# Principles of Reliable Data Transfer

WE'LL COVER THE FOLLOWING

## Network Layer Imperfections #

The transport layer must deal with the imperfections of the network layer service. There are three types of imperfections that must be considered by the transport layer:

1. Segments can be **corrupted** by transmission errors
2. Segments can be **lost**
3. Segments can be **reordered** or **duplicated**

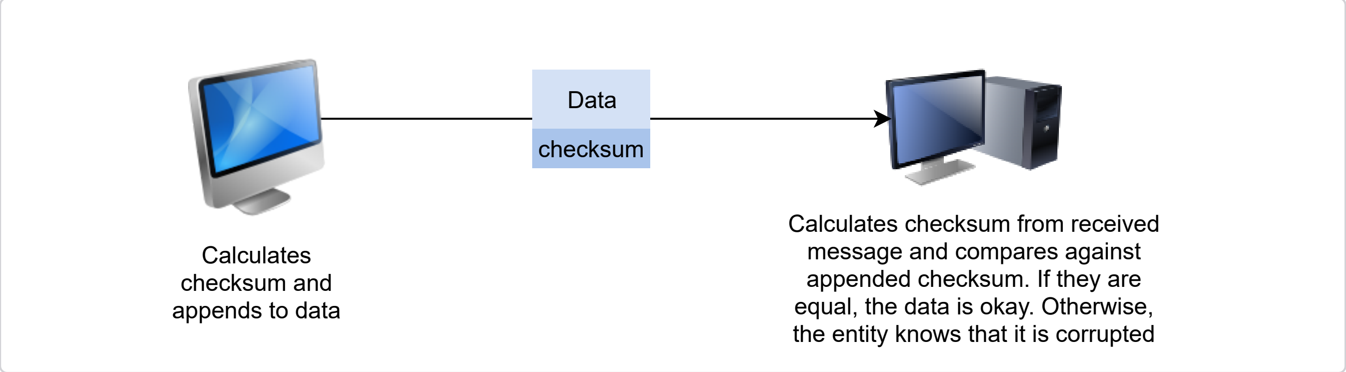
Let's look at some workarounds for these problems that the transport layer employs.

## Checksums #

The first imperfection of the network layer is that segments **may be corrupted by transmission errors**. The simplest error detection scheme is the **checksum**.

A checksum can be based on a number of schemes. One possible scheme is an arithmetic sum of all the bytes of a segment. Checksums are computed by the sender and attached with the segment. The receiver verifies it upon reception

and can choose what to do in case it is not valid. Quite often, the segments received with an invalid checksum are **discarded**.

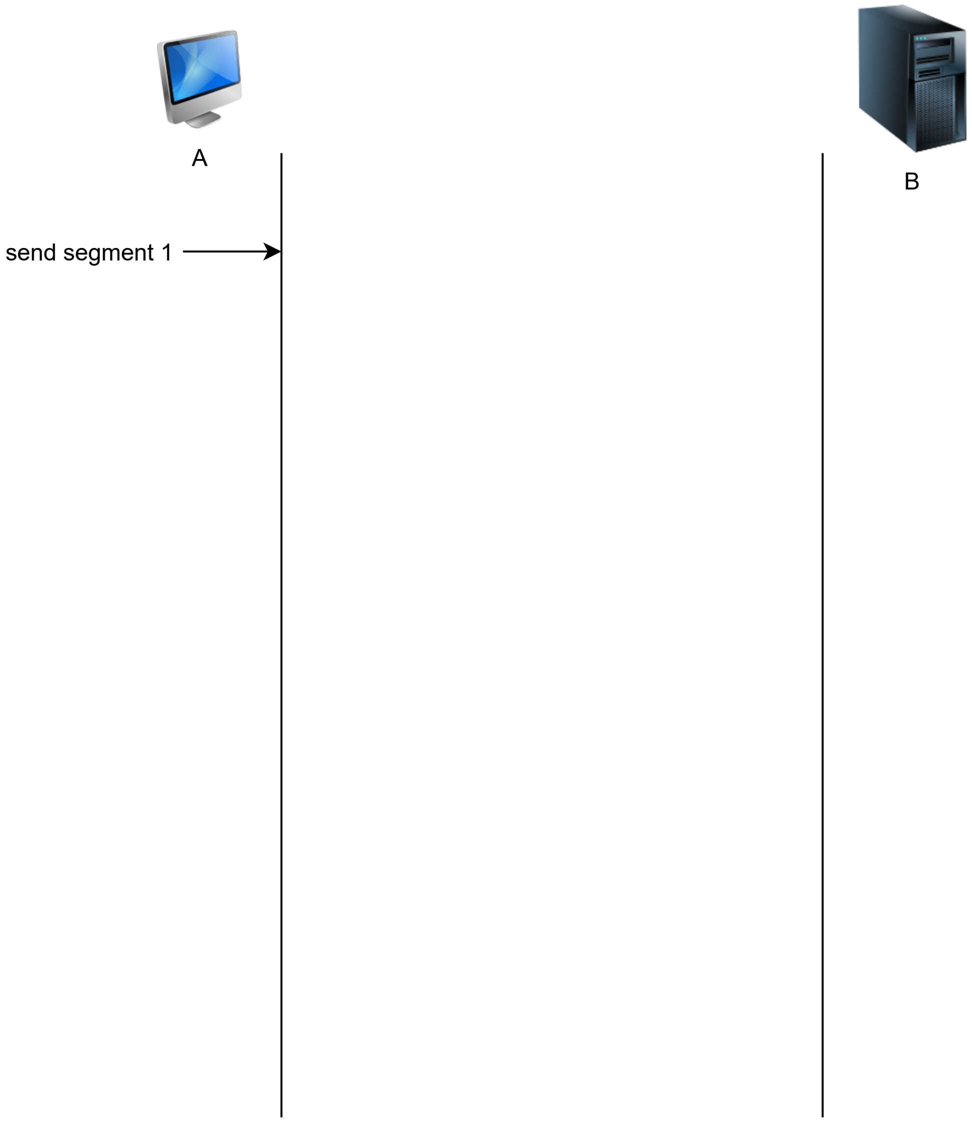

Data
checksum
Calculates checksum and appends to data
Calculates checksum from received message and compares against appended checksum. If they are equal, the data is okay. Otherwise, the entity knows that it is corrupted

## Retransmission Timers #

The second imperfection of the network layer is that **segments may be lost**. Since the receiver sends an acknowledgment segment after having received each data segment, the simplest solution to deal with losses is to use a **retransmission timer**.

