# BİLGİSAYAR AĞLARI DERSİ
# LABORATUVAR FÖYÜ

**Dr. Halil ARSLAN**

## Laboratuvar -11

## TCP

İçindekiler



Öğrencinin

Adı Soyadı : EYYÜP ERDOĞAN

Okul No : 2021141033

Teslim Tarihi : 21.05.2024

İmza :

Bu laboratuvarda, ünlü TCP protokolünün davranışını ayrıntılı olarak inceleyeceğiz. Bunu, bilgisayarınızdan uzak bir sunucuya 150 KB'lık bir dosya aktarırken gönderilen ve alınan TCP segmentlerinin izini analiz ederek yapacağız.

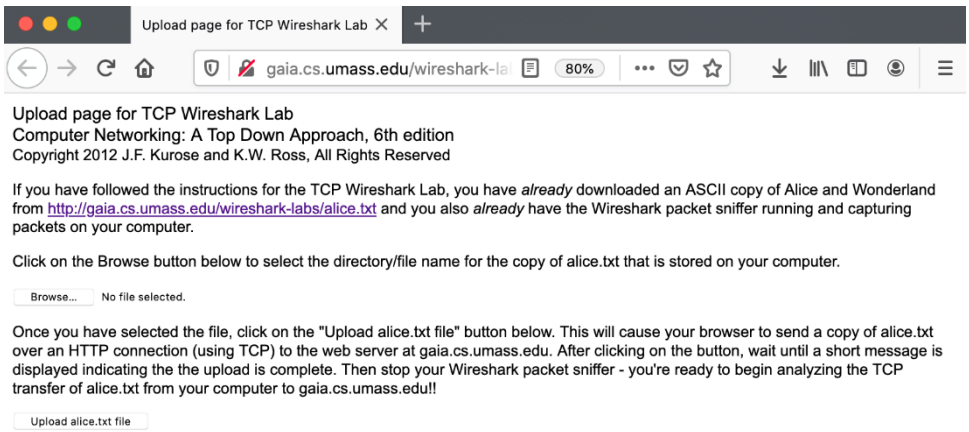
- Güvenilir veri aktarımı sağlamak için TCP'nin sıra (**sequence**) ve onay numaralarını (**acknowledgement numbers**) nasıl kullandığını inceleyeceğiz;
- TCP'nin tıkanıklık kontrol algoritmasını (**TCP's congestion control algorithm**) -yavaş başlatma (**slow start**) ve tıkanıklıktan kaçınma (**congestion avoidance**) kavramları üzerinden- çalışırken göreceğiz ve
- TCP'nin alıcı tarafından tanıtılan akış kontrol mekanizmasına (**receiver-advertised flow control mechanism**) bakacağız.
- Ayrıca kısaca TCP bağlantı kurulumunu ele alacağız ve bilgisayarınız ile sunucu arasındaki TCP bağlantısının performansını verim (**throughput**) ve gidiş-dönüş süresi (**round-trip time**) kavramları üzerinden araştıracağız.

# Bilgisayarınızdan uzak bir sunucuya toplu TCP aktarımı yakalama

TCP keşfimize başlamadan önce, bir dosyanın bilgisayarınızdan uzak bir sunucuya TCP aktarımının paket izini elde etmek için Wireshark'ı kullanmamız gerekecek. Bunu, Web sunucusunda yer alan bir Web sayfasına erişerek yapabilirsiniz. Bilgisayarımızdan başka bir bilgisayara büyük miktarda veri aktarmak istediğimiz için GET yöntemi yerine POST yöntemini kullanıyoruz. Elbette, bilgisayarınızdan gönderilen ve alınan TCP segmentlerinin izini sürmek için bu süre zarfında Wireshark'ı çalıştırmanız gerekmektedir.

Aşağıdakileri yapın:

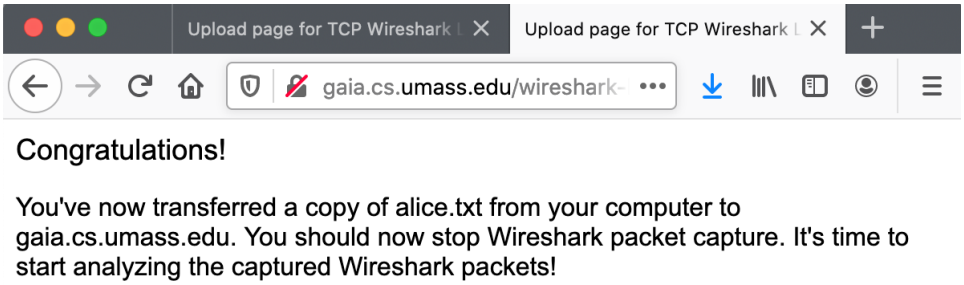
- Web tarayıcınızı başlatın. http://gaia.cs.umass.edu/wireshark-labs/alice.txt adresine gidin ve *Alice in Wonderland*'nın ASCII kopyasını alın. Bunu, bilgisayarınızda bir yerde bir .txt dosyası olarak saklayın.
- Daha sonra http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html adresine gidin.
- Şekil 1'e benzeyen bir ekran görmelisiniz.




**Şekil 1.** alice.txt dosyasını bilgisayarınızdan gaia.cs.umass.edu'ya yüklemek için sayfa

- Bilgisayarınızda az önce oluşturduğunuz ve *Alice in Wonderland*'ı içeren dosyayı bu formdaki Dosya Seç (Browse) butonu ile seçin. "*Upload alice.txt file"* butonuna henüz basmayın.
- Şimdi Wireshark'ı başlatın ve paket yakalamaya başlayın.
- Tarayıcınıza dönerek, dosyayı gaia.cs.umass.edu sunucusuna yüklemek için "*Upload alice.txt file"* butonuna basın. Dosya yüklendikten sonra, tarayıcı pencerenizde kısa bir tebrik mesajı görüntülenecektir.
- Wireshark paket yakalamayı durdurun. Wireshark pencereniz, Şekil 2'de gösterilen pencereye benzer görünmelidir.




**Şekil 2**. Başarılı! gaia.cs.umass.edu'ya bir dosya yüklediniz ve bunu yaparken bir Wireshark paket izi yakalamanız beklenmektedir.

Wireshark'ı canlı bir ağ bağlantısında çalıştıramıyorsanız, ödev ekinde verilen ***tcp-wireshark-trace1-1.pcapng*** dosyasını[1] indirip kullanabilirsiniz.

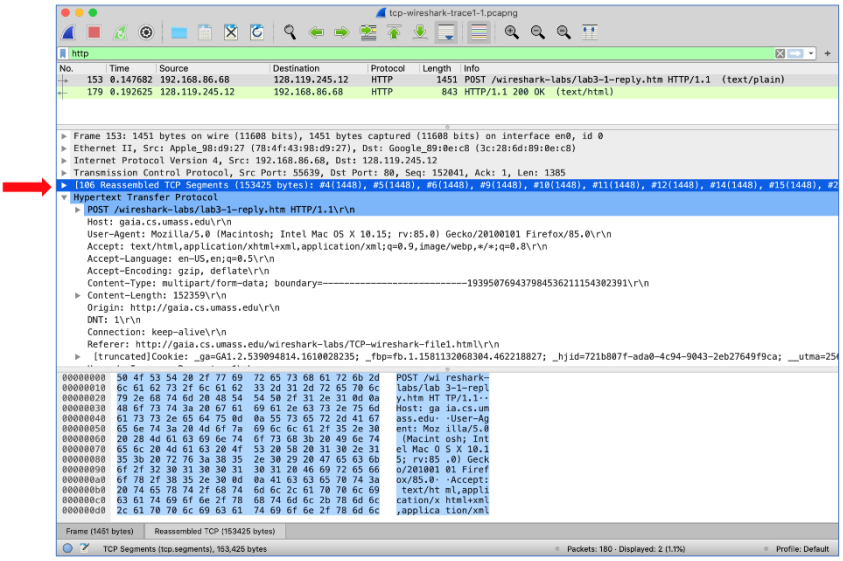
# Yakalanan ize ilk bakış

TCP bağlantısının davranışını ayrıntılı olarak incelemeden önce, izin (**trace**) üst düzey bir görünümünü ele alalım.

alice.txt dosyasını bilgisayarınızdan gaia.cs.umass.edu'ya yükleyen HTTP POST mesajına bakarak başlayalım. Wireshark izlemenizde bu dosyayı bulun ve HTTP POST mesajına daha dikkatli bir şekilde bakabilmemiz için HTTP mesajını genişletin. Wireshark ekranınız Şekil 3 gibi görünmelidir.

