

## Självständigt arbete i Datavetenskap

### HTTP/2

*Är väntans tider över på webben?*



Författare: Johanna Szepanski

Handledare: Anne Norling

Examinator: Johan Leitet

Termin: VT 2016

Ämne: Datavetenskap

Nivå: Högskoleexamen

Kurskod: 1DV42E

## Abstrakt

HTTP/1.1, webbens arbetshäst, har inte genomgått några större förändringar sedan 1999 och är inte designat för att klara dagens komplexa webbsidor. Idag är det inte ovanligt att en webbsida behöver ladda mer än 100 resurser för att rendera och fungera som den ska och det har gjort webben långsam. Utvecklare har tvingats vara kreativa och hitta sätt att öka prestandan i webbapplikationer och många lösningar har under åren kommit att bli best practice.

Arbetet med att snabba upp webben har pågått länge. 2009 annonserade Google att de arbetade på ett nytt experimentellt protokoll, SPDY. SPDY blev framgångsrikt och med det protokollet som utgångspunkt påbörjades arbetet med att uppdatera det ursprungliga HTTP-protokollet. Resultatet av dessa ansträngningar godkändes för publicering 2015 och går under namnet HTTP/2.

Syftet med den här experimentella studien var att undersöka vilken effekt HTTP/2 har på laddningstiderna i en befintlig webbapplikation, om det finns skillnader mellan webbläsare och hur skillnaderna mellan olika uppkopplingstyper ser ut. Två webbläsare ingick i experimentet, Chrome och Firefox, och två uppkopplingstyper, DSL och långsamt 3G.

HTTP/2 har en positiv effekt på den i studien testade applikationen. Som ett medelvärde av båda webbläsarna och båda uppkopplingstyperna minskade laddningstiden med drygt 6,5%. Chrome visade bäst effekt av HTTP/2 vid första laddningen på en snabbare uppkoppling medan Firefox bäst nyttjar HTTP/2 vid andra laddningen på en längsammare uppkoppling. Vid tre av fyra testparametrar är det Firefox som är vinnaren vilket är intressant då Firefox, rent generellt oavsett protokollversion, är den snabbare webbläsaren.

HTTP/2 är fortfarande ungt och det finns all anledning att tro att det inte går att fastställa dess fulla potential förrän om flera år. Utvecklare kommer att testa och finjustera implementationen av protokollet under lång tid framöver, men för den som är intresserade av webbutveckling är framtiden med HTTP/2 synnerligen intressant.

Information om HTTP/2 och resultaten från alla tester finns publicerade i rapporten.

## Nyckelord

HTTP, HTTP/1.1, HTTP/2, web loadingtimes, the effect of HTTP/2, new features in HTTP/2, experiments with HTTP/2

## Tack

Ett stort tack till handledare Anne Norling och mina motståndare Caroline Forslund, Christoffer Goude, Lisa Westlund och Mathias Claesson. Ni är guld värda!

# Innehåll

<b>1 Inledning</b>	<b>3</b>
1.1 HTTP	3
1.1.1 <i>HTTP/1.1</i>	4
1.1.2 <i>Vikten av hastighet på webben</i>	5
1.1.3 <i>HTTP/2 [7, kapitel 12]</i>	5
1.2 Syfte	7
1.3 Frågeställning	8
1.4 Tidigare forskning	8
1.4.1 <i>HTTP/2 is the future of the web, and it is here [10]</i>	8
1.4.2 <i>A Simple Comparison of HTTPS, SPDY and HTTP/2 [11]</i>	8
<b>2 Metod</b>	<b>9</b>
2.1 Laddningstid [12]	9
2.1.1 <i>Vertikal linje</i>	9
2.1.2 <i>Horisontell linje</i>	10
2.2 Testparametrar	10
2.3 Baslinje	10
2.4 Testmiljö	10
2.5 Teknisk specifikation	11
2.6 Genomförande	11
2.6.1 <i>Konfiguration av respektive webbserver</i>	11
2.6.2 <i>Testsvit</i>	11
2.6.3 <i>Verifiering av respektive protokoll</i>	11
2.7 Metoddiskussion	12
<b>3 Resultat</b>	<b>13</b>
3.1 Laddningstider i sekunder	14
3.2 Laddningstider skillnad i %	15
3.3 Medelvärde av respektive laddning	16
3.4 Medelvärde för respektive webbläsare	16
3.5 Medelvärde av den totala effekten	17
<b>4 Slutsats</b>	<b>18</b>
4.1 Fortsatt arbete	18
<b>Ordlista</b>	<b>19</b>
<b>Referenser</b>	<b>20</b>
<b>Bilagor</b>	<b>I</b>
Bilaga A Tester Chrome, DSL 1,5Mbps/38 Kbps 50 ms RTT	I
Bilaga B Tester Firefox, DSL 1,5Mbps/38 Kbps 50 ms RTT	IV
Bilaga C Tester Chrome, långsamt 3G	VII
Bilaga D Tester Firefox, långsamt 3G	X

# 1 Inledning

Sir Tim Berners-Lee<sup>1</sup> presenterade sin vision av webben för sin chef på CERN<sup>2</sup>, Mike Sendall<sup>3</sup> 1989. I dokumentet, som hade rubriken, ”Information Management: A Proposal” [1], gjorde Mike Sendall en notering om att idén var vag men intressant och gav Berners-Lee tid att arbeta med projektet i september 1990. Webben syfte var att förenkla spridning av information [2]. I oktober 1990 hade Berners-Lee skrivit tre tekniker som än idag utgör webbens grundstenar, HTML, URI och HTTP.

Den första webbsidan publicerades 1990 och 1993 släpptes webben fri för vem som helst att använda utan kostnad [2].

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.  
Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .  
[What's out there?](#)  
Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.  
[Help](#)  
on the browser you are using  
[Software Products](#)  
A list of W3 project components and their current state. (e.g. [Line Mode](#) , [X11 Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )  
[Technical](#)  
Details of protocols, formats, program internals etc  
[Bibliography](#)  
Paper documentation on W3 and references.  
[People](#)  
A list of some people involved in the project.  
[History](#)  
A summary of the history of the project.  
[How can I help ?](#)  
If you would like to support the web..  
[Getting code](#)  
Getting the code by [anonymous FTP](#) , etc.

Den första webbsidan<sup>4</sup>

## 1.1 HTTP

Kommunikationen mellan en webbläsare och en webbserver sker genom en förfrågan-svar procedur, på engelska request-response, som börjar med att användaren skriver in adressen, URL:en till en resurs i webbläsarens adressfält. Adressen kan peka mot en webbapplikation som finns publicerad på en webbserver och webbservern svarar på webbläsarens förfrågan genom att sända tillbaka alla de resurser som behövs för att webbapplikationen ska kunna renderas och fungera som den ska.

För att den här förfrågan-svar proceduren ska fungera används det protokoll som kallas HTTP. Enkelt förklarat är det en samling regler för hur kommunikationen på webben ska gå till oavsett vilken webbläsare och server som används. Det går att likna vid hur Postverket en gång satt upp regler för hur brev ska skickas. De ska ha en mottagaradress, ett frimärke och läggas på en av Postens utsatta brevlådor.

---

1. <http://webfoundation.org/about/sir-tim-berners-lee/>

2. <http://home.cern/>

3. [http://bullarchive.web.cern.ch/bullarchive/9930/art2/Text\\_E.html](http://bullarchive.web.cern.ch/bullarchive/9930/art2/Text_E.html)

4. <http://info.cern.ch/hypertext/WWW/TheProject.html>

## 1.1.1 HTTP/1.1

HTTP skrevs, som tidigare nämnts av Sir Tim Berners-Lee, för mer än 25 år sedan och dess senaste stora uppdatering HTTP/1.1 kom 1999. Sedan dess har protokollet inte genomgått några större förändringar. Då bestod en webbsida av text och länkar med minimal formattering och bilder. Idag är en webbsida mycket mer komplicerad än så med videos, animeringar, Javascript- och CSS-filer och mycket mer. I en undersökning gjord av Radware<sup>5</sup> har medianstorleken på en webbsida ökat med 13,59% bara mellan åren 2015 och 2015 [3].

Alla dessa resurser som idag utgör en webbapplikation måste skickas från webbservern till webbläsaren för att applikationen ska renderas i webbläsarfönstret och fungera korrekt. En förfrågan och ett svar skickas per resurs och det kostar i laddningstid. HTTP/1.1 är inte designat för dagens webbapplikationer och det har gjort webben långsam [4, kapitel 2]. 57% av de som shoppar på nätet menar att de skulle lämna en e-butik som tar längre tid än tre sekunder att ladda. Trots det var den mediana laddningstiden, för det 100 e-butiker Radware testade sommaren 2015, 5,2 sekunder [3].

Ju fler resurser en webbapplikation består av desto längre tid tar den att ladda och orsaken till detta går att finna i HTTP:s ”regelverk”. När en webbläsare kommunicerar med en webbserver öppnas en förbindelse, en så kallad TCP connection, på bågge sidor. Alla förfrågningar och svar skickas genom denna förbindelse. Problemet är att endast en resurs släpps igenom åt gången och förbindelsen kan inte återanvändas förrän föregående resurs laddat klart. Från början var det bara tillåtet att ha två förbindelser öppna samtidigt. Nu för tiden, beroende av webbläsare, tillåts att 4-6 förbindelser jobbar parallellt.

Ett annat problem med HTTP/1.1 har varit metadata. Med varje förfrågan och svar följer en header innehållande metadata. Mycket av informationen i headern är exakt likadan mellan olika förfrågningar och svar. Speciellt user-agent och cookies är långa och statiska och påverkar laddningstiden negativt. Det har tidigare gått att komprimera resurser men inte headern.

The screenshot shows a search form titled "Search White House Press Releases, Radio Addresses, Photos and Web Pages".  
1. A text input field labeled "TERM/PHRASE" with placeholder text: "To search White House Press Releases, Radio Addresses, Photos and Web Pages, enter a TERM or PHRASE in the box below which describes your topic of interest (for example, 'social security benefits for retired people')."  
2. A section for date filtering:

- "START DATE": dropdown menus showing "01", "19", and "1993".
- "END DATE": dropdown menus showing "12", "27", and "1996".
- "ITEMS RETURNED FROM EACH CATEGORY": dropdown menu showing "10".
- "SORT ORDER": radio buttons for "DATE" (selected) and "RELEVANCE".

