AdaptNet: An Adaptive Protocol Suite for the Next-Generation Wireless Internet

Ian Akyildiz, Yucel Altunbasak, Faramarz Fekri, and Raghupathy Sivakumar
Georgia Institute of Technology

This work was supported in part by an NSF grant ANI-0117840.

ABSTRACT

Over the last decade, the tremendous growth in the mobile Internet user population has been accompanied by an equally exciting evolution in wireless data networks. However, quite understandably, the evolution has been distinctly characterized by an increasing degree of heterogeneity along several dimensions such as the access technology, network model, device, and application requirements. This heterogeneity, in turn, imposes a significant challenge on the design of the network protocol stack, and leads to the question: *how can the protocol stack at a mobile host cater effectively to the heterogeneous characteristics of the operating environment?* In this article we provide an overview of AdaptNet, an adaptive protocol suite for next-generation wireless data networks. AdaptNet consists of protocol solutions at different layers of the protocol stack addressing several problems, including rate adaptation, congestion control, mobility support, and coding. A common underlying theme in the design of the protocols in the AdaptNet suite is adaptiveness to the operating environment. Through high-level discussions, preliminary results, and pointers to relevant related work, we show how AdaptNet achieves the goal of effectively addressing heterogeneity in next-generation wireless data networks.

INTRODUCTION

Next-/fourth-generation (NG/4G) wireless systems, currently in the design phase and scheduled to be deployed by the end of this decade, are expected to support considerably high data rates, and will be based on IP technology, making them an integral part of the Internet infrastructure. Users are expected to be able to receive the same services over NG systems as they do over wireline networks, including bandwidth-demanding applications like interactive multimedia, voice over Internet, games, and videoconferencing. Furthermore, such services are to be provided ubiquitously over a diverse set of environments including indoor home and office, outdoor pedestrian and vehicular areas, and global satellite regions. Heterogeneous environments will exhibit different data rates and handoff frequencies, with smaller cells providing significantly higher data rates than larger cells, albeit at the cost of higher handoff frequencies for the same mobility rates. It is envisioned that users will be seamlessly assigned and reassigned to the appropriate level in the cell hierarchy based on their physical locations and mobility profiles.

Thus, the NG wireless Internet (NGWI) can be expected to exhibit two defining characteristics:

- Heterogeneity in the physical network environments and architectures used
- A significant change in the nature of applications supported, from traditional low-data-rate applications to real-time and high-speed multimedia applications

In this article we argue that the heterogeneity along two dimensions of network environments and the nature of applications warrant a comprehensive rethinking of the design of the network protocol stack at the mobile stations. In particular, we make the case for adaptiveness in layers 2 (link), 4 (transport), and 5 (application), and present an adaptive protocol suite called *AdaptNet* that is adaptive to both the underlying network environment and the applications that run atop the protocol stack. We do not focus on the network layer as our goal is not to require any changes to the IP substrate of the Internet, and achieve adaptiveness in a scalable fashion. Toward this end, our solution is predicated on requiring changes only at the mobile host. The uniqueness of AdaptNet lies not only in the adaptivity of the protocols, but also in their cross-layer interactions. The specific protocol contributions of this work are summarized below.

Application: At the application layer, we specifically focus on the exciting area of real-time video streaming, and propose source and channel-adaptive coding to handle data and bit error rate fluctuations of the wireless channel.

Transport: At the transport layer, we present an adaptive mobile-host-centric transport layer

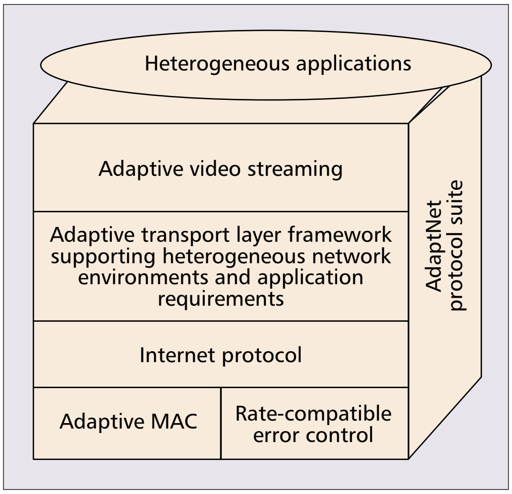


Figure 1. *AdaptNet protocol suite.*

framework called *Radial Reception Control Protocol* (R^2CP). While the goals of R^2CP are to provide an adaptive solution to the problems of heterogeneity, we also show how R^2CP provides a new dimension of functionalities that are bound to be required in the NGWI. Furthermore, we present an adaptive congestion control algorithm as part of the R^2CP framework that adapts to the specific operating environment.

Link layer: At the link layer, we present an adaptive medium access control (A-MAC) framework to perform seamless medium access control over heterogeneous networks without requiring any additional modifications in existing network infrastructures.

Data link: We also present, at the link layer, we present an adaptive error correcting system that functions with only one encoder and decoder at the sender and receiver respectively, but still can change the coding rate based on the channel conditions to maintain acceptable quality of service (QoS).

The rest of the article is organized as follows. We first present an overview of the *AdaptNet* protocol suite. Then, the specific protocols at the different layers of the protocol suite are summarized in their respective sections.. Finally, we give an overview of some related work and conclude the article.

THE ADAPTNET PROTOCOL SUITE

The development of the AdaptNet protocol suite is an ongoing effort at Georgia Institute of Technology. The goal of the project is to develop an adaptive protocol suite that can handle the vagaries of the wireless channel and the heterogeneity of technologies experienced by mobile hosts when moving from one wireless network to another. In this section we provide an outline of the AdaptNet protocol suite (Fig. 1) and highlight the interactions between the different layers of the protocol stack.

While the network protocol stack is made up of several layers, the goal of the AdaptNet project is to develop adaptive protocols at the application, transport, and link layers. The network layer, consisting of the Internet Protocol (IP), is explicitly left untouched for reasons of deployability, and leaving the network routing infrastructure as is. Several related works have focused on adaptive physical layer technologies [1], but such approaches at the physical layer are beyond the scope of this article.

