Cross-layer design optimizations in wireless protocol stacks

Vijay T. Raisinghani *,1

Tata Infotech (ATG) and KR School of IT, IIT Bombay

Sridhar Iyer

KR School of IT, IIT Bombay

Abstract

It is known that the performance of applications on mobile devices is affected by the device constraints of memory, processing power, battery life and the variations in the wireless network. The variations in the wireless network will be compounded in the next generation networks – 3G and beyond – when the devices move across heterogeneous networks. To allow interoperability with the Internet, existing standard protocol stacks would be deployed in the new networks and mobile devices. However, protocol stack *implementations* based on layering do not function efficiently in mobile wireless environments.

Cross layer feedback in the protocol stack is useful to improve the performance of mobile devices. We discuss the benefits of cross layer feedback on the mobile device and present a representative survey. We also present an architecture to enable efficient cross layer feedback.

Key words: Seamless mobility, cross layer feedback, 4G networks, protocol stacks

* Corresponding author
Email addresses: rvijay@it.iitb.ac.in (Vijay T. Raisinghani), sri@it.iitb.ac.in (Sridhar Iyer).
URLs: http://www.it.iitb.ac.in/~rvijay (Vijay T. Raisinghani), http://www.it.iitb.ac.in/~sri (Sridhar Iyer).
[1] Author is Ph.D student sponsored by Tata Infotech Ltd.

1 Introduction

The new wireless networks – 3G and beyond – would be all-IP. See [1] for a series of articles on 4G. A typical mobile computing set-up is shown in figure 1. Standard (e.g. TCP/IP [2]) protocol stacks would be used on the intermediate nodes (e.g. base stations, access points) and the mobile devices to ensure interoperability with the existing Internet.

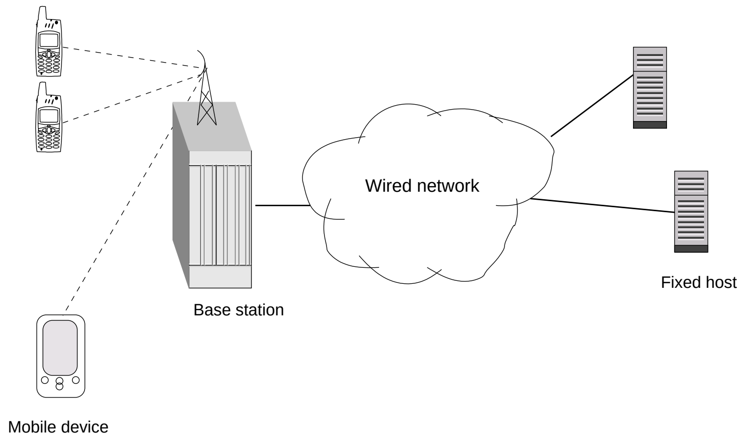

Fig. 1. Typical mobile wireless scenario

The standard protocol stacks are designed and implemented in a layered manner in accordance with the OSI [3] reference model (see figure 2). It is well known that layering is desirable since it helps in creation of standard modular software components. Also, layering cleanly separates abstraction from implementation and is thus consistent with sound software engineering principles – information hiding[4] and end-to-end principle[5]. However, in [6] it is emphasized that it is important to distinguish between the *architecture* of a protocol suite and the *engineering* of a specific end system or relay node.

Protocol stack *implementations* based on layering do not function efficiently in mobile wireless environments[7]. This is due to the highly variable nature of wireless links and the resource-poor nature of mobile devices. There have been research efforts for improving performance of the protocol stack by using cross layer feedback. For wired networks some of the early work [8,9,6] showed that information exchange between protocol layers is useful. Cross layer feedback means allowing interaction of a layer with any other layer in the protocol stack. Some examples: (1) TCP may share packet loss and available throughput information with the application layer so that the application can adapt accordingly, (2) Link/MAC layer may tune transmit power of the physical layer to control bit-error rate.

Cross layer feedback optimizations may be implemented at the intermediate nodes or mobile hosts (MH). There have been proposals for cross layer feedback at intermediate nodes, for example: link layer *snoop* on TCP packets [10], channel state dependent packet scheduling [11]. We focus on cross layer feedback in the MH since we believe that it would be easier to deploy the optimized protocol stack in the new devices at the manufacturing stage itself. Further, no changes would be required to the existing network.

We believe that improving user satisfaction is the ultimate goal of improving application performance over wireless devices. Cross layer feedback on a mobile device would help in improving application performance and thus user satisfaction. However, to further enhance user satisfaction it is essential to incorporate dynamic user requirements also into the protocol stack. For example, it would be useful to allow a user to indicate and change application priorities dynamically rather than letting the system fix application priorities. Further, user satisfaction is also affected by the time for which the mobile device can be used, after a battery recharge. This is the time for which a fully charged battery lasts. Since, it is known that a large fraction of the battery is consumed by the network interface [12], it is imperative that the various protocol layers adapt and collaborate to optimize power consumption.

