HTTP'nin Gelişimi: İnternetin Temel Taşı

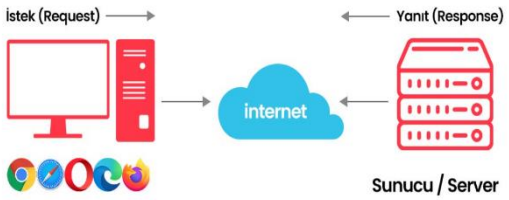
Murat Berk Yetiştirir

Öz

Bu dokümanda, HTTP protokolünün evrimini ve modern web teknolojileri üzerindeki etkisini ele alır. HTTP/2 ve HTTP/3 protokollerinin karşılaştırılması yapılarak, her iki protokolün avantajları ve dezavantajları incelenmiştir.

1.0 HTTP NEDİR?

HTTP protokolü, istemci ve sunucu arasında gerçekleşen bir istek/yanıt modeline dayanır. İstemci, sunucuya bir istek yöntemi, URI ve protokol sürümü içeren bir istek gönderir. Ardından, istek değiştiriciler, istemci bilgileri ve olası gövde içeriği içeren MIME benzeri bir mesaj gelir. Sunucu ise, mesajın protokol sürümü ve bir başarı veya hata kodunu içeren bir durum satırı ile yanıt verir. [1]



Şekil 1: Basit bir şekilde HTTP taslağı. [S1]

HTTP, bağlantısız uygulama düzeyinde, esnek bir protokolüdür. İstemcilere türden bağ ımsız tekdüze bir arayüz sağlar ve hizmetin nasıl uygulandığını gizler. Sunucular, istemcilerin amacını bilmek zorunda değildir, bu da genel amaçlı kullanımı kolaylaştırır. HTTP, bir aracı p rotokol olarak da işlev görebilir ve HTTP olmayan sistemleri genel bir arayüze dönüştürür. Pr otokol, arayüzün arkasında ne olduğuna göre tanımlanmaz; sadece iletişimin sözdizimini ve al ınan iletişimin amacını belirler. [2]

1.0.1 Caching (Önbellekleme) Nedir?

HTTP, esnek etkileşim için kendini tanımlayan mesajlar ve genişletilebilir semantik kullanan bağlantısız bir istek/yanıt protokolüdür. Önbellekleme, performansı artırmak ve yanıtları daha hızlı iletmek için kullanılır. Paylaşılan önbellekler çoklu kullanıcılar için, özel önbellekler tek bir kullanıcı için çalışır. Taze yanıtlar gecikmeyi ve ağ yükünü azaltarak kullanıcı deneyimini iyileştirir ve kaynakların verimli kullanımını sağlar. [3]

1.1 HTTP/1.1 Nedir?

Bu protokol, özelliklerinin güvenilir bir şekilde uygulanmasını sağlamak için HTTP/1.0'dan daha katı gereksinimlere sahiptir. Pratik bilgi sistemleri, basit alımın ötesinde arama, ön uç güncellemesi ve ek açıklama gibi daha fazla işlevsellik gerektirir.[4]

HTTP 1.1, HTTP 1.0'dan çok daha büyük ve karmaşıktır. Çok sayıda ayrıntı ve genişletme seçenekleri sunar, ancak bu durum birlikte çalışabilirlik sorunlarına yol açabilir. Ayrıca, HTTP 1.1, TCP'nin tüm gücünden tam olarak yararlanmakta zorlanır ve istemciler yaratıcı çözümler bulmak zorunda kalır. Bununla birlikte, HTTP 1.1'in geniş kapsamı ve esnekliği, farklı bağlamlarda ve uygulamalarda kullanılmasını mümkün kılar. Bu, geliştiricilere zengin bir araç seti sunar, ancak aynı zamanda karmaşıklığı da artırır.[5]

1.1.1 HTTP/1.1'de Caching (Önbellekleme) Nedir?

HTTP, esnek etkileşim için kendini tanımlayan mesajlar ve genişletilebilir semantik kullanan bağlantısız bir istek/yanıt protokolüdür. Bu yapı, protokolün ağ tabanlı hipermetin bilgi sistemleriyle etkileşimde esneklik sağlamasına olanak tanır. Önbellekleme, performansı artırmak ve yanıtları daha hızlı iletmek için kritik bir yöntemdir. Önbellekler, gelecekteki benzer isteklerde yanıt süresini ve ağ bant genişliği tüketimini azaltır.

