

ТРАНСПОРТНИЙ РІВЕНЬ ПРОТОКОЛИ TCP/UDP

ТРАНСПОРТНИЙ РІВЕНЬ ПРОТОКОЛИ TCP/UDP



Модель OSI

Протоколи TCP/UDP

Основні функції протоколу TCP

Порти

Заголовок сегмента TCP

Приклад заголовка TCP у Wireshark

Встановлення TCP з'єднання

Розрив TCP з'єднання

Керування втратами сегментів

Протокол UDP

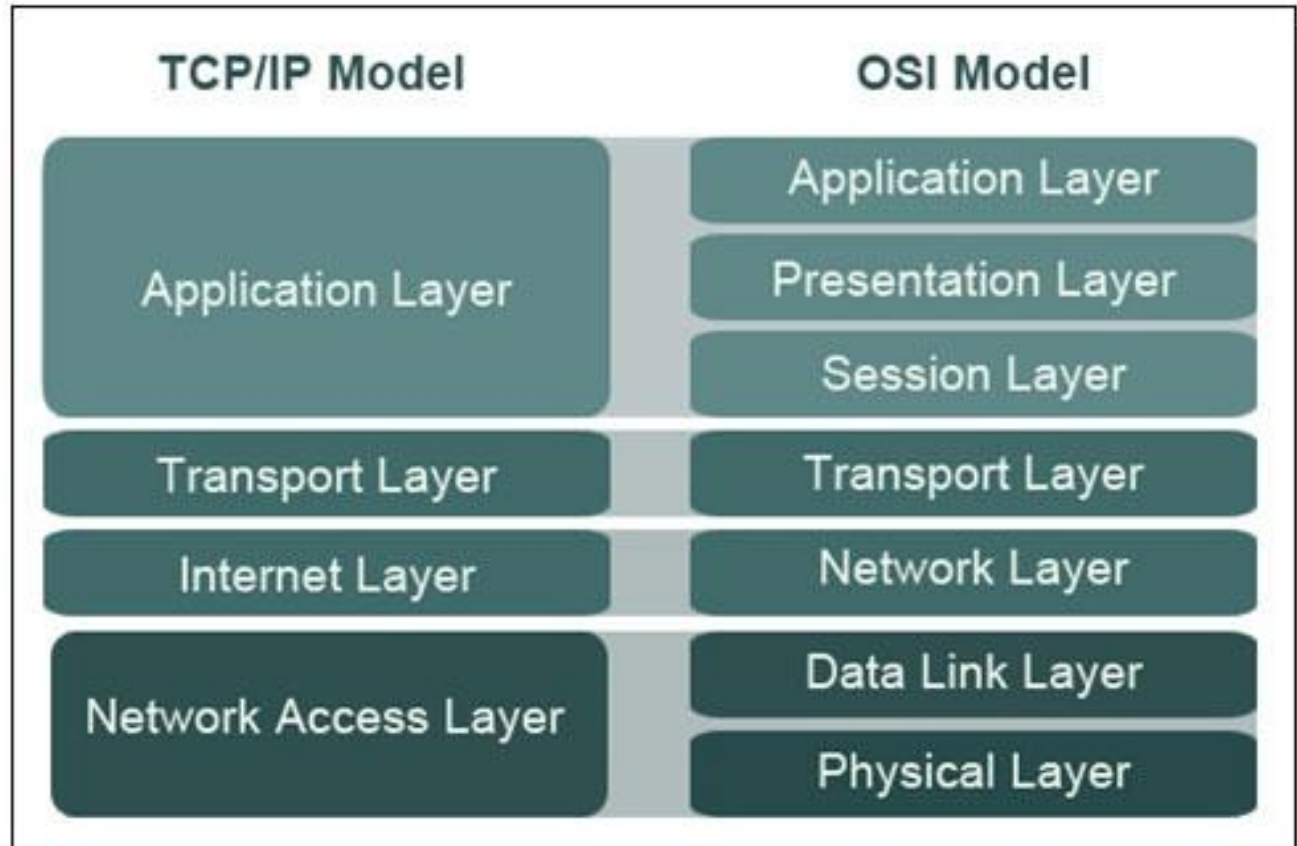
Запитання



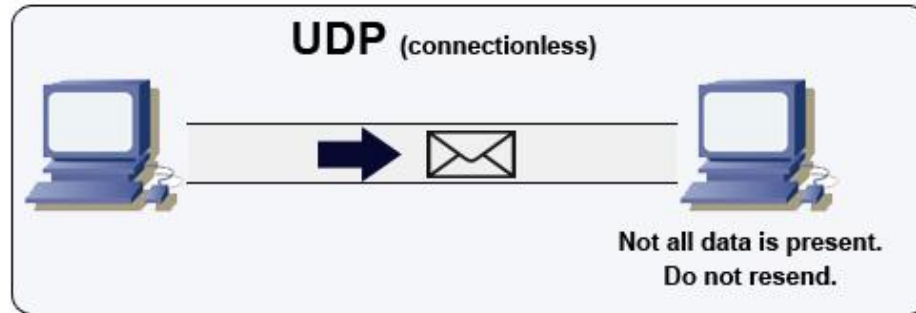
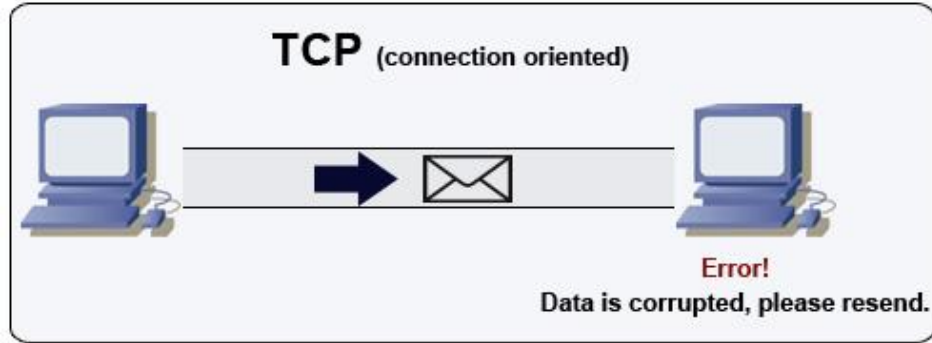
МОДЕЛЬ OSI

- ПРИКЛАДНИЙ (APPLICATION)
- ПРЕДСТАВНИЦЬКИЙ (PRESENTATION)
- СЕАНСОВИЙ (SESSION)
- ТРАНСПОРТНИЙ (TRANSPORT)
- МЕРЕЖЕВИЙ (NETWORK)
- КАНАЛЬНИЙ (DATA LINK)
- ФІЗИЧНИЙ (PHYSICAL)

МОДЕЛЬ OSI



ПРОТОКОЛИ TCP/UDP



ОСНОВНІ ФУНКЦІЇ ПРОТОКОЛУ TCP

Основні функції, які виконуються всіма протоколами транспортного рівня:

- Сегментація і реасемблінг
- Мультиплексування з'єднань
- Протокол TCP додатково виконує наступні функції
- Попереднє встановлення з'єднання (Establishing a Session)
- Надійна доставка даних (Reliable Delivery)
- Відновлення первинного порядку отриманих даних (Same Order Delivery)
- Керування передачею (Flow Control)

ПОРТИ

- **Well Known Ports** (Добре відомі порти) - порти з номерами від 0 до 1023. Ці номери зарезервовані для служб і системних програм
- **Registered Ports** (Зареєстровані порти) - порти з номерами від 1024 до 49151. Ці номери портів призначені користувацьким процесам і програмам
- **Dynamic or Private Ports** (Динамічні або приватні порти) - порти з номерами від 49152 до 65535. Ці порти переважно динамічно вибираються клієнтськими програмами при ініціюванні з'єднання

ЗАГОЛОВОК СЕГМЕНТА TCP

TCP Header

Offsets Octet		0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 0 0 0			N S	C W R E	E C G	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size																		
16	128	Checksum																Urgent pointer (if URG set)															
20	160	Options (if Data Offset > 5, padded at the end with "0" bytes if necessary)																															
...																															