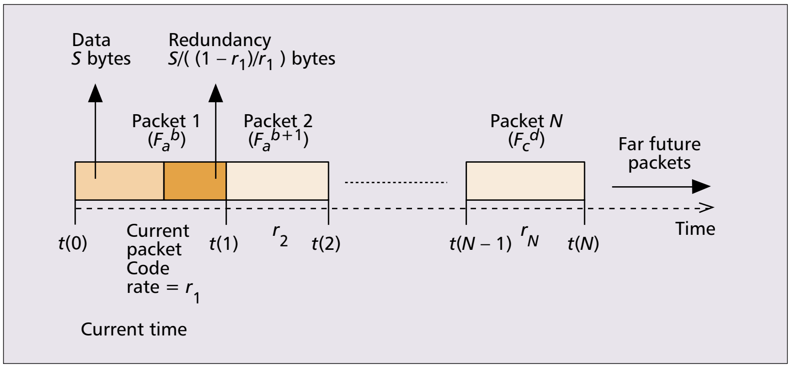
At the core of the AdaptNet protocol suite is an adaptive transport layer framework, R^2CP [2]. R^2CP is a multistate mobile-host-centric transport protocol that explicitly handles the issues of the multihomed nature of mobile hosts and heterogeneity through appropriate mechanisms. One of the key functionalities of R^2CP on which we elaborate here is adaptive congestion control. Essentially, R^2CP uses a single congestion control algorithm that can adapt to a variety of network conditions. R^2CP's congestion control is appropriately applicable to both bulk transfer and real-time applications, and changes its behavior based on the nature of the application. Finally, the congestion control module provides critical input to the adaptive video streaming application we discuss next.

While the application layer is traditionally looked upon as lying beyond the scope of the network protocol stack, the AdaptNet protocol suite explicitly addresses one class of applications we believe deserves attention due to both its popularity and resource-intensive nature. Specifically, the AdaptNet protocol suite consists of a source and channel-adaptive coding algorithm that derives input from the R^2CP transport layer protocol on the available bandwidth and loss rates, and translates the source stream to maximize the perceived video quality. The algorithm relies on scalable adaptive coding capabilities at the lower layers of the protocol stack, and the adaptive link layer of the AdaptNet suite we describe next has such capabilities.

The final component of the AdaptNet protocol suite we discuss in this article is the adaptive link layer protocol that consists of adaptive schemes for medium access control and coding. Adaptive coding refers to the ability of the link layer to change the coding performed depending on the nature of the wireless channel. While adaptive coding is by itself desirable to maintain consistent QoS, the proposed protocol requires just a single encoder and decoder at the sender and receiver, respectively, and thus is a scalable approach to supporting adaptive coding. The adaptive link layer provides an important tool for the adaptive application layer to enable channel-aware coding.

Heterogeneous wireless architectures impose challenges to the MAC layer in terms of different access schemes and resource allocation techniques, as well as diverse QoS requirements. The AdaptNet protocol suite achieves adaptivity to the architectural heterogeneity as well as diverse QoS requirements by deploying a new adaptive MAC framework in the mobile hosts. The AdaptNet suite consists of an adaptive MAC layer that handles the heterogeneity in *access schemes*, *resource allocation*, and *QoS requirements* due to the variable topology of NG wireless networks, the access techniques used by each scheme, and various QoS requirements of applications, respectively.

The AdaptNet protocol suite achieves adaptivity to the architectural heterogeneity as well as diverse QoS requirements by deploying a new adaptive MAC framework in the mobile hosts.



■ **Figure 2.** *The scheduled transmission times for the subsequent* N *packets in the send buffer.* F_a^b *denotes the bth packet of picture frame* F_a. r_i *is the code rate applied to the* i*th packet. Stronger FEC codes applied to the first packet and/or smaller quantizer values used in the source coder increase the expected quality (of the first packet) at the receiver, but shift the scheduled transmission times for the subsequent packets, which decreases their expected qualities. This trade-off must be managed so as to maximize the overall video quality rather than the individual packet qualities.*

In the rest of this article we present the individual layers in more detail. Due to lack of space, the discussions and arguments are maintained at an overview level, and appropriate references are provided for more interested readers. Nevertheless, the goals of the discussions are to both motivate the need for adaptiveness at the different layers considered and provide insights into how the AdaptNet protocol suite achieves the desired adaptiveness.

An Adaptive Application Layer

Mobile video is expected to be an important application class for NG wireless systems. However, time- and location-dependent rate and loss characteristics of the wireless links pose serious challenges for both conversational and streaming video applications [3]. Also, the heterogeneous structure of NG systems exhibits different channel data rates for mobile clients. Thus, terminals should adaptively adjust the video bit rate in order to achieve the best presentation quality. The source encoder can change the video rate in conversational applications. However, in streaming applications where the video is often pre-encoded and stored, real-time adaptive source encoding is not applicable. Transrating, transcoding, and scalable video coding are among the proposed solutions in this scenario.

The noisy and multipath nature of the radio link causes frequent packet losses in wireless systems. The resulting degradations on a picture frame may propagate to succeeding frames because of the variable length coding and the motion compensation used in the standardized video codecs [3]. During the last decade, several solutions have been offered to provide resiliency to such packet losses at the application layer. These are effective for communication scenarios where the application cannot modify the error control mechanisms deployed in the underlying system. For instance, spatial and temporal error resilience can be accomplished by the use of slice structured coding and intra-picture refresh, respectively. At the client side, a decoder may also perform error concealment in order to predict the missing parts from those that remain intact [4]. Use of multiple description coding (MDC) with diversity techniques is shown to be another effective method to achieve application layer error resiliency.

The effect of losses on presentation quality can further be mitigated by an integrated approach where, in addition to the application-layer error resiliency techniques, the lower layers of the protocol stack deploy source-aware error control methods. Thereby critical packets, those that cause more distortion when lost, can have stronger error protection, and stringent delay constraints are taken into account in packet scheduling and resource allocation to guarantee on-time packet delivery [5].

