User managed wireless protocol stacks

Vijay T. Raisinghani
Tata Infotech Ltd. (ATG)
KR School of IT
IIT Bombay
rvijay@it.iitb.ac.in

Sridhar Iyer
KR School of IT
IIT Bombay
sri@it.iitb.ac.in

Abstract

User experience on mobile devices is bound by the device constraints of memory, processing power, battery life and variations in the wireless network. Cross layer feedback in the protocol stack is useful to improve the performance of mobile devices. However, improvement in user experience is bound by the cross layer design decisions.

We propose utilizing dynamic user feedback *to enhance the performance of the stack, as per the user needs. We discuss some examples where such type of feedback will be useful. We also propose an architecture for cross layer feedback.*

1. Introduction

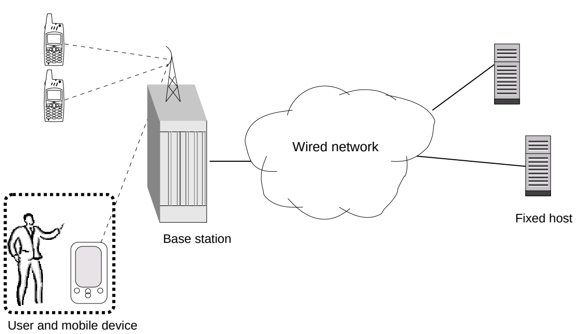

Figure 1. Typical mobile computing scenario

Mobile wireless environments are prone to packet losses, high bit error rates, and mobility induced disconnections. Much research has been done [2, 3, 5] for improving the performance of the protocol stack by using cross layer feedback. While the system based on heuristics, can take certain actions, a user may be able to take *better* decisions, which may be contrary to the system decision. For example: A system on the mobile device may run a real-time application at a higher priority than *ftp*. However, in view of an impending disconnection a user may want the *ftp* to have higher priority. Other examples where user feedback may be useful are: (1) *User authorization for socket creation* - This may help in saving battery power and prevent virus spreads. (2) *User specified acceptance of erroneous data* - User may decide to accept some *data checksum failed* packets. We are investigating the type of errors that can be tolerated. This again may be useful in conserving battery power.

To the best of our knowledge none of the approaches so far talk of bringing *dynamic user feedback* into the protocol stack.

A typical mobile computing set-up is shown in fig. 1. The area of our contribution is highlighted by the dashed box. Our first contribution is incorporation of dynamic user feedback into the protocol stack. Our second contribution is proposing an architecture for cross layer interactions. Both are described in the following sections.

2. User feedback

To validate our idea we are doing some experiments using user feedback for controlling application priority. We report here the results of some simulations. The simulations were done using ns-2 [4]. The simulation set-up is similar to the scenario in fig. 1. The user runs two *ftp* applications on the mobile device, each downloads some file from the fixed host. The simulations were run for 10s. The wireless link bandwidth was set to 2Mbps.

We mapped user specified application priorty to TCP's receiver window. The total receive buffer on the device was assumed constant. This buffer was redistributed amongst applications based on their priority. Fig. 2 shows the case where no user feedback is used. As expected both applications achieve equal throughput. In fig. 3 the effect of user feedback can be seen. In this simulation, the user has assigned application priorities in the ratio 2:1, which is mapped to receiver window allocations in the ratio 2:1. As expected, the higher priority application gets throughput in the ratio 2:1 as compared to the lower priority application. It these simulations it was assumed that there were no losses on the links. We also have done a few simulations with some losses on the links. In these simulations too the higher priority application gets higher throughput. We do not show the graphs here due to space limitations.

The simulation results demonstrate that user feedback can be used to enhance user experience on mobile devices.