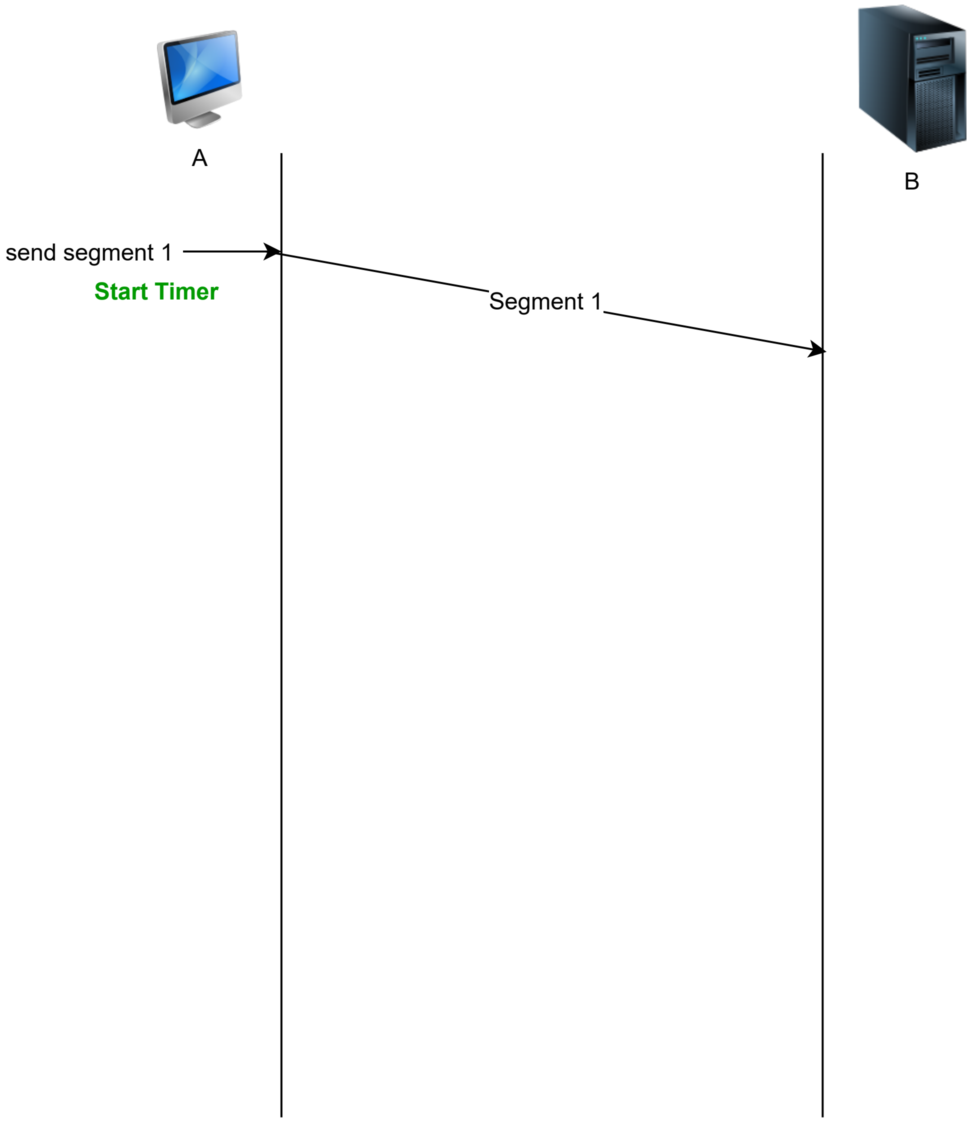
A retransmission timer starts when the sender sends a segment. The value of this retransmission timer should be *greater* than the **round-trip-time**, for example, the delay between the transmission of a data segment and the reception of the corresponding acknowledgment. Note that TCP sends an acknowledgment for almost every segment! We'll look at this in more detail in later lessons. When the retransmission timer expires, the sender assumes that the data segment has been lost and retransmits it.

This is illustrated in the figure below:



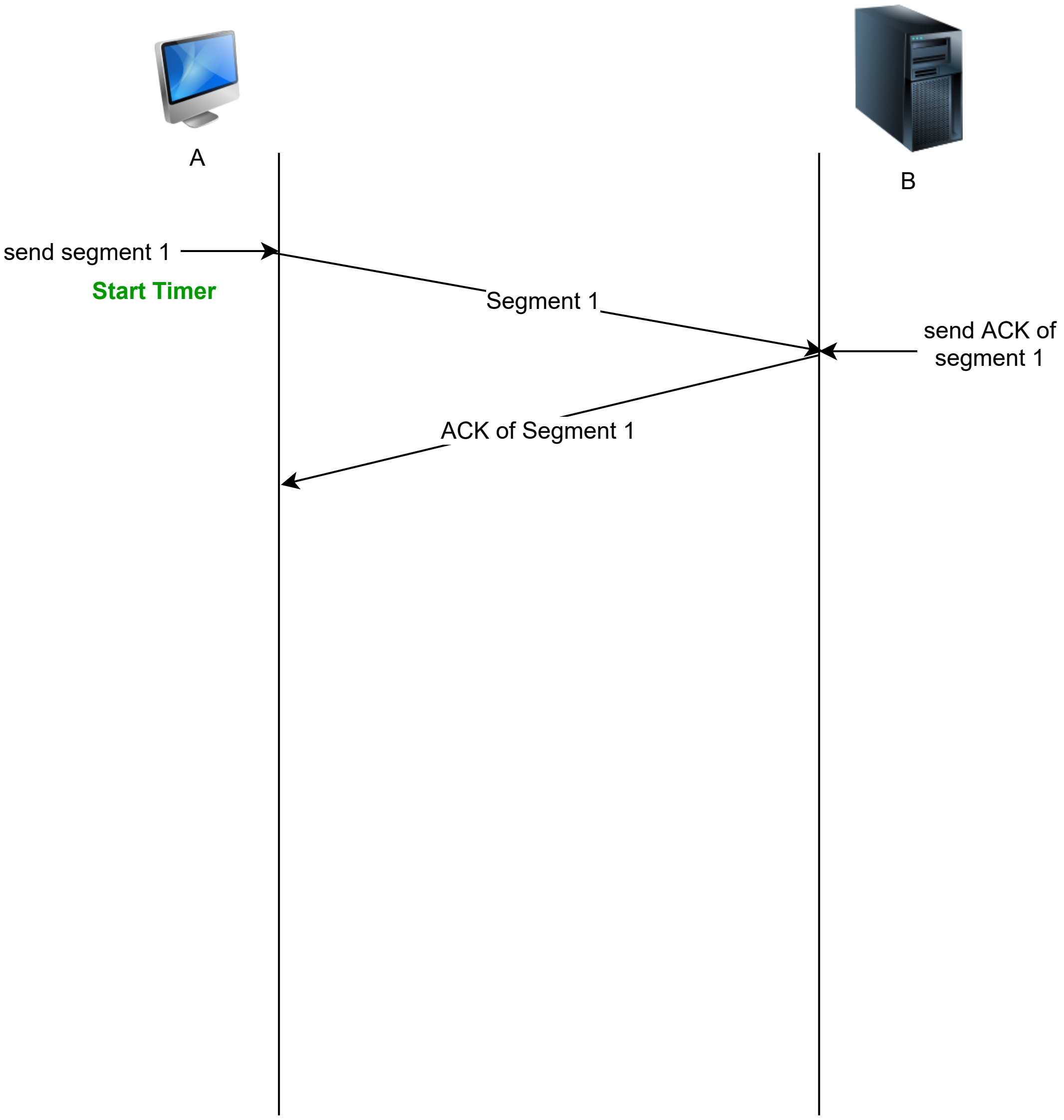
Using retransmission timers to recover from segment losses