ЗАГОЛОВОК СЕГМЕНТА TCP

Значення полів заголовка TCP, які містять прапорці, що використовуються в процесі встановлення з'єднання. TCP використовує 6 прапорців (в порядку їх появи)

URG – наявність в сегменті термінових даних

ACK – сигналізує про необхідність підтвердження

PSH – сигнал фінального сегмента в буфері відправника

RST – скидування з'єднання або його розрив

SYN – запит на встановлення з'єднання

FIN - запит на розрив з'єднання

ПРИКЛАД ЗАГОЛОВКА TCP У WIRESHARK

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: `or tcp.flags == 18 or tcp.flags == 16) and tcp.port == 80` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
61	1.999969	1.1.22.177	1.2.24.64	TCP	26737 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=873768152 TSER=0
62	0.000021	1.2.24.64	1.1.22.177	TCP	http > 26737 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=873768312 TSER=873768152
63	0.000013	1.1.22.177	1.2.24.64	TCP	26737 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=873768425 TSER=873768312
88	0.899962	1.1.86.231	1.2.88.33	TCP	26499 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=880968125 TSER=0
89	0.000019	1.2.88.33	1.1.86.231	TCP	http > 26499 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=880968266 TSER=880968125
90	0.000012	1.1.86.231	1.2.88.33	TCP	26499 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=880968374 TSER=880968266
100	0.399969	1.1.115.63	1.2.116.92	TCP	26209 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=884168126 TSER=0
101	0.000019	1.2.116.92	1.1.115.63	TCP	http > 26209 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=884168262 TSER=884168126
102	0.000011	1.1.115.63	1.2.116.92	TCP	26209 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=884168367 TSER=884168262
126	0.999971	1.1.220.199	1.2.221.211	TCP	35020 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=892168132 TSER=0
127	0.000017	1.2.221.211	1.1.220.199	TCP	http > 35020 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=892168258 TSER=892168132
128	0.000013	1.1.220.199	1.2.221.211	TCP	35020 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=892168371 TSER=892168258
149	0.899970	1.1.219.122	1.2.220.200	TCP	18269 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=899368130 TSER=0
150	0.000016	1.2.220.200	1.1.219.122	TCP	http > 18269 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=899368247 TSER=899368130
151	0.000013	1.1.219.122	1.2.220.200	TCP	18269 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=899368366 TSER=899368247
175	0.899970	1.1.28.172	1.2.29.172	TCP	18454 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=906568127 TSER=0
176	0.000018	1.2.29.172	1.1.28.172	TCP	http > 18454 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=906568256 TSER=906568127
177	0.000011	1.1.28.172	1.2.29.172	TCP	18454 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=906568355 TSER=906568256
178	0.099971	1.1.173.69	1.2.174.15	TCP	53507 > http [SYN] Seq=0 Win=3960 Len=0 MSS=1460 TSV=907368125 TSER=0
179	0.000017	1.2.174.15	1.1.173.69	TCP	http > 53507 [SYN, ACK] Seq=0 Ack=1 Win=4344 Len=0 MSS=1460 TSV=907368247 TSER=907368125
180	0.000014	1.1.173.69	1.2.174.15	TCP	53507 > http [ACK] Seq=1 Ack=1 Win=4344 Len=0 TSV=907368370 TSER=907368247

▶ Frame 89 (70 bytes on wire, 70 bytes captured)

▶ Ethernet II, Src: MS-NLB-PhysServer-26_c5:02:00:00 (02:1a:c5:02:00:00), Dst: MS-NLB-PhysServer-26_c5:01:00:00 (02:1a:c5:01:00:00)

▶ Internet Protocol, Src: 1.2.88.33 (1.2.88.33), Dst: 1.1.86.231 (1.1.86.231)

▼ Transmission Control Protocol, Src Port: http (80), Dst Port: 26499 (26499), Seq: 0, Ack: 1, Len: 0

Source port: http (80)

Destination port: 26499 (26499)

0000 02 1a c5 01 00 00 02 1a c5 02 00 00 08 00 45 00E.
0010 00 38 78 74 40 00 20 06 31 41 01 02 58 21 01 01 .8xt@. 1A.X!..
0020 56 e7 00 50 67 83 7b 4b 73 eb fb 1f 62 90 90 12 V..Pg.{K s..b..
0030 10 f8 7b 35 00 00 02 04 05 b4 01 01 08 0a 34 82 ..{5....4..
0040 82 4a 34 82 81 bd .J4...