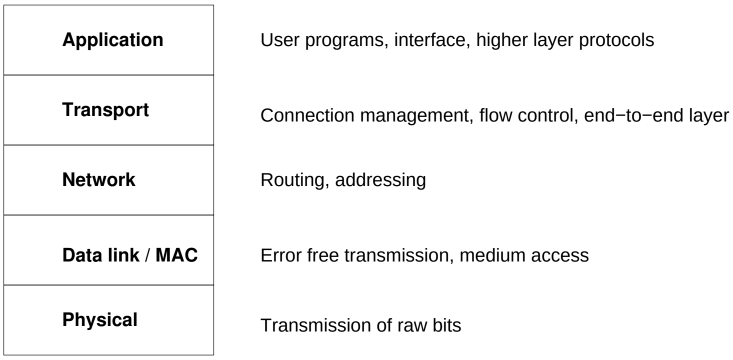

Fig. 2. TCP/IP protocol stack[13]

In section 2 we discuss the various cross layer feedback possibilities and the potential benefits. In section 3 we present our proposal for a cross layer feedback architecture.

2 Cross layer feedback

Cross layer feedback means allowing interaction of a layer with any other layer in the protocol stack. A layer may interact with layers above or below it. For the sake of convenience, cross layer feedback can be categorized as follows:

- *Upper to lower layers:* Information flowing downwards from a layer to any layer below it, is put under this category. Some examples are: application requirements (acceptable delay or acceptable packet losses) are communicated to the link layer to enable the link layer to adapt its error correction mechanisms; user defined application priority is communicated to TCP to increase the receiver window of the application with a higher priority.

- *Lower to upper layers:* Information flowing upwards from a layer to any layer above it, is put under this category. Some examples are: TCP packet loss information is given to the application layer so that the application can adapt its sending rate; physical layer transmit power and bit-error rate information is communicated to the link/MAC layer to enable it to adapt error correction mechanisms or manipulate physical layer transmit power.

We list a few examples of cross layer feedback for each layer before presenting the details:

- **Physical:** Channel condition (e.g. bit-error rate) status from the physical layer can be used by link layer to adapt its error control mechanisms. Also, the physical layer transmit power can be tuned by the MAC layer to increase the range of transmission.
- **Link / MAC layer:** The number of retransmissions at the link layer can serve as a measure of channel condition. TCP may re-estimate its retransmission timers based on this data. The link layer may adapt its error correction mechanism based on the Quality-of-Service (QoS) i.e. acceptable delay, packet losses, etc. requirements of the application layer.
- **Network:** Mobile-IP hand-off begin/end information can be used at TCP to manipulate its retransmission timer. Mobile-IP layer could use link layer hand-off events to reduce Mobile-IP hand-off latency.
- **Transport:** Packet loss data at TCP can help the application layer adapt its sending rate. Link layer can adapt its error control mechanisms based on TCP retransmission timer information.
- **Application:** An application could use information about channel conditions from the physical layer to adapt its sending rate. Also, an application could indicate to the user the throughput it requires versus the available throughput.
- **User:** A user may give the system an indication of impending disconnection. This information may be used by TCP to freeze its retransmission timer.

We now present the details of how the cross layer information can be utilized by a layer to adapt itself or tune the parameters of some other layer. We also discuss the benefits and indicate some of the disadvantages. For each layer we present its function, important cross layer information available at that layer and important information available to that layer from other layers. We also highlight the crucial cross layer information at each layer which could have large impact on user perceived QoS.

The function of the physical layer is to transmit raw bits at a certain power level to achieve a certain transmission range and reduce bit errors.

The information available at the physical layer is: transmit power, bit-error rate and coding/modulation in use.

Since the link/MAC layer is responsible for error free transmission over the wireless link, using physical layer information for adapting the link/MAC layer error control mechanisms could give large gains in throughput and power saving. By similar argument, we believe that the transmit power control instructions from the link/MAC layer to the physical layer would also help in achieving similar gains.

We list below the important interactions of the physical layer with other layers:

- *Interaction with upper layers:*
  - *Application or user:* The application layer or the device user may decide that the current physical layer characteristics are not sufficient for their requirements and tune the physical layer according to their requirements or download the software for another physical layer [14]. Further, multimedia applications like video may adapt its coding rate based on channel condition information from the physical layer (see section 2.5 for details).

    One of the disadvantages of downloading software for the physical layer could be high power consumption for downloading.
  - *Network:* Bit-error rate information may be used by the network layer to change to another interface (see section 2.3).
  - *Link/MAC:* Link/MAC layer could instruct the physical layer to increase the transmit power to reduce the bit-error rate. Conversely, the link/MAC layer may instruct the physical layer to reduce transmit power while increasing the error correction at the link layer. However, increasing transmit power increases power consumption. Increased error correction at the link/MAC layer may also increase the power consumption. Thus, a combined mechanism [15] would be required for optimum power consumption.
- *Physical layer adaptation using battery information:* Besides the interactions with the other layers, the physical layer may adapt itself based on battery status. It may also adapt its coding/modulation based on the QoS requirements of higher layers and channel conditions. See [16] for an example of adaptive modulation based on channel conditions.

