

## *Internet Geschwindigkeit.*

In Abhängigkeit der Starttaste absolviert die Internet-Geschwindigkeit-Applikation 4 mal 3 Messungen. Gestartet wird mit der Download-Messung, danach die Upload-Messung und zum Schluss die Messung eines Zeitintervalls für den Hin- und Rückweg eines Datenpakets (Ping).

Für die Download-Geschwindigkeit kontaktiert JavaScript den Server <https://www.cloudflare.com/de-de/>. Der Server liefert ein, mit Nullen gefülltes, fünf Megabyte großes Datenpaket. Für den Upload-Test wird ein einfacher HTTP-Anfrage- und Antwortdienst <https://httpbin.org/> benötigt. Die Funktion Ping kontaktiert [https://www.google.com/generate\\_204](https://www.google.com/generate_204).

Alle drei Netzwerkfunktionen sind asynchrone Dienste. In JavaScript bedeutet asynchron, dass eine Aufgabe (wie das Laden von Daten) im Hintergrund ausgeführt wird, ohne dass das gesamte Programm angehalten werden muss. Es gibt kein Blockieren, während der Browser auf die Antwort des Servers wartet, und er kann weiterhin auf Benutzereingaben reagieren oder Animationen abspielen. Wäre der Vorgang "synchron", würde die Webseite einfrieren, bis die Daten da sind.

Zwei analoge Messinstrumente visualisieren die Download- und Upload Geschwindigkeit. Meine Applikation sollte genau so animiert werden, wie es auf dieser Webseite <https://canvas-gauges.com/documentation/examples/> zu sehen ist. Gelöst habe ich das mit einem Schrittschaltwerk.

Im Fehlerfall, jede asynchrone Anfrage besitzt eine try- & catch-Methode, wird der Programmlauf gestoppt. Man muss dann die Starttaste erneut betätigen. Weiterhin wird eine Fehlernachricht visualisiert.

### *Details über Download, Upload und Ping:*

**Die Download-Geschwindigkeit** (Download-Speed) ist die Datenmenge, die pro Sekunde aus dem Internet heruntergeladen werden kann. Meist in MBit/s (Megabit pro Sekunde) angegeben. 1 Byte entspricht 8 Bit. Einflussfaktoren sind Anschlussart (DSL, Kabel, Glasfaser), Anbieter, Tageszeit und WLAN-Qualität. Weiterhin wird die Download-Geschwindigkeit durch die Bandbreite, Serverauslastung und Netzwerkauslastung bestimmt. Zusätzlich

zu den eigentlichen Nutzdaten werden Steuerdaten (Protokoll-Informationen) übertragen, was die Gesamtmenge geringfügig erhöht. Die tatsächlich übertragene Datenmenge ist immer größer als die Datei, die am Ende auf deiner Festplatte landet.

- **Header-Daten:** Protokolle wie TCP/IP fügen jedem Paket Steuerinformationen hinzu (z. B. IP-Adressen, Checksummen zur Fehlerkorrektur).
- **Verbindungsaufbau:** Bevor der eigentliche Download startet, tauschen Sender und Empfänger Signale aus, um die Verbindung zu synchronisieren.
- **Bestätigungen (ACKs):** Dein Rechner schickt ständig kleine Bestätigungen zurück, um zu sagen: „Paket 1 ist angekommen, schick Paket 2“.

**Wichtig:** Bei meiner Internet-Geschwindigkeits-Applikation wird die Netto-Rate (die nutzbaren Daten) gemessen. Dein Anbieter verkauft dir hingegen oft die Brutto-Rate – also inklusive der Steuerdaten.

```
46 // ****
47 // Test Download. blob = Speicher = Binary Large Object
48 // ****
49 async function testDownload(url) {
50     const start = performance.now(); // <-- Start Zeitstempel
51     const response = await fetch(url, { cache: "no-store" });
52     const blob = await response.blob();
53     const end = performance.now(); // <-- Ende Zeitstempel
54     return get_Mbps(blob.size, end - start);
55 }
56
```

**Die Upload-Geschwindigkeit** (Upload-Speed) ist für den „Otto Normalverbraucher“ immer geringer im Durchschnitt als die Download-Geschwindigkeit. Warum? Dass der Upload (Daten von dir ins Netz) meist deutlich langsamer ist als der Download (Daten aus dem Netz zu dir), hat weniger mit technischem Unvermögen zu tun, sondern ist eine bewusste Entscheidung der Provider. Die meisten Internetnutzer konsumieren mehr, als sie produzieren. Wenn du ein YouTube-Video schaust, lädst du gigantische Datenmengen herunter, während du nur winzige "Bestätigungspakete" hochlädst, um dem Server zu sagen: "Ja, die Daten sind angekommen, schick mehr!" Da die Kapazität der physischen Leitungen begrenzt ist, weisen