File: "/home/kkuehl/blogpostinteface... Packets: 433 Displayed: 123 Marked: 0 Profile: Default

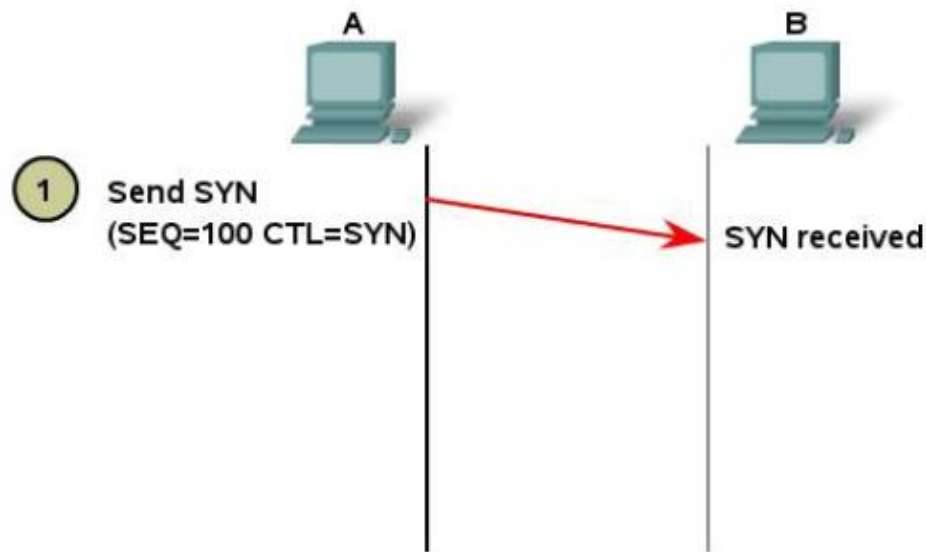
ВСТАНОВЛЕННЯ TCP З'ЄДНАННЯ

Процедура **Three-Way Handshake** виконує наступні дії:

1. Встановлює присутність і працездатність отримувача,
2. Перевіряє, чи на станції-отримувачі запущений процес, який обробляє запити до номера порта серверної програми, з якою встановлюється з'єднання
3. Інформує станцію призначення про наміри клієнта встановити з'єднання на цьому номері порта

ВСТАНОВЛЕННЯ TCP З'ЄДНАННЯ

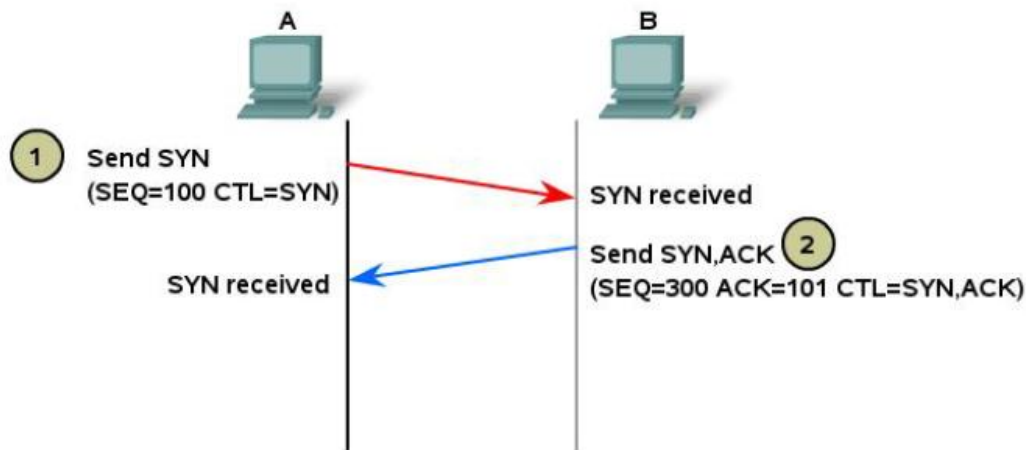
Клієнт відправляє сегмент, який містить деяке початкове значення (ISN – Initial Sequence Number) і встановлює спеціальний прапорець, який сигналізує про наміри встановити з'єднання



ctl = Which control bits in the TCP header are set to 1

A sends SYN request to B.