**1** of 7



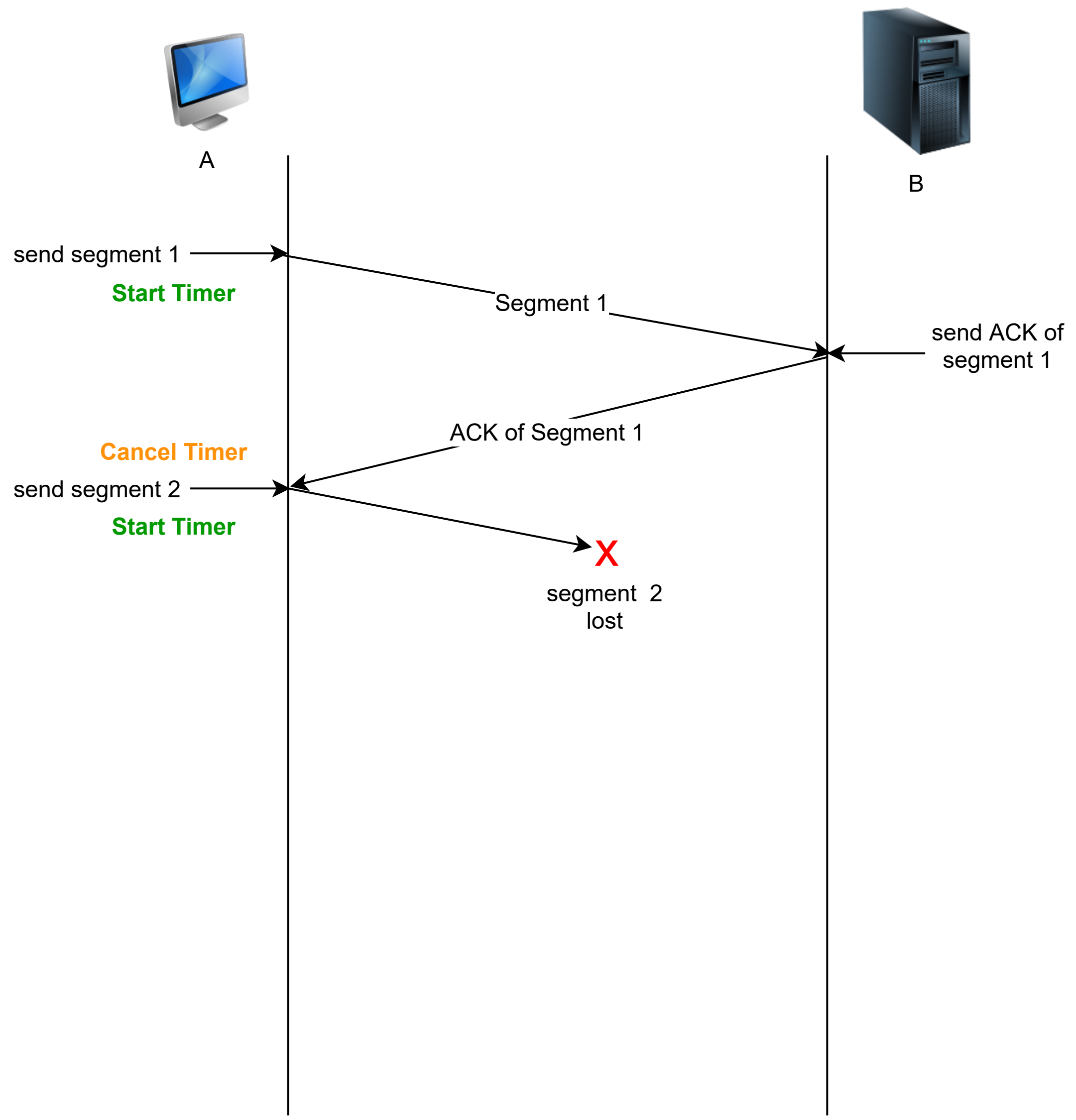
Using retransmission timers to recover from segment losses