Our work differs from previous studies in this area in:
- Its joint optimization of source and channel-code rates
- Its consideration of residual network resources for the subsequent packets

That is, we incorporate the effects of source and channel-code rate selection on the amount of channel resources consumed and the overall distortion. The most closely related work in terms of modeling the time-varying wireless channel and forward error correction (FEC) rate adaptation is proposed by Elaoud and Parameswaran [6]. In this study, transmission decisions are made considering the packet deadline and air interface status. However, they do not incorporate the packet dependencies and the effect of future transmissions in the formulation. The rest of the section discusses an adaptive and error-resilient wireless video streaming technique proposed by the authors [7]. In this technique, variable data and bit error rate (BER) characteristics of the channel are handled with the use of source and channel-adaptive coding. That is, our objective is to optimize the FEC code rate (for each packet) and transcoding parameters (for each frame) so as to maximize the expected video quality at the client. Due to the limited bandwidth and the delay requirements, the amount of channel resources spent for the transmission of a packet affects the residual resources for subsequent packets, as illustrated in Fig. 2. A sender may prefer preserving network resources for more important subsequent packets and for packets that may face noisier channel conditions. Thus, we argue that the sender should optimize the FEC code rate and transcoding parameters considering both current and *subsequent* packets. By doing so, we achieve efficient allocation of channel resources that maximizes overall quality rather than individual packet quality.

The transport layer of AdaptNet dynamically monitors the channel characteristics. The application layer keeps track of the channel BER via the information gathered from the transport layer. Utilizing finite state Markov chains (FSMCs) that characterize the channel, it then estimates the expected channel quality at a future time instant. We also model the error propagation phenomena and introduce a distortion measure for packet losses to determine the importance of each video packet. Given knowledge of the channel characteristics, the packet distortion measure, and the deadline of each packet, the sender then makes the transcoder parameter and channel code rate decisions such that on-time delivery probability of the packets is maximized. Source rate may be reduced as a result of the optimization in order to use more channel coding bits at higher BERs. Transcoding also provides adaptivity to long-term bandwidth variations due to the heterogeneity of environments in the NG systems.

The application layer requires cross-layer coordination to enable the proposed optimization. The transport layer provides the channel quality information used to estimate future BERs. QoS guarantees provided by the MAC layer can be utilized to characterize the available bandwidth and delay. Layer coordination also enables the application layer to dictate the selected channel code rate to the data link layer. The FEC codes at the selected rate are generated using the rate-adaptive low density parity check (LDPC) codes.

Figure 3 depicts the H.264-compressed video streaming simulation results, where we compare different source and channel adaptation methods. In the experiments the raw capacity of the channel (before channel coding) is set to 100 kb/s. FEC code rate adaption provides a quality improvement around 1.5 dB over the fixed FEC code rate at video rates below 75 kb/s. If the video bit rate exceeds 75 kb/s, both methods cause quality degradation due to insufficient channel bandwidth. This problem is solved by incorporating transcoding in the optimization process, and good video quality is maintained at the higher bit rates. Thus, even if the bandwidth provided to a mobile client fluctuates and/or the channel error characteristics vary, the user will still be able to receive an acceptable quality video.

Our future plans include:
- Further reducing the computational complexity through heuristics, which will be derived based on the rigorous solution
- Developing more accurate source rate distortion models (to be used in source adaptation) for H.264
- Using a hybrid combination of automatic repeat request (ARQ) and FEC where a limited number of retransmissions are allowed

AN ADAPTIVE TRANSPORT LAYER

In this section we present a new multistate transport protocol, Radical Reception Control Protocol (R^{2}CP) for NG heterogeneous wireless data networks. R^{2}CP is specifically designed for mobile hosts with multiple heterogeneous interfaces. For such a host, a transport protocol should be able to handle heterogeneity in the operating environment, even during the course of a single connection. Furthermore, if the mobile host so chooses, the transport protocol should be able to use multiple interfaces simultaneously for a single connection.

Related work in the area of wireless transport layer protocols can be classified as belonging to one of three types of protocols:

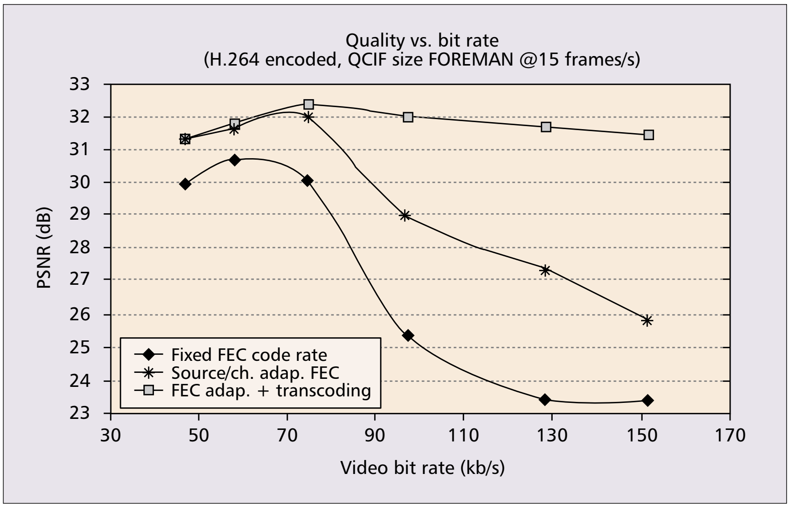


Figure 3. *An H.264 encoded FOREMAN sequence; channel capacity is set to 100 kb/s.*

- Protocols with mechanisms adapted to the wireless channel peculiarities
- Protocols that allow for multiple interfaces to be used simultaneously
- Protocols that handle mobility in a purely end-to-end fashion

R^{2}CP comprehensively addresses problems handled by the above classes of approaches, and further supports some key functionalities necessary in heterogeneous wireless environments that protocols in related work do not address.

In the rest of this section we discuss the design motivation, functional overview, and high-level protocol details of R^{2}CP.