Internetanbieter dem Download einen größeren Anteil der verfügbaren Bandbreite zu, um das Nutzererlebnis für die Mehrheit zu optimieren.

```
57 //*****
58 // Test Upload!
59 //*****
60 async function testUpload() {
61   const dataSize = 5 * 1024 * 1024; // 5 MB
62   const data = new Uint8Array(dataSize);
63   const start = performance.now();           // <-- Start Zeitstempel
64   await fetch(up_URL, {
65     method: "POST",
66     body: data,
67   });
68   const end = performance.now();             // <-- Ende Zeitstempel
69   return get_Mbps(dataSize, end - start);
70 }
```

**Was ist ein Ping?** Ping steht für „*Paket Internet Groper*“ und ist ein Werkzeug um Netzwerkverbindung zu testen. Der Ping stellt fest, ob eine Zieladresse (IP-Adresse oder Domain wie google.com) überhaupt antwortet. Weiterhin wird die Zeitspanne in Millisekunden gemessen, die ein Datenpaket für den Hin- und Rückweg benötigt (Round Trip Time, RTT). Die Zeitspanne wird hier als „*Latenz*“ bezeichnet. In der Informationstechnik beschreibt *Latenz* die Verzögerung zwischen einer Aktion und der darauffolgenden Reaktion.

**Ping (Latenz) und Jitter!** Jitter ist die Varianz oder Unregelmäßigkeit der Ping-Werte, also wie stark die Verzögerung zwischen den einzelnen Datenpaketen schwankt. Der Jitter wird in Millisekunden gemessen. Für die Ermittlung von Datenpaket-Schwankungen (Jitter) muss das Werkzeug Ping mehrere Messungen durchführen. Bei meiner Internet-Speed-Applikation sind das insgesamt 10 Messungen.

**Bewertung:** Ein konstant niedriger Ping ist gut, aber ein hoher Jitter macht die Verbindung instabil. Eine Verbindung mit niedrigem Ping, aber hohem Jitter kann sich schlechter anfühlen als eine mit etwas höherem, aber konstantem Ping, da die Unvorhersehbarkeit störend ist, besonders bei Anwendungen wie VoIP (Voice over IP) oder Online-Game. Tritt Jitter auf, liegt das Problem meist an einer ausgelasteten Netzwerkbandbreite, alten Netzwerkinfrastrukturen, Ethernet-Kabeln oder Endgeräten.


```

72  //*****
73  // Ping!
74  //*****
75  async function ping() {
76      const start = performance.now();
77      await fetch(ping_URL, { mode: "no-cors", cache: "no-store" });
78      const end = performance.now();
79      return end - start;
80  }
81
82  //*****
83  // Jitter!
84  //*****
85  function calculateJitter(times) {
86      let diffs = [];
87      for (let i = 1; i < times.length; i++) {
88          |   diffs.push(Math.abs(times[i] - times[i - 1]));
89      }
90      const jitter = diffs.reduce((a, b) => a + b, 0) / diffs.length;
91      jitter_value = jitter.toFixed(2);
92  }

```

*Liebe Leserinnen und Leser. Hallo du! Ich muss mich hier absichern. Das möchte ich im nachfolgenden Text auf humorvoller Weise tun:*

Hier ist es also: "Internet Speed" – das Programm, auf das die Welt (oder zumindest dein Router) **nicht gewartet** hat. Da wir uns schon so lange kennen und ich dich als Freund schätze, möchte ich sicherstellen, dass unsere Freundschaft diesen digitalen Geniestreich überlebt. Daher hier ein paar „freundschaftliche“ Randnotizen:

 **Keine Gewähr, keine Haftung, kein Mitleid:** Ich habe den Code nach bestem Wissen und Gewissen (und nach dem dritten Espresso) geschrieben. Wenn dein PC plötzlich meint, er sei eine Heißluftfritteuse, oder wenn er anfängt, in Binärcode rückwärts zu fluchen – ich bin unschuldig! Die Nutzung erfolgt auf eigene Gefahr.

Dein Datenvolumen hat bereits Angst: Nur damit du es weißt: Das Tool ist hungrig. Es verbraucht Internet-Volumen schneller als du Pizza am Spieleabend. Wenn dein Provider dir morgen eine Glückwunschkarte zur Beförderung zum „Platin-Nutzer“ schickt, weißt du, woran es liegt. Behalte deinen Zähler im Auge, bevor dein Breitband zur Schmalspur wird!

**Support-Level: Mein technischer Support beschränkt sich auf Schulterzucken und den Satz: „Also bei mir hat’s funktioniert.“**

**Kurz gesagt: Wenn es deinen Speed verdoppelt, bin ich ein Genie. Wenn es dein WLAN in die Steinzeit schickt, haben wir uns heute nie unterhalten. 😊  
Viel Spaß beim Testen und lass die Leitung nicht glühen!**

**Januar 2026, Hans Busche**