**2** of 7

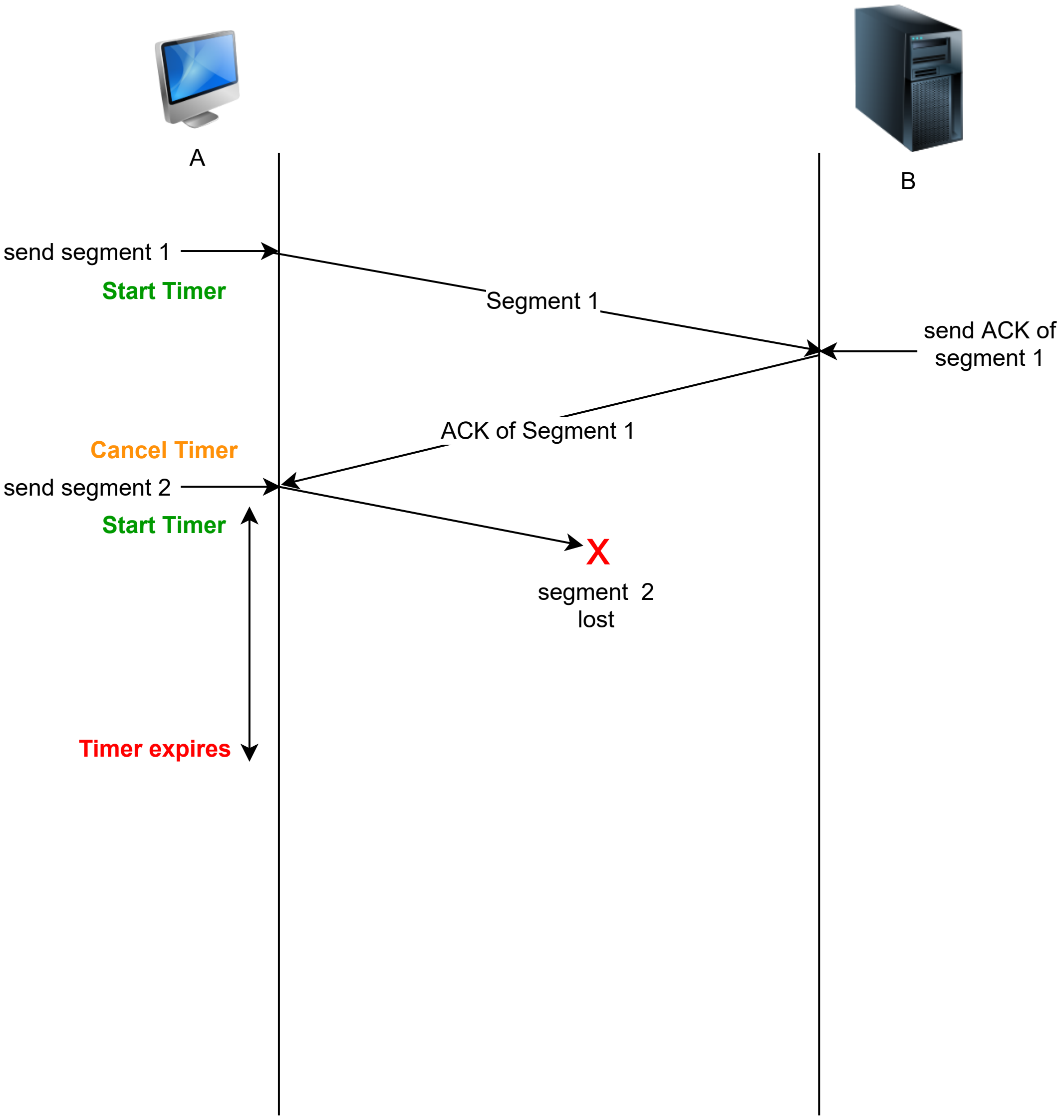


Using retransmission timers to recover from segment losses

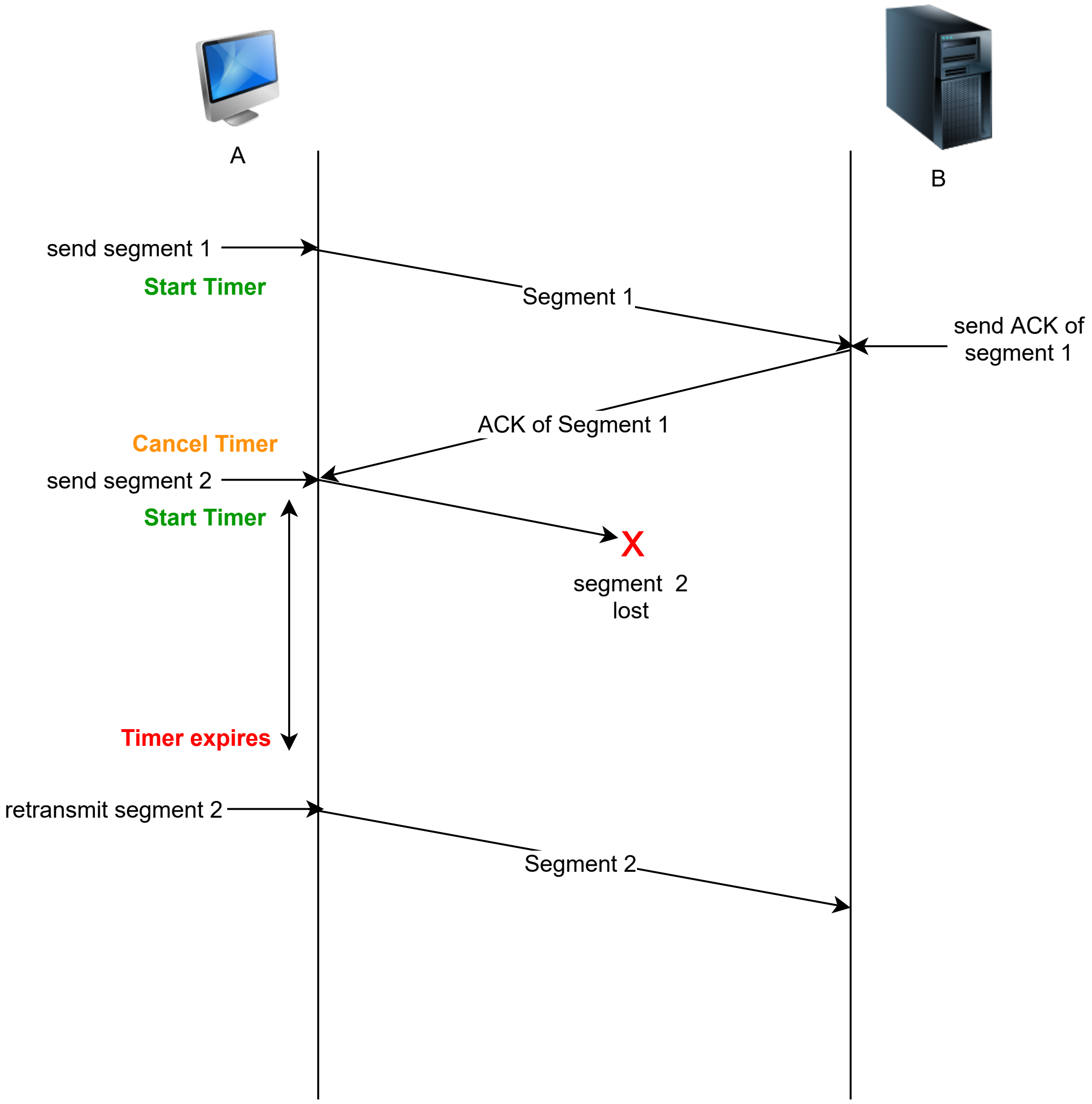
3 of 7



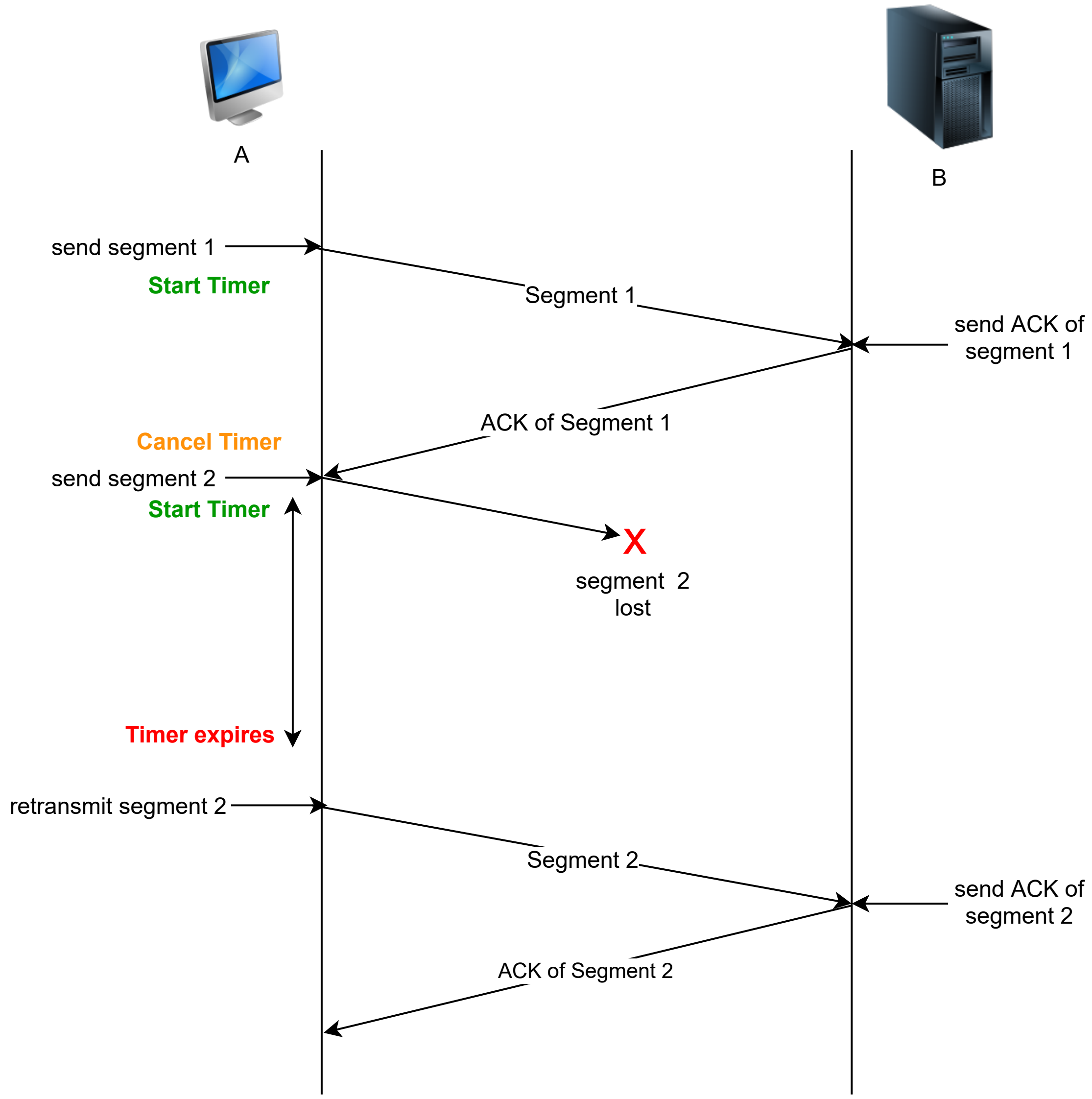
Using retransmission timers to recover from segment losses