  It should be noted that for *explicit* measurement of bit-error rate of the data transmitted to the receiver, the receiver will have to provide feedback to the sender.

The functions of link/MAC layer can be summarized as follows: link reliability through forward error correction (FEC) and Automatic Repeat reQuest (ARQ); medium access protocol to avoid/reduce collisions; framing(with a particular frame-size) of data to be transmitted to ensure reliable transmission with minimal overhead.

FEC/ARQ are related since both are used to improve link reliability. FEC requires transmission of extra bits to enable the receiver to recover the correct data in spite of bit-errors. ARQ on the other hand ensures reliability by retransmitting the corrupted frames i.e. the frames for which the receiver does not send an acknowledgment.

The information at the link/MAC layer which can be used by other layers is: current FEC scheme, number of frames retransmitted, frame length, point in time when the wireless medium is available for transmission and hand-off initiation/completion events.

The next generation wireless networks will be all-IP. Thus, we believe that the feedback about link layer hand-off to Mobile-IP would be crucial and useful in improving throughput, by reducing Mobile-IP hand-off latency. Besides this, the application feedback to the link layer about its QoS needs will be important to enhance user satisfaction. The link layer can use this information to adapt its error control mechanism.

The important interactions with link/MAC layer are:

- *Interactions with upper layers:*
  - *User:* The user can be informed about the link throughput. See section 2.6 for details.
  - *Application:* Different applications have different QoS requirements. At the link/MAC layer the frames may be treated differently based on the QoS needs of the corresponding application. For example, frames of applications with low delay requirement may be transmitted on priority [17]. Similarly, frames of application with higher reliability requirement may be transmitted with higher reliability by adapting the FEC/ARQ [17]. However, these schemes of adapting transmission priority or FEC/ARQ may lead to processing overhead and hence delays, besides higher power consumption.
  - *Transport:* When channel conditions are poor, delays due to retransmissions at the link layer may cause TCP to time-out, start retransmission and reduce its sending rate. To avoid this, TCP round-trip time or retransmission timer values can be used to control link layer retransmissions. Similarly, retransmission information from the link layer could be

used to adapt TCP retransmission timer. See [18] for the effect of link layer retransmissions on TCP round-trip time and thus its throughput. Results in [18] show a need for TCP and link layer interactions.
  · *Network:* Mobile-IP is used for IP hand-off whenever the mobile device changes sub-nets. However, Mobile-IP hand-off involves delays since the hand-off is initiated when network change is detected at the IP layer. This information may not be available as quickly as changes in the signal strength which is continually measured by the device. Thus link layer hand-off information can be used to reduce the hand-off latency for Mobile-IP [19,20].
- *Interactions with lower layers:*
  · *Physical:* Based on current channel conditions the error control mechanisms at the link layer may be adapted to reduce the transmission errors [21–23]. [21] also show that using channel condition information the frame length can be adapted to improve the throughput. For tuning physical layer power see section 2.1.
- *Battery aware link/MAC layer:* Adaptation of the FEC/ARQ mechanisms or the physical layer transmit power may lead to increased power consumption. Thus it is essential that these schemes adapt in tandem so as to optimize energy consumption [24,15].

2.3 Network layer

The function of the network layer is routing and addressing. The network layer selects an interface over which the packets are transmitted. Also, the Mobile-IP layer handles IP hand-off to transparently maintain IP connectivity in *foreign* networks.

The information available at the network layer which would be useful for cross layer feedback is: Mobile-IP hand-off initiation/completion and network interface in use (assuming that the device has multiple network interfaces).

We believe that Mobile-IP hand-off initiation and completion information to the application layer and TCP will be important to conserve battery and increase throughput. TCP can use the hand-off information to freeze its retransmission timer and the application layer can stop sending data till the hand-off is completed. Further, as discussed in section 2.2 reducing Mobile-IP hand-off delay is essential to avoid throughput degradation. Thus, to speed up Mobile-IP hand-off, feedback to the Mobile-IP layer about link layer hand-offs will be essential.

- *Interactions with upper layers:*
  · *Application or user:* A device may have multiple wireless network inter-

faces that can provide different levels of service. For example, a wireless LAN interface may provide lesser delays and higher throughput as compared to a GPRS interface on the same device.

Application layer QoS requirements i.e. information about acceptable delay or throughput, can be used to select the appropriate interface. Another scenario when an interface change may occur is when a user moving into office from his car switches the interface from GPRS to WLAN. The network layer will route the packets onto the new network interface whenever requested by the upper layers. However, continuing a session uninterrupted on the new interface is an open research area.

  · *Transport:* Mobile-IP hand-off information may be used by TCP to manipulate its timers and avoid unnecessary packet retransmissions [25]. We present the details of this in the section on TCP i.e. section 2.4.

