Mehr Durchzug

Windows: Wie Microsoft den IP-Verkehr beschleunigt

Im Anniversary Update für Windows 10 und Windows Server 2016 baut Microsoft ohne viel Tamtam gleich fünf neue Features in den Netzwerk-Stack ein. Alle dienen dazu, noch etwas mehr aus der Internet-Leitung herauszuquetschen.

Von Martin Winkler

Microsofts Updates für die aktuellen Betriebssysteme Windows 10 und Windows Server 2016 bringen viele Verfeinerungen, die die Nutzergemeinde bereits umfassend diskutiert hat. Weniger bekannt ist, dass die Updates auch Auswirkungen auf die Netzwerkleistung haben.

Es handelt sich um fünf neue Funktionen für den TCP-Stack (Transmission Control Protocol): TCP Fast Open, Initial Congestion Window 10, TCP Recent ACKnowledgment, Tail Loss Probe und TCP LEDBAT.

Alle fünf erweitern und verbessern das Verhalten beim Abruf von Daten aus dem Internet, indem sie die Latenz verkürzen, also die Frist, die vom Anstoßen eines Vorgangs bis zum Eintreten des erwünschten Effekts vergeht. Interessant ist, dass es sich bei drei der Funktionen um abgesegnete RFC-Spezifikationen handelt, aber bei zweien um experimentelle Entwürfe – TCP Recent ACKnowledgement und Tail Loss Probe (TLP).

Kürzere Latenzen

Webbrowser müssen zum Laden einer Webseite viele TCP-Verbindungen aufbauen. Für jede einzelne Verbindung ist ein Drei-Wege-Handshake erforderlich (siehe Grafik rechts). Bei üblichen TCP-Stacks beginnt die Übertragung von Nutzdaten erst nach dem Handshake. Bei einer einzelnen TCP-Übertragung spielt das keine so große Rolle, man hat dennoch das Gefühl, dass die Übertragung fast unmittelbar beginnt. Bei gängigen Webseiten summieren sich viele einzelne Latenzen jedoch zu einem spürbaren Effekt – der Seitenaufbau lahmt.

Für jede herkömmliche TCP-Verbindung beträgt die Latenz 1,5 RTTs (Round Trip Time). RTT ist ein relativer Wert, der von der Strecke zwischen dem Client und dem Server abhängt, auf dem das benötigte Seitenelement liegt. Man kann den Wert beispielsweise mit dem Ping-Kommando messen – es liefert die Zeit, die nach dem Abschicken eines Pings bis zum Empfang der Antwort vom angepingten Server vergeht. Von einem ADSL-Anschluss erreicht man einen nahen Server in 40 bis 50 ms, von einem VDSL-Anschluss in etwa 20 ms. Weiter entfernte Server antworten innerhalb von 100 ms (USA) bis 300 ms (Asien).

Mit TCP Fast Open lassen sich wiederholte Drei-Wege-Handshakes zum selben Server vermeiden; dafür muss ein Server nach dem ersten Verbindungsaufbau ein kryptografisch sicheres (d. h. möglichst nicht vorhersehbares) Cookie an den Client schicken (TCP Fast Open Cookie). Das ist für Clients ein Schlüssel zum Abkürzen von darauffolgenden TCP-Initialisierungen zum selben Server; sie müssen es bereits mit dem SYN-Paket mitschicken. Ein Server, der ein gültiges Cookie bekommt, wartet also nicht erst die SYN-ACK-Folge ab, sondern nutzt die bereits aufgebaute TCP-Verbindung, um Nutzdaten umgehend zu schicken. Das spart bis zu 1 RTT. Cookies werden periodisch im Abstand von Sekunden bis Minuten neu gebildet; ältere werden dadurch ungültig, was Replay-Attacken mittels abgegriffener Cookies erschwert.