Using retransmission timers to recover from segment losses



Using retransmission timers to recover from segment losses



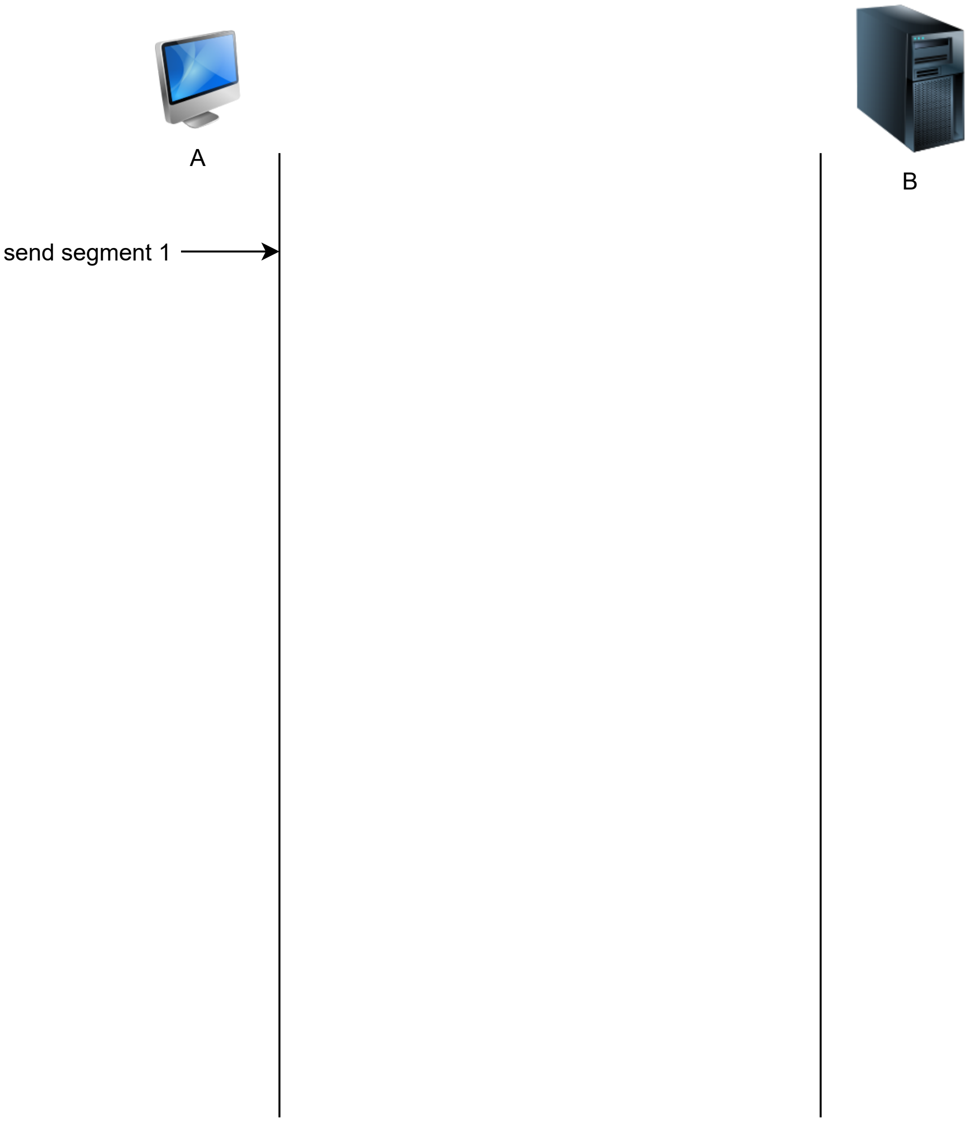
Using retransmission timers to recover from segment losses

