Observera att svaren redovisas kortfattat och hänvisas till kursboken för utförligare svar.

Vt2011

1. Se avsnitten 1.5.1-1.5.2 i kursboken (4p)

1.1 nätverkslagret, IP (Internet Protocol), datagram (paket)

1.2 länklagret, Ethernet MAC (Medium Access Control), frame (ram)

1.3 PDU är data till protokollet uppifrån plus det header som protokollet lägger till innan de tillsammans skickas nere i protokollstacken. Header är ett antal fält för bl.a. kontroll-data och adressering-info. Data uppifrån (payload) är det PDU från protokollet på ovanliggande lagret.

$\mathbf{PDU}_{N-1} = \mathbf{Header}_{N-1} + \mathbf{PDU}_{N}$ (payload)

1.4 Peer-to-peer protokollkommunikation är den **logiska** kommunikationen mellan protokoll hos olika värddatorer på samma nivå av OSI-lager. Protokollen använder och tolkar varandras kontrollfält i PDU-header. Detta innebär att protokoll vid ett lager **N** på sändarsidan kommunicerar med ”peer” lika protokoll vid samma lager **N** på mottagarsidan genom den kontrollinformation som finns i de olika fälten i header.

Ex.: Sändar-TCP skickar header + segmentets data. Mottagar-TCP tolkar alla header-fält.
Ex.: Http-klient och –server utbyter meddelande med varandra som varje inleds med header-lines som protokollet på varje sida använder för peer-to-peer protokollkommunikation

2. Se avsnitt 2.1.4, 3.2 och 3.3.2 i kursboken (4p)

2.1 Den ena typen är en tillförlitlig, förbindelse-orienterad dataöverföring i form av sekvensnumrerad byte-ström med många kontroll-mekanismer för bl.a. iordningsleverans, flöde och stockning. Denna transporttjänst används för applikationer som inte tolerar paketförlust eller datafel.

Den andra typen är en förbindelselös överföring av applikationsdata i form av datagram utan någon nämnvärd kontroll. Denna transporttjänst används för applikationer som kräver snabbhet eller är tidskänsliga.

2.2 TCP resp. UDP

2.3 HTTP kräver felfri hämtning av object och därför använder TCP. Medan DHCP (broadcast) och realtids applikationer använder UDP.

2.4 Internets transportprotokollet använder portnumer och socket för multiplexering och demultiplexering.

3. Se avsnitt 2.2.2 i kursboken (4p)

3.1 Persistent HTTP: Webklienten ges möjlighet att hämta webbsidan och de tillhörande objekten med en enda TCP-anslutning ”TCP connection”, då servern behåller anslutningen öppen (Connection: Keep-Alive) efter första svar och klienten kan därmed fortsätta begära olika refererade objekt löpande förutsatt att alla objekt ligger på samma server. På detta sätt minskar man den totala tiden och de allokerade resurserna när klienten slipper skapa en ny TCP-anslutning för varje objekt.

3.2 Persistent HTTP utan pipelining: Klienten sätter upp en TCP-anslutning till servern, begär webbsidan och sedan kan den begära de 3 objekt en efter en i serie (ett objekt i taget). Klienten behöver vänta på svar på varje begärd objekt innan den frågar efter nästa. Därmed skickar servern objekten också en efter en på klientens begäran.

3.3 Persistent HTTP med pipelining: Klienten sätter upp en TCP-anslutning till servern, begär webbsidan och sedan kan den begära de 3 objekt i följd (parallellt) utan att behöva vänta på svar på varje objekt. Servern kan därmed skicka de begärda objekten också parallellt.

3.4 Fördelen är att TCP hos klienten kan begära flera objekt med ett enda segment (paket) vilket gör nedladdningen snabbare.

4. Se Wireshark DNS-labb (4p)

4.1 Första paketet är adresserat till 129.16.1.53 som är den lokala DNS-servern som användarens dator anlitar för DNS-förfrågor. Frågan (-type = A) var om vilken IP-adress har en värddator som har hostnamnet **e3dns.ericy.com**. Användarens dator har fått svar att adressen är 198.24.6.2. (Sedan verifier datorn med PTR att adressen verkligen tillhör den efterfrågade värddatorn)

4.2 Sista paketet skickades från **e3dns.ericy.com** (198.24.6.2) som är en av de officiella (authoritative) DNS-servrarna för domännamnet **ericsson.com**. DNS-svaret innehåller sektioner för en förfråga (-type = MX), tre svar om namnen på de emailservarna, de officiella namnservrarna och slutligen extra information om dessa servrars IP-adresser.