Paylaşılan önbellekler, birden fazla kullanıcı tarafından erişilen yanıtları depolar ve genellikle bir ara sunucu veya ağ geçidinin parçası olarak kullanılır. Özel önbellekler ise yalnızca tek bir kullanıcıya hizmet eder ve genellikle bir kullanıcı aracısının bileşeni olarak işlev görür. Bu, bireysel kullanıcıların sıkça eriştiği kaynaklara daha hızlı ulaşmasını sağlar.

Taze yanıtlar, doğrulama gerektirmeden yeniden kullanılabilir ve her kullanımda gecikmeyi ve ağ yükünü azaltır. Bu sayede, kullanıcı deneyimi önemli ölçüde iyileşir ve ağ kaynaklarının verimli kullanımı sağlanır. Ayrıca, taze yanıtların yeniden kullanımı, sunucuya olan yükü de hafifletir ve genel ağ trafiğini optimize eder.[6]

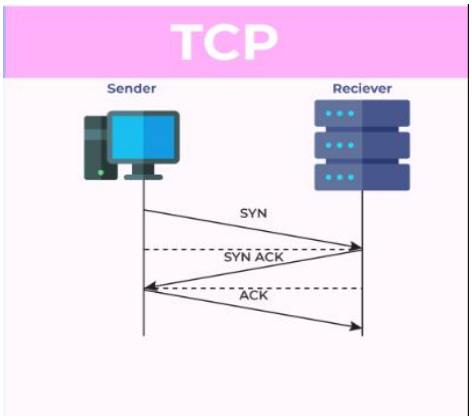
1.1.2. HTTP'yi TSL Üzerinden Kullanmak Nedir?

İstemci, bağlantıyı başlatmalı ve TLS el sıkışmasını tamamladıktan sonra HTTP isteği göndermelidir. TLS, güvenli bağlantı kapanışı sağlar ve kapanış uyarısı değişiminden sonra bağlantı kapatılabilir. Erken kapanış, alınan verilerin güvenliğini etkilemez, ancak sonraki verilerin eksik olabileceğini gösterir. [7]

HTTP/TLS, HTTP URI'lerinden, 'http' protokol tanımlayıcısı yerine 'https' protokol tanımlayıcısını kullanarak ayırt edilir. HTTP/TLS belirten örnek bir URI şu şekildedir: https://www.ornek.com/ [8]

1.1.3 TCP (Transmission Control Protocol) Nedir?

İletim Kontrol Protokolü (TCP), paket anahtarlamalı bilgisayar iletişim ağlarında ve özellikle bu ağların birbirine bağlı sistemlerinde, ana bilgisayarlar arasında son derece güvenilir bir protokol olarak kullanılmak üzere tasarlanmıştır.[9]



Şekil 2: TCP'nin (Transmission Control Protocol) çalışma prensibi. [S2]

TCP protokolü iletim hızı, gecikme, bozulma, yinelenme veya segmentlerin yeniden sıralanmasından bağımsız olarak neredeyse her tür iletim ortamında güvenilir bir şekilde çalışmak üzere tasarlanmıştır. Yıllar içinde ağ teknolojisindeki ilerlemeler, iletim hızlarının artmasına neden olmuştur ve en hızlı yollar, TCP'nin ilk tasarlandığı alanın çok ötesine geçmiştir.[10]

TCP, bağlantı kurmak için üç yönlü el sıkışma kullanır ve güvenilir veri iletimi sağlar, ancak gecikmeyi artırır. Güvenilir veri gerektirmeyen uygulamalar UDP kullanabilir. TCP, ağ tıkanıklığını önler fakat hizmet reddi ve bağlantı kaçırma gibi güvenlik açıkları vardır.[11]

1.1.4 Pipelining Nedir?

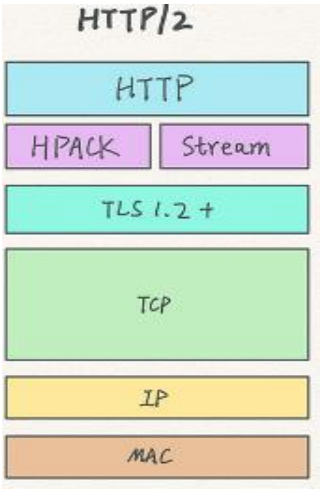
Kalıcı bağlantıları destekleyen bir istemci, yanıtları beklemeden birden fazla istek gönderebilir ("pipelining"). Bir sunucu, güvenli yöntemler kullanıldığında bir dizi ardışık isteği paralel olarak işleyebilir, ancak yanıtları alınan sırayla göndermelidir. Bir bağlantı kapanmadan önce yanıtları almayan bir istemci, yanıtlanmamış istekleri yeniden denemelidir. Başarısız bir bağlantıdan sonra ardışık istekleri yeniden denerken, istemci bağlantı kurulur kurulmaz ardışık istek göndermemelidir. [12]

