**Вариант 1:**

1. *Протоколен стек за RIP, OSPF и BGP*

RIP, OSPF, BGP са протоколи за динамична маршрутизация.

RIP - ползва се в сравнително малки и рядко менящи си топологията мрежи. Има бавна скорост на сходимост, метриката която ползва е хопове. (би предпочел пътят по-близък по брой междинни рутери, не по скорост на връзката).

OSPF - Динамичен протокол за маршрутизация, ползван в рамките на АС (автономна система). Има бърза скорост на сходимост и знае цялостната топология на мрежата. Метриката му е комплексна, един от критериите е b\*/andwidth (скоростта на връзката). Използва Shortest path tree и алгоритъм на Дийкстра за да намери оптималният маршрут.

BGP - основният протокол за маршрутизация в Internet. Поддържа таблица от IP мрежи (префикси), които определят достижимостта на мрежите между автономните системи. BGP е протокол с вектор на пътищата, path vector protocol.

\*АС - област на единно администриране. Например СУ или по-големите Интернет доставчици.  
+\*Скорост на сходимост/конвергенция -> времето за което промяна в топологията бива разпространено до всички участници в схемата.

1. *Йерархична маршрутизация*

* С увеличаването на размерите на мрежата нараства обемът на маршрутните таблици, което изисква повече памет и процесорно време за тяхната обработка.
* Това налага въвеждането на йерархично маршрутизиране, при което мрежата се разделя на области. Маршрутизаторите в една област знаят всичко за вътрешната структура на своята област, но не знаят вътрешната структура на останалите области. За по-големи мрежи може да е необходима йерархия с повече от две нива.
* ~~Обявяването на маршрути чрез OSPF е реализирано в йерархия. Има една главна област и множество подобласти. Главната област се обозначава с номер 0, подобластите с номер по-голям от 0.~~

1. *Мултиплексиране и демултиплексиране (при TCP)*

Обикновено на всеки TCP/IP хост има много процеси, като всеки от тях иска да изпраща и получава информация по мрежата. Всичката тази информация обаче трябва да бъде изпратена през същия интерфейс. Мултиплексирането е процесът на комбиниране на два или повече потока от информация, така че те да преминат през една-единствена физическа връзка. TCP предоставя възможности за мултиплексиране, като използва порт номера на източника и дестинацията. Тези порт номера позволяват на TCP да създаде няколко виртуални връзки над една физическа и да мултиплексира информационния поток през тази връзка. Някои от добре познатите порт номера са: 25 за SMTP, 23 за Telnet, 21 за FTP, 80 за HTTP, 443 за HTTPS и други. Обратния процес – приемането на много несвързани пакети и предаването им до правилните процеси – се нарича демултиплексиране.

1. *Контрол на потока и контрол на претовареността при TCP*

* Колкото и добре да е проектирана дадена мрежа, не може без загуби на данни. TCP осигурява методи за управление на тези загуби на сегменти. Един от тях е механизмът за повторно предаване на сегменти с непотвърдени данни. TCP услугата в хоста-приемник потвърждава само данни, състоящи се от непрекъсната последователност от данни. Ако липсват един или повече сегменти, потвърждават се само данните в сегментите, които попълват плътно потока. Това е познатията ви Selective Repeat. Тук Selective Acknowledge (SACK).
* TCP има и механизми за управление на потока (flow control). Flow control синхронизира скоростите на потока от данни между двете услуги в сесията. Когато източникът е информиран, че оределено количество данни в сегментите е получено, тогава може да продължи с изпращане на повече данни за дадената сесия.
* Полето Window Size определя количеството данни, което може да бъде предадено, преди да бъде получено потвърждение. Първоначалният размер на прозореца се определя в началото на сесията чрез threeway handshake.
* Механизмът за обратна връзка в TCP нагласява ефективната скорост на предаване на данните към максималния поток, който мрежата и устройството-получател могат да поддържат без загуби и без повторни предавания.