- *Interactions with lower layers:*
  · *Link/MAC:* As discussed in section 2.2, link layer hand-off information can be used to reduce the Mobile-IP hand-off latency [19,20].
  · *Physical:* The bit-error rate on an interface could be used as a guide by the network layer to switch to another network interface which has lower bit-error rate.

2.4 Transport

Transport layer is concerned with establishing end-to-end connections over the network. Mobile networks are characterized by large delays, packet losses and high bit error rates. Transport protocols like TCP interpret this as congestion losses which reduces its throughput[25].

The information available with TCP which may be of use to other layers is: round-trip time(RTT), retransmission timeout(RTO), maximum transmission unit (MTU), receiver window, congestion window, number of packets lost and actual throughput (or goodput).

We believe that the information about packet losses and actual throughput would be crucial. This information would give a feedback to the user about the kind of throughput to be expected from the connection. The user may shut-down some applications based on this information to allow other critical applications to continue.

As pointed out earlier, TCP interprets variations in mobile wireless environments as congestion. Hand-off and associated delay is one of the important characteristics of mobility that reduces TCP throughput. Thus, feedback from layers like Mobile-IP and link layer, about hand-off initiation and completion, would be essential in improving TCP throughput.

We list the important interactions of TCP with other layers:

- *Interaction with upper layers:*
  - *User:* A user may assign priorities to the running applications. In case the applications are downloading some data, this higher priority would indicate the need for higher download bandwidth. To enhance user satisfaction, TCP may map the higher priority of an application to a larger receive window [26]. Further, a user could provide information about impending disconnection. This information can be used by TCP to increase its RTO values, since the flow of TCP acknowledgments to the mobile host will be disrupted. It should be noted that receiver window control[26] on the mobile host will result in throughput improvement if TCP on the sender is not constrained by congestion.
  - *Application:* Applications may indicate their QoS requirements to TCP. Based on this information TCP may manipulate the receiver windows. On the other hand, TCP may provide packet loss and goodput information to the application. The application can use this input to adapt its sending rate.
- *Interaction with lower layers:*
  - *Network:* Mobile-IP hand-off delay may lead to reduced throughput due to TCP retransmission and back-off mechanism. TCP can be informed about the event of Mobile-IP hand-off to avoid this timeout. A *fast retransmit* [25] can be initiated using this information. This can help reduce the latency in recovering from a timeout.
  - *Link/MAC:* Delays due to link layer retransmissions may lead to decreased TCP throughput [18]. Information about packets pending retransmission at link layer may be used to adjust the RTT/RTO (also see section 2.2.) Also, see [27,28] for benefits of link layer and TCP interactions.

2.5 Application

The application layer is the interface to the user for running user tasks. For example: web browsing, downloading a file using FTP, sending e-mail, watching a video clip, etc.

The existing applications were designed for wired networks and do not perform well in wireless networks. Application adaptation based on information from lower layers would be useful in improving application performance over wireless networks.

An application layer can communicate to other layers the application's QoS needs i.e. the delay tolerance, acceptable delay variation, required throughput and acceptable packet loss rate.

We believe that interactions with the physical and link layer would be crucial in improving application performance. The errors on the physical layer can help applications adapt in various ways to reduce the bandwidth requirement. For applications like video, the application layer coding is a measure of the bandwidth required for transmission. Thus, if information about channel conditions is available from the physical or link layer, the application can adapt its coding rate so as to reduce the bandwidth required. Similarly, tuning the link layer error control mechanisms based on application QoS requirements would be essential in improving application throughput.

- *Interaction with upper layers:*
  - *User:* A user's requirement can be captured by an application and communicated to the lower layers. The mobile device could then be re-configured to satisfy user needs [14].
- *Interaction with lower layers:*
  - Transport: Application layer could provide its QoS requirements to TCP. TCP may tune the receiver window for the various applications to improve throughput for applications that require higher throughput. Also see section 2.4.
  - *Network:* Application QoS requirements could be used by the network layer to switch the network interface. See section 2.3.
  - *Link/MAC:* As discussed in section 2.2, link layer may prioritize frames of applications with higher QoS requirements.
  - *Physical:* Multi-media applications like video, use various standard coding techniques for video transmission. Such applications can use information about channel condition to adapt the coding [29]. For example, if the bandwidth is low, a lower quality video coding may be used which requires lesser bandwidth. Similarly, an email application downloading emails to the mobile device could defer downloading the file attachments when the channel conditions are poor. Some of the proposals for application adaptation are [30–32]
- *Power saving based on application information:* It may be possible to switch off the network interface card intermittently [33]. This can be done if an application can tolerate some delays or has periods when it is not transmitting data. Also, information about the type of coding used by a video-application could be used to discard some frames at the network interface to save power [29]. However, this will reduce the video quality.