1.1.4.1 Head of Line Blocking Nedir?

HTTP Pipelining, önceki bir isteğin yanıtını beklerken başka bir isteği göndermenin bir yoludur. Bu, süpermarket veya banka gişesinde sıraya girmek gibidir. Önünüzdeki kişinin hızlı mı yoksa yavaş mı olacağını bilemezsiniz: baş satır engellemesi. Doğru sırayı seçmeye dikkat edebilirsiniz, hatta kendi sıranızı başlatabilirsiniz, ancak karar verildikten sonra değiştiremezsiniz. Yeni bir sıra oluşturmak, performans ve kaynak cezasına yol açar ve sınırlı sayıda sıra için ölçeklenebilir değildir. [13]

1.2 HTTP/2 Nedir?

HTTP/2, HTTP/1.1'in tüm temel özelliklerini destekleyen ve optimize edilmiş bir protokoldür. Her isteği/yanıt değişimini kendi akışıyla ilişkilendirir, böylece akışlar bağımsız çalışır ve bir engelleme diğerlerini etkilemez. Akış kontrolü ve önceliklendirme, verilerin verimli kullanılmasını sağlar. Sunucu yönlendirmesi ile sunucu, istemcinin ihtiyaç duyacağını öngördüğü verileri proaktif olarak gönderebilir. Ayrıca, tekrar eden büyük veri içeren HTTP başlık alanları sıkıştırılır, bu da özellikle istek boyutlarını avantajlı hale getirir. (Belshe & Peon, 2015) [14]



Şekil 3: HTTP/2 protokolünün katmanlarını: HTTP, HPACK, Stream, TLS 1.2+, TCP, IP ve MAC. [S3]

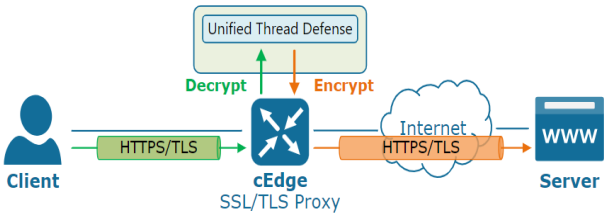
HTTP/2, TCP bağlantısı üzerinde çalışan ve HTTP/1.1'in URI şemalarını ve varsayılan port numaralarını kullanan optimize edilmiş bir protokoldür. İstemci, TCP bağlantısını başlatan taraftır. Uygulamalar, yukarı akış sunucusunun HTTP/2'yi destekleyip desteklemediğini kontrol etmelidir. HTTP/2 desteğinin belirlenme yöntemi "http" ve "https" URI'ları için farklılık gösterir:"http" için 80 ve "https" için 443. "http" URI'larında, destek keşfi farklı işlemlerle gerçekleştirilirken, "https" URI'larında güvenli bağlantının sağlanması için ek adımlar gerekebilir. [15]

HTTP/2'nin amacı, HTTP protokolünü daha verimli hale getirmektir. HTTPbis grubu, bu protokolü geliştirirken bazı kısıtlamalar koydu. Öncelikle, HTTP paradigmaları korunmalıydı; yani istemci, istekleri TCP üzerinden sunucuya göndermeliydi. http:// ve https:// URL şemaları değiştirilemezdi çünkü bunlar çok yaygındı. HTTP/1 sunucuları ve istemcileri uzun yıllar daha kullanılacak, bu nedenle HTTP/2 sunucularına geçişin mümkün olması gerekiyordu. Proxy'lerin, HTTP/2 özelliklerini HTTP 1.1 istemcilerine birebir eşlemesi gerekiyordu. Protokolden isteğe bağlı parçalar kaldırılmalı veya azaltılmalıydı; böylece her şey zorunlu hale gelerek gelecekteki uyumsuzluklar önlenmiş olacaktı. (Stenberg, 2014)[16]

