A timer-based data link control protocol for mobile computing

著者	根元　義章
journal or publication title	IEEE International Conference on Universal Personal Communications, 1998. ICUPC '98.
volume	2
page range	1339-1343
year	1998
URL	http://hdl.handle.net/10097/46912

doi: 10.1109/ICUPC.1998.733710

A Timer-based Data Link Control Protocol For Mobile Computing

Gengyu WEI and Risaburo SATO
Department of Electrical Engineering,
Graduate School, Tohoku Gakuin University

Akihiro FUJII
School of Project Design,
Miyagi University

Yoshiaki NEMOTO
Graduate School of Information Science, Tohoku University

1. Introduction

With the recent advances in portable devices and wireless networks, a new paradigm of computing has been emerged which is know as mobile computing. Studies [1-6] have been reported in literature dealing with various aspects supporting for mobile computers to access the Internet. For example, IETF has developed a complete set of specifications [2] for providing mobility in the Internet, IEEE has recommended IEEE802.11 [1, 3] for using 2.4GHz range radio access technologies, and the Infrared Data Association (IrDA) has proposed a protocol stack [9, 10] to facilitate infrared access techniques. In other existed contributions, handoff [4, 5], lousy wireless link [6], link access method [12], and their effects on the performance of reliable transport layer protocol, e.g. TCP, have been studied.

The IrDA protocols were recommended for mobile computers, such as note PC, PDA, etc. This protocol stack is coming to the industrial standards for infrared access techniques. The IrDA Link Access Protocol, IrLAP [10], is based on HDLC [7] and specified for equipment using a directed half-duplex serial physical communication medium. Nowadays, infrared techniques are often used in the point-to-point indoor communications. The Normal Response Mode (NRM) is adopted, in which a primary station exchanges frames with a secondary station. Due to the half-duplex nature, the sending permission (P/F bit in Command/Response frame of IrLAP, called token in the following) is required to exchange between the primary station and the secondary one.

The mobile computers can be arbitrarily small in size, which means restricted memory resources and limited electrical power capacity. The power consumption of mobile computer in communications has been one of the most important problems to be solved. As the power consumption is proportional to the total sending time, the transmission efficiency in name of saving electrical power could be obtained if the token is conveyed in the I-frame. So, in the case of network idle, each station should hold the token for a period of time so as to reduce the overhead of exchanging token. The previous work [12] approached this problem by scheduling the holding token time at link layer and was contributed to provide an efficient data transfer while reducing the average power consumption in communications.

In this paper, we propose a solution to the problem of interconnecting mobile computer to the legacy networks. IrDA has been suggested IrDA LAN (IrLAN) network architecture [9] that contains access point mode, hosted mode, and ad hoc mode, only the Access Point (AP) mode is considered in this work. The architecture of AP mode is shown in Fig. 1, this mode supports mobile computers to access to the Internet by an Access Point (AP) equipment. AP is a stationary equipment containing Network Interface Card (NIC) and IrDA protocol stack; AP servers as a router (or gateway) to do data transmission. It is cleat that the AP mode is a typical usage including both fixed and mobile equipment. As AP mode expects to be widely deployed, the solution to power consumption for this mode is significant.

Since the current version IrLAP has traderoff between transmission efficiency and power consumption, the transmission performance will be degraded when implementing electrical power consumption control. To overcome this tradeoff, we propose a timer based data link control protocol to be solution specific for the AP mode. The proposed protocol, as a variant of IrLAP, optimizes the power consumption of mobile host and improves the protocol efficiency during connection establishment and communications.

The rest of this paper is organized as follows. Section 2 reviews the problems related to IrDA standards. Section 3 gives the proposed protocol. Section 4 evaluates the end-to-end protocol performance and discusses the results. Finally, Section 5 is the conclusion.

2. Communication Overheads in AP Mode

According to the IrLAP specifications, the discovery procedure and connection establishment procedure are defined, in which media search time is about 500ms. It will take nearly 1.4 seconds for a station to establish a connection using the default speed of 9.6Kbps. After an connection is established, the primary station (AP) is required to exchange the token with the secondary station (mobile computer). Even if mobile computer has no frames to send, it has to send a response frame returning the token in order to maintain the link connectivity. This results in the consumption of electrical power of mobile computer. Suggestions have been proposed to disconnect the connection in order to save the power, however, resumption of the connection will incur 1.4 second delays. This is unexpected for many application running legacy protocols. So the mechanism that can reduce the consumption of electrical power while maintaining the connectivity is required.