Mobile-host-centric operation: There are several advantages to be gained in a wireless environment if the transport protocol were mobile-host-driven. In order to use network-specific congestion control schemes depending on the wireless interface the mobile host uses, without overloading the server with a plethora of congestion control mechanisms, the congestion control of the connection needs to be performed at the mobile host. In addition, since the mobile host may change the communication peer during a connection (server migration), or the number of servers it connects to (depending on the number of active interfaces) during periods of mobility, it is advantageous if the mobile host controls the reliability mechanism (which data to request from the sender). In this context, R^{2}CP is designed to operate either atop TCP or atop the Reception Control Protocol (RCP) (which is a receiver-centric clone of TCP),[1] and functions in a purely mobile-host-centric fashion.

Maintaining multiple states: An important issue in achieving seamless handoffs in a reliable connection such as TCP is to minimize the impact of handoff latency (especially for vertical handoffs), and handle packet reordering and losses during handoffs. R^{2}CP, hence, is built as a multistate extension of TCP/RCP. R^{2}CP dynamically maintains multiple states by creating and deleting RCP states depending on the number of active interfaces in use during handoffs. An RCP state created for each active interface thus only concerns the connection state of the end-to-end

[1] In the rest of the article, we use the name RCP to represent a mobile-host-centric single state transport protocol.

While using multiple congestion control protocols is an option, a more elegant and cost-effective solution is to use an adaptive congestion control approach, wherein the congestion control algorithm adapts to the operating environment. R²CP uses such an algorithm in its operation.

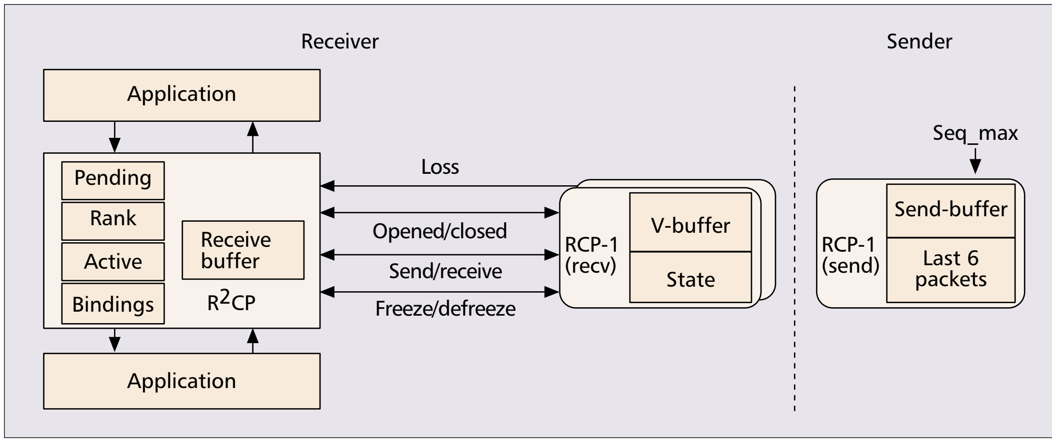


Figure 4. *R²CP architecture.*

path terminated at each interface. By virtue of its mobile-host-centric design, R²CP distinguishes itself from related approaches [8, 9] in its ability to communicate with one or multiple senders running the single-state RCP protocol. No change is necessary at the RCP sender to support R²CP at the receiver.

Decoupling of functionalities: Note that an R²CP connection with *k* active interfaces consists of *k* RCP states. R²CP minimizes overheads by decoupling the transport layer functionalities associated with the per-path characteristics from those that pertain to the aggregation connection. The congestion control mechanism is a per-path functionality, and is handled only by individual RCP states. On the other hand, reliability, as well as socket buffer management, pertains to the aggregate connection (as far as the application is concerned), and is handled by R²CP itself. Therefore, R²CP controls what data to request from each sender, and individual RCP states control how much data it can request along its path.

Adaptive congestion control: While R²CP delegates the task of congestion control to the individual RCP states, congestion control still needs to be performed in an interface-specific fashion. While using multiple congestion control protocols is an option, a more elegant and cost-effective solution is to use an adaptive congestion control approach, wherein the congestion control algorithm adapts to the operating environment. R²CP uses such an algorithm in its operation. Briefly, default TCP's congestion window increase and decrease parameters (α and β) are fixed at constant values of 1 and 0.5, respectively. In heterogeneous wireless environments, where loss rates and delay can fluctuate over a wide range of values across different network types, the use of such constant adaptation values makes the congestion control algorithm vulnerable to the vagaries of the network environment. *R²CP* uses an *adaptive congestion control* (ACL) algorithm that dynamically monitors the wireless random loss rate and delay, and adjusts its congestion control adaptation parameters in a manner that offsets the loss rate and delay components introduced by the wireless link [10]. Recall that the application layer described earlier relies on feedback from the congestion control algorithm for its operation.

Multiplexing and scheduling: When multiple RCP states coexist in a connection and collectively move data from one or multiple senders to the receiver, a challenging issue at the receiver is how to schedule the transmissions of different states and achieve maximum effect of bandwidth aggregation. Specifically, different paths have different characteristics in terms of bandwidth and delay; given that they share the same receive buffer, it is important that the slower paths do not stall the progress of the faster paths. Individual RCP states request R²CP for transmission (to request data from the sender) based on the progression of their congestion window, and R²CP schedules transmission based on the round-trip time of each path. As we mention above, R²CP maintains the binding information for all pending segments requested through individual RCP states. Any losses detected by individual RCP states (through arrival of 3 out-of-order segments or a timeout) are reported to R²CP such that the corresponding data is immediately *unbound* from the concerned RCP state. Unbound data will be scheduled by R²CP for transmission subsequently. Hence, head-of-line blocking due to segment losses, and bandwidth or delay mismatches of individual pipes are minimized.