At the bottom of the form are three small thumbnail images of documents and a link: "To comment on this service: [feedback@www.whitehouse.gov](mailto:feedback@www.whitehouse.gov)".

Vita Husets webbsida 1996 [5].

5. <https://www.radware.com/>



Vita Husets webbsida 2016 [5].

Genom åren har det utvecklats strategier för att komma runt dessa begränsningar i HTTP. Det finns en uppsjö av artiklar och föreläsningar om hur utvecklare optimerar en webbapplikation för att minska laddningstiderna. Yslow<sup>6</sup> ett testverktyg från Yahoo, listar 23 olika åtgärder att implementera i en webbapplikation för bästa möjliga prestanda [6]. Ilya Grigorik<sup>7</sup>, web performance engineer på Google, beskriver också dessa prestationshöjande strategier i sin bok, High Performance Browser Networking, [7, kap 11].

### 1.1.2 Vikten av hastighet på webben

2009 genomförde Google en undersökning där de fördröjde leveransen av sökresult till slutanvändaren mellan 100 till 400 millisekunder och såg då en minskning av antalet sökningar per person på 0,2-0,6 procent. Antalet sökningar blev färre ju längre tid användarna blev utsatta för fördräjningen och ju längre perioden var ju längre tid tog det innan antalet sökningar var tillbaka på samma nivå som före experimentet [8]. Året därpå annonserade Google att webbsidors laddningshastighet skulle komma att påverka en webbsidas ranking i sökresultatet [9].

Hastigheten blev allt viktigare för webbens användare och HTTP/1.1 var helt enkelt inte kapabelt att leverera webbsidor till webbläsarna i en för användarna tillfredsställande snabbhet och ur föregångarens brister föddes HTTP/2.

### 1.1.3 HTTP/2 [7, kapitel 12]

2009 meddelade Google att de arbetade på ett nytt experimentellt protokoll, SPDY<sup>8</sup>, som ämnade minska laddningstiderna med upp till 50%. SPDY implementerades snabbt i Chrome, Firefox och Opera och räknades som standard 2012. The HTTP Working Group<sup>9</sup> studerade under dessa år SPDY:s framgångar och med det protokollet som utgångspunkt började man bygga på det som skulle bli den officiella HTTP/2-standarden.

6. <http://yslow.org/>

7. <https://www.linkedin.com/in/igrigorik>

8. <https://www.chromium.org/spdy/spdy-whitepaper>

9. <http://httpwg.org/>

I början av 2015 godkände IESG<sup>10</sup> HTTP/2 för publicering och kort efter det annonserade Google att de skulle avveckla SPDY. HTTP/2 ersätter inte, utan utvecklar, HTTP/1.1. HTTP-metoder, statuskoder och header-fält förblir som tidigare. Det som skiljer sig åt är sättet HTTP-meddelanden skickas mellan webbläsare och webbserver. Till skillnad från HTTP/1.1 som är ett textprotokoll kommunicerar HTTP/2 genom att dela upp meddelanden i sk binära paket (binary frames). Den binära kommunikationen består av tre delar:

#### Ström

All kommunikation i HTTP/2 sker genom en enda förbindelse (TCP-connection) som kan skicka ett obegränsat antal strömmar. En ström är ett tvåvägsflöde av bytes som kan frakta ett eller flera meddelanden. Varje ström har sitt unika id.

#### Meddelande

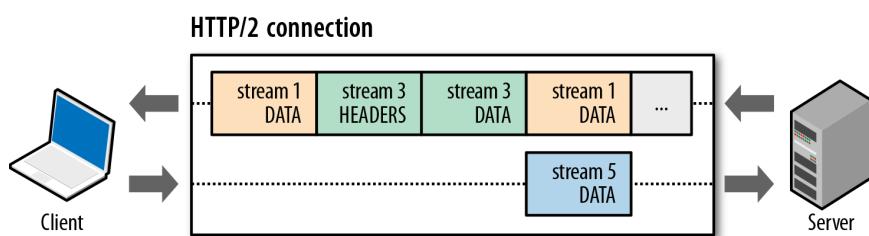
Ett meddelande är ett logiskt HTTP-meddelande, en förfrågan eller ett svar, som innehåller ett eller flera binära paket.

#### Binära paket

Varje paket fraktar en specifik typ av data och har en header som med hjälp av ett id berättar vilken ström den tillhör. Paket från olika strömmar kan blandas för att senare återförenas tack vare att de bär på information om vilken ström de tillhör.

#### *1.1.3.1 Multiplexing*

Som tidigare beskrivits är HTTP/1.1 tvungen att använda multipla förbindelser för att skicka data parallellt och dessa förbindelser kan bara skicka en resurs åt gången sk head-of-line-blocking. Genom att bryta ner HTTP-meddelanden i av varandra oberoende binära paket och låta dem blandas för att sedan till slut bygga dem samman igen elimineras problemet med head-of-line-blocking. Detta kallas multiplexing.



*Förfrågan-svar genom multiplexing och en gemensam förbindelse [7, kapitel 12, figur 12-3].*

#### *1.1.3.2 Server push*

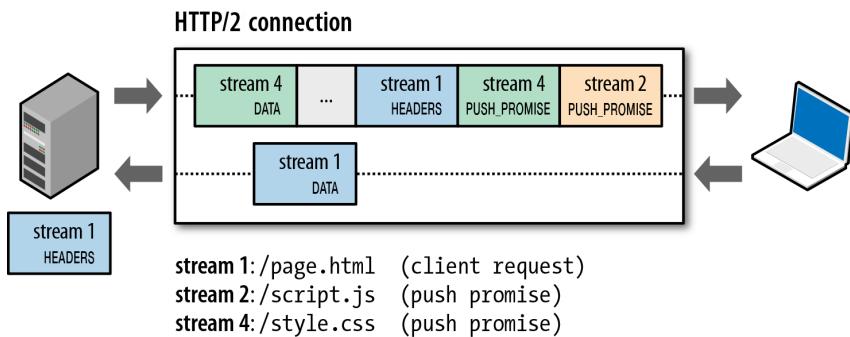
En annan nyhet med HTTP/2 är servernens möjlighet att skicka flera svar på en enda HTTP-förfrågan. Det innebär att servern kan pusha flera extra resurser utan att klienten explicit efterfrågat dem. Som det är nu, i HTTP/1.1, efterfrågar webbläsaren resurserna efterhand som DOM-dokumentet tolkas och servern svarar med att skicka en resurs per förfrågan. Om HTML-dokumentet t ex deklarerar att det behövs en style.css-fil skickar webbläsaren en förfrågan till servern efter den filen och servern svarar. För varje sådan deklaration av behövd resurs skickas en förfrågan och ett svar. En tidigare lösning för att minska laddningstider har varit att minska antalet resurser som behöver skickas till webbläsaren.

---

10. <https://www.ietf.org/wg/>

Ett sätt att göra det är genom sk resource inlining. Det vill säga att CSS- och Javascriptkod skrivs direkt i HTML-filen och på så sätt skapar utvecklaren en slags manuell push. Det finns dock flera fördelar med att HTTP/2 kan sköta push på egen hand:

- Pushade resurser kan cachas i webbläsaren.
- Pushade resurser kan återanvändas på flera ställen i applikationen.
- Pushade resurser kan multiplexas på samma sätt som andra resurser.
- Pushade resurser kan prioriteras olika av servern.
- Pushade resurser kan nekas av webbläsaren om de t ex redan är cachade.



*Servern initierar nya strömmar för push-resurser [7, kapitel 12, figur 12-5].*

#### 1.1.3.3 Header-komprimering

Alla överföringar med HTTP bär med sig en header. I HTTP/1.1 skickas denna header med metadata som ren text och kan innehålla ytterligare 500-800 bytes, ibland mer. För att dra ner på denna overheadkostnad och öka prestandan komprimerar HTTP/2 headern med HPACK-komprimering<sup>11</sup>.

HPACK tillåter att headerns olika fält kodas till statisk Huffman-kod<sup>12</sup> vilket reducerar deras individuella överförda storlek. Både webbläsaren och servern underhåller och uppdaterar en indexerad lista med tidigare kodade headerfält. Från denna lista hämtas sedan tidigare skickade och komprimerade fält och proceduren behöver alltså inte upprepas varje gång.

#### 1.1.3.4 Övergång till HTTP/2

Inga befintliga applikationer behöver skrivas om för att stödja HTTP/2 dåremot måste både webbläsaren och webbservern tala samma version för att överföringar ska ske genom HTTP/2. Är så inte fallet faller kommunikationen automatiskt tillbaka till HTTP/1.1. Vilka webbservrar och webbläsare som har stöd för HTTP/2 finns listade på Github<sup>13</sup> och caniuse.com<sup>14</sup>.

## 1.2 Syfte

Syftet med rapporten är att undersöka vilken effekt HTTP/2 har på en webbapplikations laddningstid. Hur stora blir skillnaderna i en befintlig applikation där webbläsare och webbserver kommunicerar genom HTTP/1.1 kontra HTTP/2 och skiljer sig effekten åt mellan webbläsare.

11. <https://tools.ietf.org/html/rfc7541>

12. <https://sv.wikipedia.org/wiki/Huffmankodning>

13. <https://github.com/http2/http2-spec/wiki/Implementations>

14. <http://caniuse.com/#feat=http2>

## 1.3 Frågeställning

Vilken effekt har HTTP/2 på en webbapplikations laddningstid?

## 1.4 Tidigare forskning

Då den nya versionen av protokollet är relativt ny finns ingen omfattande eller historisk forskning att ta del av. Betydelsen av HTTP/2 kommer troligtvis inte vara helt känd förrän om några år. Det finns dock mindre exeperiment att ta del av.