[1] [1] Ödev ekinde verilen ***tcp-wireshark-trace1-1.pcapng*** dosyasını indiriniz. Bu izleme dosyaları (trace file), kendi başınıza paketleri yakalamadan bu Wireshark laboratuvar sorularını yanıtlamak için kullanılabilir. Bir izleme dosyasını indirdikten sonra, onu Wireshark'a yükleyebilir ve File açılır menüsünü kullanarak, Open'ı seçerek ve ardından izleme dosyası adını seçerek izlemeyi görüntüleyebilirsiniz.




**Şekil 3.** alice.txt dosyasını bilgisayarınızdan gaia.cs.umass.edu'ya yükleyen HTTP POST mesajının geniş hali

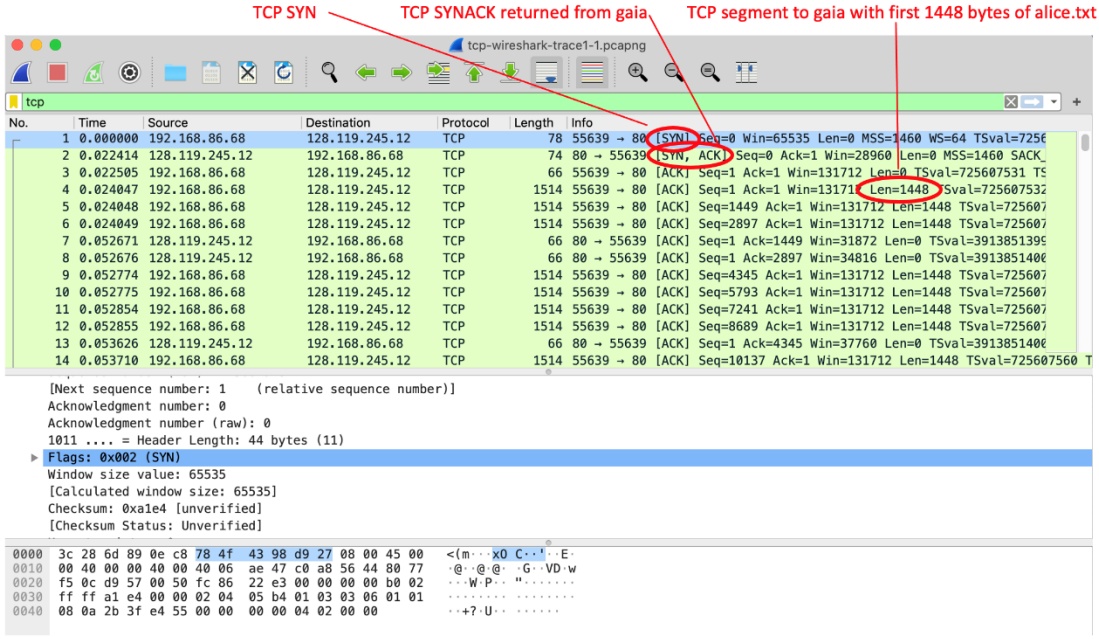
Burada dikkat edilmesi gereken birkaç nokta vardır:

- HTTP POST mesajınızın gövdesi (Uygulama katmanı (**application-layer**)), 152K bayttan büyük bir dosya olan alice.txt dosyasının içeriğini içerir. Tamam, o kadar büyük değil, ancak bu HTTP POST mesajının yalnızca bir TCP segmentinde yer alması için çok büyük olacak!
- Aslında, Şekil 3'teki Wireshark penceresinde gösterildiği gibi, HTTP POST mesajının 106 TCP segmentine yayıldığını görüyoruz. Bu, Şekil 3'te kırmızı okun yerleştirildiği yerde gösterilmiştir Buraya dikkatli bakarsanız, Wireshark'ın size gerçekten yardımcı olduğunu, POST mesajının başlangıcını içeren ilk TCP segmentinin Şekil 3'teki örnek için özel izlemede #4 numaralı paket olduğunu görebilirsiniz. 2 numaralı dipnotta not edilen *tcp-wireshark-trace1-1* izidir. İzdeki 5 numaralı paketteki POST mesajını içeren ikinci TCP segmenti vb.

Şimdi bazı TCP segmentlerine bakarak “elimizi kirletelim”.

- Öncelikle, Wireshark penceresinin üst kısmındaki görüntü filtresi belirtimi penceresine “tcp” (küçük harf, tırnak işaretleri olmadan) yazarak Wireshark penceresinde görüntülenen paketleri filtreleyin. Wireshark ekranınız Şekil 4'teki gibi görünmelidir. Şekil 4'te, SYN bit setine sahip TCP segmentini görebilirsiniz – bu, HTTP POST mesajını ve alice.txt dosyasını taşıyacak olan ve gaia.cs'ye TCP bağlantısını kuran **three-way handshake’**in ilk TCP mesajıdır. Ayrıca, POST mesajını ve alice.txt dosyasının başlangıcını taşıyan TCP segmentinin (yukarıda söylendiği gibi paket #4) yanı sıra SYNACK segmentini (TCP **three-way handshake’**in ikinci adımı) de görebilirsiniz. Elbette, kendi izleme dosyanızı alıyorsanız, paket numaraları farklı olacaktır, ancak Şekil 3 ve 4'te gösterilene benzer bir davranış görmelisiniz.