1.2.1 TLS (Transport Layer Security) Nedir?

TLS, iki uygulama arasında gizlilik ve veri bütünlüğü sağlar ve iki katmandan oluşur: TLS Kayıt Protokolü ve TLS El Sıkışma Protokolü. Kayıt Protokolü, simetrik kriptografi ile

gizlilik ve MAC kullanarak güvenilirlik sağlar. El Sıkışma Protokolü, sunucu ve istemci arasında kimlik doğrulama, şifreleme algoritması müzakeresi ve kriptografik anahtarlar oluşturur. TLS, uygulama protokolüne bağımsız olarak çalışır ve güvenlik sağlamada esneklik sunar. Ayrıca, TLS protokolü, dinlemeye ve bağlantı saldırılarına karşı güçlü bir koruma sağlar. Bu yapı, birçok üst düzey protokol tarafından kolayca benimsenebilir ve uyarlanabilir.[17]



Şekil 4: HTTPS/TLS proxy'sinin güvenliği nasıl artırdığının gösterimi. [S4]

1.2.1.1 TLS-Uygulama Katmanı Protokol Müzakeresi Uzantısı (ALPN) Nedir?

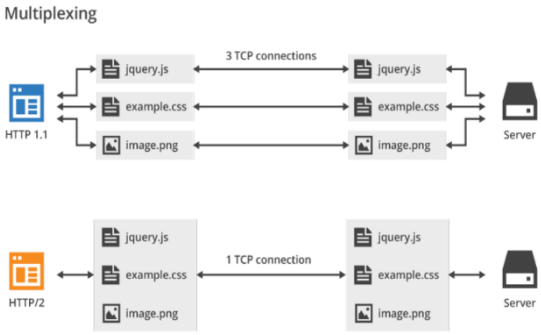
ALPN uzantısı, TLS el sıkışması sırasında müzakereyi gerçekleştirir ve protokol seçimini sunucuya bırakır. Bu, sertifika seçimi veya bağlantı yönlendirme gibi senaryoları kolaylaştırır. Ayrıca, el sıkışma sırasında protokol seçimini açık bir şekilde yöneterek, müzakere edilen protokolün gizlenmesi ile ilgili sahte bir güven oluşturmaktan kaçınır. Protokolün gizlenmesi gerekiyorsa, bağlantı kurulduktan sonra yeniden müzakere önerilir, bu da gerçek TLS güvenlik garantileri sağlar. [18]

1.2.2 URI (Uniform Resource Identifier) Nedir?

Bir Birleşik Kaynak Tanımlayıcısı (URI), bir kaynağı tanımlamak için basit ve genişletilebilir bir yöntem sağlar. URI'lar(Birleşik Kaynak Tanımlayıcıları), kaynakları tanımlamak için tekdüze, esnek ve yeniden kullanılabilir tanımlayıcılardır. Kaynak terimi, elektronik belgeler, resimler, bilgi kaynakları ve hizmetler gibi geniş bir yelpazeyi kapsar. Tanımlayıcılar, bir kaynağı diğerlerinden ayırt etmek için gereken bilgileri içerir. Bu tanımlayıcılar, bir kaynağa erişim sağlamak zorunda olmadan onu belirtmek için kullanılabilir. Tanımlanan kaynaklar tekil olmayabilir ve zamanla değişebilir. [19]

1.2.3 HTTP/2'de Akışlar ve Çoklama

Bir "akış", bir HTTP/2 bağlantısı içinde istemci ile sunucu arasında değiş tokuş edilen bağımsız ve çift yönlü bir çerçeve dizisidir. Akışların birkaç önemli özelliği vardır: Tek bir HTTP/2 bağlantısı, birden fazla eşzamanlı açık akış içerebilir ve her iki uç nokta da birden fazla akıştan gelen çerçeveleri birbirine karışabilir. Akışlar, her iki uç nokta tarafından tek taraflı olarak başlatılabilir veya paylaşılabilir ve her iki uç nokta tarafından da kapatılabilir. Çerçevelerin gönderilme sırası önemlidir, alıcılar çerçeveleri aldıkları sıraya göre işlerler. Özellikle HEADERS ve DATA çerçevelerinin sırası anlamsal olarak önemlidir. Akışlar bir tamsayı ile tanımlanır.[20]



Şekil 5: Çoklama (Multiplexing) yapısı sayesinde HTTP/2 ve HTTP/1.1 arasındaki fark. [S5]

Akışları çoklamak, TCP bağlantısının kullanımı üzerinde rekabet yaratarak akışların engellenmesine neden olabilir. Bir akış kontrolü şeması, aynı bağlantı üzerindeki akışların birbirleriyle yıkıcı şekilde etkileşime girmemesini sağlar. Akış kontrolü hem bireysel akışlar hem de bağlantının tamamı için kullanılır. HTTP/2, akış kontrolünü WINDOW_UPDATE çerçevesi kullanarak sağlar. (Thomson & Mozilla Benfield, 2022 [21])