ВСТАНОВЛЕННЯ TCP З'ЄДНАННЯ



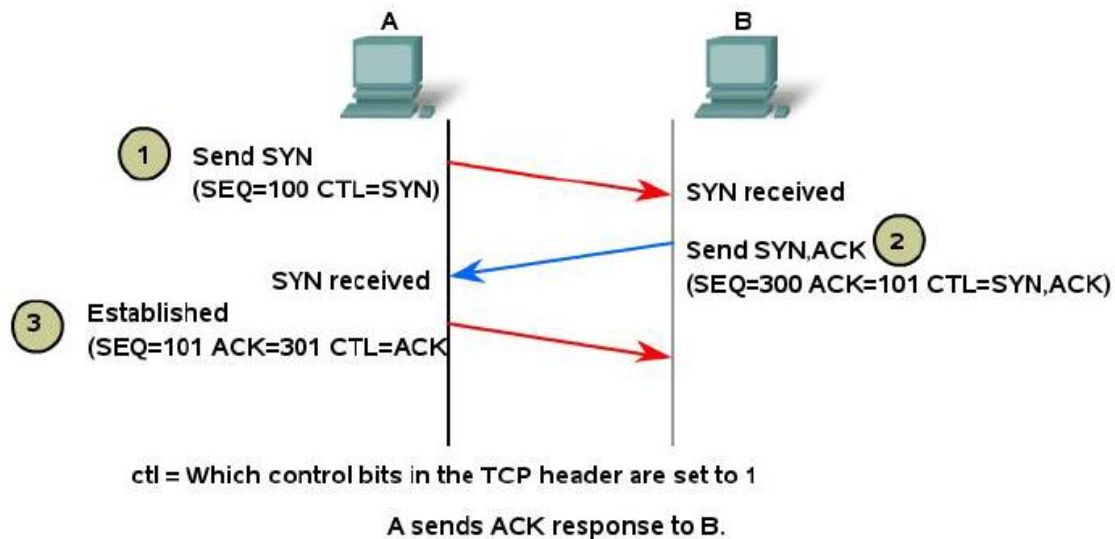
ctl = Which control bits in the TCP header are set to 1

B sends ACK response and SYN request to A.

Сервер надсилає у відповідь сегмент з підтвердженням ISN клієнта (ISN клієнта + 1) і власним ISN. Обмін цими значеннями дозволяє організувати контроль над передачею даних

ВСТАНОВЛЕННЯ TCP З'ЄДНАННЯ

Сегмент 3
підтвердженням клієнтом
ISN сервера (ISN сервера
+ 1) завершує процес
встановлення з'єднання

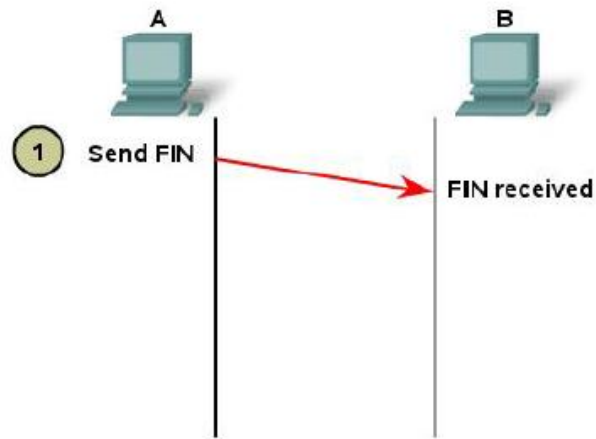


РОЗРИВ TCP З'ЄДНАННЯ

Процес розриву з'єднання проходить в чотири етапи (хоча насправді кількість етапів розриву з'єднання залежить від розробника конкретного стеку TCP/IP, стандарт передбачає саме чотири етапи)

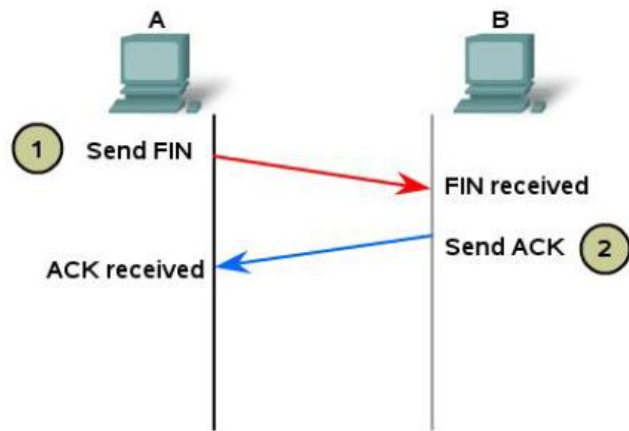
РОЗРИВ ТСП З'ЄДНАННЯ

Станція ініціює розрив з'єднання, відправляючи сегмент без корисних даних із встановленим бітом FIN



A sends FIN request to B.