In the IrLAP to control sending permission, turn around times of the token were defined. The Maximum Turn Around Time (*Max tat*) is the maximum time that a station can hold the token and the Minimum Turn Around Time (*min tat*) is the time that a station converts its mode

0-7803-5106-1/98/$10.00 © 1998 IEEE

from sending to receiving, or vice verse. The default max tat is 500ms and min tat is 10ms. The IrDA scheme states that the primary station is not required to turn the token out faster than 500ms. Whereas, in the IrLAP recommendations, the primary station can use 100ms *Max tat* to simulate full duplex link, no specific statements are for the secondary station.

Although the long holding token time can alleviate the power consumption in certain degree, it would be problematic for request-response applications to use legacy transport protocol. In this situation, after the secondary station receives some data, its link layer returns RR-response while the data are being processed by upper level protocols. The upper layer response is reluctant to wait 500ms for the next RR-command. The effects of this delay can not be ignored. Although scheduling the holding token time [12] has improvements in performance, it is not a sufficient solution to electrical power consumption since there are still existed many cases that the mobile computer turns out the token without data. It have been understood that the holding token time should be short for obtaining better response performance and that must be long to reduce the communication cost.

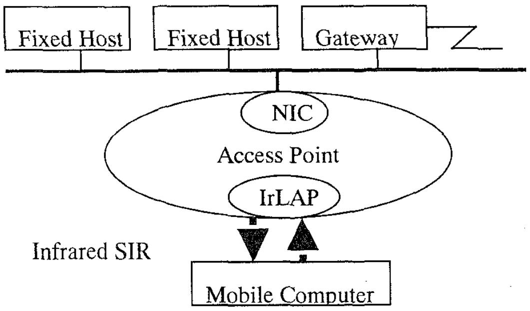

Fig. 1 The AP mode in IrLAN.

On considering the AP mode, we noticed, it is the mobile computer that we should pay our attention to the solution of electrical power consumption, not the AP. The existed IrLAP recommendations try to cover all three modes, short of considerations for each situation. The all-fit-in-one approach has seldom been successful in developing communication protocols, the anticipation of widely deployments of AP mode in IrLAN motivates us to propose a data link control protocol as a solution to electrical power consumption specific for the AP mode.

3. Timer Based Data Link Control Protocol

The proposed protocol can be regarded as enhancements and modifications of IrLAP in the following facets. The first is to define a procedure to detect the presence of mobile computer, the second is to simplify the connection establishment process, and the third is to use timer for maintaining connectivity.

For the AP mode, AP is generally more powerful in processing than the mobile computer. AP is resided in a fixed place and manages the infrared link. IrDA has standardized the communication diameter to be 1m by using infrared link. A problem occurs when AP begins to send Connection Request (CR) message. To explore the nature of infrared medium to facilitate data link control, we define a presence detection procedure for AP to detect whether the mobile computer comes near to it. The technique and mechanism using the infrared light to do detection is the common knowledge.

AP periodically runs the presence detection that needs not any link layer response from mobile computers. If AP finds a mobile computer to be present, it sends CR message with the provider information at default rate. AP continues to send CR messages in a period shorter than presence detection until the mobile computer responds the periodic CR.

Based on our consideration, although the mobile computer serves as a secondary station, it can also initiate connection establishment process. After getting media access parameters from the periodic CR frame, the mobile computer will send a Set Normal Response Mode (SNRM) to start the connection establishment procedure. As only point-to-point communication between AP and the mobile computer is considered here, contentions are not existed. So, to start SNRM from the mobile computer significantly decreases the connection establishment time. The first thing during this stage is to make an agreement on transmission speed at which information frames are conveyed.