Aber Nutzdaten wie Cookies lassen sich nun grundsätzlich in SYN-, SYN-ACK- und ACK-Paketen übertragen. Typisches Beispiel ist die TLS-Verschlüsselung (Transport Layer Security). Bisher tau-

Ab dem 2. August liefert Microsoft mit dem Anniversary Update für Windows 10 und Server 2016 etliche neue Funktionen, darunter auch Spezialitäten, die den Internet-Verkehr beschleunigen.

schen dafür Server und Client nach dem TCP-Verbindungsaufbau ein server_hello und ein client_hello aus, um TLS auszuhandeln. Nutzdaten fließen daher erst, nachdem ein RTT verstrichen ist. Clients mit TCP Fast Open stecken das client_hello schon in das SYN-Paket und sparen ein RTT.

Auch Webbrowser dürfen im SYN-Paket Nutzdaten mitschicken, und zwar für den ersten HTTP-Request. Das ist jedoch nur für HTTP-Requests erlaubt, die wiederholt werden dürfen, denn der Browser darf sich nicht darauf verlassen, dass das Betriebssystem auf der Server-Seite im SYN gelieferte Nutzdaten zum Webserver hinaufreicht. Davon profitieren also Anwendungen wie CSS oder JavaScript, deren HTTP-Requests Clients wiederholen dürfen.

Congestion Window

Die zweite wichtige Neuerung betrifft das Congestion Window; Microsoft erhöht es von 4 MSS auf 10 MSS (Maximum Segment Size – IP-Paketkapazität abzüglich Ethernet-, TCP- und PPP-Header, häufig 1452 Bytes). Das Congestion Window ist eine vom Sender berechnete Obergrenze an Daten, die ohne zugehörige Quittungen zum Empfänger unterwegs sein dürfen (ACKs). Dieser Wert dient dazu, die Puffer auf der Strecke zum Empfänger (z. B. in Routern) nicht zu verstopfen; der Sender vermeidet damit Paketverluste durch überfüllte Puffer.

Kein Internet-Host kennt die maximale Geschwindigkeit der gerade genutzten Übertragungsstrecke. Deshalb tastet er sich an das Maximum heran, indem er das Congestion Window verdoppelt, solange er Quittungen bekommt (Slow Start). Wenn eine Quittung ausbleibt, geht er von einem Übertragungsfehler aus und halbiert seine Senderate.

Je kleiner das Congestion Window zu Beginn der Übertragung, desto langsamer die Beschleunigung. Das bedeutet: Je kürzer eine Datei, desto weniger lastet sie eine Leitung bei kleinem Congestion Window aus. Das nun vergrößerte initiale Congestion Window adressiert genau diesen Fall – kurze Dateien können daher schnelle Leitungen besser ausschöpfen, weil sie in einem oder mehreren größeren Stücken übertragen werden.

Übertragungsfehler

Gelegentlich kommt es vor, dass Daten am Ende einer Übertragung das Ziel nicht erreichen (Tail Loss Probe, TLP); der Sender bekommt die Quittungen für die letzten Segmente nicht. Normalerweise muss er dann mindestens eine Sekunde warten, bevor er mit Sendewiederholungen beginnt. Diese Latenz kann er mit der Funktion Tail Loss Probe verringern: Dabei darf der Sender Wiederholungen starten, sobald ACKs für 2 RTTs ausbleiben. Sollte der Sender daraufhin Quittungen erhalten, sendet er ohne Neuberechnung des Congestion Window weiter (ähnlich wie bei SACKs).

Mittels Recent ACKnowledgement (RACK) reagiert der Server eher auf Datenverluste als mit der bisher gängigen Methode. Nach dem bisher üblichen Verfahren wartet der Empfänger eine Zeit lang und sendet dann das zum letzten empfangenen Segment gehörende ACK ein zweites Mal, um einen Paketverlust zu signalisieren (duplicate ACK). Dann schickt der Sender jene Daten erneut, er nach den doppelt bestätigten abgeschickt hatte.