1.2.2 HTTP/2'de HPACK (Başlık Sıkıştırma) Nedir?

HTTP/1.1'de başlık alanları sıkıştırılmaz ve bu, bant genişliğini tüketir ve gecikmeyi artırır. SPDY, DEFLATE ile sıkıştırma yaparak bu sorunu çözüldü, ancak CRIME saldırısı gibi güvenlik risklerini ortaya çıkardı. HPACK, tekrarlı başlık alanlarını ortadan kaldıran ve güvenlik risklerini sınırlayan yeni bir sıkıştırıcıdır. HPACK, kısıtlı ortamlar için sınırlı bellek gereksinimine sahiptir ve basit ve esnek olmayan yapısı sayesinde uygulama hatalarından kaynaklanan güvenlik sorunlarını azaltır. Değişiklikler ancak tam bir yenileme ile mümkündür. [22]

1.2.3 HTTP/2 ile TLS 1.3 Kullanımı Nedir?

TLS 1.2 ve daha önceki sürümler, bağlantı sırasında parametreleri ve anahtarları değiştirme mekanizması olan yeniden müzakereyi destekler. Bu, HTTP/1.1'de sunucunun, istemciden bir sertifika isteyip istemeyeceğine karar vermek için kullanılırdı. HTTP/2, tek bir bağlantı üzerinde birden fazla HTTP isteğini çokladığı için bu mekanizmayla uyumlu değildir. TLS 1.3, yeniden müzakereleri kaldırır ve bunun yerine bağlantı sonrası kimlik doğrulama ve anahtar güncelleme mekanizmalarını getirir. Bağlantı sonrası kimlik doğrulama, çoklanmış protokollerle aynı sorunları yaşar, ancak yasak yalnızca yeniden müzakerelere uygulanır. Burada HTTP/2 'yi güncelleyerek, TLS 1.3 sonrası kimlik doğrulamayı da yasaklar. [23]

1.3 HTTP/3 Nedir?

HTTP/3, QUIC taşıma protokolünü kullanarak HTTP semantiklerini taşır ve böylece daha hızlı, güvenli ve verimli iletişim sağlar. QUIC bağlantıları, protokol müzakeresi, akış tabanlı çoklama ve akış kontrolü sunar, bu da veri iletimini daha etkili hale getirir. Her HTTP/3

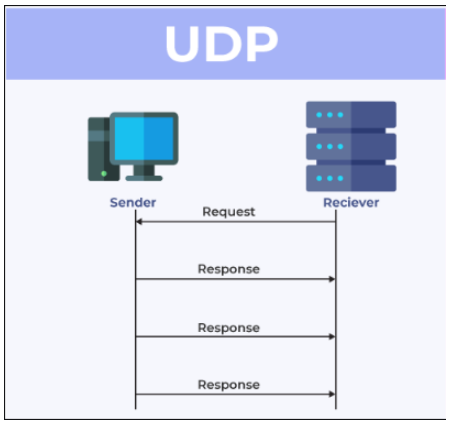
akışının temel birimi çerçevelerdir ve bu çerçeveler, akışlar bağımsız olarak çalıştığı için veri iletimini kesintiye uğratmadan gerçekleştirir.

Sunucu itme (server push) özelliği sayesinde, sunucular, istemcinin ihtiyaç duyacağı verileri öngörerek önceden gönderebilir, bu da yanıt sürelerini azaltır ve kullanıcı deneyimini iyileştirir. Ayrıca, HTTP/3, HPACK yerine QPACK kullanarak verilerin sıkıştırılmasını sağlar ve böylece verimliliği artırır. QPACK, özellikle veri sıkıştırmada daha yüksek performans ve düşük gecikme sunar.

