adapting multimedia Internet content for universal access. In *IEEE Transactions on Multimedia*, volume 1, March 1999.

[27] 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, Dec. 1998.

[28] 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. In *The Seventh International Symposium on WIRELESS PERSONAL MULTIMEDIA COMMUNICATIONS (WPMC 2004)*, September 2004.

[29] A. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, D. Rockwell, and C. Partridge. Smart packets for active networks. In *OpenArch '99*, 1999.

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

[31] S. Subramanian, P. Wang, R. Durairaj, J. Rasimas, F. Travostino, T. Lavian, and D. Hoang. Practical Active Network Services within Content-Aware Gateways. In *DARPA Active Networks Conference and Exposition (DANCE 2002)*, pages 344–355, June 2002.

[32] C. Tschudin and R. Gold. Selnet: A translating underlay network - tr2003-020. Technical report, Uppsala University, 2001.

[33] F. Vacirca, A. D. Vendictis, and A. Baiocchi. Investigating interactions between arq mechanisms and tcp over wireless links. In *European Wireless (EW '04)*, Febuary 2004.

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

[35] M. Yarvis, P. L. Reiher, and G. J. Popek. Conductor: A framework for distributed adaptation. In *Workshop on Hot Topics in Operating Systems*, pages 44–, 1999.

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