**7** of 7

## Limitations of Retransmission Timers

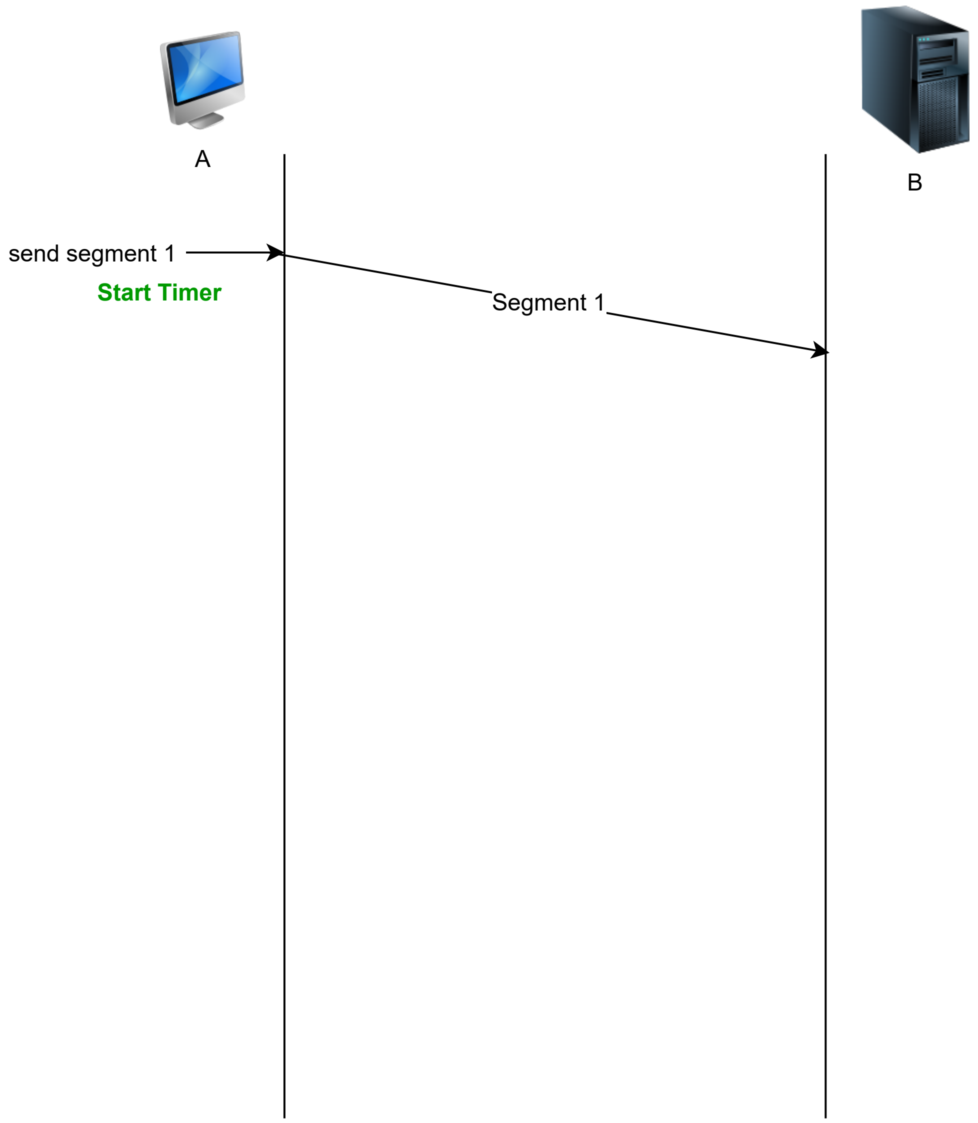
Unfortunately, retransmission timers alone are not sufficient to recover from segment losses. Let us consider the situation depicted below where **an acknowledgment is lost.** In this case, the sender retransmits a data segment that has been received correctly, but not properly acknowledged.

Unfortunately, as illustrated in the figure below, the **receiver considers the retransmission as a new segment** effectively and the segment is *duplicated*.