**Вариант 2:**

1. *Изисквания към маршрутните протоколи*

* Маршрутизиращите протоколи трябва да отговарят на множество изисквания. Да са достатъчно прости и лесни за конфигуриране и да осигуряват надеждна и стабилна работа на мрежата. Да реагират своевременно на отпадане на маршрутизатори или връзки между тях. Да бъдат в състояние да открият алтернативни пътища за доставяне на пакетите, ако такива съществуват.

1. *Каква информация съдържа маршрутната таблица на BGP*

* Поддържа таблица от IP мрежи (префикси), които определят   достижимостта на мрежите между автономните системи.
* Съдържа информация коя мрежа зад кой next-hop се намира и списък с автономни системи през които трябва да премине, също така има поле за метрика, localpreference и weight. (IP адрес на мрежа, next hop, metric, loc pref, path)

1. *С какви механизми се осигурява достоверността на данните при TCP*

* Acknowledgements    При размяната на един или повече пакети, получателя връща acknowledgement (наречено "ACK") към изпращача, показвайки, че е получил пакетите. Това се осъществява чрез полето Acknowledgement number в сегмента, който има стойност равна на номера на 1-вия бит от данните в следващия сегмент, който се очаква. Флагът в сегмента ACK валидира стойността на полето Acknowledgement number. Ако пакетите не са ACK-нати, изпращача може да преизпрати пакетите (или да спре връзката ако си мисли чр получателя е крашнал).
* Може да се използва и полето Checksum в сегмента, който се изчислява върху: TCP header-a, TCP data pseudoheader-a (src IP, dest. IP, полето length в IP header-a)
* ~~Flow control    Ако изпращача изпраща пакети прекалено бързо, получателя изпуска пакети. Тогава се изпраща съобщение за забавяне на скоростта на изпращане.~~
* ~~Packet recovery services   Получателя може да поиска преизпращане на пакетите.~~

1. *Алгоритъм на плаващия прозорец при TCP*

Плаващият прозорец при TCP се използва за управление на потока от данни (Flow Control) т.е. за оптимизация на трафика. Неговата големина се определя от полето Window Size в TCP сегмента и оказва какво количество данни могат да бъдат приети (изпратени) преди да се получи потвърждение (ACK) и без да се препълнят буферите след последния потвърден номер на байт. При получаването на данни прозореца се свива т.е. Windows size намалява и при 0 изпращача трябва да прекрати своята дейност. След обработка на данните получателя увеличава прозореца, което е знак, че е готов до приема нови данни.

**Вариант 3:**

1. *Външна и вътрешна маршрутизация*

* външна маршрутизация – Използва е от IGP. От таях най-популярен е протоколът iBGP. Ползва се за комуникация между АС (автономни системи)  
  За вътрешна маршрутизация най-популярен е протоколът OSPF. Ползва се в рамките на АС. Протоколът знае за цялостната топология на мрежата, благодарение на което може да намери оптимален маршрут, спрямо различни критерии, например широчина на лентата a.k.a bandwidth a.k.a скорост на връзката a.k.a пропускливост на комуникационния канал.

1. *Какви полета съдържа маршрутната таблица на OSPF*

* Мрежа, административна дистанция(110)+метрика(20), next-hop, интерфейс, таймер от кога знаем за мрежата от съответния интерфейс  
  Ред от ospf МТ:  
  O>\* [172.16.1.0/30](http://172.16.1.0/30) [110/20] via [10.10.10.2](http://10.10.10.2/), eth0, 00:14:14  
  \*Административна дистанция - всеки маршрутизиращ протокол има по подразбиране стойност на административна дистанция. Така ако сме научили за съществуването на дадена мрежа от протокола RIP и в същото време по протокола OSPF, ще се вземе пътя от протокола с по-добра административна дистанция (OSPF = 110, RIP = 120). Колкото по-малка е административната дистанция, толкова по-значима е. Директно вързаните мрежи към нас са с административна дистанция 0, статично конфигурираните са с АД 1.