РОЗРИВ ТСП З'ЄДНАННЯ

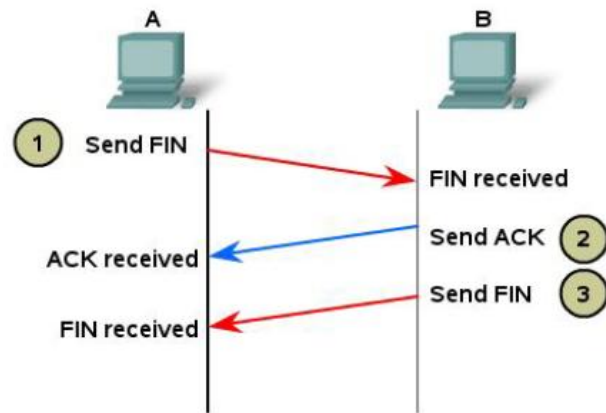


B sends ACK response to A.

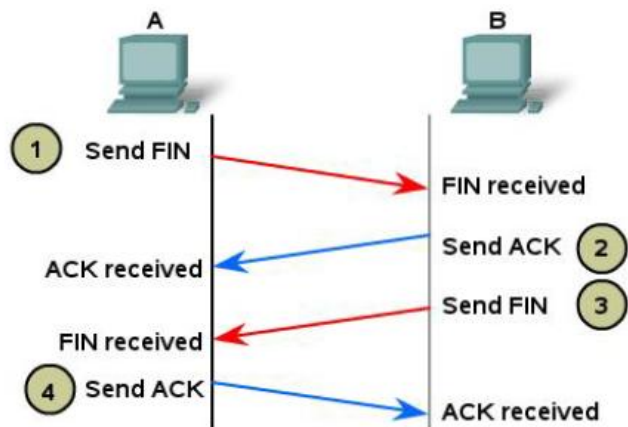
Партнер підтверджує
отримання цього
сегмента
прапорцем ACK

РОЗРИВ ТСП З'ЄДНАННЯ

Партнер відправляє власний сегмент з бітом FIN (в деяких реалізаціях стеку протоколів ТСП/IP цей крок може бути поєднаний із другим кроком)



РОЗРИВ ТСП З'ЄДНАННЯ

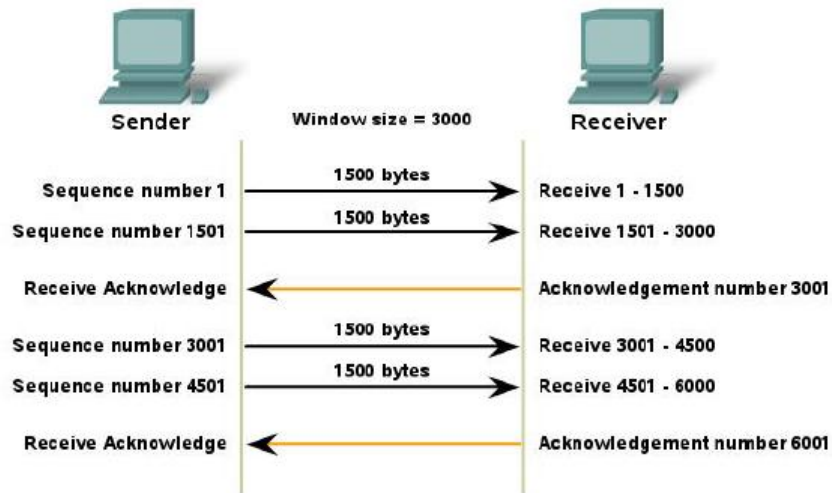


A sends ACK response to B.