### 1.4.1 HTTP/2 is the future of the web, and it is here [10]

Javier Garza gjorde 2015 en demoapplikation åt Akamai<sup>15</sup> för att demonstrera skillnaden mellan HTTP/1.1 och HTTP/2. Han delade upp en bild i 397 fyrkanter och lät två iframes ladda samma resurser från två olika domäner. Den ena med stöd för HTTP/2 den andra utan. Vid tre olika besök på sidan, i Firefox/46.0, blev resultaten så olika att den enda slutsatsen som kan dras är att HTTP/2 har en kortare laddningstid, men hur mycket kortare den var varierade kraftigt vid de här tre besöken.

### 1.4.2 A Simple Comparison of HTTPS, SPDY and HTTP/2 [11]

HttpWatch genomförde ett laddningstest med HTTPS/1.1, SPDY och HTTP/2. En enkel applikation hämtades i Firefox och följande kunde konstateras:

- Headerstorleken var betydligt mindre i HTTP/2 än i de båda andra tack vare HPACK.
- Storleken på svarsmeddelandena (response messages) var minst i HTTP/2 när det gällde bilder, men störst för textresurser. Anledningen till att svarsmeddelandena var större för textresurser i HTTP/2 kunde de häröra till att Google:s servrar lägger på padding på dessa resurser.
- SPDY och HTTP/2 använder färre förbindelser (TCP connections) än HTTPS/1.1.
- Laddningstiden är kortast för HTTP/2, följt av SPDY och sist HTTPS/1.1. Vid en jämförelse laddade HTTP/2 21,9% snabbare än HTTPS/1.1 och 19,1% snabbare än SPDY. Anledningen till att HTTP/2 är snabbare än SPDY beror troligtvis på HPACK-kompressionen.

De drar slutsatsen att HTTP/2 troligtvis kommer öka prestandan avsevärt i jämförelse med HTTPS/1.1 och SPDY, men att den padding som läggs på i svarsmeddelandena för vissa resurser är ett område för potentiell oro.

---

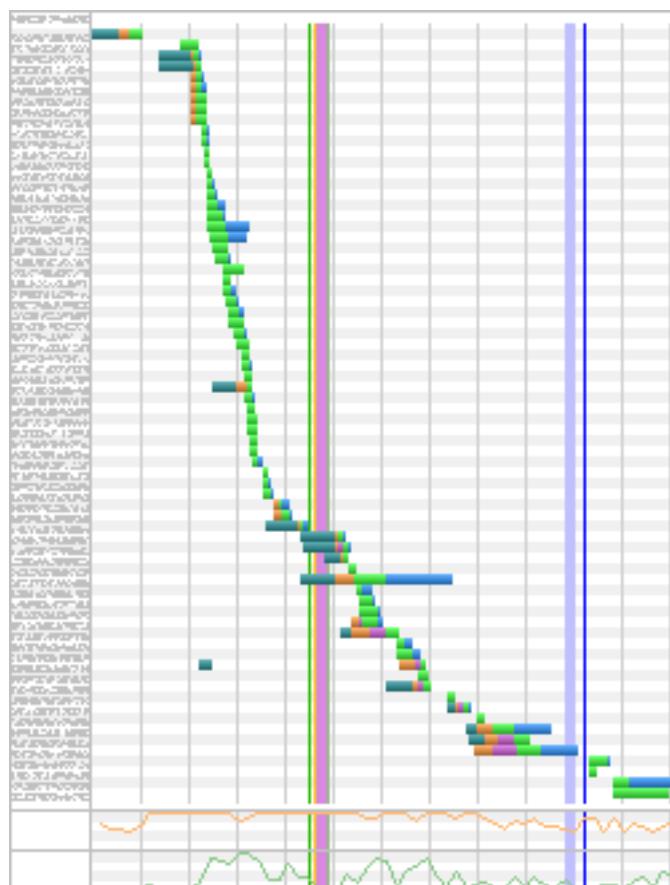
15. <https://www.akamai.com>

## 2 Metod

Eftersom det finns så pass få publicerade tester av skillnaden i laddningstid mellan HTTP/1.1 och HTTP/2 föll det sig naturligt att genomföra den här studien som ett exeriment. Syftet är att testa hur stor effekt, i sekunder HTTP/2 har i en befintlig webbapplikation. En applikations laddningstid kan delas in i flera beståndsdelar och för att kunna utreda om effekten är jämt fördelad över dessa delar eller var effekterna är högst respektive lägst behöver begreppet laddningstid definieras.

### 2.1 Laddningstid [12]

Med laddningstid menas tiden mellan det att en resurs efterfrågas och till dess att den är laddad till fullo. Varje resurs som behövs för att en applikations ska fungera har sin egen laddningstid och den delas upp i olika mätvärden. Ett diagram över en webbapplikations initiala laddning kan se ut enligt bilden nedan där laddningstiden visualiseras genom färgkodade streck och varje färg representerar en särskild beståndsdel av resursens totala laddningstid.



En skärmdump från ett test gjort med verktyget WebPageTest (se 2.4 Testmiljö).

#### 2.1.1 Vertikal linje

Det ljusgröna strecket indikerar när sidan först börjar rendera. Detta betyder inte att det som är viktigast på sidan renderas först utan det kan vara vad som helst. En bakgrundsfärg, en rubrik eller en del av en bild. För användaren skickar det dock ett viktigt budskap: något händer.

Det blåa strecket berättar när sidan är komplett laddad. I verkligheten betyder det att webbläsaren skickar ett sk onLoad event när de flesta av sidans resurser har laddats. Det är ett

mått som används för att mäta en webbapplikations snabbhet, men innebär inte nödvändigtvis att det till fullo går att interagera med sidan.

Där bilden tar slut längst ut till höger är sidan med alla dess resurser laddad till fullo.

### 2.1.2 Horisontell linje

Mörkgrön = DNS-kontroll

Den här delen avser den tid det tar webbläsaren att kolla upp domänen som resursen hämtas från.

Orange = TCP connection, förbindelse

Den tid det tar för både webbläsaren och webbservern att öppna sina respektive förbindelser för vidare kommunikation.

Ljusgrön = Tid till första byte

Avser tiden det tar för webbläsaren att ta emot resursens första byte. Användarens internetuppkoppling spelar roll här, men även tiden det tar för webbservern att räkna ut vilken resurs som ska skickas och hur långt det fysiska avståndet är mellan webbläsaren och webbservern.

Blå = Resursen laddad

Avser den tid det tar att till fullo ladda en resurs från webbservern till webbläsaren.

## 2.2 Testparametrar

Studien kommer att redovisa fyra olika parametrar.

- Tid till första byte
- Tid till renderingsstart
- Tid till komplett laddning
- Tid till fullständig laddning

## 2.3 Baslinje

Baslinjen utgörs av en befintlig webbapplikation levererad från server till klient över HTTPS/1.1. Applikationens kod finns publicerad på Github<sup>16</sup>. Inga prestandaoptimerade lösningar har implementerats förutom att vissa ramverk och bibliotek laddas via CDN, Content Delivery Network. Vilket innebär att dessa resurser automatiskt cachas av de webbläsare som avses testas.

## 2.4 Testmiljö

WebPagetest<sup>17</sup> är ett gratisverktyg främst utvecklat av Google som ett led i deras strävan att göra webben snabbare, [13]. Verktyget är enbart utvecklat för att mäta webbapplikationers laddningstid och genererar resultatet uppdelat i ovan nämnda testparametrar. På WebPagetest används testklienter placerade runt om i världen och som användare går det att välja vilken plats och vilka webbläsare som ska testas.

---

16. [https://github.com/js223kz/TRAFFICREPORTS\\_js223kz](https://github.com/js223kz/TRAFFICREPORTS_js223kz)

17. <http://www.webpagetest.org/>

Enbart en applikation kan testas mot en specifik klient åt gången för att säkerställa att resultatet inte blir missvisande på grund av trafik från olika håll. Det här experimentet kommer att använda testklient och webbserver placerade i Frankfurt och två webbläsare, Chrome/50.0.2661.102 och Firefox/46.0. För ytterligare jämförelse redovisas resultatet för två typer av uppkopplingar, långsamt 3G och DSL.

## 2.5 Teknisk specifikation

Applikationen är skriven i: Javascript  
Ramverk: Angular 1.4.5, Express 4.13.4  
Programsystem: Node.js 5.6.0  
Webbserver: Nginx 1.10  
Applikationerna publiceras på: Digital Ocean  
Protokoll: HTTP över säker HTTPS

## 2.6 Genomförande

Samma applikation kommer att publiceras under två olika domännamn. Kod och implementation är identisk. Den enda skillnaden finns i konfigurationen av webbservsen. Testerna av HTTP/1.1 och HTTP/2 kommer att köras växelvis och styrks genom skärmdumpar av testresultaten (se bilagor A-D). Alla tester kommer att genomföras tre gånger. Slutligen sker en sammanställning av resultaten.

### 2.6.1 Konfiguration av respektive webbserver

HTTP/1.1  
listen 443 ssl default\_server;  
listen [::]:443 ssl default\_server;

HTTP/2  
listen 443 ssl http2 default\_server;  
listen [::]:443 ssl http2 default\_server;

### 2.6.2 Testsvit

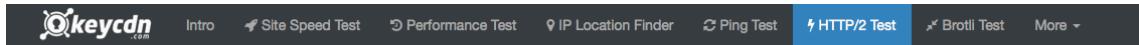
Chrome, HTTP/1.1, DSL - 3 gånger  
Chrome, HTTP/2, DSL - 3 gånger  
Firefox, HTTP/1.1, DSL - 3 gånger  
Firefox, HTTP/2, DSL - 3 gånger  
Chrome, HTTP/1.1, långsamt 3G - 3 gånger  
Chrome, HTTP/2, långsamt 3G - 3 gånger  
Firefox, HTTP/1.1, långsamt 3G - 3 gånger  
Firefox, HTTP/2, långsamt 3G - 3 gånger

### 2.6.3 Verifiering av respektive protokoll

Verifiering av respektive protokoll har skett med ett testverktyg på keycdn.com<sup>18</sup> (se nästa sida).

---

18. <https://tools.keycdn.com/http2-test>



## ⚡ HTTP/2 Test Verify HTTP/2.0 Support

Online **HTTP/2 test** - Verify if your server or CDN supports HTTP/2.