Mit RACK wartet der Sender nicht auf doppelte ACKs, sondern führt Buch darüber, wann ACKs eintreffen sollten. Trifft ein ACK für ein Segment ein, das deutlich später gesendet wurde als noch unquittierte Daten, geht er davon aus, dass die Segmente vor diesem ACK verloren gegangen sind, und sendet sie neu. So wird der Versand unbestätigter Segmente eher wiederholt. Microsoft aktiviert dieses Feature nur bei Verbindungen, deren Latenz über 10 ms beträgt; bei LAN-Verbindungen bleibt es also abgeschaltet. Das erscheint sinnvoll, weil aufgrund der geringen Latenzen die Berechnungen zu hohen Abweichungen unterworfen sind.

Absichtliches Bremsen

Speziell für zeitlich unkritische Datenübertragungen im Hintergrund wurde TCP LEDBAT entwickelt (Low Extra Delay Background Transport). Sendet ein Teilnehmer zu schnell, laufen Puffer voll, was zu einer Erhöhung der Latenz führt, da alle nachfolgend im Puffer ankommenden Pakete warten müssen, bis sie an der Reihe sind. TCP LEB DAT setzt der Sender ein, um Datenströme geringer Priorität entsprechend langsam zu senden.

Dafür schätzt er periodisch die Latenz der Verbindung, indem er Änderungen des Congestion Windows auswertet. Eine einfachere Möglichkeit, die Latenz zu bestimmen, besteht darin, die Zeit zwischen dem Senden eines Packets und der Ankunft des zugehörigen ACK heranzuziehen. Nimmt dieser Abstand zu, geht der Sender davon aus, dass sich die Puffer füllen, weil andere TCP-Verbindungen die Leitung nutzen. Dann reduziert er seine Sendegeschwindigkeit, bis die Latenz wieder sinkt.

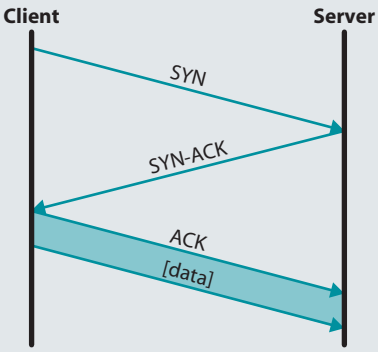
Es gibt kommerziell erhältliche Traffic Shaper, die Pufferauslastungsmessungen schon seit einigen Jahren beherrschen. Dazu gehört beispielsweise cFosSpeed, an dem der Autor dieses Beitrags mitgearbeitet hat; es misst die Latenz zum nächsten Hop.

Microsoft hat diese Funktion als Socket-Option für Applikationsentwickler implementiert. Nützlich wäre es für Filesharing- oder auch Mail-Programme, wenn diese lange Mails verschicken. Wie Entwickler diese Funktion nutzen können, ist bisher freilich undokumentiert; Details erklärt Microsoft auf Nachfrage.

(dz@ct.de)

Drei-Wege-Handshake

Bisher konnte ein Windows-Client erst dann Nutzdaten von einem Web-Server anfordern, wenn der Handshake abgeschlossen war. Das führt bei heutigen Webseiten zu erheblichen Verzögerungen beim Seitenaufbau – es gibt jedoch einige Verfahren, diese Latenzen zu verkürzen.



Literatur

[1] TCP Fast Open for zero RTT TCP connection setup, RFC 7413, ct.de/netze/rfc/rfcs/rfc7413.shtml
[2] Initial Congestion Window 10 (ICW10) by default for faster TCP slow start, ct.de/netze/rfc/rfcs/rfc6928.shtml
[3] Y. Cheng et al, RACK: a time-based fast loss detection algorithm for TCP, experimental IETF draft, ietf.org/archive/id/draft-cheng-tcpm-rack-00.txt
[4] N. Dukkipati et al., Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail Losses, experimental IETF draft, tools.ietf.org/html/draft-dukkipati-tcpm-tcp-loss-probe-01
[5] TCP LEDBAT for background connections, RFC 6817, ct.de/netze/rfc/rfcs/rfc6817.shtml