4.3 Användaren frågade **e3dns.ericy.com** om vilka emailservrar (-type = MX) som tillhör domännamnet **ericsson.com**. Kommandot var:

>nslookup -type=MX ericsson.com e3dns.ericy.com

5. Se avsnitt 3.7 i kursboken (4p)

5.1 Det är timeout för att vänta på ACK som ”sannerligen” indikerar TCP-sändaren om paketförlust pga en allvarlig stockning på vägen till TCP-mottagaren.

5.2 Händer det timeout när sändar-TCP väntar på ett ACK återgår TCP i alla fall till ”Slow Start”, men om ett trippel duplikat ACK tas emot för ett och samma segment vilket indikerar en kortvarig stockning. Detta innebär att ett tidigare segmentet bland de sända segmenten är förlorat. TCP kan övergå till ”Congestion Avoidance” i den **Reno**-implementationen av stockningskontroll i stället för att börja på nytt med ”Slow Start” som i den tidigare versionen **Tahoe**. I båda fallen sätts tröskeln till halva storleken på ”Congestion Window” före den upptäckta händelsen.

5.3

”Congestion Window” är antalet segment som TCP kan sända parallellt utan att behöva vänta på ACK.

Slow Start är det då sändar-TCP börjar försiktigt genom att sända bara ett segment först och sedan ökar sändingshastighet genom att fördubbla antalet segment efter varje RTT dvs. exponentiell ökning (om det får ACK på alla tidigare sända segmenten) tills det når en tröskel.

Congestion Avoidance är det då sändar-TCP har nått tröskeln och börjar öka ”Congestion Window” med ett enda segment i taget efter varje RTT dvs. linjär ökning (om det får ACK på alla tidigare sända segmenten) för att undvika stockningen. Tröskeln räknas som hälften av det tidigare ”Congestion Window” innan stockningen har skett. Vid början av sändningen sätts tröskelen till ett default värde.

Se problem P32 fig. 3.57 (kap 3) i kursboken

Vid omgångar ”rounds” (1-6) & (23-26) befinner sig TCP i s.k. ”slow start”.

Vid omgångar ”rounds” (7-16) & (17-22) befinner sig TCP i s.k. ”Congestion Avoidance”.

Vid round 16 får TCP indikation på paketförlust i form av trippel duplikat ACK, då sätts tröskeln till 21 segment och TCP startar om ”Congestion Avoidance” vid detta tröskelvärde och sända om segmentet vid round 17 (fast recovery) och sedan ökar ”congestion window” med ett segment per omgång.

Vid round 22 får TCP indikation på paketförlust men denna gång i form av timeout, då sätts tröskeln till 13 segment och TCP återgår till ”slow start” att sända om segmentet vid round 23 och sedan fortsätter på samma vis som i början tills det når den nya tröskeln på 13 segment vid round 27.

Nya värdet på tröskeln anger storleken i antal segment på halva “Congestion Window” vid senaste händelse. Det är 32 i början (default) och sedan sätts det till 21 efter trippel duplikat ACK och sedan sätts det till 13 segment efter timeout.

6. Se avsnitt 6.3.2 i kursboken

En positiv bekräftelse ”ACK” används av länkprotokollet MAC 802.11 för trådlösa LAN för att informera sändaren om lyckad överföring över radiolänken. Utebliven ACK är en indikation för sändaren om att försöka sända om samma ram.

Detta är nödvändig dels med anledning av att radiolänken är mer utsatt för störningar, brus och interferens så att de sända ramarna kan lätt drabbas av bitfel, till skillnad från Ethernet-kablar. CRC-kontrollen hos mottagaren hjälper att avgöra om ramen mottagits felfri eller inte. Om inget bitfel inträffats då bekräftar mottagaren detta med ACK till sändaren. I annat fall får sändaren inte ACK inom rimlig väntetid och därmed skall denna sändare försöka sända om samma ram eftersom bitfel över radiolänken kan förekomma oftare än på kablar and det är effektivare att göra omsändning här på denna länk än att låta t.ex. TCP göra det på end-to-end basis.

En annan anledning är att MAC 802.11 är baserat på CSMA/Collision Avoidance och möjligheten att upptäcka kollisioner över radiolänken är mycket svårt pga s.k. ”Hidden Terminal” problemet och svårigheten att hårdvaran skulle kunna avlyssna kanalen samtidigt som den sänder egen ram. Därför behöver sändaren indikation om det har hänt kollision eller inte. Det är kostsamt med kollisioner på radiolänken eftersom trådlös sänadare skickar hela ramen oavsett om det händer kollision eller inte. Däremot tillämpar Ethernet Collision Detection och sändaren kan avgöra om det har hänt kollision och avbryta sändningen.