2.6 User

We consider the user to be the uppermost layer of the protocol stack. We believe that user requirements should be taken into account to enhance user perceived QoS. The motivation for this is that the user decision may be contrary to the system decision but would lead to improved user satisfaction. For example: (1) for a user a FTP download may be more important than a streaming video, (2) a user may know that a disconnection is imminent in an approaching tunnel while the system will *know* it only after the signal is affected, (3) the system may decide to conserve battery by not downloading some information while the user may, at that instant, feel that the information (e.g. a *stock quote*) is more important than saving battery.

The user will need information from the lower layers to use the mobile device effectively. We believe the most crucial one will be link throughput information from the link layer. This will help the user decide about the applications that can be run and also indicate to the user the kind of performance that should be expected.

- *Interaction with lower layers:*
  - · *Application:* The application can be used to capture user requirements and communicating it to lower layers [14].
  - · *Transport:* One example of using dynamically defined user defined application priority is receiver window control. Higher priority applications are given larger receive windows [26,34]. Information about impending disconnection, from the user, can be used by TCP to manipulate its retransmission timer(see section 2.4).
  - · *Network:* A user may instruct the network layer to switch the network interface from, say WLAN to GPRS, since the user may be moving out from office in the next few minutes (Also see section 2.3).
  - · *Link/MAC:* Link throughput information will help the user decide the applications that can be run and also indicate to the user the kind of performance that should be expected.
  - · *Physical:* The user requirements could be used to reconfigure the terminal's physical layer [14]. Also see section 2.1.
- *Battery status:* If the battery is low, the user may decide to switch off the device; instruct the system to optimize power consumption or override all power saving optimizations in case some information is required on priority. If the battery is almost fully charged, the user may allow higher power consumption in order to improve performance of some applications like video. However, the user decisions will be based on other factors also. For example, how long the user wants the battery to last.

We summarize the various cross layer optimizations in table 1. We also list some additional references in the table.

It can be seen from the various ideas discussed that cross layer feedback is quite complex than just sharing information between layers. Also, there may be other ideas for cross layer feedback which may come up as the new generation wireless networks are deployed. We believe that an architecture for cross layer

Information users → creators ↓	**(User)**	**Application**	**Transport (TCP)**	**Network (IP)**	**Link / MAC**	**Physical**
(User)		- User QoS requirements [14]	- Application priority[26] - Impending disconnection	- Interface selection		- User QoS requirements [14]
Application	- Goodput		- QoS	- QoS [31]	- QoS [31], [17] - Coding [29]	- QoS, Bandwidth request [31] - Power control [31], [33]
Transport (TCP)	- Packet loss / goodput	- Packet loss / goodput			- RTT/RTO information	
Network (IP)			- Hand-off [25] - Interface change[35]		- QoS [24]	
Link / MAC	- Errors [36] - Goodput	- Errors [36], [30]	- Errors [36] - Interface queue [28] - Hand-off	- Errors [36] - Channel condition [28] - Route failure [37] - Hand-off [19], [20] - RTS/CTS information [38]		- Errors [36] - Power [15]
Physical		- Bandwidth available [30]		- Capability information [39] - Bandwidth [40] - Interface / card removed [35]	- Channel conditions / Signal strength [21], [22], [24], [41], [42]	
(Battery / Power / Energy)	- Battery status	[43], [44], [30], [29]	[43], [44]	[43], [44]	[43], [44], [15]	[43], [44], [15]

Table 1. Cross layer information exchange

feedback would enable efficient implementation of such feedback. Further, user perceived QoS and the need for energy saving would drive the cross layer architecture.

In the next section we present our proposal for a cross layer architecture [45].

3 Cross layer architecture

Cross layer feedback would involve much more complexity and information exchange than shown in table 1. This indicates that a well defined architecture for cross layer feedback would be useful. Proposals like [35] discuss some ideas about cross layer feedback across the stack, but do not suggest a generic architecture for cross layer feedback.

One of the important constraints for deciding on an architecture for a wireless device is the resources. However, the various wireless devices used for access would have differences in resources i.e. differences in CPU clock speed, hard-disk capacity, RAM, etc. Thus, it may be useful to define a few architectures which cater to the different categories of devices. Finally, for deployment the architecture may be selected according to the available resources on the device. For example: for a high-end laptop a highly modular architecture and thus high overheads may be acceptable. For a PDA it may be necessary to sacrifice some modularity to reduce overheads in information exchange.