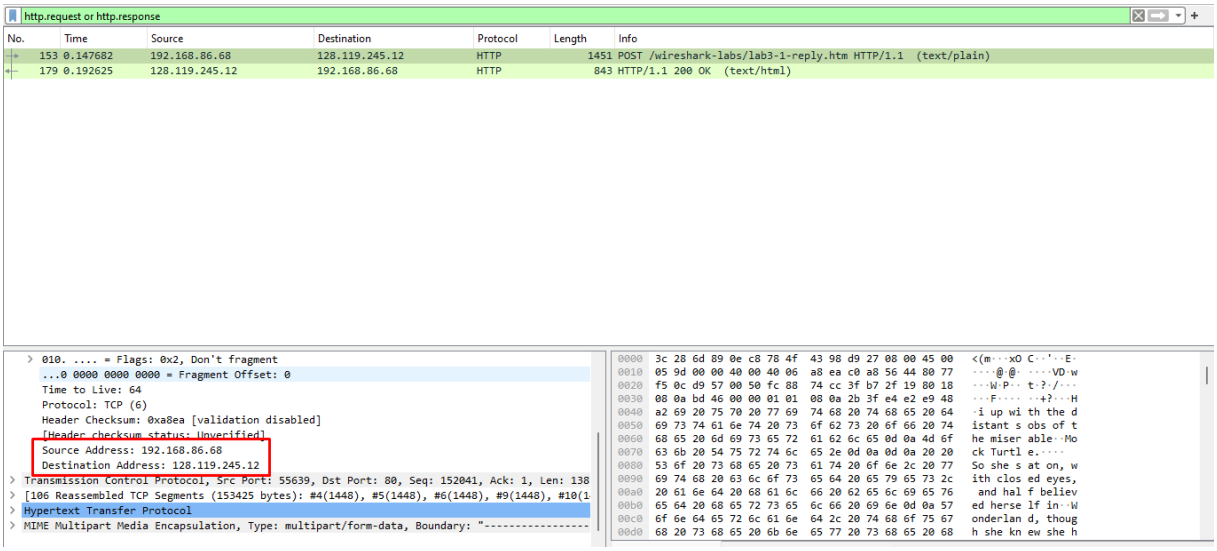
 



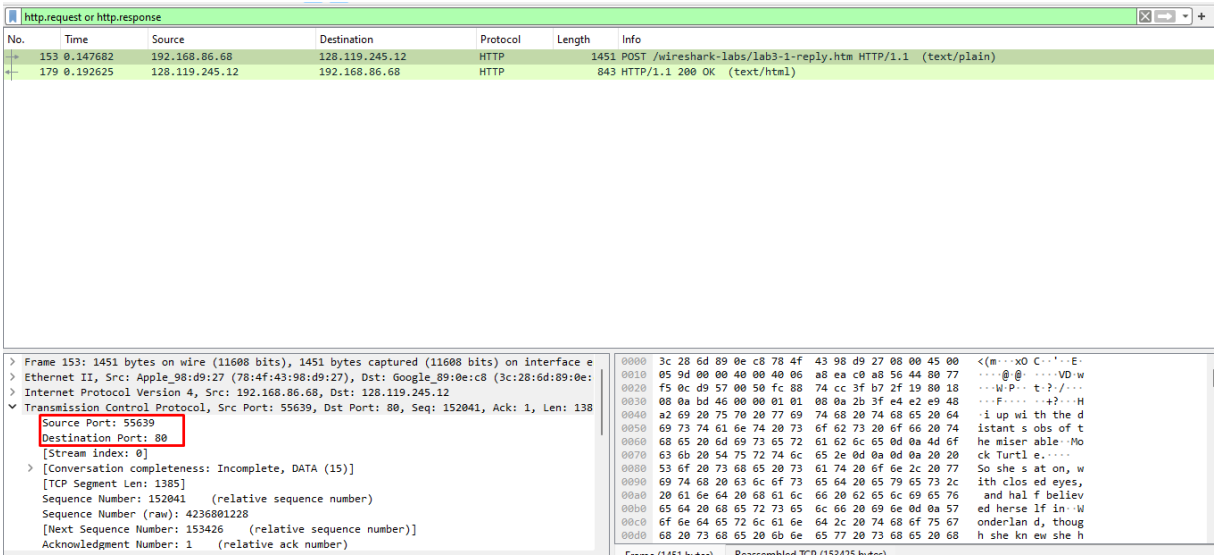

**Şekil 4.** HTTP POST mesajının (alice.txt dosyası dahil) gaia.cs.umass.edu'ya gönderilmesiyle ilgili TCP segmentleri

Aşağıdaki soruları kendi canlı izlemeniz ile veya ödev ekinde verilen Wireshark paket dosyasını kullanarak yanıtlayınız.

1. alice.txt dosyasını gaia.cs.umass.edu'ya aktaran istemci bilgisayar (kaynak) tarafından kullanılan IP adresi ve TCP port numarası nedir? Bu soruyu yanıtlamak için, bir HTTP mesajı seçmek ve bu HTTP mesajını taşımak için kullanılan TCP paketinin ayrıntılarını "seçilen paket başlık penceresinin ayrıntılarını (details of the selected packet header window)" kullanarak keşfetmek muhtemelen en kolay yoldur.



2. gaia.cs.umass.edu'nun IP adresi nedir? Bu bağlantı için hangi port numarası üzerinden TCP segmentleri gönderiyor ve alıyor?



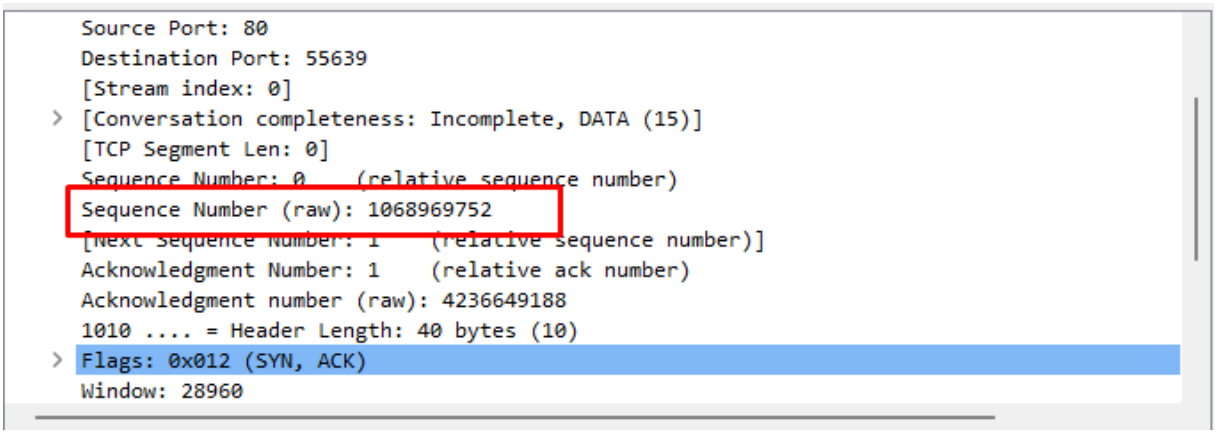

Bu laboratuvar HTTP yerine TCP ile ilgili olduğundan, Wireshark'ın “yakalanan paketlerin listesi” penceresini, yukarıdaki Şekil 4'te gösterildiği gibi HTTP mesajları yerine HTTP mesajlarını içeren TCP segmentleri hakkında bilgi gösterecek şekilde değiştirin. Aradığımız şey bu- bilgisayarınız ile gaia.cs.umass.edu arasında gönderilen bir dizi TCP segmenti!

