ETH zürich

Networked Systems
ETH Zürich — seit 2015

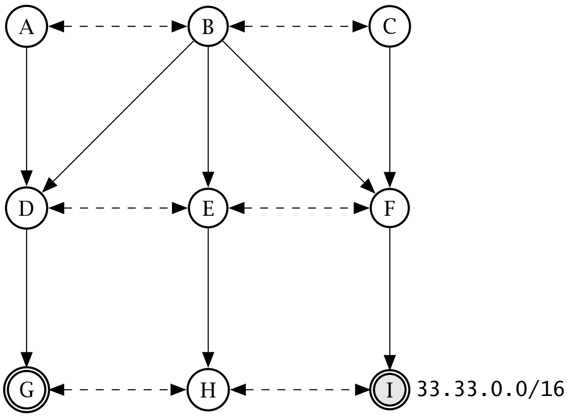
Communication Networks

Prof. Laurent Vanbever

Exercise 9 – BGP Hijack and Reliable Transport Concepts

Return of the BGP Hijack

9.1 BGP Security (Exam Question 2020)



An Internet topology of 9 ASes in which AS I announces a prefix and AS G tries to hijack it.

Consider the Internet topology consisting of 9 Autonomous Systems (ASes) in the Figure above. Single-headed plain arrows point from providers to their customers (AS A is the provider of AS D) while double-headed dashed arrows connect peers (AS A and AS B are peers). Each AS is made up of a single BGP router and applies the default selection and exportation BGP policies based on their customers, peers and providers.

In this task, the routers break ties using the AS number of the neighbor: in case multiple routes are equally good, the router selects the route of the neighbor with the lowest AS number (in alphabetical order; e.g., a route from AS A is preferred over AS B in case of a tie).

AS I is the origin of prefix `33.33.0.0/16` and advertises it to its neighbors. Independently of what the external advertisements are, AS I ***always*** prefers its internal route to reach any IP destination in `33.33.0.0/16`.

a) AS *G* wants to hijack the traffic going to AS *I* for `33.33.0.0/16`. It starts advertising the exact same prefix with itself, AS *G*, as origin. From which ASes is it able to hijack the traffic?

b) The ASes notice the hijack and, as a counter-measure, deploy Resource Public Key Infrastructure (RPKI) Internet-wide. After that, from which ASes is the attacker able to hijack the traffic by still advertising the exact same prefix with itself as origin?

c) RPKI has a flaw. What is the problem of RPKI? How can AS *G* hijack the prefix `33.33.0.0/16` despite RPKI? From which ASes is AS *G* able to hijack the traffic?

d) In response, the ASes switch to BGPSec (Secure BGP). Explain what security it provides and how AS *E* can detect that the announcement from AS *G* has a forged AS path.

Reliable Transport Concepts

9.2 Reliable versus Unreliable Transport

In the lecture, you have learned how a reliable transport protocol can be built on top of a best-effort delivery network. However, some applications still use an unreliable transport protocol.

a) What are the characteristics of best-effort and of reliable transport?

b) What could be advantages of using an unreliable transport protocol?

c) What type of applications are suitable to use unreliable transport protocols?

d) The User Datagram Protocol (UDP) only provides unreliable transport. Assume you are forced to use a network which only supports UDP as a transport protocol. You must transmit an important document which eventually should be correctly transmitted. Do you see a way to implement some of the reliable transport mechanisms despite using UDP?

9.3 Negative Acknowledgments

In the lecture, we have mainly looked at transport protocols using (positive) Acknowledgments (ACKs). However, we could also use so called Negative Acknowledgments (NAKs or NACKs). In this case, the receiver is sending a NAK for every packet that it *did not* receive. To detect lost packets, the receiver looks at the sequence numbers of all the received packets and sends NAKs for every missing sequence number. After receiving a NAK, the sender will retransmit the corresponding packet.

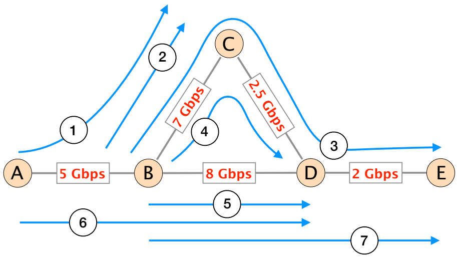
a) Assuming a network with nearly no packet loss, what could be the main advantage of using NAKs?

b) Assume now that the receiver will immediately send a NAK as soon as it detects a gap in the received packet numbers. E.g. for the following packet number sequence [4, 5, 7] the receiver would immediately send a NAK for packet 6. Can you see a problem with this implementation? How could you (partially) mitigate the problem?

c) So far, NAKs look like a good alternative to (positive) ACKs. Nonetheless, TCP – the currently most-widely used transport protocol – is *not* using NAKs. There has to be a problem. Assume that the sender is transmitting 5 packets (with sequence number 1 to 5). Find at least two sequences of packet or NAK losses such that the **sender** wrongly assumes that the 5 packets were correctly received.

9.4 Fairness

Consider the network on the left consisting of 5 nodes (A to E). Each link has a maximal bandwidth indicated in red. 7 flows (1 to 7) are using the network at the same time. You can assume that they have to send a lot of traffic and will use whatever bandwidth they will get. Apply the max-min fair allocation algorithm discussed in the lecture to find a fair bandwidth allocation for each flow. You can use the table below. In the top row, indicate which link is the current bottleneck. The other rows contain the corresponding bandwidth distribution for each flow.



A network with shared links and 7 flows.

Bottleneck link						
Flow 1 A - B - C						
Flow 2 B - C						
Flow 3 B - C - D - E						
Flow 4 B - C - D						
Flow 5 B - D						
Flow 6 A - B - D						
Flow 7 B - D - E						

9.5 Go-Back-N Warm-Up Questions

Sender and receiver keep separate windows and buffers for sent and received segments.

a) Compare how the sender and the receiver advance their respective windows.

b) Which segments does the *sender* buffer? When can segments be removed from the buffer?

c) A *receiver* typically buffers out-of-order segments. What is the advantage of such a buffer?

Cumulative ACKs acknowledge that all segments *up to* the acknowledged segment have been received.

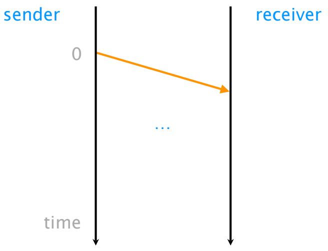
d) Why are cumulative ACKs used? Do they help with lost data segments, lost ACKs, or both?

When *Fast Retransmit* is used, the sender retransmits a segment after duplicate ACKs.

e) How does Fast Retransmit improve performance?

f) Compare Fast Retransmit in the case of mild congestion (some segment losses) and heavy congestion (nearly all segments lost).

9.6 Reliable Transport (Exam Style Question)



How long would a transfer take?

Consider a Go-Back-N sender and receiver directly connected by a 10 Mbps link with a propagation delay of 100 milliseconds. The retransmission timer is set to 3 seconds and the window has a length of 4 segments.

Draw a time-sequence diagram (see left) showing the transmission of 10 segments (each segment contains 10 000 bits). An ACK is transmitted as soon as the last bit of the corresponding data segment is received. The size of an ACK is very small, that means they have a negligible transmission delay.

a) Draw the time-sequence diagram for the case where there are no losses.

b) Draw the time-sequence diagram for the case where the 3rd and the last segment are lost once.