URL   Public

HTTP/2 Test Result http-1-1.tk

Negative! http-1-1.tk does not support HTTP/2.0. Supported protocols: http/1.1

*Skärmdump av test med HTTP/1.1*



## ⚡ HTTP/2 Test Verify HTTP/2.0 Support

Online **HTTP/2 test** - Verify if your server or CDN supports HTTP/2.

URL   Public

HTTP/2 Test Result bythebeach.se

Yeah! bythebeach.se supports HTTP/2.0. Share via: [Twitter](#)

*Skärmdump av test med HTTP/2*

## 2.7 Metoddiskussion

Då rapporten avser utreda effekterna av HTTP/2 i sekunder och eftersom det finns så lite presenterade tester föll det sig naturligt att besvara frågeställningen med en experimentell metod. Genom att använda ett testverktyg som WebPagetest går det att avläsa flera kristiska punkter i laddningstiden och få dessa presenterade på ett överskådligt och tydligt sätt och på så vis eliminera personliga tolkningar som annars utgör en risk om applikationen testas i den egna webbläsaren. Eftersom internetuppkoppling kan variera i styrka kommer varje test att utföras tre gånger och både individuella resultat och sammanställningar kommer att redoviseras. Genom att inte utveckla en applikation enkom för experimentets syfte riskerar inte applikationens beståndsdelar baseras på utförarens personliga preferenser. Samtidigt kan det utgöra en risk då det inte tagits hänsyn till eventuella, för HTTP/2, prestandahöjande implementeringar som möjligtvis kan generera ett annorlunda resultat.

### 3 Resultat

Sammanställningen av resultaten baseras på enskilda tester (se bilaga A-D) och alla testkörningar gjordes under samma dag, 2016-05-19. Vid de tillfällen effekterna av testerna presenteras i procent innebär ett positiv värde en minskning av laddningstiden med HTTP/2. Negativa värden betyder således att laddningstiden ökat med HTTP/2.

Genomgående för båda uppkopplingarna och båda laddningarna uppvisar Chrome helt klart bättre effekt av HTTP/2 vid testparametern tid till första byte än Firefox. Firefox visade mycket liten eller till och med negativ effekt vid samma testparameter. I Firefox blir effekten av HTTP/2 absolut bäst vid den andra laddningen på en långsam 3G-uppkoppling medan Chrome minskar sina laddningstider bäst på en snabbare DSL-uppkoppling vid första laddningen. Oavsett uppkoppling uppvisar Chrome bäst effekt av HTTP/2 på första laddningen medan det tvärtom för Firefox gäller den andra laddningen.

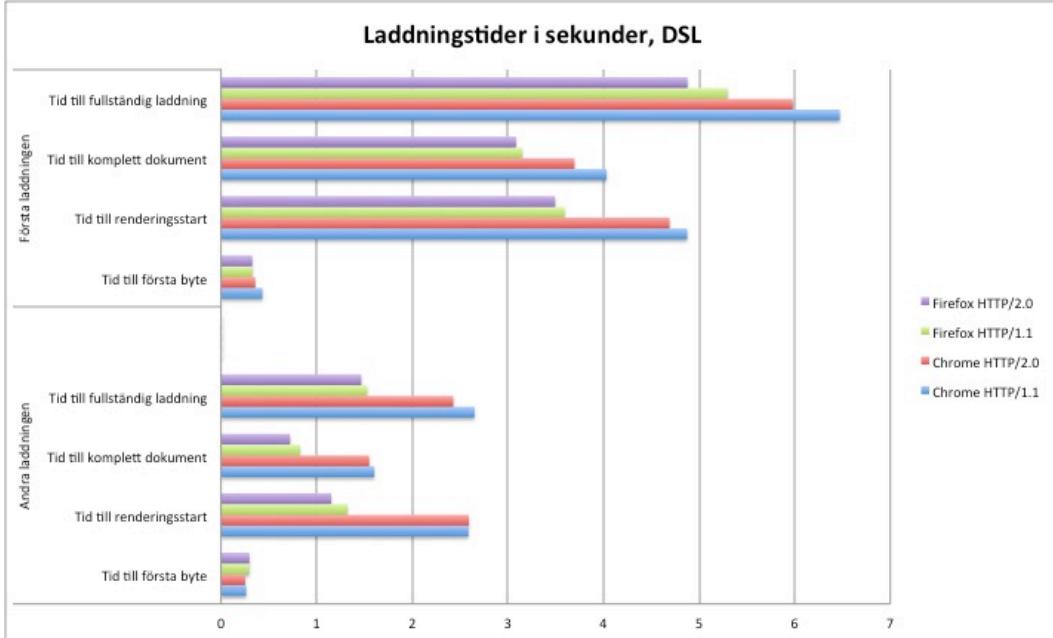
Vid tre av fyra testparametrar oavsett uppkoppling är det Firefox som minskar laddningstiderna mest med HTTP/2. Intressant att notera är att oavsett protokollversion är Firefox den snabbare webbläsaren.

Den testparameter som uppvisar bäst effekt av HTTP/2 är tid till komplett dokument vilket är det värde som ofta får representera en webbapplikations snabbhet om tester genomförs i den egna webbläsaren. Totalt sett, oavsett webbläsare, uppkoppling eller laddning minskar laddningstiden till komplett dokument med strax över 6,5%.

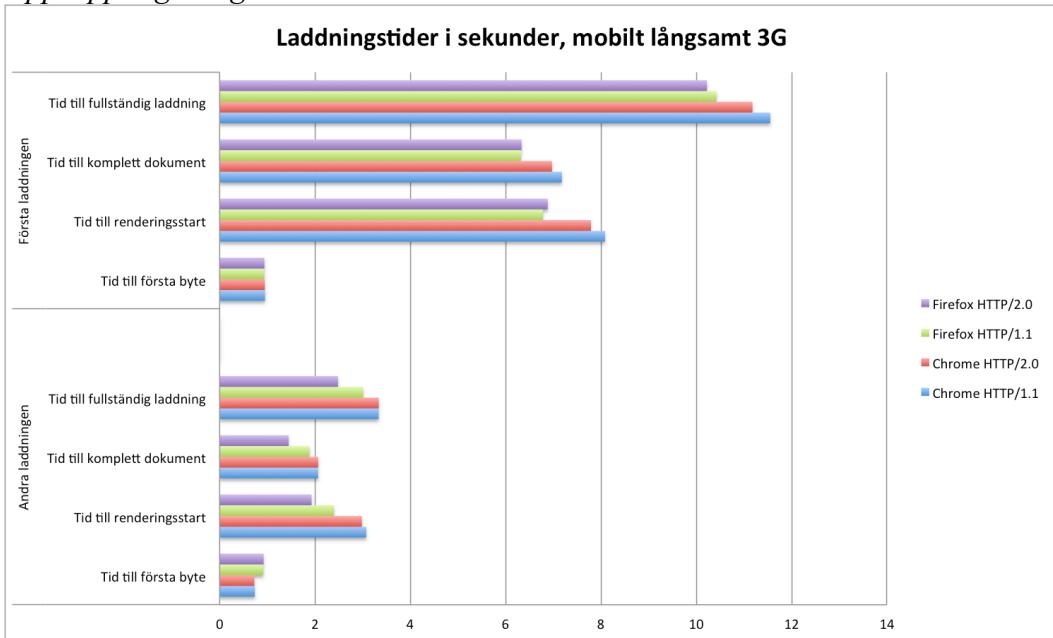
### 3.1 Laddningstider i sekunder

Resultatet presenteras per uppkopplingstyp. Diagrammen är uppdelade i två sektioner, första och andra laddningen. Varje stapel representerar en webbläsare och ett protokoll. Resultatet är ett medianvärde av alla tre testkörningar. Medianvärde innebär mindre risk för ett missvisande slutresultat om ett enskilt testresultat är markant avvikande.