Son olarak, QUIC, güvenilir teslimat ve tıkanıklık kontrolü sağlamak için gerekli geri bildirimi sunar ve bağlantılar yeni ağ yollarına kesintisiz olarak taşınabilir. Bu, özellikle mobil cihazlar ve değişken ağ koşullarında daha stabil ve güvenilir bağlantılar sağlar. QUIC'in bu yetenekleri, HTTP/3'ü modern web protokolleri arasında ön plana çıkarır ve internet trafiğini daha güvenli ve verimli hale getirir. (Bishop, 2022) [24]

1.3.1 UDP (User Datagram Protocol) Nedir?

Kullanıcı Datagram Protokolü (UDP), bilgisayar ağlarında paket anahtarlamalı iletişim sağlamak için tanımlanmıştır ve İnternet Protokolü (IP) üzerine inşa edilmiştir. Bu protokol, uygulama programlarının minimum protokol mekanizmasıyla mesaj göndermesine olanak tanır. UDP, işlem odaklıdır ve teslimat veya çoğaltma koruması garanti edilmez. Verilerin sıralı ve güvenilir teslimatını gerektiren uygulamalar, İletim Kontrol Protokolü (TCP) kullanmalıdır.[25]



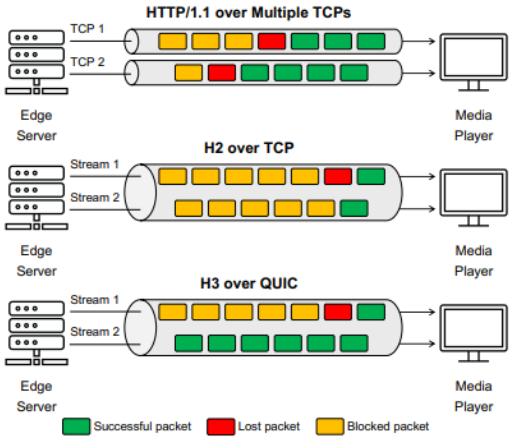
Şekil 6: UDP'nin (User Datagram Protocol) Çalışma Prensibi. [S6]

UDP modülü, internet başlığından adresleri ve protokol alanını belirleyebilir. Bir UDP/IP arayüzü, tüm internet başlığını içeren internet datagramını döndürebilir ve IP'ye gönderilmek üzere tam datagramı iletebilir. IP, alanları doğrular ve başlık toplamını hesaplar.[26]

1.3.2 HTTP/3'te QUIC Nedir?

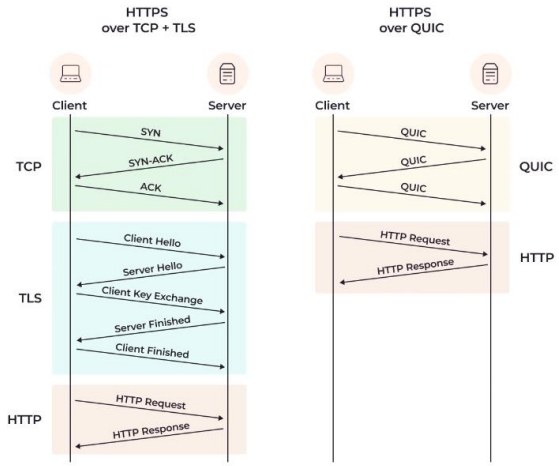
QUIC, güvenli ve genel amaçlı bir taşıma protokolüdür. QUIC, istemci ve sunucu arasında durumsal etkileşim yaratan bağlantı odaklı bir protokoldür ve TLS el sıkışmasını

entegre eder. QUIC, uygulama verilerini mümkün olan en kısa sürede değiş tokuş etmeyi sağlar. Uç noktalar, QUIC paketlerini değiş tokuş ederek iletişim kurar ve bu paketler UDP datagramları içinde taşınır. Uygulama protokolleri, sıralı bayt dizileri olan akışlar aracılığıyla bilgi değiş tokuşu yapar. Akışlar çift yönlü veya tek yönlü olabilir. QUIC, güvenilir teslimat ve tıkanıklık kontrolü sağlar ve bağlantılar yeni ağ yollarına taşınabilir. Bağlantı sonlandırma için birden fazla seçenek sunulmuştur. [27]



Şekil 7: Verilerin birden fazla TCP bağlantısı üzerinden (buradaki örnek iki TCP içindir) veya tek bir TCP veya QUIC bağlantısı üzerinden gönderilmesi (Demir et al., 2024) [S7]

QUIC, uygulamalar için hızlı bağlantı kurulumu sağlar ve tek bir bağlantı üzerinden çoklanmış, akış kontrollü veri akışlarını kullanır. Ayrıca, ağ yolu değişikliği sırasında bağlantıyı kesmeden akış kesintilerini en aza indirir. HTTP/3, HTTP/2'den çoğu özelliği devralırken, QUIC sayesinde sıfır tur süresi (0-RTT) bağlantı kurulumu gibi ek yetenekler sunar. HTTP/3, QUIC'e entegre edilen TLS 1.3 ile bağlantı kurulumu için en az bir RTT tasarruf sağlar. Ayrıca, TCP'nin satır başı engelleme sorunlarını ortadan kaldırır. Akış çoklama ile her istek-yanıt çifti bağımsız veri dizileri olarak işlenir, böylece farklı istekler aynı anda işlenebilir. (Demir et al., 2024) [28]