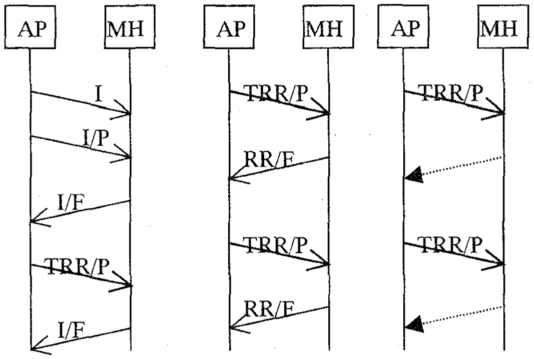

Fig. 2 Cases of exchanging tokens

After the connection is established, AP exchanges frames with the mobile computer by obeying the procedures defined in IrLAP. The AP turns out the token by a command of I- or RR-frame with P=1; the mobile computer returns a response of I- or RR-frame with F=1. It is required in the IrLAP recommendations that the response from mobile computer must respond the command within the holding token time. The process to deliver command and response frames are depicted in Fig. 2 (a). This command and response frame-based mechanism to exchange sending permission results in overheads of consumption of electrical power. We introduce a timer-based mechanism to exchange sending permission into AP mode. A new command frame is defined, named as Token exchange RR (TRR) command frame. TRR command is used for AP to pass the token to the mobile computer. Whereas, the TRR command, unlike the RR-command, requires an optional response from the mobile computer. The procedures using TRR command are depicted in Fig. 2.

In the following we explain how the TRR command works and reduces the consumption of electrical power for mobile computer. When there are data, the token is conveyed in I-frames. However, if AP has not any I-frames to be sent, it will turn out the token by the TRR command. AP delivers the TRR command each time when the holding token time expires and then converts from sending mode to receiving mode. If AP receives a RR-response after delivery of TRR, AP and mobile computer exchange token following a normal process. If AP does not receive any frames within the mobile holding token interval, it returns to the sending mode automatically. In this case, AP interprets no response as that the mobile computer has no frame to send.

On the other hand, the mobile computer has three choices to react to the TRR command. a) It sends an I-frame with F=1 if there are any frames waiting in the buffer. b) It delivers a RR-frame with F=1 if it prefers to return the token by a response frame. c) It is just keeping silent if it has no frames to send during the holding token time. In the case of (c), we say that mobile computer works in a saving electrical power mode. It is easily understood that if using the TRR command, the mobile computer will not consume its electrical power to turn out the token when the network is idle. In this way, the overheads of electrical power consumption to send RR-response are reduced and even omitted.

Whereas, we do not intend to violate the existing handshake mechanisms defined in IrLAP. To guarantee the original command and response mechanism still works as well, the mobile computer is required to response every command frames except for TRR. So the mobile computer can use the RR-response to respond the RR-command. The TRR command functions only if the mobile computer understands the semantics of saving electrical power mechanism. Since the mobile computer is not obligatory to response each TRR, AP has to use the existed RR handshake mechanism to tackle exception situations. AP will send a RR-frame in the following cases: 1) no frames are from mobile computer for a long time; 2) the mobile computer is likely to be absent. AP will disconnect the link if the mobile does not respond RR-commands.

We restate that the presence detection is still being carried out during communications. The period to detect should be shorter than that in initial phase. The presence detection facilitates to determine whether to send TRR in communication stage. When the mobile computer is in AP's communication range, it is detectable. So the TRR is available to work. Otherwise, AP can not detect it when the mobile computer moves out of communication diameter. And if AP gets no response for RR-command, AP can decide that the link was broken.

The protocol semantic is based on the fact that transmissions between AP and the mobile computer is reliable if the mobile computer is in the AP's communication range. So any transmission of command and response would be mutually arrived. The mobile computer is in saving electrical power mode if it does not deliver response to TRR command. And no response just means that the mobile computer has not any frames to send, it does not mean that the link is broken. To improve the response performance, we prefer that AP should frequently send TRR frames when the network is idle. Although IrLAP suggested the 100ms holding token time to simulate full-duplex link, it will not guarantee the optimal performance for the legacy transport. However, if the timer-based mechanism to exchange the token is adopted, TRR command can be transferred more frequent than 100ms, improvements of transport layer protocol performance will be apparent.

