Wireless Access to Internet via Bluetooth: Performance Evaluation of the EDC Scheduling Algorithm

Abstract

In [10], we have extensively studied the EDC protocol, proving its correctness. In [11], the algorithm was also compared by studying its performance in terms of delay and overhead.

Keywords

Wireless, EDC, Scheduling, Medium Access Control

1. Introduction

Networks require a higher level of security, especially for the exchange of sensitive information. In [12], a novel approach is proposed, which enables a secure communication between two devices. The approach is based on the use of a specific protocol that ensures the confidentiality of the transmitted data. This protocol is designed to provide a secure communication channel, allowing two devices to exchange information securely.

2. Related Work

Several studies have been conducted to evaluate the performance of the EDC protocol. In [13], the protocol was compared with other existing protocols, and it was found to be more efficient in terms of delay and overhead. In [14], the protocol was evaluated in terms of its security properties, and it was found to be highly secure.

3. Performance Evaluation

The performance of the EDC protocol was evaluated using simulation tools. The results showed that the protocol is able to provide a secure communication channel, with low delay and overhead. The protocol was also found to be robust against various types of attacks.

4. Conclusion

The EDC protocol is a promising solution for secure communication. The results of our evaluation show that the protocol is able to provide a secure communication channel, with low delay and overhead. The protocol is also robust against various types of attacks. We recommend further research in this area.
The 802.11 MAC layer is responsible for the MAC protocol and is intended to provide a wireless MAC layer that is compatible with the 802.3 Ethernet. The 802.11 MAC protocol is based on the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol used by Ethernet. However, because of the differences between wired and wireless networks, the 802.11 MAC protocol must be modified to accommodate the unique characteristics of wireless communication.

The 802.11 MAC protocol uses a distributed coordination function (DCF) to control access to the wireless medium. The DCF is similar to the CSMA/CD protocol used by Ethernet, but it is modified to account for the additional delays and overhead associated with wireless transmission.

In the DCF, each station listens to the channel to determine if it is free before transmitting. If the channel is free, the station waits a random period of time before transmitting. If the channel is busy, the station waits a longer random period of time before trying again. This process is repeated until the channel is free, at which time the station can transmit.

The DCF provides a simple and efficient method for controlling access to the wireless medium. It is designed to work well in a wide range of environments, including highly congested networks. However, it is not very efficient in terms of throughput, since each station must wait a random period of time before transmitting.

To improve efficiency, the 802.11 MAC protocol also supports a point coordination function (PCF) that allows for polled access to the network. In this mode, the access point (AP) polls each station to determine if it has data to transmit. If the AP determines that a station has data, it allows the station to transmit. This mode can be used to provide higher throughput in networks with low contention.

In addition to the DCF and PCF, the 802.11 MAC protocol also supports a hybrid coordination function (HCF) that combines elements of both DCF and PCF. The HCF allows for both polled and non-polled access to the network, depending on the needs of the network.

The 802.11 MAC protocol also includes a number of other features, such as support for roaming between access points, support for Quality of Service (QoS), and support for encryption to provide security.

The 802.11 MAC protocol is used by a wide range of wireless LAN products, including access points, wireless adapters, and wireless routers. It is also supported by a number of operating systems and software applications.

The 802.11 MAC protocol is specified in the IEEE 802.11 standard, which is a series of standards for wireless local area networks (WLANs). The first version of the IEEE 802.11 standard was published in 1990, and it has since undergone several revisions.

The 802.11 MAC protocol is used in a wide range of applications, including wireless LANs in homes and offices, wireless networks in hospitals and schools, and wireless networks in transportation systems.

In the future, the 802.11 MAC protocol is expected to continue to evolve to meet the needs of new applications and technologies. The IEEE 802.11 Working Group is currently working on a number of new standards, including IEEE 802.11ax, which is expected to provide higher throughput and improved security.
3.1 EDC specification

A detailed EDC specification is given in [10]. Due to the space constraints, we provide only an EDC specification for the downlink (uplink) packet. A detailed explanation of the EDC specification is given in [10].

4. Simulation Model and Performance Evaluation

The network model consists of a single pipeline classifier that classifies data packets into one of several classes. The classifier is instantiated at the beginning of each cycle. The classifier output is used to determine the order in which the packets are sent to the network layer. The network layer receives the packets and determines the order in which they are sent to the MAC layer. The MAC layer receives the packets and determines the order in which they are sent to the physical layer.

**Figure 1: Protocol stack**

The protocol stack consists of the following layers:

- Protocol stacks
- MAC layer
- IP layer
- TCP layer
- L2CAP layer
- LLC layer
- MAC layer
- Physical layer

**Figure 1. Protocol stack**

The traffic sources are generic TCP and UDP packet sizes. The packets are sent to the network layer, which adds its header and queues the packets for transmission. The MAC layer receives the packets and determines the order in which they are sent to the physical layer. The physical layer receives the packets and determines the order in which they are sent to the physical medium.
3. If it is larger than the size of a slot-based packet, send it at a 1-slot-based packet.
4. All the packets are sent by the S3 scheduler.