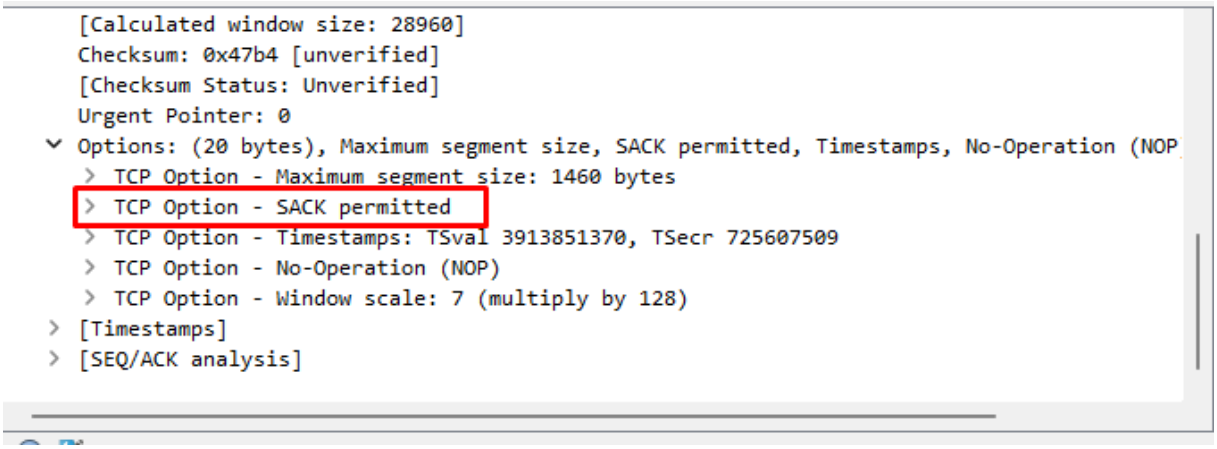
 

# TCP Temelleri

TCP segmentleri için aşağıdaki soruları yanıtlayın:

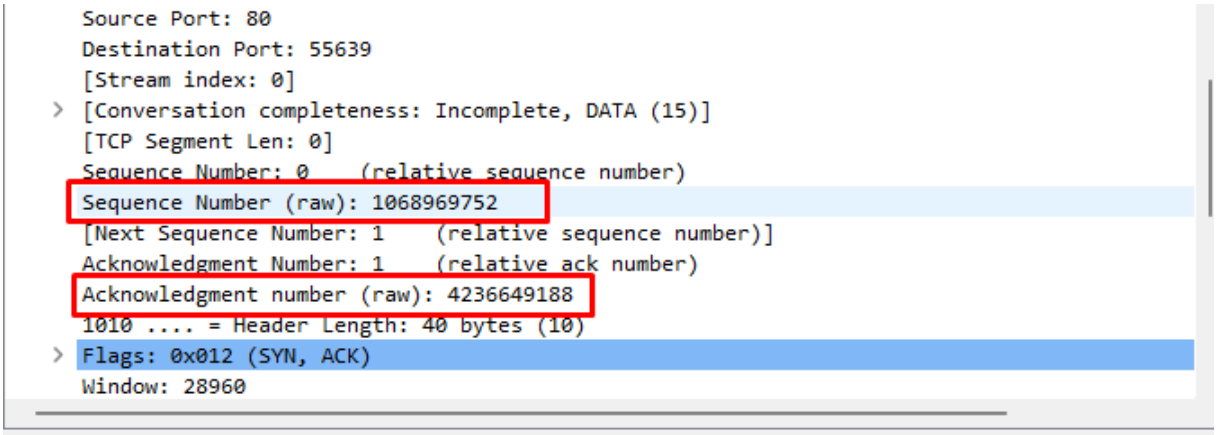
3. İstemci bilgisayar ile gaia.cs.umass.edu arasında TCP bağlantısını başlatmak için kullanılan TCP SYN segmentinin *sequence number* 'ı nedir? (Not: Bu, TCP segmentinin kendisinde taşınan "ham (**raw**)" sıra numarasıdır; Wireshark penceresindeki "No." sütunundaki paket # DEĞİLDİR. TCP veya UDP'de "paket numarası" diye bir şey olmadığını daha önce söylemiştik. Ayrıca bunun, bu TCP oturumunun başlangıç sıra numarasına göre göreceli sıra numarası olmadığını görebilirsiniz). Segmenti bir SYN segmenti olarak tanımlayan bu TCP segmentinde ne var? Bu oturumdaki TCP alıcısı Selective Acknowledgments'ları kullanabilecek mi?



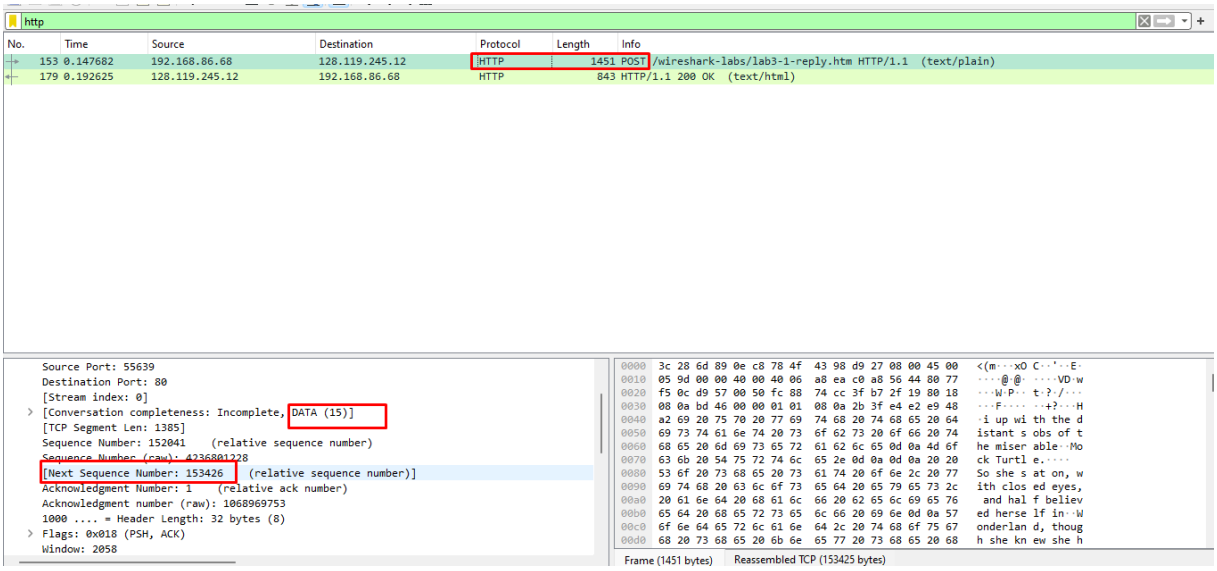



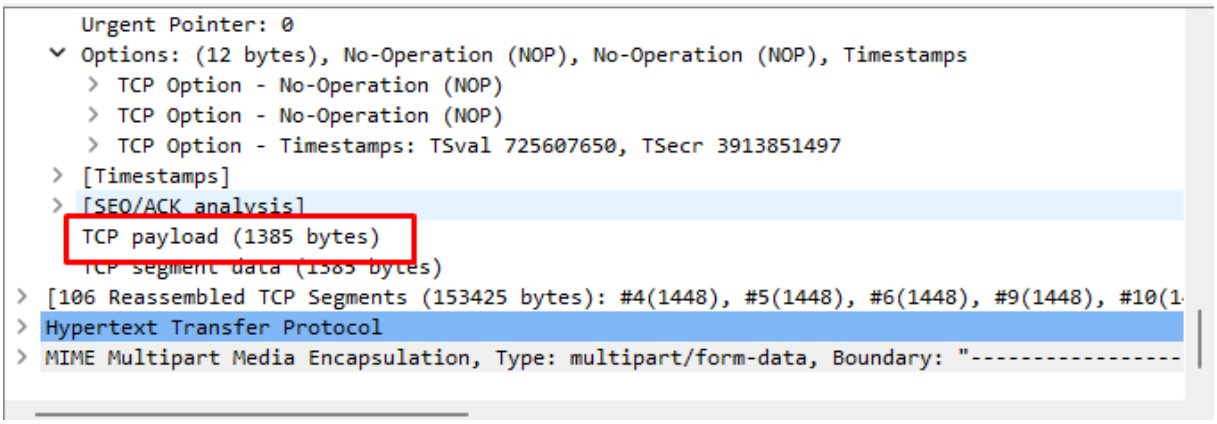
4. SYN'ye yanıt olarak gaia.cs.umass.edu tarafından istemci bilgisayara gönderilen SYNACK segmentinin *sequence number* 'ı nedir? Segmenti bir SYNACK segmenti olarak tanımlayan segmentte ne var? SYNACK segmentindeki Acknowledgement alanının değeri nedir? gaia.cs.umass.edu bu değeri nasıl belirledi?