*Uppkoppling: DSL*



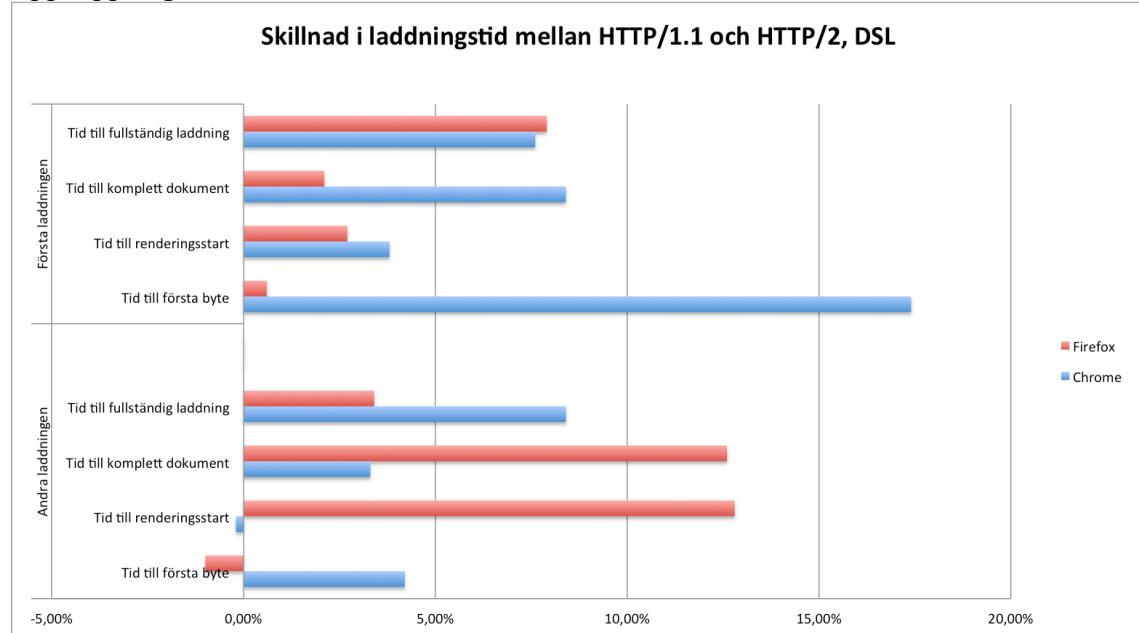
*Uppkoppling: långsamt 3G*



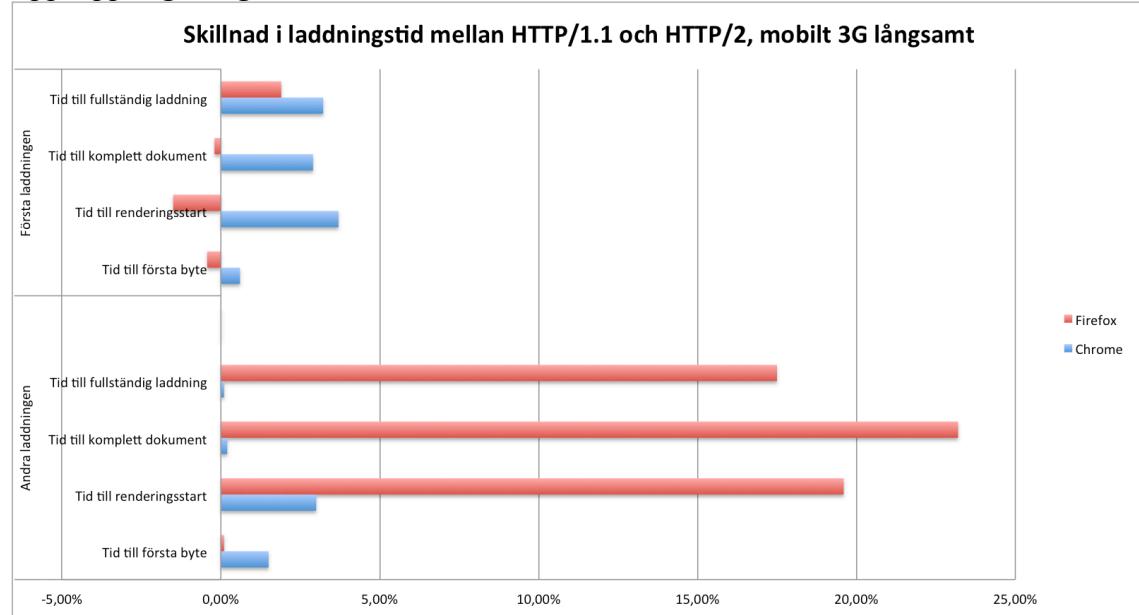
### 3.2 Laddningstider skillnad i %

Diagrammen visa skillnaden i laddningstider, vid implementation av HTTP/2, uttryckt i procent. Resultaten är uppdelade efter uppkopplingstyp och presenterar både första och andra laddningen för respektive webbläsare. Decimaler är avrundade till närmaste 10-tal.

#### Uppkoppling: DSL

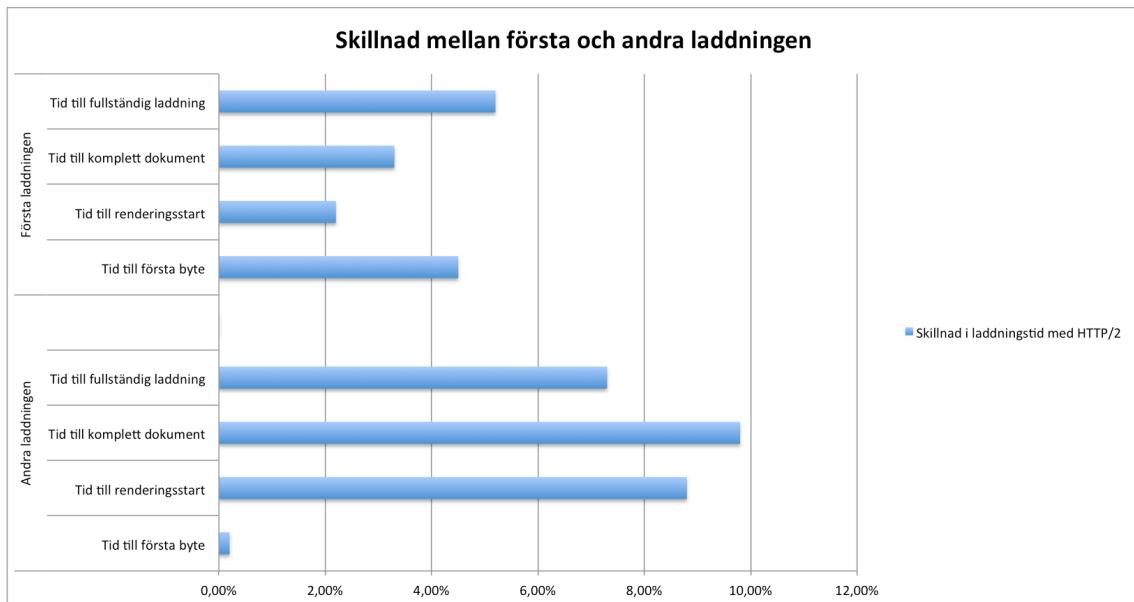


#### Uppkoppling: långsamt 3G



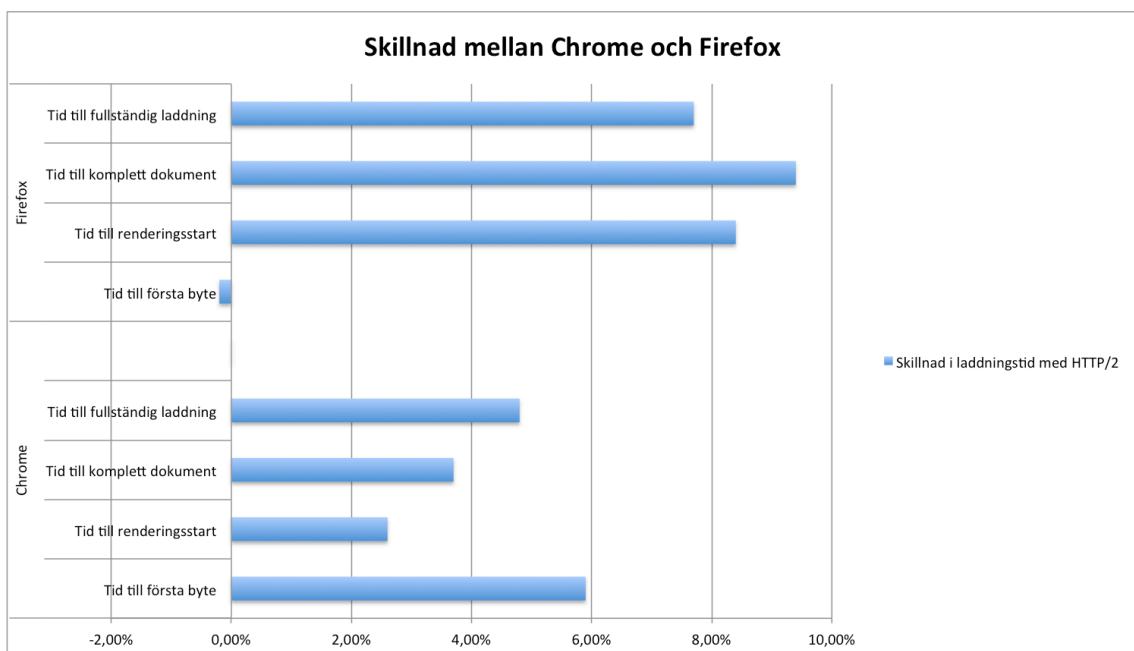
### 3.3 Medelvärde av respektive laddning

Resultatet är ett medelvärde av samtliga webbläsare och uppkopplingstyper. Diagrammet visar effekten vid en implementation av HTTP/2 och är uppdelat i två sektioner, första och andra laddningen. Decimaler är avrundade till närmaste 10-tal.



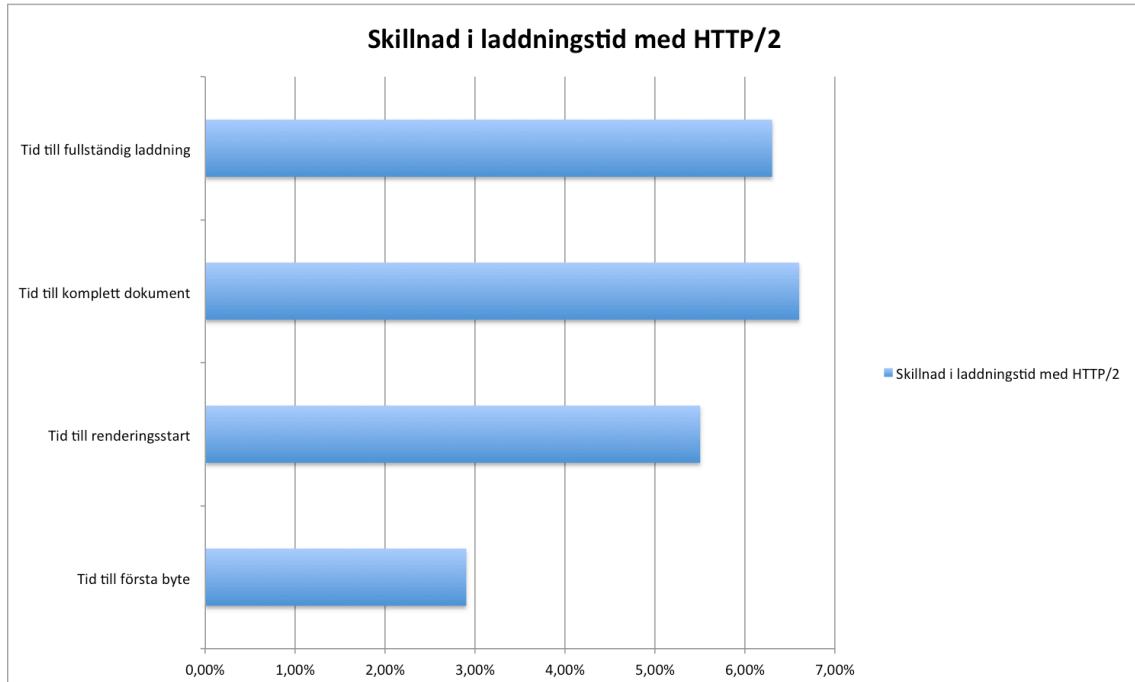
### 3.4 Medelvärde för respektive webbläsare

Resultatet är ett medelvärde av båda laddningarna och båda uppkopplingstyperna uppdelat per webbläsare. Decimaler är avrundade till närmaste 10-tal.