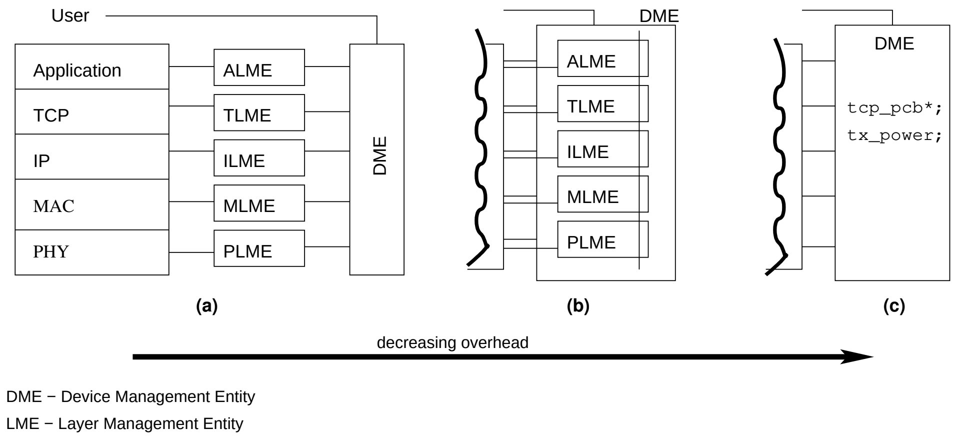

Fig. 3. Cross layer architecture

Figure 3 shows the three main architectures we propose for cross layer feedback, which could be selected according to device resources. We believe that the three architectures should cater to a broad category of devices.

As shown in figure 3, our architecture consists of various *management entities*. The Device Management Entity(DME) serves as the protocol stack interface and control entity for the device. It has access to device information like display, battery status, etc., which could be used by the protocol stack layers. Each protocol layer has a Layer Management Entity(LME). A LME would have access to the internal data structures of its layer. The LME can read/write to those data structures. Also, the LME may have additional algorithms to process cross layer information.

The DME and LMEs are combined in various ways to give the three architectures as shown in figure 3. Figure 3(a) shows a highly modular architecture wherein all cross layer communication is through the DME only. In figure 3(b) the LMEs can communicate directly with each other. Figure 3(c) shows the DME as a global data-structure i.e. the DME and the protocol stack layers can communicate directly.

For standardizing the interfaces, APIs will be required for communication between the DME and LMEs. Next, we briefly discuss the possible APIs for the DME and LME.

DME API. The DME service access points (SAP) can be abstracted into a set of two simple functions DME.SET and DME.GET. For example: DME.SET (LINK, RETX, <number>) can be used to set the number of retransmissions at the link layer. DME.GET (BANDWIDTH) can be used to get the current bandwidth. The link layer may set the transmission power level of the physical layer through DME.SET (PHY, POWER, <level>). TCP may read the current bandwidth through DME.GET (BANDWIDTH). The DME can internally redirect the calls to various LMEs or directly write or read variables according to the architecture selected (see figure 3).

LME functionality. Besides the data items shown in table 1, the LMEs may include some additional functionality. For example: the TCP management entity could have some functionality to map user specified application priorities to TCP specific parameters.

Additionally, some set (or all) LMEs may register with DME for events, to enable LMEs to receive instant updates to important data like bandwidth.

We believe that no single architecture would be optimal for any device i.e. certain positive aspects of one architecture may help improve other architectures. For example, from figure 3(c) it can be seen that there could be some common functionality which can be put in a *core* – a globally accessible data-structure – to reduce access overheads. In the following paragraph we give an overview of this core.

Defining the core of the architecture. We believe that the core will help in reducing the overheads involved in cross layer feedback. Also the core will serve as the central object for interfacing with entities outside the stack, such as the display or the user. The core may contain information which has a high *access score* a_s. In general, a_s is a function of: the number of layers reading an information item and each layer's frequency of reading that information. A simple implementation example could be: $a_s = \sum f_i$, where f_i is the probable number of times that data item is accessed over some time period by the i^{th} layer. In this case, if the hand-off information from the link layer is accessed twice ($f_{tcp} = 2$) by TCP and once ($f_{ip} = 1$) by IP, then $\sum f_i$ is 3 for hand-off information (see table 1).

We presented above the different possibilities of cross layer feedback and a representative survey and motivated the need for a cross layer architecture. We also presented our proposal for cross layer architecture.

4 Related surveys

See [43] for a survey of work addressing energy efficient and low-power design within all layers of the wireless network protocol stack. See [36] for the impact of higher order error statistics on the various layers of the protocol stack. The authors discuss the impact of such information for adaptive protocol design throughout the stack. Power aware protocols in ad hoc networks are discussed in [44]. References therein provide insight into the various power aware protocol proposals and design issues. [46] present a conceptual framework for network and client adaptation. They survey the various proposals for application adaptation and map it to the conceptual framework. [47] present a survey on the various suggestions to improve TCP behavior over heterogeneous networks.

5 Conclusion and future work

Many surveys predict that in the next few years the number of wireless devices accessing the Internet will far exceed the number of wired devices. The need for a paradigm shift from strictly layered protocol stacks to cross layer feedback is clear from the benefits of cross layer feedback.

Currently, there does not exist a complete and well defined architecture for cross layer feedback across the protocol stack. We believe that an architecture is required for implementing efficient feedback in protocol stacks. We presented our proposal for a cross layer feedback architecture. We are currently in the process of formalizing and refining the architecture.