An architectural overview of R²CP, its key data structures, and its cross-layer interaction between the adaptive application layer and lower networking layers are illustrated in Fig. 4. R²CP is a transport protocol that interacts with the application and IP at the receiver. The adaptive congestion control functionality incorporated by R²CP provides the adaptive application layer with path resource availability feedback. Thus, the adaptive application layer can use this information to accurately adapt its media encoding rate in order to maximize link utilization and media reception quality. Furthermore, the adaptive congestion control mechanism also closely interacts with the adaptive link layer in order to obtain wireless access link information such as access delay and packet error rate to be used in adapting the TCP configuration [10]. On the other hand, R²CP dynamically creates and maintains multiple RCP states depending on the number of active interfaces in use. Each RCP state created at the receiver will set up a connection with a remote RCP sender. R^2CP allows different RCP states to connect to the same sender (unicast) or different senders (multipoint-to-point).[2] The sender side of an R^2CP connection is a plain RCP sender, and is oblivious to whether it is one endpoint of a multipoint-to-point or unicast connection. Note that since each RCP pipe may request noncontiguous data (depending on the transmission schedule at R^2CP) from its peer, the request is always transmitted in a unique *pull* mode. All senders of the R^2CP connection transmit whatever data is requested in an incoming REQUEST message independent of each other. The throughput performance results for TCP, TCP with explicit loss notification, and R^2CP are shown in Fig. 5. There is only one path between the sender and the receiver, and the loss rate on the wireless link is varied. The performance improvement shown by R^2CP is due to its mobile-host-centric design. For more information on R^2CP, see [2].

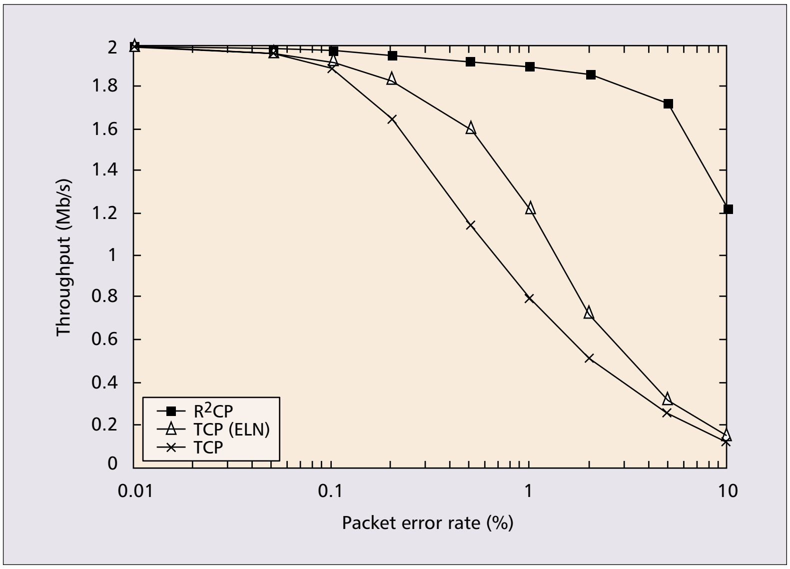


Figure 5. *R^2CP performance.*

ADAPTIVE MEDIUM ACCESS CONTROL

In this section we present the adaptive MAC (A-MAC) component of the AdaptNet suite [11]. In NG wireless networks, the MAC layer may encounter different protocols such as time-division multiple access (TDMA), code-division multiple access (CDMA), wideband CDMA (WCDMA), and carrier sense multiple access (CSMA) schemes as well as their hybrids. In addition to architectural heterogeneity, NG wireless networks are also expected to provide a diverse set of services to mobile users.

There exist several studies in the literature to address the integration of existing wireless systems [12, 13], which, however, require either significant modifications to the existing infrastructure and base stations or a complete new architecture. Therefore, these approaches lead to integration problems in terms of implementation costs, scalability, and backward compatibility.

As part of our AdaptNet protocol suite, we aim to integrate the existing wireless architectures without requiring any modifications in the base stations. We propose a new two-layered A-MAC, as shown in Fig. 6. We introduce a novel *virtual cube* concept that serves as a basis for comparison of different network structures. Based on the virtual cube concept, A-MAC provides *architecture-independent decision and QoS-based scheduling algorithms for efficient multinetwork access*.

The virtual cube concept defines a unit structure based on the resource allocation techniques used in existing networks. We model the resource in a three-dimensional space with time, frequency, and power/code dimensions that model the time it takes to transfer information, the data rate of the network, and the power consumed in transmitting information through the specific network, respectively. Furthermore, the power dimension is also used to capture the effect of multicode transmissions in CDMA networks. Based on the virtual cube concept, underlying access schemes are modeled as a three-dimensional structure called a *resource bin*. As a result, resource bins capture the capacity of the network access unit, as well as timing information, data rate, and power requirements (Fig. 7).

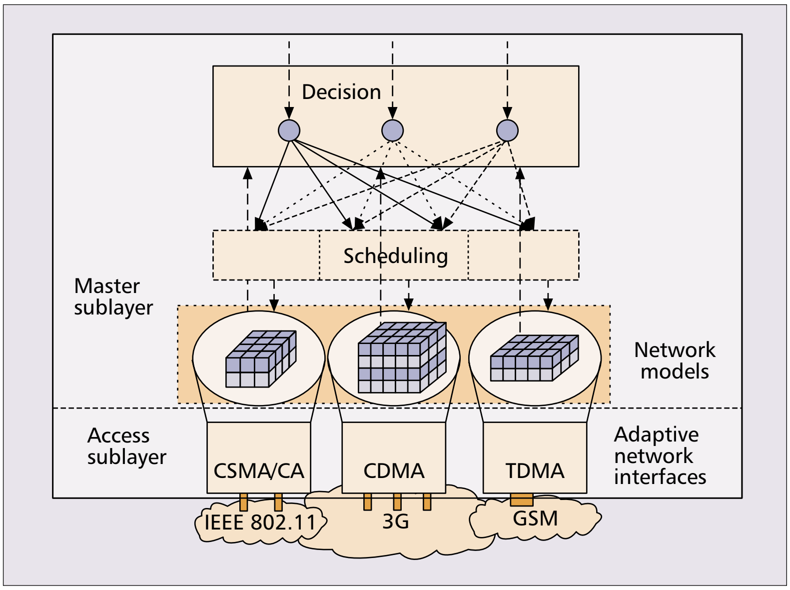


Figure 6. *Main components of A-MAC.*

A-MAC uses the virtual cube concept to accomplish adaptivity to both architectural heterogeneities and diverse QoS requirements using a two-layer structure as shown in Fig. 6. We discuss the functionality of each block in the following.