5. HTTP POST komutunun başlığını içeren TCP segmentinin *sequence number*'ı nedir? POST mesaj başlığını bulmak için Wireshark penceresinin alt kısmındaki paket içerik alanını araştırmanız ve DATA alanında ASCII metni "POST" olan bir segment aramanız gerekmektedir. Bu TCP segmentinin yük/veri (payload/data) alanında kaç bayt veri bulunur? Aktarılan alice.txt dosyasındaki tüm veriler bu tek parçaya sığdı mı?

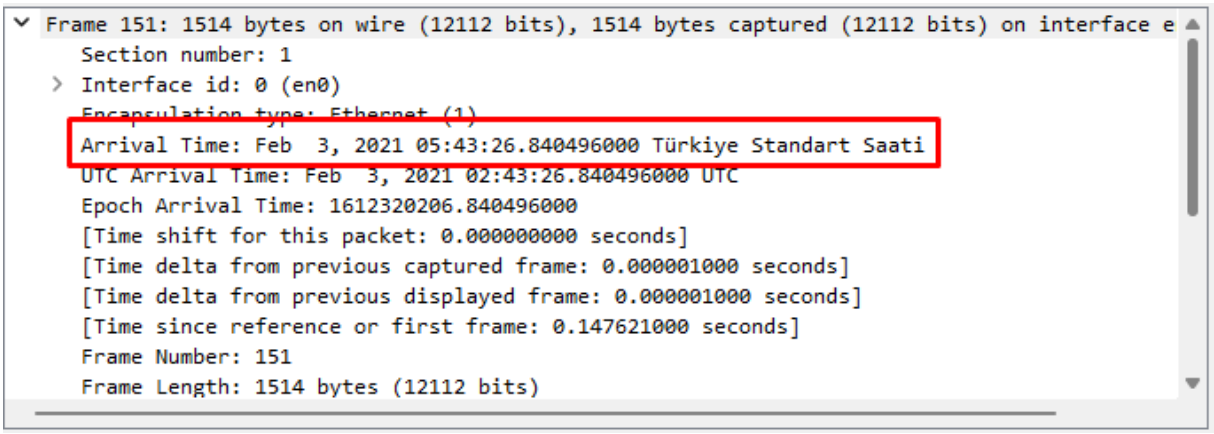





6. TCP bağlantısının veri aktarım kısmında HTTP "POST" içeren TCP segmentini ilk segment olarak kabul edin.
   - TCP bağlantısının veri aktarım kısmındaki ilk segment (HTTP POST'u içeren) ne zaman gönderildi?
   - Bu ilk veri içeren segment için ACK ne zaman alındı?
   - Bu ilk veri içeren bölüm için RTT nedir?
   - İkinci veriyi taşıyan TCP segmentinin ve onun ACK'sının RTT değeri nedir?
   - İkinci veriyi taşıyan segment için ACK alındıktan sonra EstimatedRTT değeri nedir?

   *Not*: Wireshark, gönderilen her TCP segmenti için RTT'yi çizmenize izin veren güzel bir özelliğe sahiptir. İstemciden gaia.cs.umass.edu sunucusuna gönderilen "yakalanan paketlerin listesi" penceresinde bir TCP segmenti seçin. Sonra şunu seçin: *Statistics->TCP Stream Graph->Round Trip Time Graph*



RTT (Round-Trip Time), bir segmentin gönderildiği andan onun ACK'sinin alındığı ana kadar geçen süredir.

**RTT Hesaplamaları**

**İlk Segment İçin RTT**

RTT1=ACK alınma zamanı−Go¨nderilme zamanı

RTT1=05:43:26.900557−05:43:26.840557=0.060 saniyeRTT1
=05:43:26.900557−05:43:26.840557=0.060 saniye

**İkinci Segment İçin RTT**

RTT2=ACK alınma zamanı−Go¨nderilme zamanıRTT2
RTT2=05:43:27.200000−05:43:27.000000=0.200 saniyeRTT2
=05:43:27.200000−05:43:27.000000=0.200 saniye

**EstimatedRTT Hesaplamaları**

TCP'nin RTT tahmin algoritması kullanılarak EstimatedRTT şu şekilde hesaplanır:

EstimatedRTTyeni=(1−$\alpha$)×EstimatedRTTeski+$\alpha$×SampleRTTEstimatedRTTyeni
=(1−$\alpha$)×EstimatedRTTeski+$\alpha$×SampleRTT

Genellikle $\alpha\alpha$ değeri 0.125 olarak alınır.

İlk segment için EstimatedRTT başlangıç olarak RTT_1 ile aynıdır:
EstimatedRTT1=0.060 saniyeEstimatedRTT1=0.060 saniye

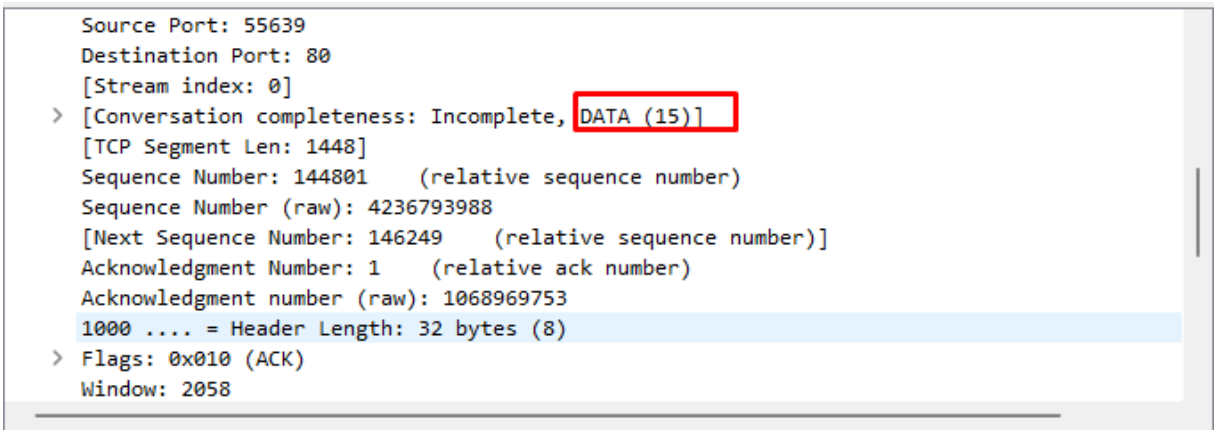
İkinci segmentten sonra EstimatedRTT şu şekilde hesaplanır:

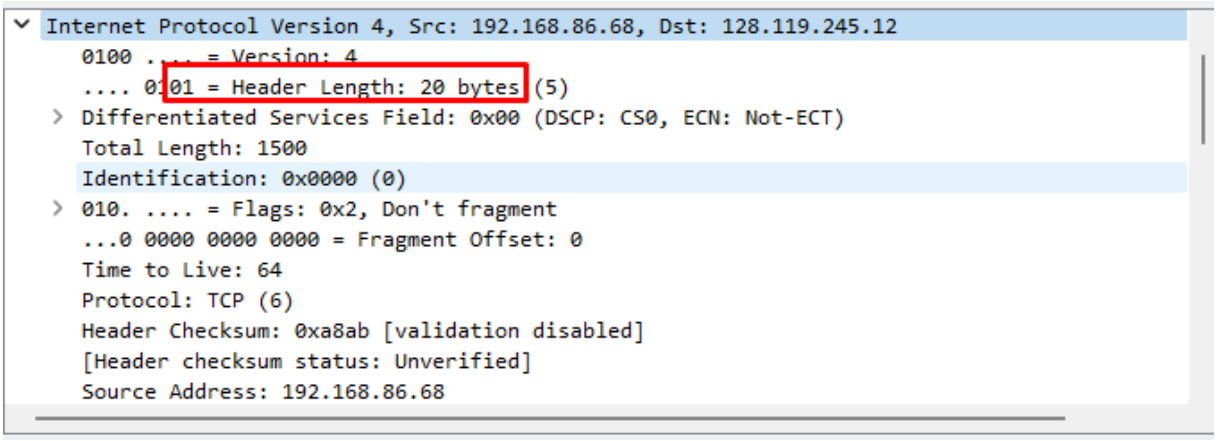
EstimatedRTTyeni=(1−0.125)×0.060+0.125×0.200EstimatedRTTyeni
=(1−0.125)×0.060+0.125×0.200
EstimatedRTTyeni=0.0525+0.025=0.0775 saniyeEstimatedRTTyeni
=0.0525+0.025=0.0775 saniye

1. İlk segmentin gönderilme zamanı: Feb 3, 2021 05:43:26.840557 Türkiye Standart Saati
2. İlk segment için ACK alınma zamanı: Eksik
3. İlk segment için RTT: 0.060 saniye
4. İkinci segment ve onun ACK'sı için RTT: 0.200 saniye
5. Yeni EstimatedRTT: 0.0775 saniye

7. Veri taşıyan ilk dört TCP segmentinin her birinin uzunluğu (başlık + yük) nedir?




1.20 +5 =25

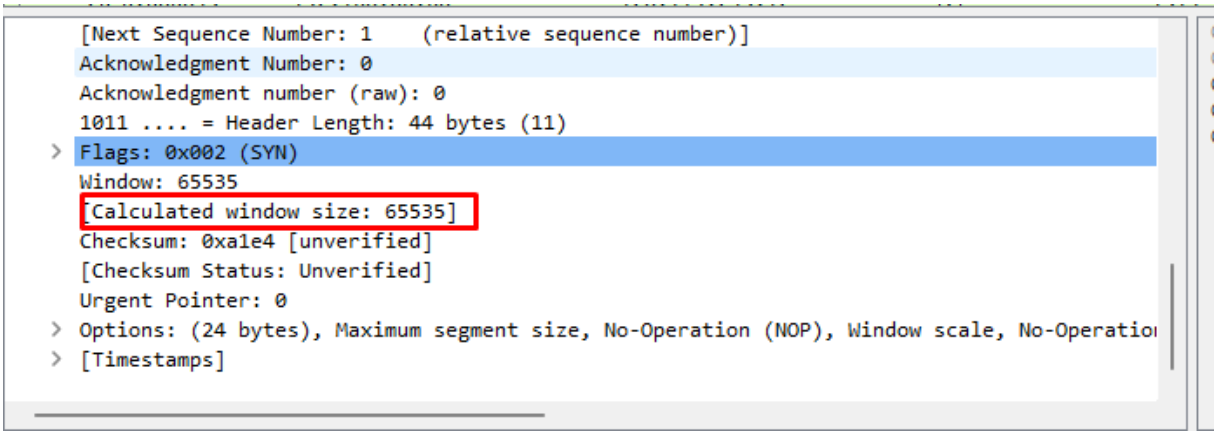
2.20+15=35

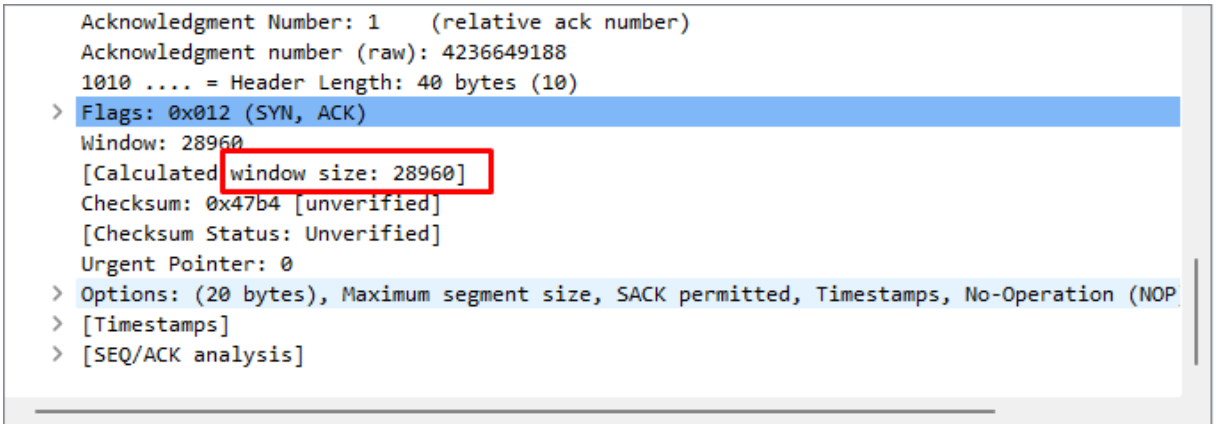
3.20+15=35

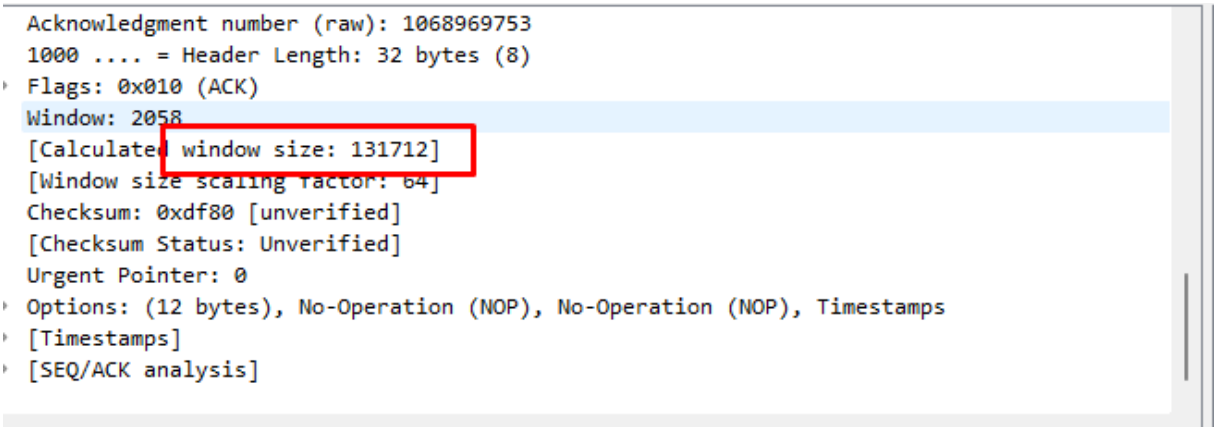
4.20+15=35

8. Veri taşıyan bu ilk dört TCP segmenti arasında gaia.cs.umass.edu tarafından istemciye bildirilen minimum kullanılabilir arabellek alanı (**buffer space**) miktarı nedir? Arabellek alanının olmaması, göndericiyi bu ilk dört veri taşıma bölümü için hiç kısıtlıyor mu?