### 3.5 Medelvärde av den totala effekten

Resultatet är ett medelvärde av samtliga webbläsare, laddningar och uppkopplingstyper uppdelat per testparameter. Decimaler är avrundade till närmaste 10-tal.



## 4 Slutsats

Som svar på frågeställningen: Vilken effekt har HTTP/2 på en webbapplikations laddningstid så är svaret att effekten är positiv. I några enstaka fall uppvisar testparametrar ett negativt resultat, men de kan konstateras vara små i jämförelse med vinningen. Implementationen av HTTP/2, åtminstone i det här fallet, är enkel och kräver ingen omfattande insats av en utvecklare. Det finns därför ingen anledning att avstå från HTTP/2. Något som dock behöver utredas är behovet av att webbläsare och webbserver kommunlicerar via det krypterade HTTPS-protokollet, som krävs för att HTTP/2 ska fungera i alla webbläsare, då detta i sig ger ökade laddningstider.

Effekten av HTTP/2 för den här webbapplikationen blev inte så höga som vid testerna i HttpWatch:s undersökning (se avsnitt 1.42), men det kan bero på implementationen och/eller applikationens beståndsdelar. Nginx stödjer ännu inte server push och det är omöjligt att säga hur mycket det skulle påverkat laddningstiderna. Implementationen stödjer heller inte ALPN, Application-Layer Protocol Negotiation, som HTTP/2 använder för att minska laddningstiderna över HTTPS. Applikationen som testades i denna studie laddar en del resurser via CDN och en del av dem hämtas från Google:s servrar. Det framgår inte av HttpWatch:s studie hur många, om ens några, externa resurser deras applikation hämtar, men det kan vara så att applikationen i den här studien får större svarsmeddelanden på grund av den padding Google:s servrar lägger på textresurser över HTTP/2.

Ser man till skillnaderna mellan webbläsarna och det faktum att Chrome får bäst effekt av HTTP/2 vid första laddningen medan Firefox uppvisar bäst effekt vid andra laddningen kan det vara så att Chromes implementation av HTTP/2 snabbare laddar externa resurser via CDN medan Firefox implementation drar mer nytta av att alla resurser kommer från samma server.

Det är lätt att ryckas med i den allmänna entusiasmen och tro att HTTP/2 är en magisk lösning för att dramatiskt minska laddningstiderna i en applikation. Personligen tror jag att det är en bra början, men det kommer ta flera år av testande och tweakande av implementationen på webbservrar, i webbläsare och webbapplikationer innan vi på riktigt kan dra nytta av fördelarna med HTTP/2.

### 4.1 Fortsatt arbete

- Blir effekten av HTTP/2 större ju fler resurser som behöver laddas?
- Blir effekten av HTTP/2 större om ALPN implementeras?
- Blir effekten större om webbservern stödjer server push?

# Ordlista

## Cookie

Även kallad webbkaka, kakfil och kaka är en liten textbaserad datafil som en webbserver kan be att få spara på webbplatsbesökarens dator. Kakornas innehåll skickas från webbläsaren till webbservern med varje förfrågan för att servern ska kunna hålla reda på information om webbplatsbesökaren. Syftet kan vara att en besökare inte ska behöva logga in mellan varje förfrågan eller ställa in språkval och liknande varje gång de besöker sidan.

## Cache

Från franska *cacher*, gömma, innebär att data mellanlagras i webbläsarens minne så att webbläsaren inte behöver hämta all data på nytt varje gång en användare besöker en webbsida. Vanligt är att man cachar filer som inte ändras så ofta för att på så sätt minska en webbsidas laddningstid.

## Header

En header bär med sig information om en förfrågan eller ett svar. Den kan innehålla en stor mängd data om allt från vilken webbläsare användaren har, vilken typ av förfrågan det rör sig om eller vilken typ av svar den accepterar att få tillbaka.

## HTML

HyperText Markup Language. Det språk som används för att bygga upp en webbsidas struktur.

## HTTP

Hypertext Transfer Protocol. Det protokoll som används för att skicka resurser över internet.

## Iframe

En Iframe (Inline Floating Frames) innebär att man placerar en ram på en webbsida i vilken man kan visa en annan HTML-sidas innehåll.

## Metadata

Information om information eller data om data. Normalt används metadata eller metainformation för att beskriva innehållet och/eller strukturen för en viss datasamling ur något perspektiv. Det kan t ex vara en beskrivning av en webbsidas innehåll.

## TCP-connection

TCP står för *Transmission Control Protocol*. TCP är vad man brukar kalla för *connection oriented*, det vill säga det skapas en förbindelse (connection) mellan sändare och mottagare, precis som under ett telefonsamtal. Genom denna förbindelse transporteras sedan alla förfrågningar och svar mellan t ex en webbläsare och en webbserver.

## User-agent

På svenska, användaragent, och finns t ex i webbläsare. När en användaragent besöker en webbsida, identifierar den sig själv, genom att skicka med information i ett HTTP-anrop. Informationen består bland annat av webbläsartyp och operativsystem.

## URI

Unified Resource Identifier också känt som URL. En webbsidas unika adress.

# Referenser

- [1] info.cern.ch ([Copyright](#) CERN 2008 - Web Communications). Information Management: A Proposal. Internet. Tillgänglig: <http://info.cern.ch/Proposal.html>
- [2] World Wide Web Foundation. History of the Web. Internet. Tillgänglig: <http://webfoundation.org/about/vision/history-of-the-web/>
- [3] Radware (2015, 23:e mars. 2015 Spring State of the Union: Ecommerce Page Speed & Web Performance Infographic. Internet. Tillgänglig: <https://www.radware.com/Documents/Infographics/SOTU-EcommercePageSpeed-spring-2015/>
- [4] Daniel Stenberg, Gitbook. http2 explained. Internet. Tillgänglig: <https://bagder.gitbooks.io/http2-explained/content/sv/index.html>
- [5] Jillian D'Onfro och Alyson Shontell (Business Insider UK, 2016, 31:e januari). What 25 popular websites used to look like. Internet. Tillgänglig: <http://uk.businessinsider.com/popular-websites-then-and-now-2016-1?r=US&IR=T>
- [6] YSlow. Web Performance Best Practices and Rules. Internet. Tillgänglig: <http://yslow.org/>
- [7] Ilya Grigorik, O'Reilly Media, Inc. (2013). High Performance Browser Networking. Internet. Tillgänglig: <http://chimera.labs.oreilly.com/books/123000000545/index.html>
- [8] Jake Brutlag, Google, Inc. (2009, 22:e juni). Speed Matters for Google Web Search. Internet. Tillgänglig: [http://services.google.com/fh/files/blogs/google\\_delayexp.pdf](http://services.google.com/fh/files/blogs/google_delayexp.pdf)
- [9] Official Google Webmaster (2010, 9:e april). Using site speed in web search ranking. Internet. Tillgänglig: <https://webmasters.googleblog.com/2010/04/using-site-speed-in-web-search-ranking.html>
- [10] Javier Garza (2015, 25:e augusti). HTTP/2 is the future of the Web, and it is here! Internet. Tillgänglig: <https://http2.akamai.com/demo/>
- [11] HttpWatch (2015, 16:e januari). A Simple Performance Comparison of HTTPS, SPDY and HTTP/2. Internet. Tillgänglig: <http://blog.httpwatch.com/2015/01/16/a-simple-performance-comparison-of-https-spdy-and-http2/>
- [12] Tammy Everts, Web Performance Today (2014, 18:e mars). Waterfalls 101: How to use a waterfall chart to diagnose your website's performance pains. Internet. Tillgänglig: <http://www.webperfomancetoday.com/2014/03/18/waterfalls-101-how-to-use-a-waterfall-chart-to-diagnose-performance-pains/>
- [13] Webpagetest. About WebPagetest.org. Internet. Tillgänglig: <http://www.webpagetest.org/about>

# Bilagor

Bilaga A Tester Chrome, DSL 1,5Mbps/38 Kbps 50 ms RTT

Testkörning 1, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 08:34:46

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot																																																
<p>Tester: i-6ed452af <a href="#">Re-run the test</a></p> <p><a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a></p>																																																					
<table border="1"><thead><tr><th colspan="5">Document Complete</th><th colspan="5">Fully Loaded</th></tr><tr><th>Load Time</th><th>First Byte</th><th>Start Render</th><th>Speed Index</th><th>DOM Elements</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Cost</th></tr></thead><tbody><tr><td>First View</td><td>4.032s</td><td>0.448s</td><td>4.875s</td><td>5928</td><td>163</td><td>4.032s</td><td>18</td><td>437 KB</td><td>6.471s</td><td>38</td><td>692 KB</td><td>\$\$--</td></tr><tr><td>Repeat View</td><td>1.602s</td><td>0.262s</td><td>2.589s</td><td>2756</td><td>163</td><td>1.602s</td><td>9</td><td>11 KB</td><td>2.651s</td><td>17</td><td>14 KB</td><td></td></tr></tbody></table>						Document Complete					Fully Loaded					Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	First View	4.032s	0.448s	4.875s	5928	163	4.032s	18	437 KB	6.471s	38	692 KB	\$\$--	Repeat View	1.602s	0.262s	2.589s	2756	163	1.602s	9	11 KB	2.651s	17	14 KB	
Document Complete					Fully Loaded																																																
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost																																										
First View	4.032s	0.448s	4.875s	5928	163	4.032s	18	437 KB	6.471s	38	692 KB	\$\$--																																									
Repeat View	1.602s	0.262s	2.589s	2756	163	1.602s	9	11 KB	2.651s	17	14 KB																																										

