ТЕМА № 1: УВОД В TCP/IP

Моделът OSI осигурява идентичност на използваните формати на данни и използваните протоколи на съответните нива на изпращащия и получаващия възел от мрежата.

Протокол – набор от правила, определящи взаимодействието между системите в рамките на едно ниво от модела. Стекът TCP/IP има 4 нива, които условно отговарят на нивата от модела OSI.

За именуването на единиците данни, които се предават по мрежата се използват различни наименования: поток, сегмент, дейтаграма, пакет, кадър.

HOST LAYERS	**DATA**	**APPLICATION**
	DATA	**PRESENTATION**
	DATA	**SESSION**
	SEGMENTS	**TRANSPORT**
MEDIA LAYERS	**PACKETS**	**NETWORK**
	FRAMES	**DATA LINK**
	BITS	**PHYSICAL**

Всеки слой на OSI модела изпълнява конкретна задача в процеса на мрежовата комуникация и след това предава данните нагоре или надолу към следващия слой. Тъй като данните се предават през слоевете, всеки слой добавя своя собствена информация под формата на хедъри, които биват добавяни пред оригиналните данни.

Тази слоеста структура на мрежата предполага използването на комутация на пакети (packet switching)

host layers – тези нива на модела OSI се реализират в крайните устройства (възлите) на мрежата (освен media layers). Тези слоеве от модела OSI (слой 4-7) осигуряват връзката от приложение до приложение (взаимодействието между процеси и услуги, работещи на тези крайни устройства) и затова са реализирани всичките в крайните устройства.

media layers – чрез тези слоеве на модела OSI (слой 1-3) си взаимодействат самите мрежови устройства.

Ниво 7 от модела осигурява интерфейс към приложенията, участващи в комуникацията между възлите в межата.

Ниво 6 – това ниво има за цел да представи данните в разбираем за приемащата ги страна вид. Тук например става конвертирането на различни кодови таблици, прилагат се методи за запис на цели или с плаваща запетайка числа. Целта е приложения на различни машини и

различни идеологии да могат да си комуникират безпрепятствено.

Ниво 5 – има за цел да осигури механизми данните за различни приложения да могат да бъдат разпознати от страна на приемащата страна. За целта са въведени нови адреси. Също така осигурява методи за установяване на нова сесия, прекратяването и, повторното ѝ отваряне ако е необходимо. Дефинирани са механизмите за Full Duplex и Half Duplex комуникация, както и механизми за Flow Control (те са дефинирани и в 4-то ниво, но по различен начин)

Ниво 4 – Транспортното ниво има една единствена дефинирана функционално цел. То е да осигури сигурна комуникация и прозрачен пренос за поток от данни. В OSI спецификацията са дефинирани 5 различни класа на транспорт (транспортен протокол) маркирани като TP0, TP1, TP2, TP3, TP4, като всеки по-горен включва функционалността на по-долния. Класовете на транспорт няма да бъдат разглеждани подробно.

Ниво 3 – това е първото напълно независимо от физическата среда ниво. Основното тук е, че разполагаме с адреси, които не са обвързани пряко с адресите от второто ниво, или можем да предаваме данни между различни сегменти с различни протоколи от второ ниво, и различни физически среди. На практика това е първото напълно независимо от физическата среда ниво, даващо възможност на две или повече машини да си препредават данни без значение към каква физическа мрежа са свързани. Предаването на данни на базата на тези адреси се нарича routing (маршрутизация), а форматирането на данните се нарича пакет.

Ниво 2 – Това ниво е обвързано с физическото ниво, но добавя допълнителна функционалност. Първо то добавя физически адреси на различните мрежови устройства, наречени MAC (Media Access Control) адреси, имащи за цел отделянето на комуникацията между отделните двойки от останалите, излъчвани по еднакъв начин в една и съща споделена физическа среда. Второ, то има грижата да открие възможни грешки при предаването на данни по физическата среда, които тя не е успяла да установи и поправи. Или казано функционално:

- Разделя комуникиращите възли чрез адреси (дава възможност на приемащите данните да разпознаят дали са били за тях на базата на адресни идентификатори)
- Проверява предадените на физическо ниво данни за коректност (и дава възможност на приемащия да предприеме мерки за коригирането им)

Тази среда е пряко обвързана с физическия пренос, защото адресите, възстановяването и проверката на данните може да го изискват. От друга страна не е толкова обвързана с физическата медия (оптика, радио, меден кабел) и кодировката, както и наличието на адреси позволяват препредаване на данни между различни физически среди използващи един и същи Data Link Layer протокол. Препредаването на

данните се нарича bridging (напоследък switching) а самите данни Frames (фреймове). Един домейн от множество устройства и физически мрежи позволяващи на участниците им да си комуникират пряко чрез MAC адресите се нарича сегмент.