Adaptive network interfaces: The access sublayer consists of adaptive network interfaces (ANIs) that are responsible for the adaptivity of the mobile host to the underlying heterogeneous architectures. Based on the underlying physical capabilities of the mobile host, each interface is capable of performing *environment awareness, access and communication,* and *network modeling*.

[2] *We assume the decision of the destination address to connect for each pipe is external to R^2CP. Such information can be provided to R^2CP, for example, by an orthogonal application running a server selection algorithm, or simply a DNS lookup for replicated servers.*

For practical reasons, we do not want to switch between multiple encoders and decoders, but to have one encoder and decoder that is modified according to the rate without changing the main structure of the code.

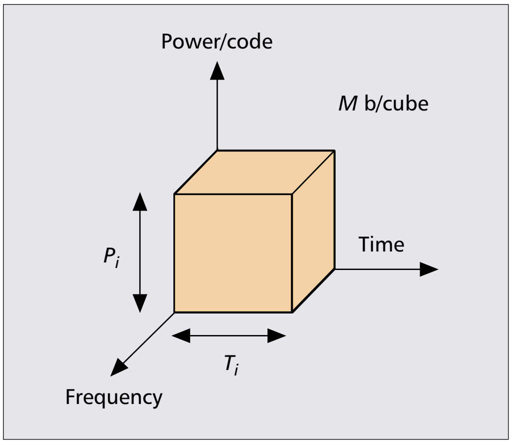


Figure 7. *The virtual cube model.*

Scheduling and decision: The master sublayer aims to forward multiple flows with various QoS requirements to appropriate networks by effectively utilizing the wireless medium and guaranteeing the QoS requirements of each flow. The scheduler is responsible for transmission of multiple flows interleaved into a single connection. For a specific traffic type, the decision block chooses the best connection. After the decision process, the bandwidth share of each traffic type in each connection is provided to the appropriate schedulers; accordingly, the scheduling is performed in the ANIs where multiple flows are directed.

For the cross-layer integration of the AdapNet suite, the MAC layer provides the information about the underlying physical network architectures to the higher layers. The capabilities of the network to which the mobile node is connected are used for the higher-layer protocols such as adaptive coding and adaptive congestion control. In addition, once the most efficient connection for traffic is selected, the mobile host performs adaptive error correction based on the channel and traffic properties. A-MAC chooses the best connection to guarantee the power considerations of the mobile host and the bandwidth requirements of the traffic, while adaptive error correction performs the most efficient error correction throughout the connection, achieving performance efficiency in all aspects of wireless communication in heterogeneous network architectures. We explore the adaptive error correction techniques next.

An Adaptive Data Link Layer

In designing an error correcting system for a time-invariant channel, we choose a code with a fixed rate and correction capability that adapts to the worst channel condition. However, in a mobile adaptive network the channel is time-varying, or different types of data have different error protection needs. Therefore, to maintain an acceptable QoS, we need to change the coding rate during transmission. For practical reasons, we do not want to switch between multiple encoders and decoders, but have one encoder and decoder that is modified according to the rate without changing the main structure of the code.

To construct rate-adaptive codes using one encoder and decoder, punctured convolutional codes are used historically. By puncturing, a higher-rate code is constructed from a low-rate parent code by eliminating some of the parity bits. Accordingly, the decoder of the parent code that knows the location of the punctured bits in the codeword can still decode the higher-rate code. The restriction of rate compatibility may also be applied by which all code bits of a high-rate punctured code are used by lower-rate codes. Therefore, if the higher-rate code is not powerful enough to correct the errors, only a supplementary set of bits needs to be transmitted.

Puncturing has adapted to turbo codes due to their good performance. Here, we propose two new methods: for lower-complexity decoding we propose rate-adaptive wavelet convolutional codes, and for higher-complexity decoding but near Shannon limit performance we investigate punctured LDPC codes. In the rest of this section we describe the two codes in more detail.

Rate-adaptive wavelet convolutional codes: In [14] we proposed using wavelet convolutional codes for rate-adaptive coding. To construct a rate K/L code, we split the message into K submessages. Then we apply those submessages to K out of L channels of an L-band orthogonal inverse wavelet system and feed zero inputs to the rest of the $L - K$ channels. The maximum achievable rate from this system is K/L. To reduce the rate, we simply split the message to fewer submessages (less than K submessages) and feed the rest of the channels of the inverse wavelet system with zero inputs. The lowest rate, $1/L$, is generated when only the first channel receives the message sequence. Therefore, the set of achievable rates is $[1/L, 2/L, \dots, K/L]$.

Since the decoding complexity of a convolutional encoder of rate K/L increases exponentially as K grows, we propose using the syndrome decoding technique. Although the wavelet convolutional encoder produces different rates, we are still able to use one trellis for its decoding. Because of the wavelet encoder structure, we draw the trellis for the highest rate, which has the maximum number of states. Then a lower-rate code is decoded by a subtrellis of the higher-rate code's trellis [14].