Although, a number of cross layer ideas hold promise, we believe that *user* *feedback* and *battery constraints* will drive the algorithms for cross layer interactions. Further research efforts are needed to combine subjective user inputs with the quantitative inputs from the various layers to enhance user satisfaction in wireless networks.

References

[1] A. Jamalipour, S. Tekinay (Eds.), Fourth Generation Wireless Networks and Interconnecting Standards, Vol. 8 of IEEE Personal Communications, 2001.

[2] W. Richard Stevens, TCP/IP Illustrated, Volume I, The Protocols, AWL, 1994.

[3] International Telecommunication Union, Information technology - Open Systems Interconnection - Basic Reference Model: The basic model, ITU-T Recommendation X.200 (Jul. 1994).

[4] D. L. Parnas, On the Criteria to be Used in Decomposing Systems into Modules, Comunications of the ACM 15 (12) (1972) 1053–1058.

[5] J. H. Saltzer, D. P. Reed, D. D. Clark, End-To-End Arguments in System Design, ACM Transactions on Computer Systems 2 (4) (1984) 277–288.

[6] D. D. Clark, D. L. Tennenhouse, Architectural Considerations for New Generations of Protocols, in: ACM SIGCOMM, Phildelphia, PA, 1990, pp. 200 – 208.

[7] G. Xylomenos, G. C. Polyzos, Internet Protocol Performance over Networks with Wireless Links, IEEE Network 13 (4) (1999) 55 – 63.

[8] D. D. Clark, The Structuring of Systems using Upcalls, in: ACM Symposium on Operating Systems, 1985, pp. 171 – 180.

[9] G. H. Cooper, The Argument for Soft Layer of Protocols, Tech. Rep. TR-300, Massachussets Institute of Technology, Cambridge, MA (May 1983).

[10] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz, A Comparison of Mechanisms for Improving TCP Performance over Wireless Links, IEEE/ACM Transactions on Networking 5 (6) (1997) 756–769.

[11] P. Bhagwat, P. Bhattacharya, A. Krishna, S. K. Tripathi, Using Channel State Dependent Packet Scheduling to Improve TCP Throughput over Wireless LANs, Wireless Netwoks 3 (1) (1997) 91–102.

[12] M. Stemm, R. H. Katz, Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices, IEICE Transactions on Communications E80-B (8) (1997) 1125–31.

[13] A. S. Tanenbaum, Computer Networks, 3rd Edition, Prentice-Hall, 1996.

[14] A. Aghvami, T. Le, N. Olaziregi, Mode Switching and QoS Issues in Software Radio, IEEE Personal Communications 8 (5) (2001) 38–44.

[15] J.-P. Ebert, A. Wolisz, Combined Tuning of RF power and Medium Access Control for WLANs, Mobile Networks and Applications 6 (5) (2001) 417–426, special issue on Mobile Multimedia Communications (MoMuC '99).

[16] W. T. Webb, R. Steele, Variable rate QAM for Mobile Radio, IEEE Transactions on Communications 43 (7) (1995) 2223–2230.

[17] G. Xylomenos, G. C. Polyzos, Quality of Service Support over Multi-Service Wireless Internet Links, Computer Networks 37 (5) (2001) 601–615.

[18] A. DeSimone, M. Chuah, O. Yue, Throughput Performance of Tranport-layer Protocols Over Wireless Lans, in: IEEE GLOBECOM, 1993.

[19] A. Sanmateu, F. Paint, L. Morand, S. Tessier, P. Fouquart, A. Sollund, E. Bustos, Seamless Mobility across IP Networks using Mobile IP 40 (1) (2002) 181–190.

[20] J. C.-S. Wu, C.-W. Cheng, N.-F. Huang, G.-K. Ma, Intelligent Handoff for Mobile Wireless Internet, in: Cáceres et al. [48].

[21] P. Lettieri, M. B. Srivastava, Adaptive Frame Length Control for Improving Wireless Link Throughput, Range and Energy Efficiency, in: INFOCOM (2), 1998, pp. 564–571.

[22] R. Ludwig, A. Konrad, A. D. Joseph, R. H. Katz, Optimizing the End-to-End Performance of Reliable Flows over Wireless Links, Wireless Networks 8 (2–3) (2002) 289 – 299.

[23] I. F. Akyildiz, I. Joe, H. Driver, Y.-L. Ho, An Adaptive FEC Scheme for Data Traffic in Wireless ATM Networks, IEEE/ACM Transactions on Networking 9 (4) (2001) 419–426.

[24] P. Lettieri, C. Fragouli, M. B. Srivastava, Low Power Error Control for Wireless Links, in: MobiCom, Budapest, Hungary, 1997, pp. 139–150, also in WINET 5 (1999).

[25] R. Cáceres, L. Iftode, Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments, IEEE Journal on Selected Areas in Communications 13 (5) (1995) 850–857.

[26] V. T. Raisinghani, A. K. Singh, S. Iyer, Improving TCP Performance over Mobile Wireless Environments using Cross Layer Feedback, in: IEEE International Conference on Personal Wireless Communications, New Delhi, India, 2002.