1. *End-to-end предаване и end-to-end контрол при TCP - ???*

* TCP управлява комуникациите между приложения, работещи на компютри в Интернет и e от типа “от край до край ” (end to end).

1. *Таймери при TCP*  – Той поддържа 4 различни таймера за всяка връзка:

* Таймер за препредаване (retransmission timer)- задава интервала време, в който трябва да се получи потвърждение от другата страна.
* Таймер за прозореца (persist timer)- съдържа информация за размера на прозореца на другата страна, дори и в случай, че другата страна затвори своя прозорец.
* Таймер за съществуване на връзката (keepalive timer)- чрез този таймер се открива дали другият край на връзк ата, по която има обмен на данни, е отказал или рестартирал.
* Таймер 2MSL- измерва времето, през което връзката е била в състояние TIME\_WAIT.

*Таймери при RIP*

* На всеки 30 sec изпраща копие от маршрутизиращата таблица към съседните маршрутизатори;
* hold down timer – 180 sec. Това е таймерът за невалиден маршрут;
* flush timer – съобщение за изтриване на маршрут – пътя се изтрива окончателно от маршрутната таблица.

**Вариант 4:**

1. *Свойства на маршрутните алгоритми*

* Маршрутен алгоритъм е част от софтуера на мрежовото ниво, която определя по коя от изходните линии да се изпрати пристигнал пакет.
* Маршрутните алгоритми биват два вида неадаптивни и адаптивни:
  + неадаптивни - статични, администраторът ръчно попълва МТ
  + адаптивни - МТ се попълва от динамичен протокол за маршрутизация). Примерно свойство за адаптивните алгоритми е да определят оптимален маршрут до дадена мрежа.

1. *Как се оценява маршрутът при RIP, OSPF и BGP*

* RIP - спрямо брой междинни възли (hops)  
  OSPF - спрямо bandwidth-a от точка до точка.  
  BGP - избира пътя с най-висок weight и localpref и най-малко на брой автономни системи през които трябва да премине.

1. *Инструменти за откриване на грешки при TCP*

* Поле Checksum (гарантира точността на сегмента). Изчислява се върху: - TCP header, TCP data и - псевдозаглавие (pseudoheader) – source IP, destination IP и поле дължина (length) в IP заглавието. (гарантира че няма промяна в IP адресите) checksum е задължително в TCP и в IPv4, и в IPv6.

1. *TCP съденинение*

* Когато два хоста комуникират по TCP, първо трябва да се оформи съединение, преди да започне обмена на данни. След приключване на комуникацията сесиите се затварят - терминират. Всичките тези процедури реализират функциите по надеждност в TCP. За установяване на връзка хостовете изпълняват трипосочна процедура – “ръкостискане” (three-way handshake). Контролните битове в TCP заглавието регистрират прогреса и състоянието на връзката.
* Three-way handshake прави:
  + Установява, че отсрещното устройство съществува в мрежата;
  + Уверява се, че отсрещното устройство има активна услуга и приема заявки на отдалечения порт, който инициатора смята да използва за сесията
  + Информира отсрещното устройство, че клиент източник има намерение да установи сесия с този номер на порт
* При TCP съединенията хостът-клиент инициира сесията към сървъра. За да си изясним как работи “three-way handshake”, нека видим какви стойности на параметри си разменят двете старани. Трите стъпки в установяване на TCP съединение са:
  + Клиентът-инициатор изпраща сегмент SYN, съдържащ начална стойност на последователността SEQ, и представляващ заявка за начало на сесия.
  + В отговор сървърът изпраща сегмент, съдържащ стойност за потвърждение ACK (SEQ + 1). Освен това - собствената си синххронизираща стойност на последователност SYN SEQ, която е по- голяма от получения и потвърден номер ACK – следващия очакван байт.
  + Клиентът-инициатор отговаря със стойност ACK, равна на (получена SEQ + 1). С това съединението е установено.
* По 4-стъпкова процедура се разменят флагове за терминиране на TCP съединение.