Şekil 8: HTTP/2 ve HTTP/3 katmanları [S8]

1.3.3 HTTP/3'te QPACK Nedir?

HPACK gibi, QPACK de alan satırlarını ("başlıklar") indekslerle ilişkilendirmek için iki tablo kullanır. Statik tablo (Bölüm 3.1) önceden tanımlanmıştır ve bazıları boş değer içeren yaygın başlık alanı satırlarını içerir. Dinamik tablo (Bölüm 3.2) bağlantı süresince oluşturulur ve kodlayıcı tarafından başlık ve ek alan satırlarını kodlanmış alan bölümlerinde indekslemek için kullanılabilir. QPACK, kodlayıcıdan kod çözücüye ve tersi yönde talimatlar göndermek için tek yönlü akışlar tanımlar. [29]

QPACK, sıkıştırmanın neden olabileceği satır başı engellemesini kontrol etme olanağı sağlar, böylece sıkıştırma verimliliği ile gecikme arasında denge kurar. HTTP/3, başlık ve ek bölümlerini sıkıştırmak için QPACK kullanır. Cookie başlık alanı, sıkıştırmadan önce ayrı ala n satırlarına bölünmelidir. Bir alan bölümü birden fazla çerez alanı satırı içeriyorsa, bunlar birl eştirilerek iletilmelidir. [30]

1.4 HTTP/2 ve HTTP/3 Arasındaki Farklılıklar

HTTP/2'nin benimsenme oranı %45,2'de sabit kalırken, HTTP/3'ün oranı %25'e yükseldi. Bu, HTTP/2'nin yerini yavaş yavaş HTTP/3'e bıraktığını gösterir. HTTP/2 güvenlik açıklarının kapsamlı bir analizi yapılmamış olup, HTTP/3'e geçişin getireceği sorunların değerlendirilmesi önemlidir. Bu değerlendirmeler, yeni protokolün potansiyel risklerini anlamak ve gerekli önlemleri almak için kritik önem taşır. Aynı zamanda, mevcut HTTP/2 dağıtımlarının güvenliğinin sağlanması için de dikkatli bir inceleme gereklidir. [31]

HTTP/2, TCP üzerinde çalışırken, HTTP/3 UDP üzerinde çalışan QUIC kullanır. TCP'nin kullanımı, HTTP/2'nin güvenilir bağlantılar ve sıra düzenlemesi sağlarken, bayt akışı soyutlamasındaki kısıtlamalar nedeniyle baş satır engelleme (HOL) sorunlarına neden olur. Buna karşılık, HTTP/3, UDP'nin sırasız teslimi ve QUIC protokolü sayesinde HOL engellemesi olmadan çoklama sağlar, bu da veri akışlarının daha verimli ve kesintisiz olmasını sağlar.

Hata yönetimi açısından, HTTP/2 daha az gelişmiş yeteneklere sahipken, HTTP/3 QUIC'in gelişmiş hata yönetimi özelliklerini kullanarak daha iyi performans ve güvenilirlik sunar. Bu, veri iletiminde meydana gelen hataların daha hızlı tespit edilmesini ve düzeltilmesini sağlar, böylece kullanıcı deneyimini artırır.

TLS şifreleme açısından, HTTP/2'de TLS kullanımı isteğe bağlıdır ve TCP üzerine entegre edilir. HTTP/3'te ise TLS, QUIC protokolüne gömülüdür ve varsayılan olarak kullanılır, bu da daha güvenli bir veri iletimi sağlar. Bu, güvenli bağlantıların daha kolay ve hızlı bir şekilde kurulmasını sağlar.

HTTP/3 ayrıca bağlantı kimlikleri aracılığıyla sorunsuz bağlantı göçü sağlar, yani bir bağlantı kaybı veya değişikliğinde veri akışı kesintiye uğramadan devam edebilir. HTTP/2 ise bağlantı göçünü desteklemez ve bağlantı kesildiğinde yeni bir bağlantı kurulması gerekebilir.[32]

1.5 Sonuç

Bu karşılaştırma, HTTP/3'ün önemli iyileştirmeler sunduğunu ve modern internet protokollerinin verimliliğini ve güvenliğini artırdığını gösterir. Yani, HTTP/3, HTTP/2'den daha iyi performans ve güvenlik sunar, çünkü HOL engelleme sorunlarını çözmekte, TLS entegrasyonunu iyileştirmekte ve bağlantı göçlerini sorunsuz bir şekilde yönetmektedir. Bu da daha hızlı ve güvenilir internet bağlantıları sağlar.