4. Performance and Discussion

To evaluate the proposed protocol while addressing link layer performance, we firstly briefly analyze the link layer performance affected by the parameters. Then we mainly concern how it affects the transport layer protocol because the end-to-end TCP performance is what the application can be obtained. The reason why to evaluate both link layer and transport layer protocol is because the enhancement at link layer is not certainty to make the transport protocol performance optimal.

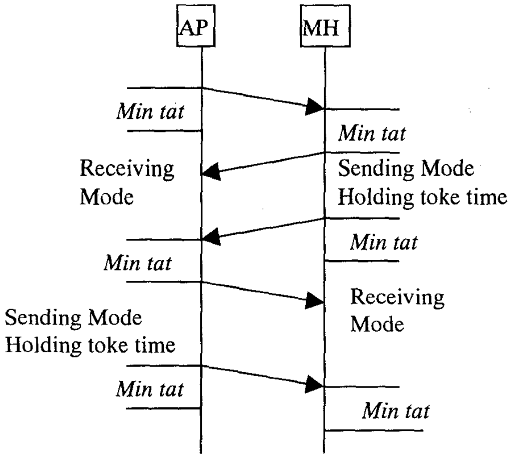

Fig. 3 Relations of link layer parameters.

In order to understand how link layer parameters affect the protocol performance, the relations of these parameters are shown in Fig. 3. It is easy to see that the link layer throughput can be expressed as a function of the holding token time (*Max tat*), *min tat*, and propagation delays. Among them, the propagation delays can be ignored since it is about 3.3ns (the standard distance between AP and the mobile computer is only 1m). If *min tat* is constant, the transmission efficiency is proportional to the holding token time. It seems that the long holding token time is required so as to get better link layer performance. However, we have to say that simply adopting long holding token time will result in poor TCP performance since the long holding token time postpones acknowledgements.

We mainly concern the obtained TCP performance on the proposed link layer mechanism. The IrDA protocols in AP were simplified, IrLAP is treated the same as PPP and SLIP, TCP/IP is directly run on IrLAP. And because Web-access services is regarded as the main service in mobile computing, we assume that the mobile computer runs Web accesses on TCP/IP. The mobile computer communicates with a proxy server resided in the same network segment to access to the Internet. Since such an application is of request-response, a burst of data will be transferred from the server to the mobile host time by time. The networks are idle when there are not any frames containing data or acknowledgement along the half-duplex link and only the token is exchanged between AP and the mobile host. According to the proposed protocol, AP sends TRR faster than 100ms and the mobile computer needs not to response each TRR command.

To understand the effects of proposed data link control protocol on the TCP performance in quality and quantity, we run simulation of TCP over a half-duplex link. The simulation configuration is shown in Fig. 4. This represents for the mobile/wireless LAN shown in Fig. 1. To run the simulation, we modified a version of REAL, a network simulator [11], and added half-duplex link supports. During the simulation, parameters related to TCP are set to obey RFC2001 [8]. The duration of simulation run is 10 seconds. The performance metric is the TCP throughput measured at the destination TCP layer, the throughput is defined as the total number of bytes delivered to the destination application divided by the simulation time. The results are reported in Mbps.

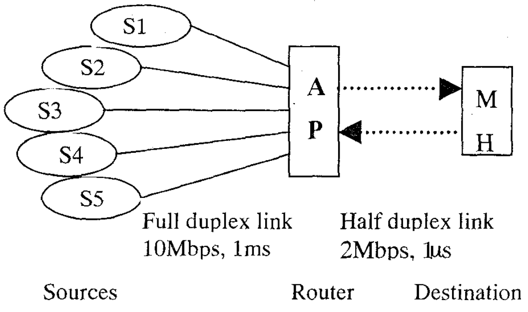

Fig. 4 Simulation model.

The simulation results are shown in the following figures. About one TCP connection situation, Fig. 5 and Fig. 6 presents how the holding token time affects the TCP performance. That time varies from 10ms to 50ms. Fig. 5 shows the TCP *tahoe* performance and Fig. 6 shows the TCP *reno* performance.