Ниво 1 – това ниво се занимава с физическата среда и физическата комуникация. В него няма физическа маршрутизация и комуникацията е или между две страни директно свързани помежду си, или между група участници, получаващи едни и същи данни „едновременно” в даден момент.

Основните действия, извършвани на това ниво са:

- Осигуряване на свързаността и комуникацията с физическата среда;
- Осигуряване на преноса на данни между физическата среда, както и правилното им прочитане;
- Осигуряване на механизъм за разрешаване на конфликтите при употреба на физическата среда от много участници (например радио ефир, ако всички излъчват едновременно, как да се различи кой какви данни предава и те да бъдат възстановени без повреда).

Очевидно е, че средата за комуникация е хомогенна, защото всички участници в комуникацията споделят една и съща физическа среда. При липса на пряка физическа свързаност не може да бъде осъществена комуникация.

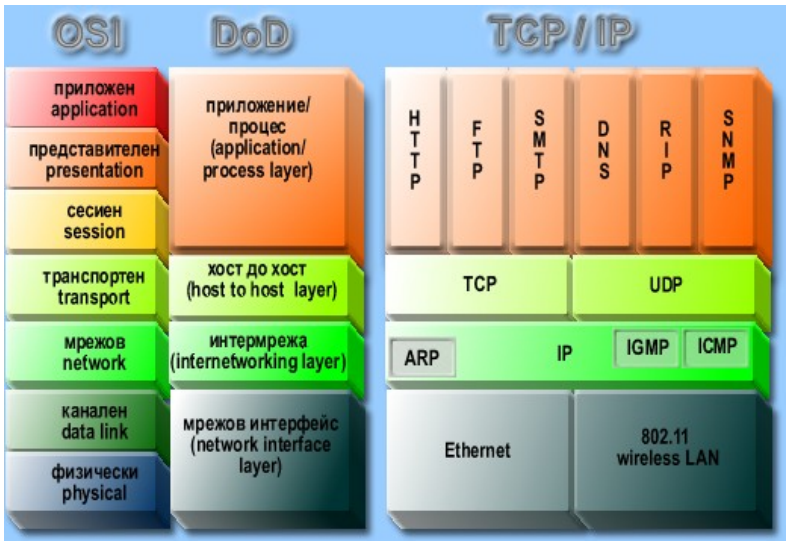
TCP/IP стек

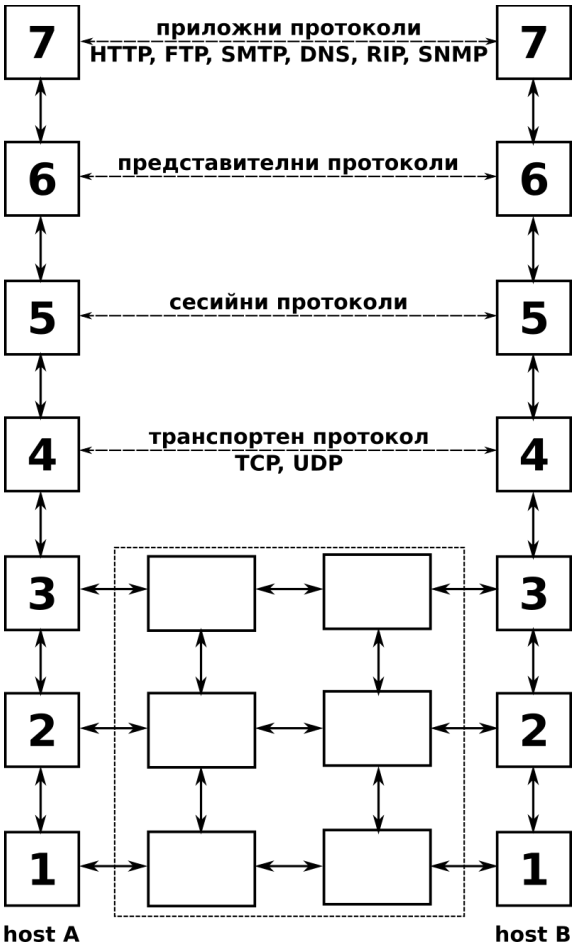
TCP/IP (Transmission Control Protocol/Internet Protocol) е протоколен стек, който е специално разработен за големи мрежи, състоящи се от много мрежови сегменти, свързани чрез рутери (routers). TCP/IP е крайъгълният камък на Интернет комуникациите. Той се превърна в най-използваното мрежово/транспортно решение за мрежи с всякакъв размер и конфигурации.