7. Se avsnitt 6.3.3 i kursboken (2p)

7.1 ARP-förfrågan är en broadcast på den fysiska länken.

Adress 1	Adress 2	Adress 3
AP-BSSID	Trådlösa STA	Broadcast
00-26-5A-30-34-92	00-1B-77-D3-20-B9	FF-FF- FF-FF- FF-FF

7.2 ARP-svaret är en unicast från standard-gateway till den trådlösa stationen.

Adress 1	Adress 2	Adress 3
Trådlösa STA	AP-BSSID	Standard-gateway
00-1B-77-D3-20-B9	**00-26-5A-30-34-92**	00-04-23-08-5B-1C

8. Se avsnitt 4.5.1 i kursboken (4p)

Varje nod informerar alla andra noder om identitet, status och kostnad av alla aktiva länkar som noden är direkt ansluten.

8.1 Linkstatus algoritm

Steg	N'	D(s),p(s)	D(t),p(t)	D(u),p(u)	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	x	∞	∞	∞	3,x	4,x	∞	**2,x**
1	xz	∞	7,z	∞	**3,x**	4,x	5,z	
2	xzv	∞	7,z	7,v		**4,x**	4,v	
3	xzvw	∞	7,z	7,v			**4,v**	
4	xzvwy	∞	**6,y**	7,v				
5	xzvwyt	9,t		**7,v**				
6	xzvwytu	**9,t**						
7	xzvwytus							

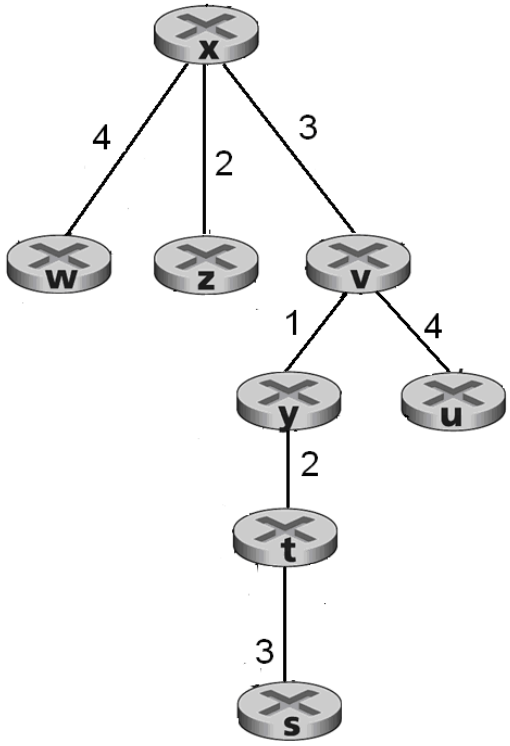
8.2

Routingtabell för nod x:

Destination	Next hop	Cost
s	v	9
t	v	6
u	v	7
v	v	3
w	w	4
y	v	4
z	z	2

8.3

Graf: nod x är root



9. Se Lab-2 (4p)

9.1

”Standard-gateway” är det närmaste router-interface som har direkt länk med värddatorn på samma subnät och som anlitas av denna dator för vidarebefordran av paket som skall levereras utanför eget subnät.

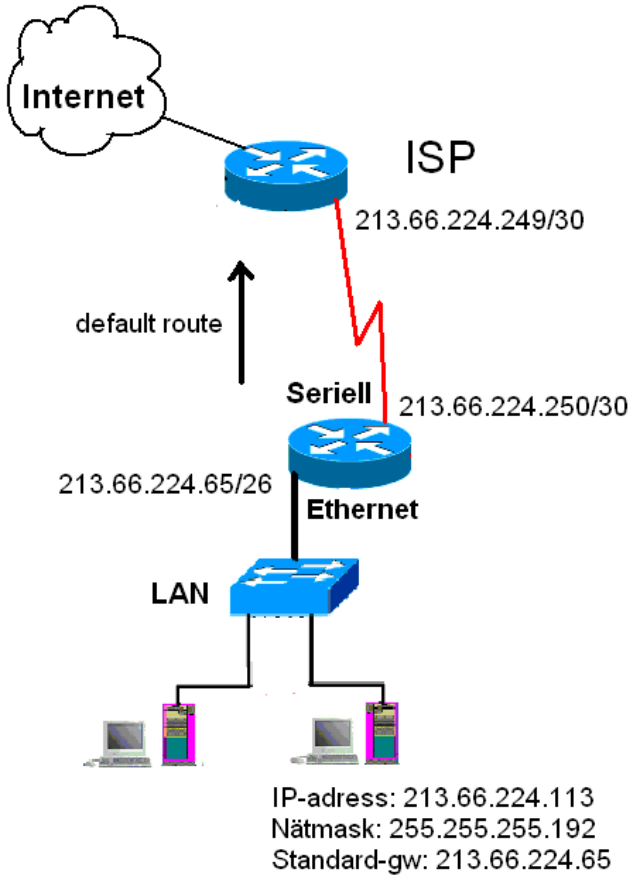
9.2

Konfigurationsparametern anger IP-adressen för denna routers Ethernet-interface som ligger på samma subnät (och det fysiska nätverket som i detta fall är Ethernet LAN).