Rate-compatible LDPC codes: LDPC codes were first proposed by Gallager. Recently, these codes were rediscovered and improved. An LDPC code is defined as a linear block code with a sparse parity check matrix $H=[h_{ij}]$; that is, most of the elements of H are equal to 0 and a few of them are equal to 1. For an (n,k) binary linear block code, the parity-check matrix has $m = n - k$ rows and n columns. The codewords x are binary vectors of length n that satisfy the equation $Hx = 0$. Each row of H corresponds to a parity check equation, and each column corresponds to one bit of the codewords. An LDPC code can also be represented by a bipartite graph called a Tanner graph. A Tanner graph is a bipartite graph with bipartition V and C, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of variable (message) nodes and $C = \{c_1, c_2, \dots, c_m\}$ is the set of check nodes.Nodes c_i and v_j are adjacent (connected by an edge) if and only if $h_{ij} = 1$. LDPC codes can be decoded by iterative algorithms called message-passing algorithms. In these algorithms, messages are exchanged between variable nodes and check nodes iteratively. In each iteration, every check node c receives messages from all its neighbor variable nodes (two vertices are neighbors if they are adjacent). Based on these messages, the check node computes new messages and sends them to its neighbors. A message that the check node c sends to the variable node v is a function of the incoming messages from all neighbors of c except v. Similarly, variable nodes send messages to their neighbor check nodes. We consider a message passing algorithm called *belief propagation*. To perform decoding, we need to know the update equations for the belief propagation algorithm in which the log likelihood ratios (LLRs) (the ratio of the probability of a variable node being equal to zero to the probability that a variable node is one) are estimated iteratively.

To construct rate-compatible LDPC codes, we take a low-rate LDPC code, puncture a subset of the bits in the codeword, and send the unpunctured bits to the receiver. It is assumed that the decoder knows the position of the punctured bits in the original codeword. To start the decoding, we need to compute LLRs in the decoder. The LLRs of the punctured bits are set to zero, and we may use the improved iterative decoding technique [15] to compensate for the performance gap of the finite-length LDPC codes from the Shannon limit.

In [16] the authors evaluated the performance of several punctured LDPC codes and optimized the puncturing pattern to get the best performance. Their simulations showed that the performance of LDPC codes degrades for high rates because of puncturing. In [17] we study the threshold effect of punctured codes and show that it plays a central role in the performance of the LDPC codes. We obtained the puncturing capacity of the LDPC code ensembles and showed that any code has a puncturing threshold p^*. We realized that if the puncturing fraction p is smaller than p^*, the punctured code is good. On the other hand, if $p > p^*$, error probability is bounded away from zero, independent of the communication channel. We found the threshold p^* for both random and intentional puncturing. As an example, we consider the (3,6) regular ensemble as a parent code. It has a puncturing threshold $p_{th} = .4294$ (note that for regular codes the random and intentional puncturing thresholds are the same). Its cutoff rate, obtained by $R_{th} = R_p/(1 - p_{th})$, is 0.8763. Thus, we cannot obtain rates higher than .8763.

Our research work shows that the highest rate we need to achieve plays an important role in the performance of the punctured code [17]. A simple design method is to choose the parent code to be good for the binary erasure channel (BEC). We need to mention that the code with good performance over BEC is also somewhat optimal over other channels. Our simulation results show that a randomly punctured code for the range of .5–.91 has less than .7 dB gap from the capacity. It is also worth noting that random puncturing is more suitable than intentional puncturing for rate-compatible LDPC coding. This is because one chooses a fraction p_1 of the bits at random for the first rate. For the next rate more bits are chosen at random from the unpunctured bits, and so on. Thus, optimization is not required for puncturing, and one can do the puncturing in a rate-compatible way.

We note that the coding rate is selected by the input provided by the application layer protocol described earlier. This input would suggest a coding rate for the transmission that matches the heterogeneous environment needs such as the data type and channel characteristic.

Although LDPC codes have an efficient decoding algorithm, their encoding complexity is quadratic in the code length. To overcome this problem, ongoing work is investigating the application of puncturing to another type of codes called turbo-like codes or repeat-accumulate (RA) codes. These codes are a special case of LDPC codes with low encoding complexity. This makes them attractive for mobile and handheld devices whose processing power might be very limited.

To construct rate-compatible LDPC codes, we take a low-rate LDPC code and puncture a subset of the bits in the codeword and send the unpunctured bits to the receiver. It is assumed that the decoder knows the position of the punctured bits in the original codeword.

SUMMARY

A case has been made for the rethinking of network protocol design for NG wireless data networks. We have argued that the high degree of heterogeneity in future wireless data networks necessitates adaptive solutions at the different layers of the protocol stack. Finally, we have provided an overview of AdaptNet, an adaptive protocol suite for NG wireless data networks. AdaptNet consists of protocol solutions at the application, transport, and link layers, respectively. For each layer we have provided insights into the adaptiveness of the protocol and its implications.

Ongoing work on the AdaptNet protocol suite is focusing on two aspects:

- More tightly coupled cross-layered interactions for environments where such interactions are permissible
- Prototype implementation of the AdaptNet protocol suite in an NGWI testbed

More information on the AdaptNet protocol suite and developments in the project can be found at http://users.ece.gatech.edu/~akan/APS.

REFERENCES

[1] V. Lau and Y.-K. Kwok, "On the Synergy between Adaptive Physical Layer and Multiple-Access Control for Integrated Voice and Data Services in a Cellular Wireless Network," *IEEE Trans. Vehic. Tech.*, vol. 51, no. 6, Nov. 2002.

[2] H.-Y. Hsieh and R. Sivakumar, "A Receiver-centric Transport Protocol for Mobile Hosts with Heterogeneous Wireless Interfaces," *Proc. ACM MOBICOM*, San Diego, CA, Sept. 2003.

[3] B. Girod and N. Farber, "Wireless Video," *Compressed Video over Networks*, M.-T. S. A. Reibman, Ed., M. Dekker, 2000.

[4] Y. C. Lee, Y. Altunbasak, and R. M. Mersereau, "Multi-frame Error Concealment for MPEG-coded Video Delivery over Error-prone Networks," *IEEE Trans. Image Processing*, vol. 11, no. 11, Nov. 2002, pp. 1314–31.

[5] J. Chakareski and P. A. Chou, "Application Layer Error Correction Coding for Rate-Distortion Optimized Streaming to Wireless Clients," *IEEE Int'l. Conf. Acoustics, Speech and Sig. Proc.*, 2002.

[6] M. Elaoud and P. Ramanathan, "Adaptive Use of Error-correcting Codes for Real-time Communication in Wireless Networks," *IEEE INFOCOM*, 1998,pp. 548–55.

AdaptNet consists of protocol solutions at the application, transport, and link layers respectively. For each layer, we have provided insights into the adaptiveness of the protocol, and its implications.