Станція ініціатор розриву підтверджує отримання запиту на розрив з'єднання від партнера

КЕРУВАННЯ ВТРАТАМИ СЕГМЕНТІВ

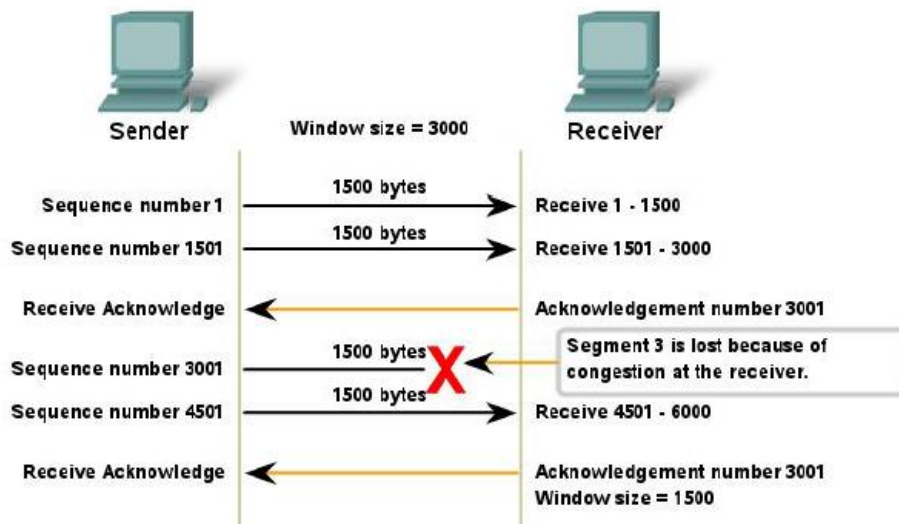
TCP Segment Acknowledgement and Window Size



The **window size** determines the number of bytes sent before an acknowledgment is expected.

The **acknowledgement** number is the number of the next expected byte.

TCP Congestion and Flow Control

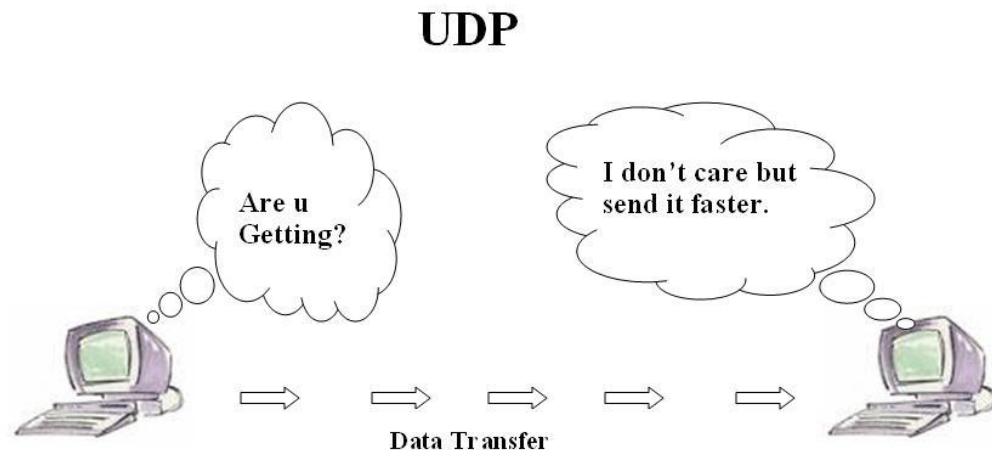


If segments are lost because of congestion, the Receiver will acknowledge the last received sequential segment and reply with a reduced window size.

ПРОТОКОЛ UDP

Протокол UDP використовують такі сервіси та протоколи вищого рівня:

- TFTP (англ. Trivial File Transfer Protocol, найпростіший протокол передачі файлів),
- SNMP (англ. Simple Network Management Protocol, простий протокол управління мережею),
- DHCP (англ. Dynamic Host Configuration Protocol, протокол динамічної конфігурації вузла),
- DNS (англ. Domain Name System, служба доменних імен).



ДЕ ПОЧИТАТИ ?

Джерела отримання додаткової інформації:

- http://en.wikipedia.org/wiki/Transmission_Control_Protocol
- <http://www.ietf.org/rfc/rfc793.txt>
- <http://citforum.ru/internet/tifamily/tcpspec.shtml>
- http://it.iut.ac.ir/sites/fsites/it/files/u4/uploads/Networking%20Class-Wireshar%20Labs-Solutions/05-Wireshark_TCP_Solution_July_22_2007.pdf
- <http://packetlife.net/blog/2010/jun/7/understanding-tcp-sequence-acknowledgment-numbers/>

ЗАПИТАННЯ ?