Testkörning 2, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 09:20:05

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot																																																
<p>Tester: i-6ed452af <a href="#">Re-run the test</a></p> <p><a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a></p>																																																					
<table border="1"><thead><tr><th colspan="5">Document Complete</th><th colspan="5">Fully Loaded</th></tr><tr><th>Load Time</th><th>First Byte</th><th>Start Render</th><th>Speed Index</th><th>DOM Elements</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Cost</th></tr></thead><tbody><tr><td>First View</td><td>4.206s</td><td>0.333s</td><td>5.084s</td><td>6204</td><td>163</td><td>4.206s</td><td>18</td><td>437 KB</td><td>6.613s</td><td>38</td><td>688 KB</td><td>\$\$--</td></tr><tr><td>Repeat View</td><td>1.630s</td><td>0.260s</td><td>2.481s</td><td>2714</td><td>163</td><td>1.630s</td><td>9</td><td>11 KB</td><td>2.705s</td><td>17</td><td>14 KB</td><td></td></tr></tbody></table>						Document Complete					Fully Loaded					Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	First View	4.206s	0.333s	5.084s	6204	163	4.206s	18	437 KB	6.613s	38	688 KB	\$\$--	Repeat View	1.630s	0.260s	2.481s	2714	163	1.630s	9	11 KB	2.705s	17	14 KB	
Document Complete					Fully Loaded																																																
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost																																										
First View	4.206s	0.333s	5.084s	6204	163	4.206s	18	437 KB	6.613s	38	688 KB	\$\$--																																									
Repeat View	1.630s	0.260s	2.481s	2714	163	1.630s	9	11 KB	2.705s	17	14 KB																																										

Testkörning 3, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 11:07:38

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot																																																
<p>Tester: i-6ed452af <a href="#">Re-run the test</a></p> <p><a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a></p>																																																					
<table border="1"><thead><tr><th colspan="5">Document Complete</th><th colspan="5">Fully Loaded</th></tr><tr><th>Load Time</th><th>First Byte</th><th>Start Render</th><th>Speed Index</th><th>DOM Elements</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Time</th><th>Requests</th><th>Bytes In</th><th>Cost</th></tr></thead><tbody><tr><td>First View</td><td>3.805s</td><td>0.432s</td><td>4.687s</td><td>5691</td><td>163</td><td>3.805s</td><td>18</td><td>437 KB</td><td>6.052s</td><td>38</td><td>688 KB</td><td>\$\$--</td></tr><tr><td>Repeat View</td><td>1.600s</td><td>0.276s</td><td>2.587s</td><td>2766</td><td>163</td><td>1.600s</td><td>9</td><td>11 KB</td><td>2.571s</td><td>17</td><td>16 KB</td><td></td></tr></tbody></table>						Document Complete					Fully Loaded					Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	First View	3.805s	0.432s	4.687s	5691	163	3.805s	18	437 KB	6.052s	38	688 KB	\$\$--	Repeat View	1.600s	0.276s	2.587s	2766	163	1.600s	9	11 KB	2.571s	17	16 KB	
Document Complete					Fully Loaded																																																
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost																																										
First View	3.805s	0.432s	4.687s	5691	163	3.805s	18	437 KB	6.052s	38	688 KB	\$\$--																																									
Repeat View	1.600s	0.276s	2.587s	2766	163	1.600s	9	11 KB	2.571s	17	16 KB																																										

## Testkörning 1, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 08:40:43

Summary Details Performance Review Content Breakdown Domains Screen Shot

Tester: i-6ed452af  
Re-run the test

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	3.662s	0.375s	4.599s	5604	163	3.662s	18	435 KB	5.979s	38	685 KB	\$\$—
Repeat View	1.549s	0.251s	2.592s	2660	163	1.549s	9	9 KB	2.428s	17	12 KB	

## Testkörning 2, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 09:29:27

Summary Details Performance Review Content Breakdown Domains Screen Shot

Tester: i-6ed452af  
Re-run the test

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	3.695s	0.319s	4.690s	5349	163	3.695s	18	435 KB	5.674s	38	685 KB	\$\$—
Repeat View	1.552s	0.233s	2.494s	2710	163	1.552s	9	9 KB	2.425s	17	12 KB	

## Testkörning 3, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - DSL  
2016-05-19 11:13:51

Summary Details Performance Review Content Breakdown Domains Screen Shot

Tester: i-6ed452af  
Re-run the test

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	4.006s	0.357s	5.080s	5987	163	4.006s	18	435 KB	6.271s	38	685 KB	\$\$—
Repeat View	1.544s	0.257s	2.595s	2807	163	1.544s	9	9 KB	2.583s	17	13 KB	

## Sammanställning protokoll: HTTP/1.1

Testkörning 1 kl 08:34	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	0,448 s	4,875 s	4,032 s	6,471 s
2:a laddning	0,262 s	2,589 s	1,602 s	2,651 s
Testkörning 2 kl 09:20				
1:a laddning	0,333 s	5,084 s	4,206 s	6,613 s
2:a laddning	0,260 s	2,481 s	1,630 s	2,705 s
Testkörning 3 kl 11:07				
1:a laddning	0,432 s	4,687 s	3,805 s	6,052 s
2:a laddning	0,276 s	2,587 s	1,600 s	2,571 s
Medianvärden				
1:a laddning	0,432 s	4,875 s	4,032 s	6,471 s
2:a laddning	0,262 s	2,587 s	1,602 s	2,651 s

## Sammanställning protokoll: HTTP/2

Testkörning 1 kl 08:40	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	0,375 s	4,599 s	3,662 s	5,979 s
2:a laddning	0,251 s	2,592 s	1,549 s	2,428 s
Testkörning 2 kl 09:29				
1:a laddning	0,319 s	4,690 s	3,695 s	5,674 s
2:a laddning	0,233 s	2,494 s	1,552 s	2,425 s
Testkörning 3 kl 11:13				
1:a laddning	0,357 s	5,080 s	4,006 s	6,271 s
2:a laddning	0,257 s	2,595 s	1,544 s	2,583 s
Medianvärden				
1:a laddning	0,357 s	4,690 s	3,695 s	5,979 s
2:a laddning	0,251 s	2,592 s	1,549 s	2,428 s

## Bilaga B Tester Firefox, DSL 1,5Mbps/38 Kbps 50 ms RTT

### Testkörning 1, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 09:02:32

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?					
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN
						A	A	A	N/A	F	X

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

						Document Complete			Fully Loaded								
						Time	Requests	Bytes In	Time	Requests	Bytes In	Cost					
First View						3.142s	0.347s	3.591s	4381	26	3.142s	19	408 KB	5.010s	39	631 KB	\$\$—
Repeat View						0.752s	0.286s	1.192s	1288	26	0.752s	11	13 KB	1.592s	18	39 KB	

### Testkörning 2, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 09:34:38

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?					
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN
						A	A	A	N/A	F	X

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

						Document Complete			Fully Loaded								
						Time	Requests	Bytes In	Time	Requests	Bytes In	Cost					
First View						3.551s	0.328s	3.797s	4765	26	3.551s	19	408 KB	5.298s	39	631 KB	\$\$—
Repeat View						0.914s	0.293s	1.395s	1462	26	0.914s	11	20 KB	1.497s	18	38 KB	

### Testkörning 3, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 11:18:46

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?					
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN
						A	A	A	N/A	F	X

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

						Document Complete			Fully Loaded								
						Time	Requests	Bytes In	Time	Requests	Bytes In	Cost					
First View						3.150s	0.319s	3.415s	4421	26	3.150s	19	408 KB	5.383s	39	631 KB	\$\$—
Repeat View						0.825s	0.297s	1.321s	1613	26	0.825s	11	11 KB	1.526s	18	39 KB	

## Testkörning 1, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 08:56:17

Need help improving?

A	A	A	N/A	F	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

Tester: i-6ed452af  
[Re-run the test](#)

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded				
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	3.084s	0.326s	3.493s	4274	26	3.084s	19	406 KB	4.878s	39	627 KB	\$\$--
Repeat View	0.701s	0.285s	1.152s	1282	26	0.701s	10	6 KB	1.196s	18	34 KB	

## Testkörning 2, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 09:32:01

Need help improving?

A	A	A	N/A	F	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

Tester: i-6ed452af  
[Re-run the test](#)

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded				
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	3.040s	0.340s	3.308s	4241	26	3.040s	19	406 KB	4.849s	39	627 KB	\$\$--
Repeat View	0.737s	0.300s	1.227s	1334	26	0.737s	10	6 KB	1.465s	18	34 KB	

## Testkörning 3, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - DSL  
2016-05-19 11:25:16

Need help improving?

A	A	A	N/A	F	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

Tester: i-6ed452af  
[Re-run the test](#)

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Document Complete			Fully Loaded				
					Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	4.600s	0.319s	5.335s	5910	26	4.600s	19	416 KB	6.426s	39	637 KB	\$\$--
Repeat View	0.721s	0.296s	1.076s	1188	26	0.721s	10	6 KB	1.468s	18	34 KB	

## Sammanställning protokoll: HTTP/1.1

Testkörning 1 kl 09:02	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	0,347 s	3,591 s	3,142 s	5,010 s
2:a laddning	0,286 s	1,192 s	0,752 s	1,592 s
Testkörning 2 kl 09:34				
1:a laddning	0,328 s	3,797 s	3,551 s	5,298 s
2:a laddning	0,293 s	1,395 s	0,914 s	1,497 s
Testkörning 3 kl 11:18				
1:a laddning	0,319 s	3,415 s	3,150 s	5,383 s
2:a laddning	0,297 s	1,321 s	0,825 s	1,526 s
Medianvärden				
1:a laddning	0,328 s	3,591 s	3,150 s	5,298 s
2:a laddning	0,293 s	1,321 s	0,825 s	1,526 s

## Sammanställning protokoll: HTTP/2

Testkörning 1 kl 08:56	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	0,326 s	3,493 s	3,084 s	4,878 s
2:a laddning	0,285 s	1,152 s	0,701 s	1,196 s
Testkörning 2 kl 09:32				
1:a laddning	0,340 s	3,308 s	3,040 s	4,849 s
2:a laddning	0,300 s	1,227 s	0,737 s	1,465 s
Testkörning 3 kl 11:25				
1:a laddning	0,319 s	5,335 s	4,600 s	6,426 s
2:a laddning	0,296 s	1,076 s	0,721 s	1,468 s
Medianvärden				
1:a laddning	0,326 s	3,493 s	3,084 s	4,878 s
2:a laddning	0,296 s	1,152 s	0,721 s	1,465 s

## Bilaga C Tester Chrome, långsamt 3G

### Testkörning 1, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 12:29:22

Tester: i-6ed452af  
[Re-run the test](#)

Need help improving?