TCP/IP е не само протоколен стек, състоящ се от протокол от мрежовия слой и протокол от транспортния слой, но и пълен комплект от протоколи, работещи в много слоеве на мрежовия модел. Понятието комплект от протоколи (protocol suite) е по-широко от понятието протоколен стек и включва и елементи, които не се изискват за мрежовата комуникация (например, помощни програми от приложния слой, които са част от комплекта TCP/IP). Много от протоколите, включени в комплекта, функционират като инструменти за събиране на информация и отстраняване на проблеми.

Архитектура на TCP/IP

Архитектурата на комплекта протоколи TCP/IP отговаря на четирислойния мрежови модел DoD (известен още като модел DARPA), но всеки един от четирите му слоя може да бъде съпоставен на един и няколко от слоевете на референтния OSI модел. Това е илюстрирани на схемата по-долу.





фигура 2. Действие на TCP/IP стека

Мрежов интерфейс

Мрежовият интерфейс изпраща и получава TCP/IP пакетите от мрежовата преносна среда. TCP/IP е проектиран така, че да бъде независим от

методите за достъп до мрежата, формата на кадрите и преносната среда. Следователно, може да бъде използван с различни LAN технологии, като Ethernet и 802.11 wireless LAN и лесно може да бъде адаптиран към бъдещи нови технологии.

Address Resolution Protocol (ARP) - транслира логическите адреси в MAC адреси. Тази транслация е необходима, защото по-долните слоеве от модела могат да обработват само MAC адреси.

PDU – protocol data unit: общият блок от данни на протокола – заглавие + полезни данни;

SDU – service data unit: блок от данни, преминал от по-горно към по-долното ниво и все още не капсулиран в PDU на това ниво;

MTU – maximum transmission unit: максимален размер на блока от полезни данни, който може да бъде предаден без фрагментация.

Важна особеност на протокола IP е способността му да изпълнява динамична фрагментация на пакетите при тяхното предаване в мрежи с различни стойности на MTU. Това е основна характеристика на каналния слой. Фрагментите от пакетите, предадени по мрежата се събират от IP модула на възела, получател на пакетите, но понякога това се прави и от междинните маршрутизатори по техния път.

Какво ще се случи, ако при предаване на фрагментиран пакет, един от фрагментите не достигне до получателя на пакета и времето за дефрагментация на целия пакет изтече? IP модулът на получателя ще отхвърли всички останали получени фрагменти от този пакет. IP модулът на източника няма да предприеме действия за повторно предаване на този пакет. Оставя решаването на този проблем на по-горните нива (на транспортно при TCP-базирани приложения и на приложно при UDP-базирани).

Причини за фрагментацията:

1. за намаляване на времето за повторно предаване на пакета в случай на загубване или повреждане;
2. ако се работи в режим half duplex да не може възела да заема за дълго време канала;
3. колкото е по-голям пакета (респективно MTU), толкова по-дълго е изчакването на другите пакети да бъдат изпратени (особено при последователни интерфейси).

Задачка: Изпраща се IP пакет (дейтаграма), съдържащ UDP пакет с големина 8192 байта потребителски данни. Колко фрагмента ще се предадат и какви ще са стойностите на отместването и дължината на всеки фрагмент (MTU=1480)?

Решение: Добавяме 8 байта (UDP header) към размера на IP дейтаграмата и тя става 8200 байта.