9.3

Routingtabellen består av två självklara vägar, (dvs. routes); dels en route till det lokala IP-subnätet som är direktanslutet till routerns Ethernet-interface mot LAN och dels en route till point-to-point WAN-subnätet som är direktanslutet till routerns seriella interface mot ISP. Slutligen krävs det en manuellt konfigurerad default route (all annan destination) via det seriella interfacet som routern har uppkopplingen mot Internet via ISP.

Destination	Mask	Nästa hopp	Via interface
0.0.0.0	**0.0.0.0**	213.66.224.**249**	Seriell
213.66.224.**64**	255.255.255.**192**	direkanslutet	Ethernet
213.66.224.**248**	255.255.255.**252**	direkanslutet	Seriell



10. Se Lab-2 och extra materialet om IP-adressering (4p)

10.1

Eftersom antalet IP adresser för företagets nätverk skall vara minst:

2*125 + 3*60 + 1*30 = 460 är det närmaste binära storleken på ett subnät 512.

$512 = 2^9$ som innebär att prefixet skall ha 9 bitar för företagets hostadresser.

Nätverksdelen blir 32 – 9 = 23 bitar och således blir prefixet **201.94.112.0/23** och mask 255.255.254.0 (här väljes subnet-zero av adressblocket men man kan välja 201.94.114.0 eller 201.94.116.0 eller 201.94.118.0).

10.2

Tredje byte (från vänster) i adressen är 112 decimalt som är **0111000**0 binärt där de första sju bitarna tillhör prefixen. Genom att använda sista bit i denna byte och första bit i fjärde byte får man fyra stycken stora subnät med 7 bitar för hostdelen (126 adresser). Subnäten är som följande:

201.94.112.0/25 är det **första** stora subnätet, subnätmask 255.255.255.128 som skall användas för det stora subnätet ifrågan.

201.94.112.128/25 är det **andra** stora subnätet, subnätmask 255.255. 255.128

201.94.113.0/25, är det trdje stora subnätet, subnätmask 255.255. 255.128 som skall subnättas ytterligare till två mindre subnät vilket ger:

201.94.113.0/26 det **första** mindre subnätet, subnätmask 255.255.255.192

201.94.113.64/26 det **andra** mindre subnätet, subnätmask 255.255.255.192

201.94.113.128/25, är det fjärde stora subnätet, subnätmask 255.255. 255.128 som skall subnättas ytterligare till två mindre subnät vilket ger:

201.94.113.128/26 det **tredje** mindre subnätet, subnätmask 255.255.255.192

201.94.113.192/26 subnätet, subnätmask 255.255. 255.192 skall subnättas ytterligare till två mindre subnät vilket ger:

201.94.113.192/27 det **minsta** subnätet av företagets subnät, subnätmask 255.255.255.224

10.3

201.94.113.224/27 subnätet med subnätmask 255.255.255.224 skall subnättas till 8 stycken 30-bitars subnät (255.255.255.252) som följande (bara sista byte):

201.94.113**.224, .228, .232, .236, .240, .244, .248, .252**

11. (4p)

a) **Maximum Segment Size (MSS)**
Den maximala storleken (antalet byte) på applikationsdata som varje TCP kan sända som ett segment exklusive header (räknas inte med) och som TCP-sändare och -mottagare har kommit överens om vid skapandet av TCP-anslutningen. Detta antal begränsas av länktyperna och deras maximala mängd av data (kallas MTU) som kan inkapslas i en ram och skickas över länken. TCP anpassar segmentets storlek med hänsyn tagen till att segmentet skall bli IP-paket som skall inkapslas i en ram utan att överskrida MTU.

b) **Nonce**
En slumpgenererat binärt tal som används vid autentisering av en slutanvädare. Talet får användas av autentiseringsprotokollet bara en gång ”once in a life time”. Nonce-användning syftar till att verifiera den som påstår sig vara en pålitlig användare och att den är aktuell (live).

c) **Beacon**
En trådlös accesspunkt måste göra närliggande enheter medvetna om dess närvaro genom att den skickar ut så kallade ”beacon frames" med regelbundna intervall (ofta 10 frames per sekund). En sådan frame innehåller bland annat tidsstämpel, SSID dvs namnet på det trådlösa nätverket, frekvenskanalen och bithastigher.

d) **Jitter**
Det är variationen i fördröjningen som drabbar paket över Internet, så att paket tillhörande samma multimediaströmmen kommer till mediaspelaren hos mottagare olika fördröjda trots att de skickades i konstant takt från sändaren. Fördröjningen är olika pga olika kötider på routrarna mellan sändaren och mottagren. Effekten blir hackning i t.ex. ljud om mediaspelaren spelar upp paket som de kommer utan att kompensera för jitter.