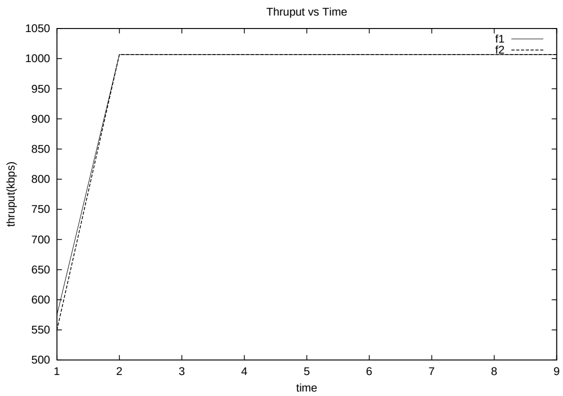

Figure 2. Application throughput without user feedback

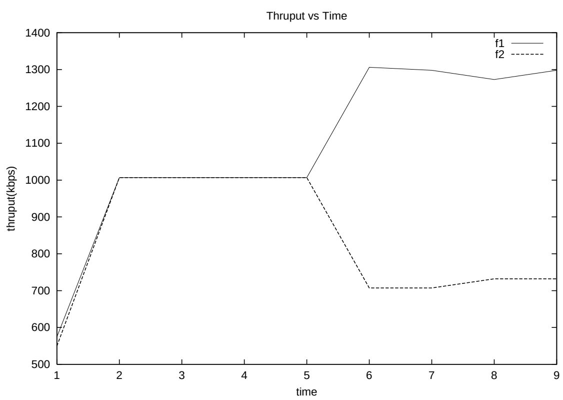

Figure 3. Application throughput with user feedback

User feedback and in general cross layer feedback need to be incorporated into the stack. To the best of our knowledge there is no generic architecture for cross layer feedback. In the next section we present our architecture for cross layer feedback.

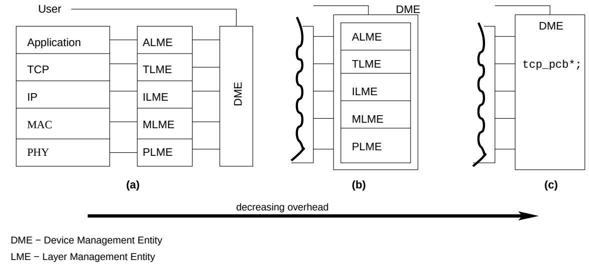

Figure 4. Cross layer architecture

3. Cross layer architecture

Fig. 4 shows the architectures we propose for cross layer feedback. Common to all the cases is a a Device Management Entity(DME) which serves as the protocol stack interface and control entity for the device. Fig. 4(a) shows an independent Layer Management Entity(LME) for each layer. All the communication to/from LMEs is through the DME. The DME may have pointers to the individual LMEs. In fig. 4(b) the LMEs are merged into the DME. Now, the LMEs can communicate directly. Also the DME can directly access the LMEs, rather than through pointers. In fig. 4(c), the DME has direct access to each layer's information. For example, the DME may have direct access to the TCP process control block pointer. The communication overhead between the various management entities may be lower in case of fig. 4(b) as compared to that in fig. 4(a). The overhead may be the least in the case of 4(c).

We are investigating issues like implementation ease, processing time, memory requirements, etc. for all the cases. We shall define the various APIs and implement user feedback using the selected architecture.

References

[1] R. Cáceres, L. F. Chang, and R. Jain, editors. *Special Issue on Wireless Internet and Intranet Access*, volume 6 of *Mobile Networks and Applications*. ACM/Kluwer, 2001.

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

[3] C.-F. Chiasserini and M. Meo. Improving TCP over Wireless through Adaptive Link Layer Setting. In *IEEE GLOBECOM, Symposium on Internet Performance (IPS 2001)*, San Antonio, TX, 22, Nov. 2001.

[4] The network simulator ns-2. `http://www.isi.edu/nsnam/ns/`,.

[5] J. C.-S. Wu, C.-W. Cheng, N.-F. Huang, and G.-K. Ma. Intelligent Handoff for Mobile Wireless Internet. In Cáceres et al. [1].