1. 1480 @0+ (MF=1);

2. 1480 @1480+ (MF=2);
3. 1480 @2960+ (MF=3);
4. 1480 @4440+ (MF=4);
5. 1480 @5920+ (MF=5);
6. 800 @7400 (MF=6).

Проверка: 1480x5 + 800 = 8200

Команди ip, дефинирани от пакета **iproute2**:

1. *ip addr add 192.168.11.17 dev eth0 – добавя ip адрес към интерфейс eth0*
2. *ip addr show – показва адресите*
3. *ip addr del 192.168.11.17/24 dev eth0 – премахва ip адрес*
4. *ip link set eth0 up*

   ip link set eth0 down
5. *ip route show – показва маршрутизиращата таблица*
6. *ip route add 10.10.20.0/24 via 192.168.11.17 dev eth0 – добавя статичен маршрут*
7. *ip route del 10.10.20.0/24 – премахва статичен маршрут*
8. *ip route add default via 192.168.11.50 – добавя шлюз (gateway) по подразбиране*

команда ifconfig:

Тази команда е остаряла и е заменена от командата ip от пакета iproute2 (вж. по-горе). Препоръчва се избягването на наейната употреба.

1. *ifconfig eth0 – показва мрежовите настройки на определен интерфейс;*
2. *ifconfig -a – показва мрежовите настройки на всички интерфейси;*
3. *ifconfig – показва информация за всички **активни** интерфейси;*
4. *ifconfig eth0 up*
5. *ifconfig eth0 down*
6. *ifconfig eth0 192.168.2.2 – присъединява мрежовия адрес 192.168.2.2 на интерфейс eth0;*
7. *ifconfig etho 255.255.255.0 – присъединява мрежова 255.255.255.0 на интерфейса eth0;*
8. *ifconfig eth0 broadcast 192.168.2.255 – променя адреса за broadcast на интерфейса eth0;*
9. *ifconfig eth0 mtu #### – определя размера на максималния размер на пакета в байтове (####), който може да бъде предаден без фрагментация. По подразбиране mtu=1500 байта;*

10. *ifconfig eth0:0 172.16.25.127 – дава на интерфейса eth0 alias eth0:0 и присъединява към него мрежовия адрес 172.16.25.127;*
11. *ifconfig eth0 hw aa:bb:cc:dd:ee – сменя MAC адреса на интерфейс eth0.*

Повече примери за използването на командата ifconfig може да се намерят в нейната помощна страница (man ifconfig).

Deprecated command	Replacement command(s)
arp	**ip n** (**ip neighbor)**
ifconfig	**ip a** (**ip addr**), **ip link**, **ip -s** (**ip -stats)**
iptunnel	**ip tunnel**
iwconfig	**iw**
nameif	**ip link**, **ifrename**
netstat	**ss**, **ip route** (for **netstat-r**), **ip -s link** (for **netstat -i**), **ip maddr** (for **netstat-g**)
route	**ip r** (**ip route**)

Broadcast address – условен, неприсвоен на никое устройство в мрежата адрес, който се използва за изпращане на broadcast пакети (пакети, предназначени за получаване от всички възли в мрежата) в компютърната мрежа.

Първото появяване на broadcast адреси в IP мрежи е през 1982 г., Robert Gurwitz & Robert Hinden.

Видове broadcast адреси в зависимост от слоя на модела OSI:

L2 – ff:ff:ff:ff:ff:ff. Използва се за предване на служебна детайли (например при запитвания по протокола arp).

L3 – зависят от използвания протокол в мрежовия слой.

Инверсия на мрежовата маска – всички 0 в нея се установяват на 1 (нарича се wildcard маска и се прилага в рутерите на Cisco Systems при конфигуриране на протокола OSPF)

команда **ping** – средство за диагностика на мрежата. Командата измерва общото време (RTT) в милисекунди за изпращане на пакет до целта и получаване на отговор от нея по мрежата. За тази цел се използва протокола ICMP – изпращане на echo request пакет и получавае на echo reply packet. Протоколът ICMP, заедно с пртокола IP осигуряват възможността за проверка за грешки, както и функционалността за тяхното докладване.