[7] M. U. Demircin and Y. Altunbasak, "Finite-Horizon FEC-Rate Adaptation for Realtime Wireless Multimedia," *IEEE Wireless Commun. and Net. Conf.*, 2004.

[8] H.-Y. Hsieh and R. Sivakumar, "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multihomed Mobile Hosts," *Proc. ACM MOBICOM*, Atlanta, GA, Sept. 2002, pp. 83–94.

[9] M. Riegel and M. Tuexen, "Mobile SCTP," IETF Internet draft, draft-riegel-tuexen-mobile-sctp-02.txt, Feb. 2003.

[10] O. B. Akan and I. F. Akyildiz, "ATL: An Adaptive Transport Layer for Next Generation Wireless Internet," to appear, *IEEE JSAC*.

[11] I. F. Akyildiz and M. C. Vuran, "A-MAC: Adaptive Medium Access Control for Next Generation Wireless Terminals," BWN Lab Tech. Rep., School of ECE, GA Inst. of Tech., Nov. 2003.

[12] C. Yeh, "ACENET: Architectures and Protocols for High Throughput, Low Power, and QoS Provisioning in Next-Generation Mobile Communications," vol. 2, 2002, pp. 779–86.

[13] H. L. Z. Naor, *A Centralized Dynamic Access Probability Protocol for Next Generation Wireless Networks*, vol. 2, 2001, pp. 767–75.

[14] F. Fekri *et al.*, "Convolutional Codes Using Finite-Field Wavelets; Time-Varying Codes and More," Feb. 2003,submitted to *IEEE Trans. Sig. Proc.*

[15] H. Pishro-Nik and F. Fekri, "On Decoding of LDPC Codes over the Erasure Channel," Sept. 2003, accepted in *IEEE Trans. Info. Theory*.

[16] J. Ha and S. McLaughlin, "Optimal Puncturing of Low-Density Parity-Check Codes," *Proc. IEEE ICC*, 2003.

[17] H. Pishro-Nik and F. Fekri, "Results on Punctured Low-Density Parity-Check Codes and Improved Iterative Decoding Techniques," Aug. 2003, submitted to *IEEE Trans. Info. Theory*.

Biographies

IAN F. AKYILDIZ [F '95] (ian@ece.gatech.edu) received his B.S., M.S., and Ph.D. degrees in computer engineering from the University of Erlangen-Nuernberg, Germany, in 1978, 1981, and 1984, respectively. Currently, he is the Ken Byers Distinguished Chair Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, and director of the Broadband and Wireless Networking Laboratory. He is Editor-in-Chief of *Computer Networks* (Elsevier) and the newly launched *AdHoc Networks Journal* (Elsevier). He is an ACM Fellow (1996). He received the ACM Outstanding Distinguished Lecturer Award for 1994, the 1997 IEEE Leonard G. Abraham Prize award (IEEE Communications Society), and the 2002 IEEE Harry M. Goode Memorial award (IEEE Computer Society) with the citation "for significant and pioneering contributions to advanced architectures and protocols for wireless and satellite networking." He also received the 2003 IEEE Best Tutorial paper award (IEEE Communications Society) and 2003 ACM SIGMOBILE award for his significant contributions to mobile computing and wireless networking. His current research interests are in wireless networks, sensor networks, and InterPlaNetary Internet.

YUCEL ALTUNBASAK (yucel@ece.gatech.edu) received M.S. and Ph.D. degrees from the University of Rochester, New York, in 1993 and 1996, respectively. He joined Hewlett-Packard Research Laboratories in July 1996. Meanwhile, he taught at Stanford and San Jose State Universities as a consulting assistant professor. He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology in 1999. He has worked on industrial- and government-sponsored projects in the areas of multimedia processing, communications, and networking. His research efforts to date have been published in over 110 peer-reviewed publications and 15 patents/patent applications. He is an associate editor for *IEEE Transactions on Image Processing*, *IEEE Transactions on Signal Processing*, *Signal Processing: Image Communications*, and the *Journal of Circuits, Systems and Signal Processing*. He served as the lead guest editor of an *Image Communications* special issue on wireless video. He has been selected as the Vice President of the IEEE Communication Society's Multimedia Communications Technical Committee. He has also been elected to the IEEE Signal Processing Society's Image and Multidimensional Signal Processing (IMDSP) Technical Committee. He served as co-chair of the Advanced Signal Processing for Communications Symposia at ICC '03. He will serve as the technical program chair for ICIP 2006. He also served as a track chair at ICME '03 and ICME '04, as a panel sessions chair at ITRE '03, as a session chair at several international conferences, and as a regular panel reviewer for government funding agencies.

FARAMARZ FEKRI (fekri@ece.gatech.edu) received B.Sc. and M.Sc. degrees from Sharif University of Technology, Tehran, Iran, in 1990 and 1993, respectively, and a Ph.D. degree from Georgia Institute of Technology in 2000. From 1995 to 1997, he was with the Telecommunication Research Laboratories (TRLabs), Calgary, Alberta, Canada where he worked on multicarrier spread spectrum systems. Since 2000 he has been on the faculty of the School of Electrical and Computer Engineering at Georgia Institute of Technology. His current research interests lie in the general field of signal processing and communications, in particular, wavelets and filter banks, error control coding, cryptography, and communication security. In the past, he conducted research on speech and image processing. His work on finite-field wavelets and their application to error control coding received the Best Ph.D. Thesis Award of Georgia Institute of Technology in 2000. He received the National Science Foundation CAREER award in 2001.

RAGHUPATHY SIVAKUMAR (siva@ece.gatech.edu) is an assistant professor in the School of Electrical and Computer Engineering at Georgia Tech where he leads the Georgia Tech Networking and Mobile Computing (GNAN) Research Group. He received his Ph.D. and M.S. degrees in computer science from the University of Illinois at Urbana-Champaign in 2000 and 1998, respectively, and his B.E. degree in computer science from Anna University, Chennai, India, in 1996. His research interests are in wireless networking, mobile computing, and computer networks.