The results show that *tahoe* monotonically increases the throughput as long as AP increases the buffer size, and that *tahoe* can provide high throughput with much more buffers. On the other hand, *reno* seems to be more sensitive to parameter changes. The increment of throughput is not determined with increasing the AP buffer size. In other word, large buffer size would not definitely guarantee *reno* with high throughput if it runs over a half duplex link. So, the gradual increment of throughput means that *tahoe* is stable and robustness in terms of performance according to the simulation results in this work.

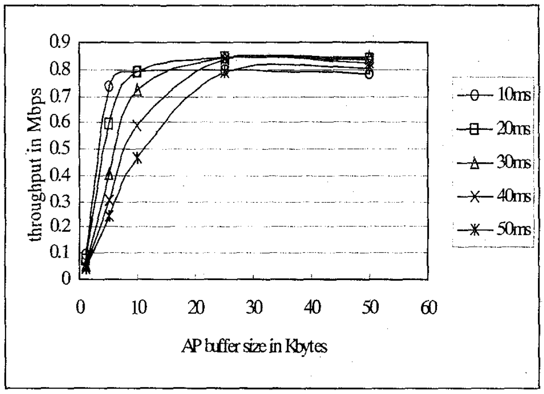

Fig. 5 TCP *tahoe* over symmetric schduling.

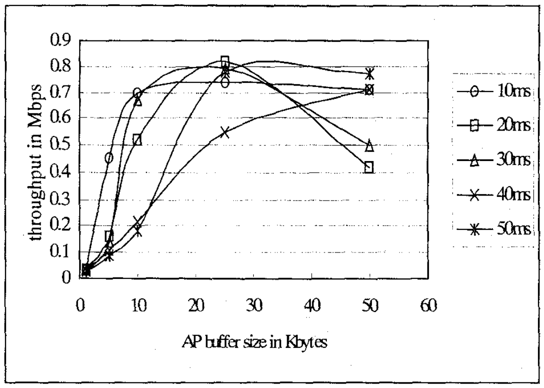

Fig. 6 TCP *reno* over symmetric scheduling.

Fig. 7 and Fig. 8 presents the TCP performance over the proposed timer-based data link control protocol. Fig. 7 shows the TCP *tahoe* performance and Fig. 8 does the TCP *reno* performance. By comparing Fig. 7 and Fig 8, it is easy to get the same result as comparison of Fig. 5 and Fig. 6. However, the proposed data link control protocol can guarantee TCP with higher throughput if comparing Fig. 7 with Fig. 5. (Notice the y-axile scales). The results in Fig. 7 shows that the timer-based scheme significantly improves the TCP *tahoe* performance by scheduling the holding token time algorithm if compared with the results in Fig. 6. Whereas, improvements of throughputs for *reno* were also obtained although they were not so cleanly and determined as *tahoe*.

At last, multiple TCP connections were simulated, the results are shown in Fig. 9. In this case, five connections simultaneously start transmission to the same destination. It is easy to see that the total throughput is identical to the simuation when only single connection is existed.

Seeing the simulation results, we understood that the link layer control mechanisms are a critical factor to guarantee TCP with optimal throughputs. A moderate TCP throughput could be obtained only if well implementation of scheduling token algorithm, because the shortest the holding token time does not always provide the highest throughput. Fig. 5 shows that when buffer size is larger than 25K bytes, *tahoe*'s throughput of 10ms holding token time is less than those of 20ms, 30ms, and 40ms; while a longer holding token time means fewer communication overheads. However, results in Fig. 7 showed that, the timer based data link control protocol greatly improves TCP performance in context of throughput and communication cost even if the token is frequent exchanged.

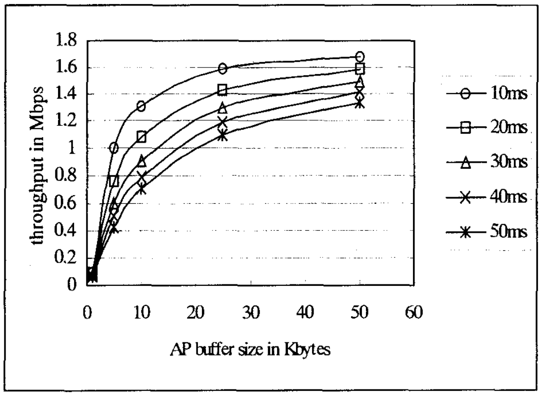

Fig.7 TCP *tahoe* over asymmetric scheduling.