KAYNAKÇA

[1] https://www.rfc-editor.org/rfc/rfc1945.html

[2] https://www.rfc-editor.org/rfc/rfc9110.html

[3] https://www.rfc-editor.org/rfc/rfc9111.html

[4] https://www.rfc-editor.org/rfc/rfc2068

[5] Stenberg, D. (2014). HTTP2 explained. *Computer Communication Review*, *44*(3), 120–128. https://doi.org/10.1145/2656877.2656896

[6] https://www.rfc-editor.org/rfc/rfc7540.html

[7] https://www.rfc-editor.org/rfc/rfc2818.html

[8] https://dl.acm.org/doi/pdf/10.17487/RFC2818

[9] https://www.rfc-editor.org/rfc/rfc761

[10] https://dl.xkwy2018.com/rfcs/rfc7323.txt.pdf

[11] https://en.wikipedia.org/wiki/Transmission_Control_Protocol

[12] https://www.rfc-editor.org/rfc/rfc9112

[13] Stenberg, D. (2014). HTTP2 explained. *Computer Communication Review*, *44*(3), 120–128. https://doi.org/10.1145/2656877.2656896

[14] Belshe, M., & Peon, R. (2015). *RFC 7540: hypertext transfer protocol version 2 (HTTP/2)*. https://dl.acm.org/doi/abs/10.17487/RFC7540

[15] https://www.rfc-editor.org/rfc/rfc7234.html

[16] Stenberg, D. (2014). HTTP2 explained. *Computer Communication Review*, *44*(3), 120–128. https://doi.org/10.1145/2656877.2656896

[17] https://www.rfc-editor.org/rfc/rfc5246

[18] https://www.rfc-editor.org/rfc/rfc7301.html

[19] https://www.hjp.at/doc/rfc/rfc3986.html

[20] https://www.hjp.at/doc/rfc/rfc9113.pdf

[21] Thomson, M., & Mozilla Benfield, E. C. (2022). *RFC 9113: HTTP/2*. https://www.rfc-editor.org/info/rfc9113

[22] https://www.rfc-editor.org/rfc/rfc7541.html

[23] https://www.rfc-editor.org/rfc/rfc8740.html

[24] Bishop, M. (2022). *RFC 9114 HTTP/3 Abstract*. https://www.rfc-editor.org/info/rfc9114

[25] https://www.rfc-editor.org/rfc/rfc9000.html

[26] https://www.rfc-editor.org/rfc/rfc768.html

[27] https://www.rfc-editor.org/rfc/rfc768.html

[28] Demir, A. B., Parlak, M., Gurel, Z., Ugur, D., & Begen, A. C. (2024). Impact of Prioritized HTTP/3 Transport on Low-Latency Live Streaming. 2024 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2024. https://doi.org/10.1109/ICMEW63481.2024.10645374

[29] https://www.rfc-editor.org/rfc/rfc9204.html

[30] https://www.rfc-editor.org/rfc/rfc9114.html

[31] https://www.sciencedirect.com/science/article/pii/S0167404822004436#sec0001

[32] https://gcore.com/learning/what-is-http-3/

[S1] https://www.tercihyazilim.com/blog/http-nedir

[S2] https://www.geeksforgeeks.org/differences-between-tcp-and-udp/

[S3] https://cabulous.medium.com/http-3-quic-and-how-it-works-c5ffdb6735b4

[S4] https://www.networkacademy.io/ccie-enterprise/sdwan/https-tls-proxy

[S5] https://www.hosting.com.tr/bilgi-bankasi/http-2-nedir-ne-ise-yarar/

[S6] https://www.geeksforgeeks.org/differences-between-tcp-and-udp/

[S7] Demir, A. B., Parlak, M., Gurel, Z., Ugur, D., & Begen, A. C. (2024). Impact of Prioritized HTTP/3 Transport on Low-Latency Live Streaming. 2024 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2024. https://doi.org/10.1109/ICMEW63481.2024.10645374

[S8] https://gcore.com/learning/what-is-http-3/