[27] M. Methfessel, K. F. Dombrowski, P. Langendörfer, H. Frankenfeldt, I. Babanskaja, I. Matthaei, R. Kraemer, Vertical Optimization of Data Transmission for Mobile Wireless Terminals, IEEE Wireless Communications 9 (6) (2002) 36–43.

[28] D. A. Maltz, J. Broch, D. B. Johnson, Lessons from a Full-Scale Multihop Wireless Ad Hoc Network Testbed, IEEE Personal Communications 8 (2001) 8–15.

[29] P. Agrawal, S. Chen, P. Ramanathan, K. Sivalingam, Battery power sensitive video processing in wireless networks, in: Proceedings IEEE PIMRC'98, Boston, 1998.

[30] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, K. R. Walker, Agile Application-Aware Adaptation for Mobility, in: 16th ACM Symposium on Operating System Principles, ACM, St. Malo, France, 1997.

[31] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, J. Villasenor, Adaptive Mobile Multimedia Networks, IEEE Personal Communications 3 (2) (1996) 34–51.

[32] H. Liu, M. E. Zarki, Adaptive Source Rate Control for Real-time Wireless Video Transmission, Mobile Networks and Applications 3 (1) (1998) 49–60.

[33] R. Kravets, P. Krishnan, Application-driven Power Management for Mobile Communication, Wireless Networks 6 (4) (2000) 263–277.

[34] V. T. Raisinghani, S. Iyer, User Managed Wireless Protocol Stacks, Poster accepted for 23^{rd} International Conference on Distributed Computing Systems (ICDCS) (2003).

[35] J. Inouye, J. Binkley, J. Walpole, Dynamic Network Reconfiguration Support for Mobile Computers, in: ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), Budapest, Hungary, 1997.

[36] M. Zorzi, R. Rao, Perspectives on the Impact of Error Statistics on Protocols for Wireless Networks, IEEE Personal Communications 6 (5) (1999) 32–40.

[37] J. G. Proakis, J. A. Rice, M. Stojanovic, Shallow Water Acoustic Networks, in: Haas [49], pp. 114–119.

[38] Y. Fang, A. McDonald, Cross-layer Performance Effects of Path Coupling in Wireless Ad hoc Networks: Power and Throughput Implications of IEEE 802.11 MAC, in: 21st IEEE International Performance, Computing, and Communications Conference, 2002, pp. 281–290.

[39] V. Bose, D. Wetherall, J. Guttag, Next Century Challenges: Radioactive networks, in: ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), ACM, Seattle, WA, 1999.

[40] D. S. Phatak, T. Goff, A Novel Mechanism for Data Streaming Across Multiple IP Links for Improving Throughput and Reliability in Mobile Environments, in: IEEE INFOCOM, Vol. 21, 2002, pp. 773–781.

[41] L. J. Williams, Technology Advances from Small Unit Operations Situation Awareness System Development, IEEE Personal Communications 8 (2001) 30–33.

[42] E. A. Brewer, R. H. Katz, Y. Chawathe, S. D. Gribble, T. Hodes, G. Nguyen, M. Stemm, T. Henderson, E. Amir, H. Balakrishnan, A. Fox, V. N. Padmanabhan, S. Seshan, A Network Architecture for Heterogeneous Mobile Computing, IEEE Personal Communications 5 (5) (1998) 8 – 24.

[43] C. E. Jones, K. M. Sivalingam, P. Agrawal, J. C. Chen, A Survey of Energy Efficient Network Protocols for Wireless Networks, Wireless Networks 7 (4) (2001) 343–358.

[44] A. Goldsmith, S. Wicker, Design Challenges for Energy-constrained Ad hoc Wireless Networks, IEEE Wireless Communications 9 (4) (2002) 8–27.

[45] V. T. Raisinghani, S. Iyer, Architecture for Cross Layer Feedback in Wireless Protocol Stacks, submitted for Doctoral Symposium at 23^{rd} International Conference on Distributed Computing Systems (ICDCS) (Feb. 2003).

[46] B. Badrinath, A. Fox, L. Kleinrock, G. Popek, P. Reiher, M. Satyanarayanan, A Conceptual Framework for Network and Client Adaptation, Mobile Networks and Applications 5 (4) (2000) 221–231.

[47] C. Barakat, E. Altman, W. Dabbous, On TCP Performance in a Heterogeneous Network: A Survey, IEEE Communications Magazine 38 (1) (2000) 40–46.

[48] R. Cáceres, L. F. Chang, R. Jain (Eds.), Special Issue on Wireless Internet and Intranet Access, Vol. 6 of Mobile Networks and Applications, ACM/Kluwer, 2001.

[49] Z. J. Haas (Ed.), Design Methodologies for Adaptive and Multimedia Networks, Vol. 39 of IEEE Communications Magazine, IEEE Communications Society, 2001.