Enhancing Cable Modem TCP Performance

Lior Storfer

Cable Broadband Communications Group, Texas Instruments

Abstract

Industry standard Data Over Cable Service Interface Specification (DOCSIS™) cable modems provide end users with an always-on, broadband connection to the Internet. Most applications delivered via that high-speed connection are transmitted via transmission control protocol (TCP). The interaction that takes place between TCP and the DOCSIS-defined media access control (MAC) protocol can have a negative impact on the performance of certain applications, and, as a result, negatively affect end users' experience with those applications.

This article discusses the inherent bi-directional behavior of TCP and its performance when used in conjunction with the DOCSIS MAC protocol. It provides typical examples of degradation in TCP throughput and describes how the addition of application awareness and acknowledgment filtering software can be used to improve TCP throughput.

TCP characteristics

TCP is the most commonly used transport protocol for Internet applications. Because it is a connection-based protocol, TCP is able to guarantee that each data packet transmitted from a server reaches its intended destination client.

The handshake protocol is a multi-step process. A server sends several packets at a time to a client. The server then waits to receive an acknowledgement (ACK) from the client, which confirms that the packets have been received. If the client does not send an ACK to the server within a given time period, the server will determine that the packets were not delivered successfully. The server "stalls" the transmission of further packets and retransmits the unacknowledged packets until it receives acknowledgement of their receipt.

The number of packets the server sends while waiting for an ACK to arrive is determined by a system parameter known as the "window size." The window size has a significant impact on TCP performance. The smaller the window size, the more likely it is that a server will stall the transmission of the pending (or queued) packets while waiting for ACKs to arrive.

Even when the physical transport channel is large enough to support a high data transmission rate, actual throughput achieved by an application may be limited to a fraction of that potential rate. It is important to understand that it is actual packet throughput, not raw data rate, which has the most dramatic impact on how poorly or how well TCP-based applications perform.

DOCSIS fundamentals

The DOCSIS cable modem standard specifies both the physical layer (PHY) aspects of cable modem transmissions and the MAC protocol used to access the cable transmission channel. The specification covers the characteristics of downstream transmissions, which are sent from the cable modem termination system (CMTS) in the headend to cable modems located at the customer premises, and upstream transmissions, which are sent from cable modems to the CMTS. The CMTS is responsible for receiving packets from the "Internet cloud" and sending them to cable modems at the customer premises via the cable network. The CMTS determines the order and priority of packet transmission across the cable network to the client. Because the CMTS has full ownership of downstream traffic, access to the transmission channel does not need to be negotiated by the cable modem for transmission of downstream traffic.

Access to the upstream channel is, however, different. In the upstream channel, all modems sharing media must compete for access to the upstream channel. For this reason, cable modems that desire to send data upstream to the CMTS must first send a request to the CMTS in order to obtain a transmission opportunity.

The CMTS collects these requests and sends a message back to all modems allotting each a time slot during which it can send its data packet back to the CMTS via the upstream channel. Cable modems may only make requests for one transmit opportunity at a time, therefore, there is a limit to the number of upstream transmissions the modem can perform in a given second.

This difference between the downstream channel and the upstream channel causes transmission between a cable modem and the CMTS to be asymmetrical. This asymmetry is the key cause of TCP throughput degradation in certain scenarios in which DOCSIS cable modems are used.

Differing scenarios that affect TCP performance

<u>Single modem, single user, no rate limit</u> – When a single cable modem is connected to the CMTS and the modem is not rate limited by the CMTS, end users will experience maximum TCP throughput. For example, a downstream channel with capacity of ~40 Mbit/sec is likely to achieve TCP throughput, which exceeds 30 Mbit/sec.

<u>Single modem, single user, upstream rate limit</u> **–** When a cable provider has configured the network to limit the rate of its cable modems' upstream transmissions, TCP throughput is decreased. For example, if a cable modem's upstream is rate limited to 128 Kbit/sec and the downstream bandwidth is not rate limited, TCP throughput drops from 30 Mbit/sec. to approximately 6 Mbit/sec.

The TCP degradation is caused by the upstream rate limit reducing the number of acknowledgements that the modem is capable of transmitting. ACKs must be transmitted from the client to the server before another batch of packets can be transmitted between the server and the client. When the rate of the ACKs that are going between the client and the server is limited, the rate of TCP throughput is also limited.

Single modem – with a home network – In a home networking environment, more than one PC can be competing for bandwidth on the cable modem connection simultaneously. TCP-based applications running in this kind of environment can experience throughput degradation of up to 20x.

For example, in a home network with two users, if User #1 is downloading files while User #2 is sending e-mails with attachments, the modem has two sources of data competing for access to the upstream channel – ACKs from User #1 and data from User #2. This competition results in ACKs accumulating in the upstream queue, thereby causing TCP throughput to drop sharply. The TCP throughput experienced by User #1 can drop from 30 Mbit/sec to 1.5 Mbit/sec.