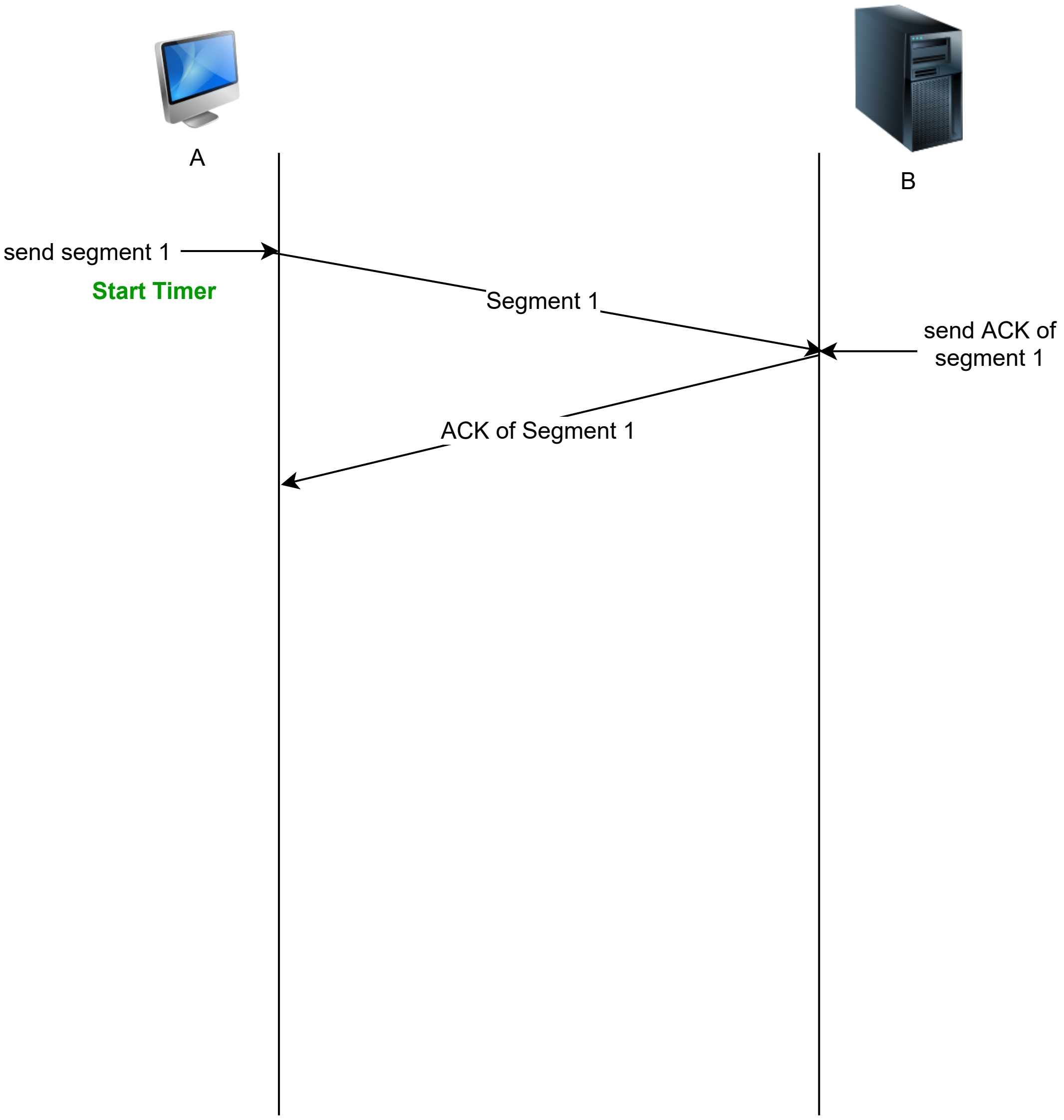


Using retransmission timers to recover from segment losses

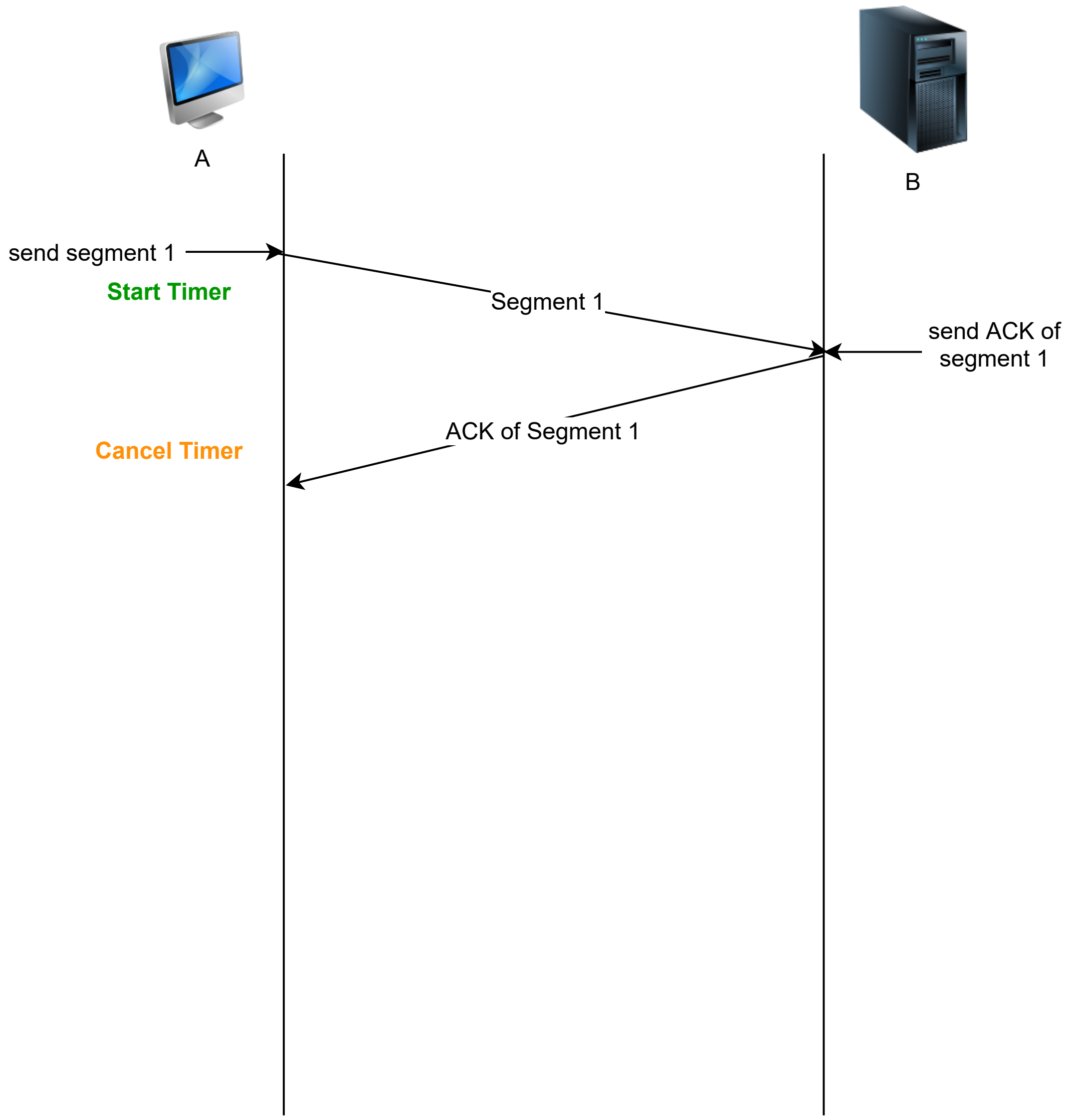
**1** of 7



Using retransmission timers to recover from segment losses

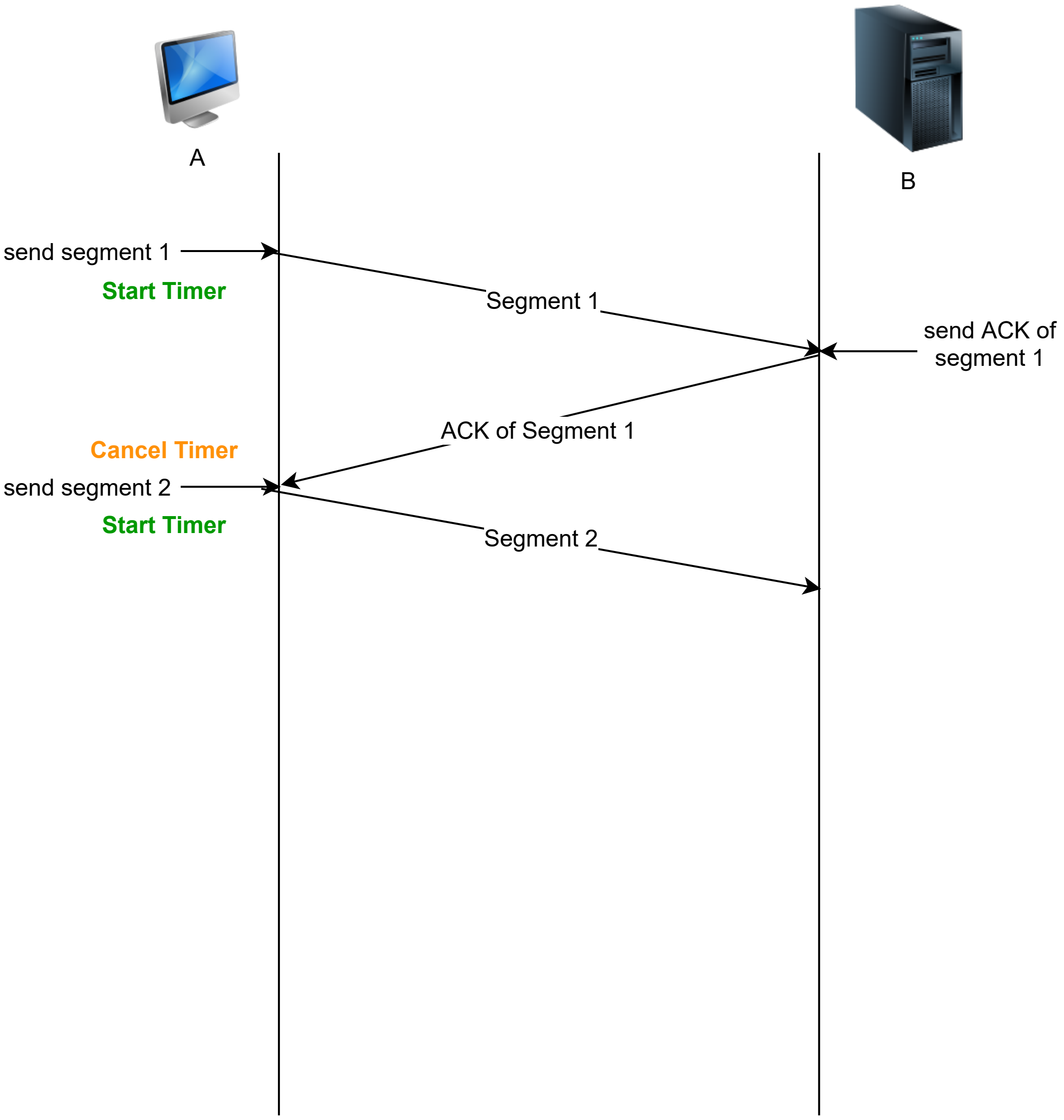


Using retransmission timers to recover from segment losses



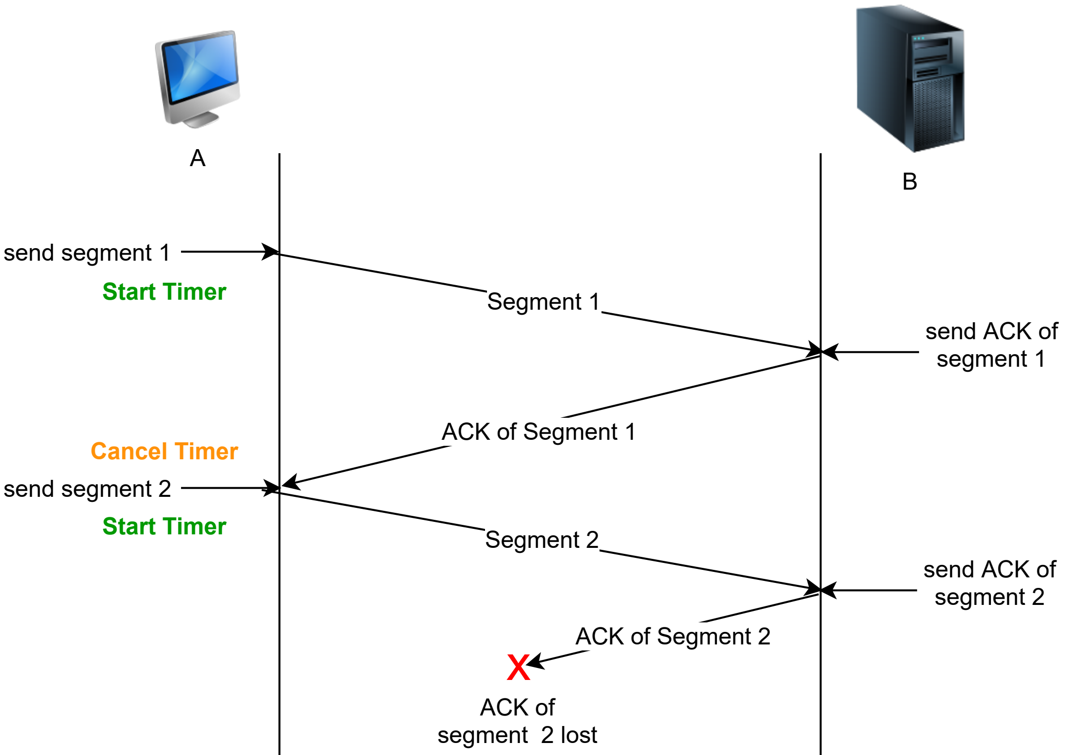
Using retransmission timers to recover from segment losses

**4** of 7

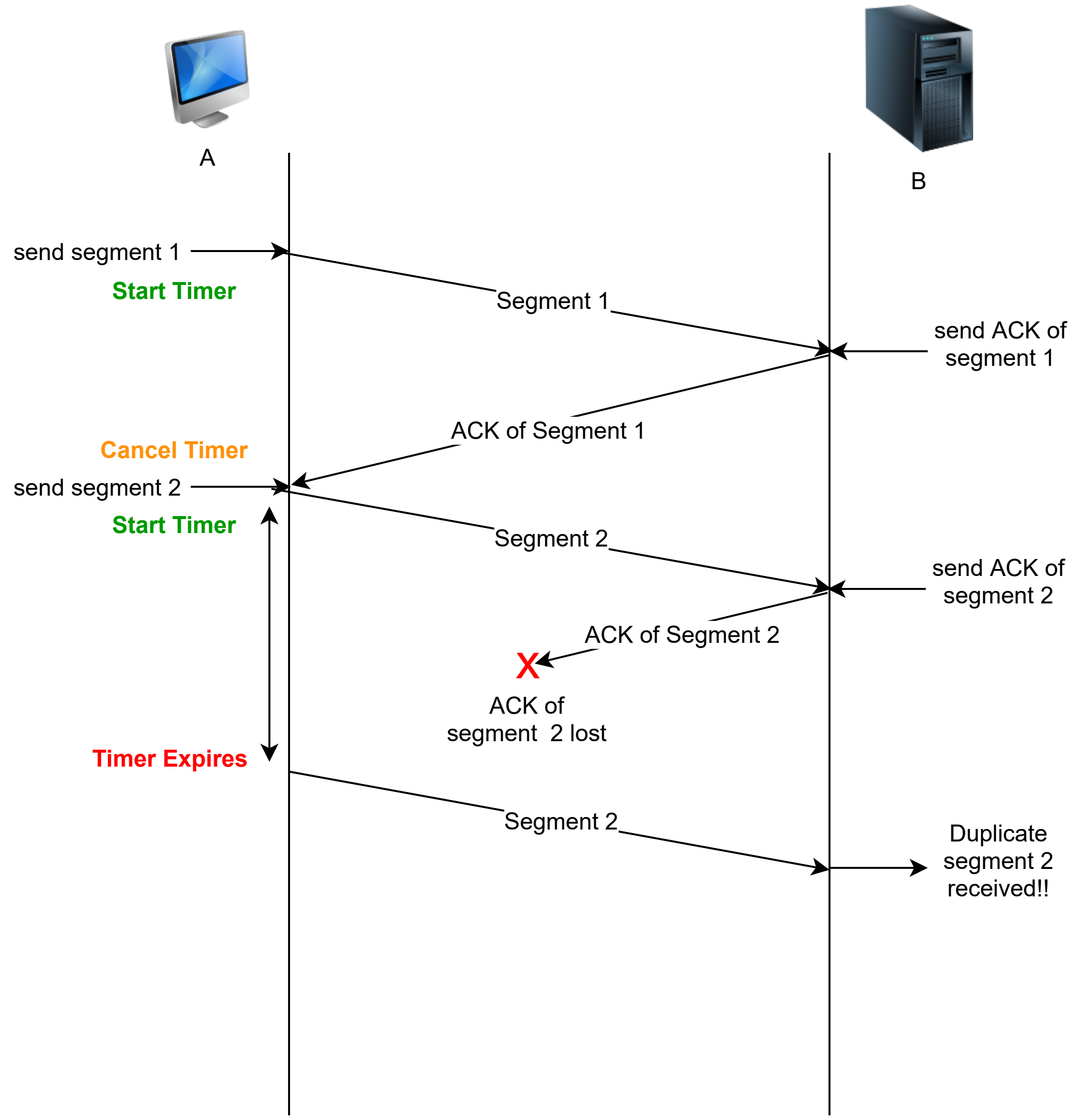


Using retransmission timers to recover from segment losses

**5** of 7



Using retransmission timers to recover from segment losses



Using retransmission timers to recover from segment losses

**7** of 7

## Sequence Numbers

To identify duplicates, transport protocols associate an *identification number* with each segment called the **sequence number**. This sequence number is prepended to the segments and sent. This way, the end entity can identify duplicates.

## Quick Quiz!

1

Checksums address which imperfection of the network layer?

COMPLETED 0%

1 of 5 ‹ ›

Let’s continue our discussion of reliable transfer data in the next lesson!