**1. نسخهٔ HTTP/2 حاوی داده‌های باینری (دودویی) است:**HTTP/1.1 از داده‌های متنی استفاده می‌کند و این در حالی است که داده‌های متنی به طور کلی در سراسر شبکه از بازدهی کمتری نسبت به داده‌های باینری برخوردارند.

**2. هِدِرهای HTTP/2 فشرده‌ شده هستند:**به طور کلی منظور از Header اطلاعاتی است که در پاسخ به یک ریکوئست ارسال می‌شود که شامل دیتا، مبدأ، نوع، حجم، مدت زمان کَش و موارد دیگر است. برخلاف HTTP/1.1، این داده‌ها در نسخهٔ HTTP/2 فشرده‌سازی می‌شوند تا پرفورمنس ارتقاء یابد.

**3. نسخهٔ HTTP/2 اصطلاحاً Asynchronous است:**در HTTP/1.1، سرور باید به همان ترتیبی که ریکوئست‌ها را دریافت کرده است، ریسپانس‌ها را ارسال کند اما نسخهٔ HTTP/2 اصطلاحاً Asynchronous است؛ بنابراین پاسخ‌های سریع‌تر و در عین حال با حجم کمتری می‌تواند در زمان کوتاه‌تری از سمت سرور ارسال شود.

**4. نسخهٔ HTTP/2 مولتی‌پلکس است:**در HTTP/1.1، فقط یک درخواست روی یک کانکشن اینترنتی TCP در آن واحد می‌تواند به کار گرفته شود و مرورگرها به طور عادی قادر به ایجاد 4 تا 8 کانکشن با سرور هستند و این در حالی است که ریکوئست‌هایی با حجم زیاد می‌توانند سرعت دانلود فایل‌های دیگر را به تأخیر بیندازند! HTTP/2 اجازهٔ ارسال چندین ریکوئست (درخواست) و دریافت ریسپانس (پاسخ) از سمت سرور را به طور هم‌زمان بر روی یک کانکشن امکان‌پذیر می‌سازد.

**5. نسخهٔ HTTP/2 امکان استفاده از Server Push را فراهم می‌سازد: ‌‌**با استفاده از این نسخه از پروتکل اچ‌تی‌تی‌پی، سرور می‌تواند فایل‌ها -و به طور کلی هر نوع داده‌ای- را قبل از آنکه ریکوئستی ارسال شود، برای مرورگر بفرستد که به این فناوری اصطلاحاً **Server Push** گفته می‌شود. برای مثال، ممکن است شما در پایین صفحهٔ خود به یک اسکریپت لینک دهید. در HTTP/1.1، مرورگر کدهای HTML را دانلود می‌کند، تجزیه می‌کند و سپس فایل جاوااسکریپت را بارگذاری می‌کند (این بارگذاری هنگامی است که با تگ <script> روبه‌رو شویم). سروری که HTTP/2 را ساپورت کند، می‌تواند چنین فایلی را قبل از اینکه نیاز آن را تشخیص دهد، برای مرورگر ارسال کند که در نتیجه در صورت نیاز، کاربر معطل دانلود شدن فایل‌های جی‌اس نخواهد شد (که این به معنی UX بهتر است).