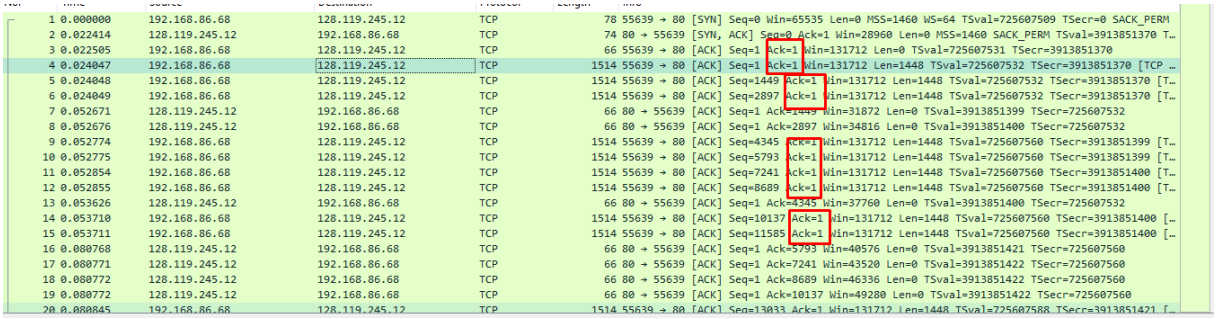

 

```
Sequence Number: 1    (relative sequence number)
Sequence Number (raw): 4236649188
[Next Sequence Number: 1449    (relative sequence number)]
Acknowledgment Number: 1    (relative ack number)
Acknowledgment number (raw): 1068969753
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)
Window: 2058
[Calculated window size: 131712]
[Window size scaling factor: 64]
Checksum: 0xbd21 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
```

pencere boyutunun geçici bir süre için azalması, ağdaki bir tıkanıklığı veya diğer performans sorunlarını işaret edebilir. Daha sonra, pencere boyutunun artması, ağdaki koşulların düzelmesi veya alıcının daha fazla veri alabilecek duruma gelmesi anlamına gelir.

9. İzleme dosyasında yeniden iletilen bölümler var mı? Bu soruyu cevaplamak için neyi kontrol ettiniz?

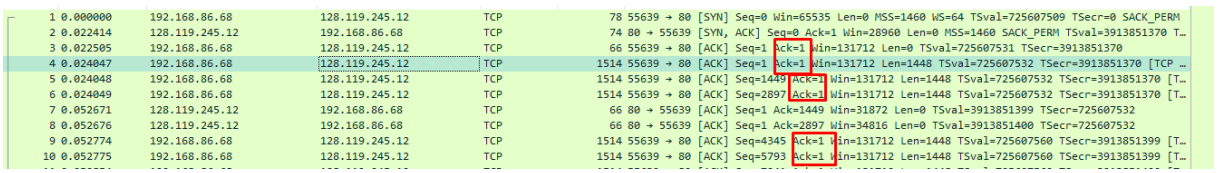
TCP Acknowledgment (ACK) Numaraları: Aynı veri dizisini taşıyan farklı TCP segmentlerinin ACK numaralarını kontrol ederek, aynı veri dizisinin birden fazla kez alındığına dair işaretler arayabilirsiniz.




10. Alıcı, istemciden gaia.cs.umass.edu'ya gönderilen ilk on veri taşıyan bölüm arasında bir ACK'de tipik olarak ne kadar veri kabul eder? Bu ilk on veri taşıyan bölüm arasında, alıcının alınan diğer her bölümü ACK'ladığı durumları belirleyebilir misiniz?

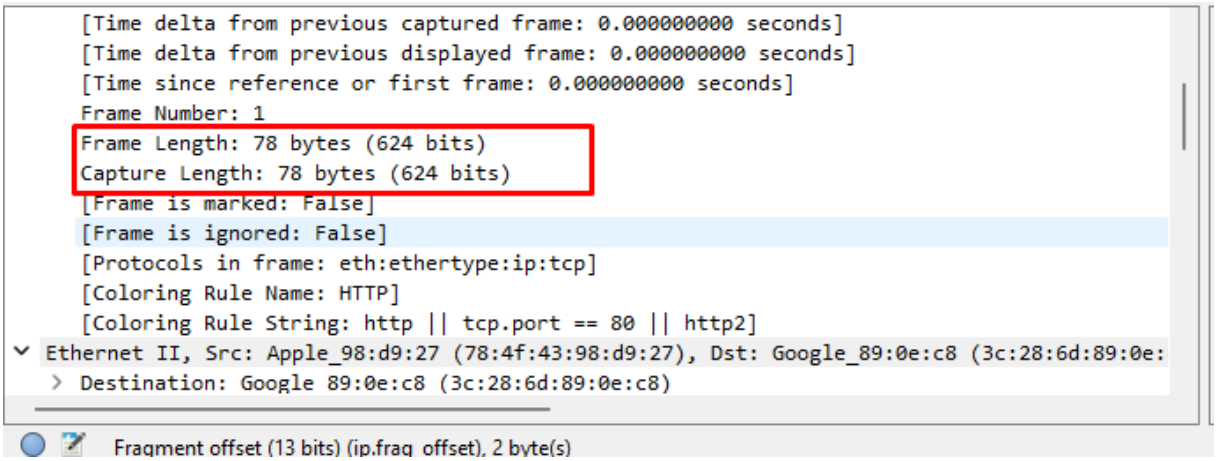
ACK numarasının, alıcının beklediği bir sonraki veri dizisinin dizin numarasına eşit ise

ACK'landığı drumları görebiliriz.



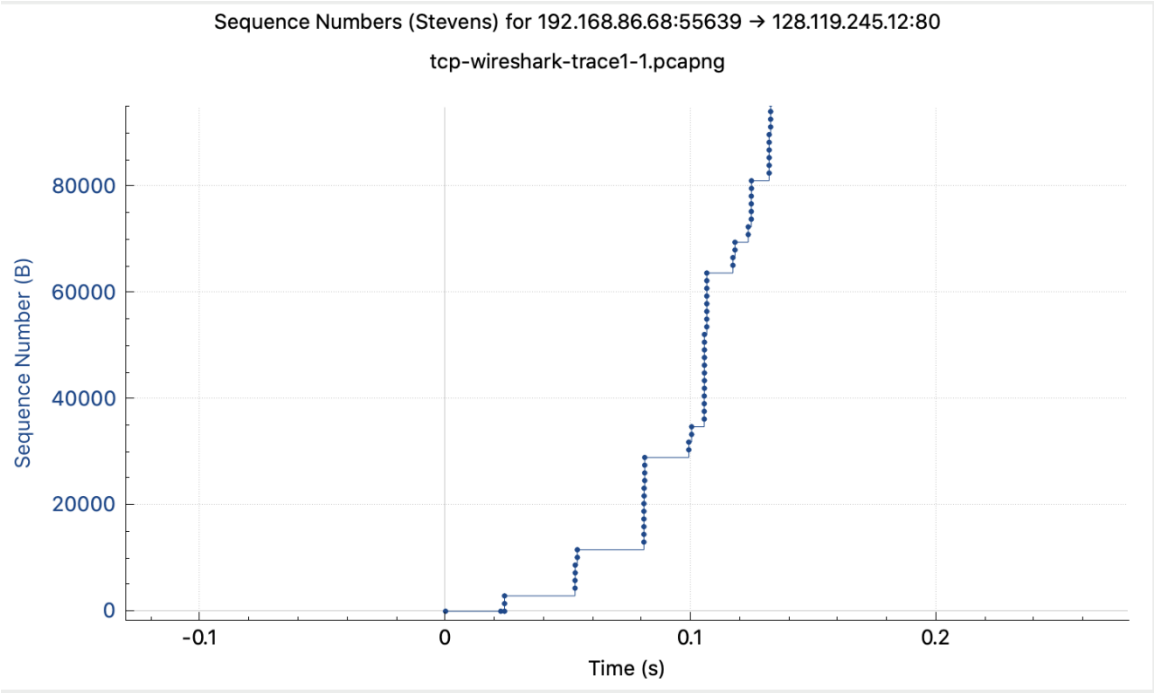
11. TCP bağlantısı için verim (**throughput**) -birim zamanda aktarılan bayt sayısı- nedir? Bu değeri nasıl hesapladığınızı açıklayınız.