RTT – Round Trip Time – включва времето за разпространение, очакване и

обработка на заявката.

```
[nick@lascar ~]$ ping google.bg
PING google.bg (216.58.208.99) 56(84) bytes of data.
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=1 ttl=57 time=1.06 ms
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=2 ttl=57 time=1.07 ms
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=3 ttl=57 time=1.07 ms
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=4 ttl=57 time=1.11 ms
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=5 ttl=57 time=1.05 ms
64 bytes from sof01s11-in-f3.1e100.net (216.58.208.99): icmp_seq=6 ttl=57 time=0.916 ms

[nick@lascar ~]$ ping 62.44.96.142
PING 62.44.96.142 (62.44.96.142) 56(84) bytes of data.
64 bytes from 62.44.96.142: icmp_seq=1 ttl=61 time=0.753 ms
64 bytes from 62.44.96.142: icmp_seq=2 ttl=61 time=0.801 ms
64 bytes from 62.44.96.142: icmp_seq=3 ttl=61 time=0.689 ms
64 bytes from 62.44.96.142: icmp_seq=4 ttl=61 time=0.641 ms
64 bytes from 62.44.96.142: icmp_seq=5 ttl=61 time=0.647 ms
64 bytes from 62.44.96.142: icmp_seq=6 ttl=61 time=0.585 ms
^C
--- 62.44.96.142 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5000ms
rtt min/avg/max/mdev = 0.585/0.686/0.801/0.072 ms
```

Когато ping-вате дестинация по име на хост, отговорите съдържат IP адреса на хоста, броят на изпратените байтове, RTT и Time to Live (TTL) на пакета. Когато ping-вате дестинация по IP address, получавате същите отговори с изключение на името на хоста (освен ако не сте въвели параметъра **-a**). Ping тества не само достижимостта, но и верифицира дали TCP/IP стека е инсталиран правилно и дали DNS резолвинга работи правилно.

- ping localhost – проверява състоянието на TCP/IP стека на локалната машина
- ping 127.0.0.1 – същото
- ping local_IP_address – проверява състоянието на мрежовата карта
- ping gateway_IP_address – проверява състоянието на връзката до шлюза на локалната мрежа

команда ss – замества командата netstat. ss – socket statistics.

```
[nick@sakurajima ~]$ ss -t
State       Recv-Q Send-Q                                     Local Address:Port
Peer Address:Port
ESTAB       0      0                                        192.168.11.102:36568
208.68.163.218:xmpp-client
CLOSE-WAIT  1      0                                        192.168.11.102:42815
152.19.134.142:https
ESTAB       0      0                                        192.168.11.102:49139
88.221.211.11:http
```

```
CLOSE-WAIT  1      0                                          192.168.11.102:57670
157.249.32.164:http
ESTAB       0      0                                          192.168.11.102:48127
84.43.191.101:xmpp-client
```

Програма **wireshark**

3 прозореца:

- списък на събраните пакети от мрежата с кратко описание;
- дърво на протоколите, инкапсулирани в кадъра;
- дъмп на пакета в шестайсетичен или текстов вид.

Tshark – конзолна версия на wireshark.

tshark -R "ip.addr == 192.168.0.1" -r /tmp/capture.cap

tshark -f "udp port 1812" -i eth0 -w /tmp/capture.cap

- Флагът -f се използва за дефиниране на филтъра. Пакетите, които не удовлетворяват условието, дефинирано с -f флага, няма да бъдат прихванати. В горния пример се прихващат само IP пакетите, които са с UDP порт 1812 (източник или дестинация).
- Флагът -i се използва за дефиниране на интерфейса, от който се очаква да видим RADIUS пакети. На мястото на 'eth0' се поставя конкретния интерфейс.
- Флагът -w flag се използва за дефиниране на файла, където ще се запише прихванатия трафик.

tshark -z "proto,colinfo,tcp.srcport,tcp.srcport" -r /tmp/capture.cap

ping abv.bg – първият пакет ползва DNS. С wireshark наблюдаваме енкапсулирането, като сме направили DNS филтриране.

ping abv.bg -s 65000 -M – със забрана на фрагментацията на пакетите

ping abv.bg -s 65000 – с wireshark се наблюдава фрагментацията на пакетите