5. The TCP sources used in Scenario A are:

- TCP 1 with traffic as shown in Figure 2.
- TCP 2 with traffic as shown in Figure 2.
- TCP 3 with traffic as shown in Figure 2.
- TCP 4 with traffic as shown in Figure 2.

Figure 3: TCP throughput of Slave and Master

Figure 4: UDP throughput of Slave and Master

Figure 5: TCP and UDP throughput of Slave and Master for different scheduling algorithms.

Figure 6: TCP and UDP throughput of Slave and Master for different scheduling algorithms.

4.1 Numerical comparison between EDC and RR behavior

We consider sources with different sources of activity to capture the dynamic behavior of the scheduling algorithm. We analyze the performance of the algorithm in the presence of multiple sources of activity. We also analyze the performance of the algorithm in the absence of multiple sources of activity. We consider two scenarios: (1) with no sources of activity and (2) with multiple sources of activity.

In Scenario A, the sources are characterized by their arrival rate and their size. The arrival rate is kept constant at 1 packet per second. The size of the packet is varied from 100 bytes to 1000 bytes.

In Scenario B, the sources are characterized by their arrival rate and their size. The arrival rate is varied from 1 packet per second to 10 packets per second. The size of the packet is kept constant at 1000 bytes.

In Scenario C, the sources are characterized by their arrival rate and their size. The arrival rate is varied from 1 packet per second to 10 packets per second. The size of the packet is varied from 100 bytes to 1000 bytes.

Results show that the EDC algorithm provides a better utilization of resources compared to the RR algorithm. The EDC algorithm is able to adapt to the changes in the traffic pattern and provide a higher throughput.

Figure 7: Comparison of EDC and RR throughput in Scenario A.

Figure 8: Comparison of EDC and RR throughput in Scenario B.

Figure 9: Comparison of EDC and RR throughput in Scenario C.
The effect of maximum segment size over 4.2

Figure 5. TCP throughput vs MSS

Figure 6. TCP throughput of Slave connection with respect to TCP connection

Figure 7. Performance of EDC vs other connection
able to adapt its polling frequency to the traffic characteristics when a source alternates periods of activity with periods of inactivity. Hence, the analysis with asymptotic TCP flows is shown to be adequate with the older MSSs.

In conclusion, we can assess that TCP connections with large MSSs are more sensitive to loss than those with a large number of packets; while a large number of packets requires to introduce a fair behavior of the scheduler. Also towards TCP connections with different MSSs, the effect of the ARQ scheme adopted in Bluetooth over TCP behavior is investigated. The effectiveness of the unnumbered ARQ scheme adopted by the Bluetooth MAC layer to hide the transport layer's errors is shown to be significant when the error rate is constant, but less so when the error rate is time-varying.

The error rate is shown to be significantly higher than in the other cases when the Goodput is equal to 2×10^{-6}, and the BRR in the Bad state equal to 10^{-4}. We also assume that the BRR in the Bad state is expressed as the probability of time slots in which both the BRR remains constant a during the packet transmission. We have also considered a distributed error rate, in this case, the BRR has been chosen equal to the average bit error rate experienced in the worst-case model. This model is used to investigate the impact of error recovery mechanism over the TCP both in the presence of burst of errors, and uniformly distributed errors.

It is worth pointing out that the main objective of this section is to compare the performance of burst loss on the wireless channel and the error recovery mechanism. Our characterization is not yet available, and the experimental values measured in the paper are obtained from the same fragment, say if

\[ A \cdot \sin B \cdot \cos C \neq 0 \]

Figure 10. TCP packet loss probability with MSS-132

Table 10. TCP packet loss probability with MSS-132

Table 9. TCP throughput of connections with different MSSs

Figure 9. TCP throughput of connections with different MSSs

Figure 8. TCP throughput of connections with different MSSs

Figure 7. TCP throughput of connections with different MSSs

Figure 6. TCP throughput of connections with different MSSs
6. Conclusions and Future Work

In this paper we have investigated the importance of the MSS parameter in determining the TCP throughput. In particular, we have shown that different MSS values can significantly improve the fairness of TCP flows with different congestion windows. In this work we have only considered sources placed in different订购 at any point in the Internet. The impact of the delays introduced when the TCP flows cross the real Internet network will be studied in future works.

Figure 11 shows the same experiments as in Figure 10, but with a new scheduling algorithm called EDC. EDC significantly improves the throughput of TCP connections when compared to a RR scheduler.

Figure 11. TCP packet loss probability with MS-315 layers.