# TCP tıkanıklık kontrolü

Şimdi istemciden sunucuya birim zamanda gönderilen veri miktarını inceleyelim. Bunu Wireshark penceresindeki ham verilerden hesaplamak yerine, verileri çizmek için Wireshark'ın TCP grafik yardımcı programlarından biri olan *Time-Sequence-Graph(Stevens)* kullanacağız.

- Wireshark'ın "yakalanan paketlerin listesi" penceresinde alice.txt dosyasının istemciden gaia.cs.umass.edu'ya aktarılmasına karşılık gelen istemci tarafından gönderilen bir TCP segmenti seçin. Ardından menüyü seçin: *Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens)*. Şekil 5'teki grafiğe benzer bir grafik görmelisiniz. Grafiğinizin Şekil 5'teki gibi görünmesini sağlamak için eksenlerde gösterilen aralıkları genişletmeniz, küçültmeniz ve karıştırmanız gerekebilir.




**Şekil 5.** TCP segmentlerinin sıra numarasına karşı zaman grafiği (Stevens formatı).

Burada her nokta, gönderilen bir TCP segmentini temsil eder ve segmentin sıra numarasını, gönderildiği zamana karşı çizer. Üst üste yığılmış bir dizi noktanın, gönderici tarafından arka arkaya gönderilen bir dizi paketi (bazen paket "filo" olarak adlandırılır) temsil ettiğini unutmayın.

Paket izleme trace *tcp-wireshark-trace1-1*'deki TCP segmentleri için aşağıdaki soruyu yanıtlayın

12. Gönderilen segmentlerin sıra numarasına karşı zaman grafiğini görüntülemek için *Time-Sequence-Graph(Stevens*) çizim aracını kullanın istemciden gaia.cs.umass.edu sunucusuna $t$ = 0.025, $t$ = 0.053, $t$ = 0.082 ve $t$ = 0.1. civarında gönderilen paketlerin "fleets"lerini düşünün. Bunun TCP'nin yavaş başlatma aşamasında (**slow start phase**) mı, tıkanıklıktan kaçınma aşamasında (**congestion avoidance phase**) mı yoksa başka bir aşamada mı göründüğünü yorumlayın. Şekil 6, bu verilerin biraz farklı bir görünümünü göstermektedir.

Yavaş başlatma aşamasında, TCP penceresi (cwnd) hızla büyür ve segmentlerin sıra numarası zamanla hızla artar. Bu durumda, grafikte dik bir eğim gözlemlenebilir.

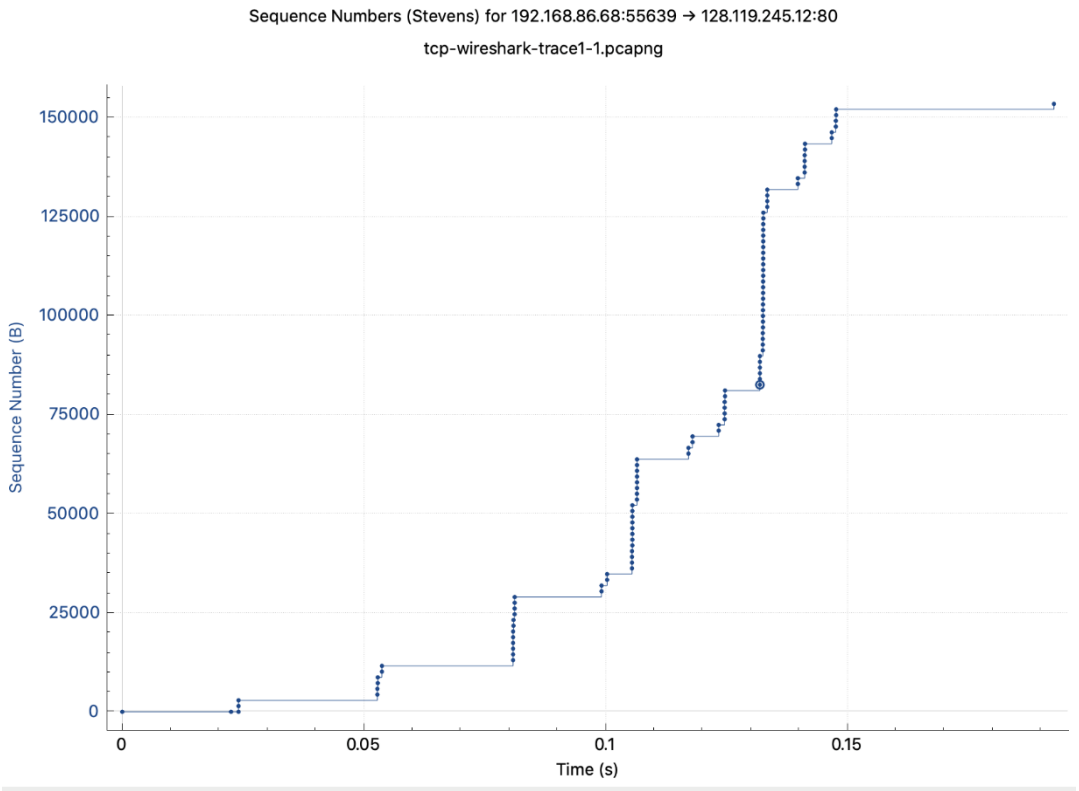
Tıkanıklıktan kaçınma aşamasında, pencere büyümesi daha lineer hale gelir ve segmentlerin sıra numarası daha yumuşak bir şekilde artar. Bu durumda, grafikte daha düşük bir eğim gözlemlenebilir.

13. Bu segment "filolarının" bir miktar periyodikliği var gibi görünüyor. Period hakkında neler söyleyebilirsiniz?

"Filoların" belirli bir periyodiklik gösterdiği durumlar, genellikle TCP'nin tıkanıklıktan kaçınma mekanizmalarının işlediği durumları yansıtabilir. Bu periyodiklik, ağda oluşan tıkanıklık durumlarına yanıt olarak TCP akış kontrol mekanizmalarının davranışını yansıtabilir.

TCP bağlantılarında periyodik davranışlar, ağda tıkanıklık olup olmadığını belirlemek ve tıkanıklıktan kaçınmak için kullanılan akış kontrol algoritmalarının bir sonucu olabilir. TCP'nin tıkanıklıktan kaçınma mekanizmaları, ağdaki trafiğin durumunu izler ve uygun önlemleri alarak ağdaki tıkanıklığı azaltmaya veya önlemeye çalışır. Bu önlemler genellikle pencere boyutu ayarlamaları, ACK (onay) paketlerinin alınma hızı ve paketlerin gönderilme hızı gibi parametrelerin dinamik olarak ayarlanmasını içerir.




**Şekil 6.** Şekil 5'teki ile aynı verilerin başka bir görünümü.