	Document Complete	Fully Loaded		
	Time	Requests	Bytes In	Cost
First View	7.179s	18	437 KB	\$---
Repeat View	2.050s	9	11 KB	3.335s 17 14 KB

Raw page data - Raw object data  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

### Testkörning 2, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 12:58:07

Tester: i-6ed452af  
[Re-run the test](#)

Need help improving?

	Document Complete	Fully Loaded		
	Time	Requests	Bytes In	Cost
First View	7.226s	18	437 KB	\$---
Repeat View	2.125s	9	11 KB	3.438s 17 14 KB

Raw page data - Raw object data  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

### Testkörning 3, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 13:12:33

Tester: i-6ed452af  
[Re-run the test](#)

Need help improving?

	Document Complete	Fully Loaded		
	Time	Requests	Bytes In	Cost
First View	7.053s	18	437 KB	\$---
Repeat View	2.064s	9	11 KB	3.265s 17 14 KB

Raw page data - Raw object data  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

## Testkörning 1, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 12:48:38

Tester: i-6ed452af

Need help improving?

	A	A	A	N/A	F	X
First Byte Time						
Keep-alive Enabled						
Compress Transfer						
Compress Images						
Cache static content						
Effective use of CDN						

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

	Document Complete			Fully Loaded								
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	6.972s	0.946s	7.789s	9952	163	6.972s	18	435 KB	10.926s	38	685 KB	<del>\$\$--</del>
Repeat View	2.081s	0.721s	2.983s	3301	163	2.081s	9	9 KB	3.428s	17	13 KB	

## Testkörning 2, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 13:00:28

Tester: i-6ed452af

Need help improving?

	A	A	A	N/A	F	X
First Byte Time						
Keep-alive Enabled						
Compress Transfer						
Compress Images						
Cache static content						
Effective use of CDN						

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

	Document Complete			Fully Loaded								
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	7.234s	0.950s	8.089s	10445	163	7.234s	18	435 KB	11.466s	38	685 KB	<del>\$\$--</del>
Repeat View	2.067s	0.726s	2.982s	3379	163	2.067s	9	9 KB	3.337s	17	12 KB	

## Testkörning 3, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Chrome - 3GSlow  
2016-05-19 13:15:53

Tester: i-6ed452af

Need help improving?

	A	A	A	N/A	F	X
First Byte Time						
Keep-alive Enabled						
Compress Transfer						
Compress Images						
Cache static content						
Effective use of CDN						

**Summary** **Details** **Performance Review** **Content Breakdown** **Domains** **Screen Shot**

[Raw page data - Raw object data](#)  
[Export HTTP Archive \(.har\)](#)  
[View Test Log](#)

	Document Complete			Fully Loaded								
	Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost
First View	6.941s	0.915s	7.789s	10200	163	6.941s	18	435 KB	11.178s	38	685 KB	<del>\$\$--</del>
Repeat View	2.010s	0.730s	3.085s	3352	163	2.010s	9	9 KB	3.293s	17	13 KB	

## Sammanställning protokoll: HTTP/1.1

Testkörning 1 kl 12:29	Tid till första byte	Tid till renderingsstart	Tid till komplett dokument	Tid till fullständig laddning
1:a laddning	0,963 s	8,084 s	7,179 s	11,550 s
2:a laddning	0,726 s	2,992 s	2,050 s	3,335 s
Testkörning 2 kl 12:58				
1:a laddning	0,952 s	8,184 s	7,226 s	11,595 s
2:a laddning	0,741 s	3,076 s	2,125 s	3,438 s
Testkörning 3 kl 13:12				
1:a laddning	0,935 s	7,892 s	7,053 s	11,159 s
2:a laddning	0,737 s	3,189 s	2,064 s	3,265 s
Medianvärden				
1:a laddning	0,952 s	8,084 s	7,179 s	11,550 s
2:a laddning	0,737 s	3,076 s	2,064 s	3,335 s

## Sammanställning protokoll: HTTP/2

Testkörning 1 kl 12:49	Tid till första byte	Tid till renderingsstart	Tid till komplett dokument	Tid till fullständig laddning
1:a laddning	0,946 s	7,789 s	6,972 s	10,926 s
2:a laddning	0,721 s	2,983 s	2,081 s	3,428 s
Testkörning 2 kl 13:00				
1:a laddning	0,950 s	8,089 s	7,234 s	11,466 s
2:a laddning	0,726 s	2,982 s	2,067 s	3,337 s
Testkörning 3 kl 13:15				
1:a laddning	0,915 s	7,789 s	6,941 s	11,178 s
2:a laddning	0,730 s	3,085 s	2,010 s	3,293 s
Medianvärden				
1:a laddning	0,946 s	7,789 s	6,972 s	11,178 s
2:a laddning	0,726 s	2,983 s	2,067 s	3,337 s

## Bilaga D Tester Firefox, långsamt 3G

### Testkörning 1, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 12:26:24

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?							
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN		
						<b>B</b>	<b>A</b>	<b>A</b>	<b>N/A</b>	<b>F</b>	<b>X</b>		
						Tester: i-6ed452af	Re-run the test	<a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a>					
					Document Complete			Fully Loaded					
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost		
First View	6.485s	1.045s	6.961s	8910	26	6.485s	17	410 KB	10.420s	39	633 KB	\$\$--	
Repeat View	1.883s	0.921s	2.436s	2662	26	1.883s	11	18 KB	3.080s	18	41 KB		

### Testkörning 2, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 12:55:48

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?							
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN		
						<b>A</b>	<b>A</b>	<b>A</b>	<b>N/A</b>	<b>F</b>	<b>X</b>		
						Tester: i-6ed452af	Re-run the test	<a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a>					
					Document Complete			Fully Loaded					
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost		
First View	6.271s	0.934s	6.783s	8690	26	6.271s	19	409 KB	10.736s	39	633 KB	\$\$--	
Repeat View	1.948s	0.910s	2.374s	2541	26	1.948s	9	16 KB	2.898s	18	38 KB		

### Testkörning 3, HTTP/1.1

Web Page Performance Test for <https://http-1-1.tk/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 13:09:54

Need help improving?

Summary	Details	Performance Review	Content Breakdown	Domains	Screen Shot	Need help improving?							
						First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN		
						<b>A</b>	<b>A</b>	<b>A</b>	<b>N/A</b>	<b>F</b>	<b>X</b>		
						Tester: i-6ed452af	Re-run the test	<a href="#">Raw page data - Raw object data</a> <a href="#">Export HTTP Archive (.har)</a> <a href="#">View Test Log</a>					
					Document Complete			Fully Loaded					
Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost		
First View	6.321s	0.923s	6.741s	8777	26	6.321s	19	409 KB	10.385s	39	633 KB	\$\$--	
Repeat View	1.850s	0.915s	2.396s	2576	26	1.850s	11	18 KB	3.008s	18	41 KB		

X

## Testkörning 1, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 11:57:51

Tester: i-6ed452af

Need help improving?

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	6.333s	0.932s	6.882s	8830	26	6.333s	19	407 KB	10.222s	39	642 KB	\$ <del>...</del>
Repeat View	1.469s	0.911s	1.927s	2040	26	1.469s	11	6 KB	2.483s	18	34 KB	

## Testkörning 2, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 12:52:45

Tester: i-6ed452af

Need help improving?

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	6.426s	0.938s	6.842s	8905	26	6.426s	19	408 KB	10.439s	39	642 KB	\$ <del>...</del>
Repeat View	1.437s	0.922s	1.923s	2092	26	1.437s	10	6 KB	3.172s	18	34 KB	

## Testkörning 3, HTTP/2

Web Page Performance Test for <https://bythebeach.se/>

From: Frankfurt, DE - Firefox - 3GSlow  
2016-05-19 13:06:01

Tester: i-6ed452af

Need help improving?

Load Time	First Byte	Start Render	Speed Index	DOM Elements	Time	Requests	Bytes In	Time	Requests	Bytes In	Cost	
First View	6.310s	0.939s	6.884s	8878	26	6.310s	19	407 KB	10.180s	39	641 KB	\$ <del>...</del>
Repeat View	1.446s	0.923s	1.960s	2162	26	1.446s	10	6 KB	2.438s	18	34 KB	

## Sammanställning protokoll: HTTP/1.1

Testkörning 1 kl 12:26	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	1,045 s	6,961 s	6,485 s	10,420 s
2:a laddning	0,921 s	2,436 s	1,883 s	3,080 s
Testkörning 2 kl 12:55				
1:a laddning	0,934 s	6,783 s	6,271 s	10,736 s
2:a laddning	0,910 s	2,374 s	1,948 s	2,898 s
Testkörning 3 kl 13:09				
1:a laddning	0,923 s	6,741 s	6,321 s	10,385 s
2:a laddning	0,915 s	2,396 s	1,850 s	3,008 s
Medianvärden				
1:a laddning	0,934 s	6,783 s	6,321 s	10,420 s
2:a laddning	0,915 s	2,396 s	1,883 s	3,008 s

## Sammanställning protokoll: HTTP/2

Testkörning 1 kl 11:57	Tid till första byte	Tid till renderingsstart	Tid till komplettdokument	Tid till fullständig laddning
1:a laddning	0,932 s	6,882 s	6,333 s	10,222 s
2:a laddning	0,911 s	1,927 s	1,469 s	2,483 s
Testkörning 2 kl 12:52				
1:a laddning	0,938 s	6,842 s	6,426 s	10,439 s
2:a laddning	0,922 s	1,923 s	1,437 s	3,172 s
Testkörning 3 kl 13:06				
1:a laddning	0,939 s	6,884 s	6,310 s	10,180 s
2:a laddning	0,923 s	1,960 s	1,446 s	2,438 s
Medianvärden				
1:a laddning	0,938 s	6,882 s	6,333 s	10,222 s
2:a laddning	0,922 s	1,927 s	1,446 s	2,483 s