

# TCP delayed acknowledgment

**TCP delayed acknowledgment** is a technique used by some implementations of the **Transmission Control Protocol** in an effort to improve **network performance**. In essence, several **ACK** responses may be combined together into a single response, reducing protocol overhead. However, in some circumstances, the technique can reduce application performance.

waiting for acknowledgment replies. Alternatively, Nagle's algorithm may be disabled by the application on the sending side.

## 3 References

[1] <http://tools.ietf.org/html/rfc1122#page-96>

## 1 Method and advantages

As described in **RFC 1122**, a host may delay sending an **ACK** response by up to 500 ms. Additionally, with a stream of full-sized incoming segments, **ACK** responses must be sent for every second segment.

Delayed **ACKs** can give the application the opportunity to update the **TCP receive window** and also possibly to send an immediate response along with the **ACK**. For certain protocols such as **Telnet**, delayed **ACKs** can reduce the number of responses sent by the server by a factor of 3, by combining the **ACK**, window update and the response data into one segment.<sup>[1]</sup>

## 2 Problems

The additional wait time introduced by the delayed **ACK** can cause further delays when interacting with certain applications and configurations. If **Nagle's algorithm** is being used by the sending party, data will be queued by the sender until an **ACK** is received. If the sender does not send enough data to fill the **maximum segment size** (for example, if it performs two small writes followed by a blocking read) then the transfer will pause up to the **ACK** delay timeout.

For example, consider a situation where Bob is sending data to Carol. Bob's socket layer has less than a complete packet's worth of data remaining to send. Per Nagle's algorithm, it will not be sent until he receives an **ACK** for the data that has already been sent. At the same time, Carol's application layer will not send a response until it gets all of the data. If Carol is using delayed **ACKs**, her socket layer will not send an **ACK** until the timeout is reached.

If the application is transmitting data in smaller chunks and expecting periodic acknowledgment replies, this negative interaction can occur. To prevent this delay, the application layer needs to continuously send data without

## 4 Text and image sources, contributors, and licenses

### 4.1 Text

- **TCP delayed acknowledgment** *Source:* [https://en.wikipedia.org/wiki/TCP\\_delayed\\_acknowledgment?oldid=545869173](https://en.wikipedia.org/wiki/TCP_delayed_acknowledgment?oldid=545869173) *Contributors:* Edward, James A. Donald, Intgr, Malcolma, Kgutwin, Abdo57, Jmajeremy, Addbot, Ironholds, Yobot, AdventurousSquirrel, Makecatbot and Anonymous: 5

### 4.2 Images

### 4.3 Content license

- Creative Commons Attribution-Share Alike 3.0