Multiple cable modems sharing the cable – As additional modems in a neighborhood are connected to the same cable system, there is competition for access to the upstream channel. The time required to gain access to the upstream channel is increased as more users log on to the network. Though the downstream channel may still have sufficient bandwidth to continue delivering large amounts of data to the user, the delay in sending TCP ACKs in the upstream channel limits the actual throughput achievable.

It is in cable providers' best interest to implement a solution to this problem. The solution they choose should allow the number of modems sharing a line to be increased while enabling TCP throughput to be sustained. Solving the problem by a CPE software solution will likely delay or eliminate the requirement to "split" the cable node.

Considerations for accelerating TCP performance

When cable network engineers evaluate solutions intended to solve the problem described above, it is important that they ensure that the following aspects are addressed:

Compliance with DOCSIS – The acceleration mechanism must work independent of the DOCSIS MAC protocol and not cause any changes to its behavior.

System level simplicity – The acceleration mechanism must be simple from a system perspective. Therefore the acceleration algorithm should reside on the CPE only. This means that the service provider will not have to make upgrades to head-end equipment -- the CMTS or server.

No additional traffic on the upstream – The acceleration mechanism must not burden the upstream channel with additional traffic – data or management overhead.

Application awareness improves TCP throughput performance

The critical element of any method proposed to enhance TCP throughput lies in the TCP protocol itself. When a server receives an ACK for a specific packet, it assumes that all previous packets have been properly received by the client. This is by definition, as each ACK includes all the information conveyed by all the ACKs preceding it.

When the upstream channel becomes clogged with traffic, ACKs start to accumulate at the client. The situation is improved if the client sends only the last ACK in the queue and discards all previous, un-transmitted ACKs. Using this method, the server still confirms that all previous packets were received by the client, but the amount of data being sent and response time would be greatly reduced.

This process of sending only the most recent ACK and discarding the ACKs that have not yet been transmitted is called "ACK Filtering." By inspecting and filtering data that is being processed by the modem, the modem is made "application aware," and is able to discard redundant ACKs.

ACK filtering meets the requirements listed above for a proper solution to this problem. ACK filtering is compliant with the DOCSIS standard. Because ACK filtering is implemented in the CPE, the solution also keeps things simple on a system level. When ACK filtering is applied, the CMTS remains unaware of any pre-processing the cable modem is performing on the data before sending it upstream.

Finally, ACK filtering is a plant-friendly technique for boosting throughput. The amount of traffic that the modem needs to send on the upstream channel is reduced significantly. This increases the amount of users that can be served on a plant while maintaining the plant's performance.

Achieving higher TCP throughput using application awareness

When ACK filtering is implemented, it produces improved throughput in all the previously mentioned scenarios, except for the "Single modem, single user, and no rate limit" case.

<u>Single modem, single user, and no rate limit</u> – In this "ideal" scenario, ACK filtering does not come into play. Because all ACKs are transmitted from the client to the server, bottlenecks do not occur on the upstream channel. Throughput of 30 Mbit/sec is maintained as before.

<u>Single modem, single user with upstream rate limit</u> – When the upstream channel has a rate limit of 128 Kbit/sec and there is no rate limit imposed on the downstream channel, TCP throughput drops to 6 Mbit/sec. Activation of ACK filtering almost completely removes any upstream bottleneck in this scenario, and TCP throughput increases to around 30 Mbit/sec.

<u>Single modem – with a home network</u> – ACK filtering in the scenario in which two or more users are sharing the same broadband cable connection results in the most noticeable improvement of TCP throughput. As described above, when one user is running an application over TCP and the second is sending data onto the upstream channel, TCP throughput drops sharply to approximately 1.5 Mbit/sec. Activation of ACK filtering makes it possible for TCP throughput to increase to 16 Mbit/sec.

<u>Multiple modems on the line</u> – ACK filtering reduces the amount of traffic on the upstream channel in this scenario as well. Therefore, the number of modems running TCP-based applications concurrently may be increased when ACK filtering is applied in this scenario. The increase is dependent on the rate limit imposed on the modems as well as the desired performance level of the plant.

Conclusion

Due to the inherent bi-directional nature of TCP and the asymmetrical properties of the DOCSIS MAC protocol, TCP throughput degrades significantly in certain cable modem usage scenarios. Activation of ACK filtering introduces application awareness on top of the DOCSIS MAC protocol and provides significant increases in TCP throughput. ACK filtering reduces the amount of time it takes to acknowledge multiple packets and also reduces unnecessary redundancy in ACK messaging, which in turn reduces the amount of traffic vying for upstream bandwidth.

To learn more about how Texas Instruments has addressed these challenges go to: www.ti.com/turbodox

Lior Storfer is the product marketing and architectures manager for Texas Instruments' Cable Broadband group. Lior holds a bachelor's degree in computer engineering from Technion (Israeli Institute of Technology) and a master's degree in business from Tel-Aviv University and can be reached at slior@ti.com