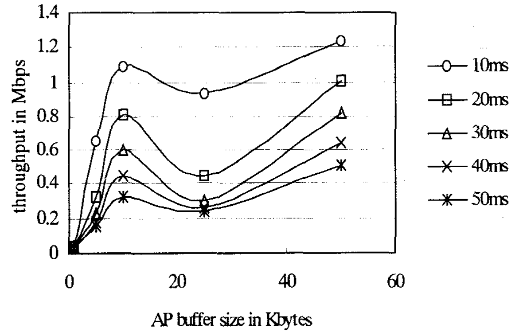

Fig. 8 TCP *reno* over asymmetric sheduling.

But, it is not so clearly for *reno*. As all figures were presented by throughput vs. AP buffer size, it means that the TCP performance rely on both link control and network resources. As a major result of this work, the timer based link control is significant to improve TCP performance.

5. Conclusion

We have proposed a timer based link layer protocol to alleviate power consumption in mobile computing environments. The proposed protocol introduced AP presence detection procedure and timer-based mechanism to exchange the token. With these mechanisms and newly defined TRR command, AP can maintains the link connectivity while saving electrical power of mobile computer. The proposed protocol can improve the TCP performance without additional power consumption for the mobile computer. We believe that, deployment of proposed protocol will improve the transmission efficiency for mobile computer.

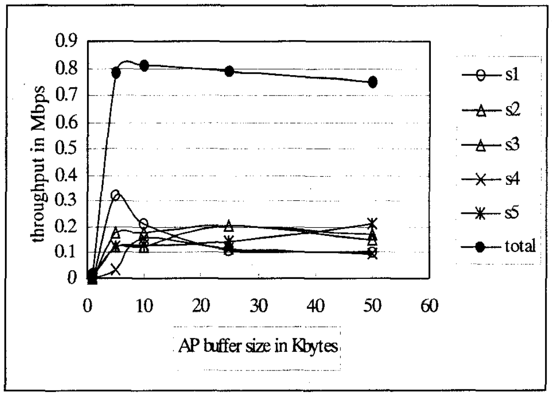

Fig. 9 The Performance of multiple TCP *tahoe* connections.

References

[1] M. Tsukamoto, "Communication Network Infrastructure for Mobile Computing", IEICE Mag., vol. 80, no. 4, pp. 338-343, 1997.

[2] C. Perkins, "IP Mobility Support", RFC2002, Oct. 1996.

[3] Vic. Hayes, "Tutorial on 802.11 to 802", Mar. 1996

[4] R. Caceres and L. Iftode, "Improving the performance of Reliable Transport Protocols in Mobile Computing Environments", IEEE JSAC., vol. 13, no. 5, pp. 850-857, June 1995.

[5] P. Manzoni and D. Ghosal, "Impact of Mobility on TCP/IP: An Integrated Performance Study", IEEE JSAC., vol. 13, no. 5, pp. 858-867, June 1995.

[6] H. Balakrishnan, et al., "A Comparison of Mechanism for Improving TCP Performance over Wireless Links", proc. ACM SIGCOM'96, Aug. 1996.

[7] C. A. Sunshine, "Computer Network Architectures and Protocols", second edition, Plenum, 1989.

[8] W. Stevens, "TCP slow start, congestion control, fast retransmit, fast recovery", RFC2001, Jan. 1997.

[9] D. Axtman, A. Ogus, and J. Reily, "LAN Access Extensions for Link Management Protoocol, IrLAN", Infrared Data Association, Jan. 1997.

[10] T. William, et al., "Serial Infrared Link Access Protocol (IrLAP) ver. 1.1", Infrared Data Association, Jun. 1996.

[11] S. Keshav, "REAL: A Network Simulator", Computer Science Department Technical Report 88/472, UC Berkeley, Dec. 1988.

[12] G. Wei, et al., "Study on Performance of TCP over Half-duplex Link in Mobile Computing Environments", Technical Report, vol. 33, Tohoku Gakuin University, 1998.