

Θέμα 1 (15%)

Θεωρείστε δίκτυο IP με έναν δρομολογητή και τέσσερα υποδίκτυα A, B, C και D με 200, 100, 50 και 20 hosts αντίστοιχα. Έχετε στη διάθεσή σας το μπλοκ διευθύνσεων 100.8.4.0/23. Ορίστε, χρησιμοποιώντας συμβολισμό CIDR, διευθύνσεις υποδικτύου ώστε οι διευθύνσεις IP που διατίθενται σε κάθε υποδίκτυο να επαρκούν για τις ανάγκες του.

100.8.4.0/23 → 32-23 = 9 bits for hosts → $2^9 = 512$ total addresses

Range: 100.8.4.0 - 100.8.5.255

Usable hosts = $2^n - 2$ (network + broadcast excluded) → prefix = $\lceil \log_2(32-n) \rceil$

Υποδίκτυο	Απαιτούμενοι hosts	Υπολογισμός	Prefix	Network Addresses	Usable hosts
A	≥ 200	$2^8 - 2 = 254$	/24	100.8.4.0/24	4.0 – 4.255
B	≥ 100	$2^7 - 2 = 126$	/25	100.8.5.0/25	5.0 – 5.127
C	≥ 50	$2^6 - 2 = 62$	/26	100.8.5.128/26	5.128 – 5.191
D	≥ 20	$2^5 - 2 = 30$	/27	100.8.5.192/27	5.192 – 5.223

Εάν ζητηθεί μπορείτε να δημιουργήσετε ένα επιπλέον δίκτυο E. Ποιο είναι το μέγιστο πλήθος host που μπορεί να περιέχει και ποια η αντίστοιχη διεύθυνση υποδικτύου;

Μένει το διάστημα 100.8.5.224/27 (5.224 – 5.255) με μέγιστο πλήθος host = $2^5 - 2 = 30$.

Θέμα 2 (10%)

Θεωρείστε τοπικό δίκτυο Ethernet το οποίο απαρτίζεται από 7 κόμβους συνδεδεμένους σε μεταγωγέα (switch). Ο κόμβος A θέλει να επικοινωνήσει με τον κόμβο B, του ίδιου τοπικού δικτύου, αλλά δεν γνωρίζει την IP διεύθυνσή του. Στο ίδιο τοπικό δίκτυο λειτουργεί εξυπηρετητής DNS του οποίου η διεύθυνση IP είναι γνωστή στον A.

(α) Ποια πακέτα ποιων πρωτοκόλλων θα ανταλλαχθούν μέχρι την ολοκλήρωση της εγκατάστασης σύνδεσης TCP από A στον B;

Βήμα	Πρωτόκολλο	Είδος πακέτου	Πηγή → Προορισμός	Σκοπός	Frames
1	ARP	ARP Request (broadcast)	A → LAN	Εύρεση MAC του DNS	7
2	ARP	ARP Reply (unicast)	DNS → A	DNS IP → DNS MAC	2
3	DNS / UDP / IP	DNS Query	A → DNS	Ερώτημα για IP του B	2
4	DNS / UDP / IP	DNS Response	DNS → A	Επιστροφή IP B	2
5	ARP	ARP Request (broadcast)	A → LAN	Εύρεση MAC του B	7
6	ARP	ARP Reply (unicast)	B → A	B IP → B MAC	2
7	TCP / IP	SYN	A → B	Έναρξη TCP σύνδεσης	2
8	TCP / IP	SYN-ACK	B → A	Αποδοχή σύνδεσης	2
9	TCP / IP	ACK	A → B	Ολοκλήρωση 3-way handshake	2

(β) Πόσα πλαίσια Ethernet θα μεταδοθούν συνολικά σε όλο το τοπικό δίκτυο; Υποθέστε ότι αυτή είναι η μοναδική κίνηση στο δίκτυο, ο πίνακας προώθησης του διακόπτη και οι πίνακες ARP των κόμβων είναι αρχικά άδειοι και δεν υπάρχουν λάθη ή απώλειες.

Frames = 9 (ARP για DNS) + 4 (DNS) + 9 (ARP για B) + 6 (TCP handshake) = 28 frames

Θέμα 3 (20%)

Υπολογιστής χρησιμοποιεί **TCP Tahoe** για να στείλει αρχείο 90 Kbyte. Αν δεν υπήρχαν απώλειες, θα απαιτείτο η αποστολή ακριβώς 90 τεμαχίων για τη μεταφορά ολοκληρού του αρχείου. Ωστόσο, λόγω συμφόρησης χάνονται το **3ο**, **33ο** και **90ο** τεμάχιο που στέλνει ο πομπός (λόγω επαναμεταδόσεων το 90ο τεμάχιο που στέλνει ο πομπός δεν είναι το 90ο τεμάχιο του αρχείου).

Συμπληρώστε τον πίνακα που ακολουθεί με τις τιμές του παραθύρου συμφόρησης (cwnd), του κατωφλίου αργής αρχής (sssthresh) όπου υπάρχει αλλαγή, το πλήθος τεμαχίων που έχουν αποσταλεί (send) και αυτών που έχουν παραληφθεί επιτυχώς (read). Θεωρείστε ότι το cwnd μετριέται σε τεμάχια και ότι ο χρόνος εκπομπής (timeout) είναι ίσος με 3RTT.

Σχεδιάστε το παράθυρο συμφόρησης (cwnd) σε συνάρτηση με το χρόνο, προσδιορίστε στο διάγραμμα όπου υπάρχουν οι φάσεις αργής αρχής, αποφυγής συμφόρησης και εκπομπής χρόνου.

RTT	cwnd	sssthresh	send	read	Σχόλια
1	1	-	1	1	Slow start
2	2	-	3	2	Slow start - Απώλεια 3 ^{ου} πακέτου (1ACK + χαμένο) -> Timeout
3-5	2	-	3	2	Timeout
6	1	1	4	3	Tahoe reset -> Slow start
7	2	1	6	5	Cong avoidance
8	3	1	9	8	Cong avoidance
9	4	1	13	12	Cong avoidance
10	5	1	18	17	Cong avoidance
11	6	1	24	23	Cong avoidance
12	7	1	31	30	Cong avoidance

13	8	1	39	31	Απώλεια 33 ^{ον} πακέτου (1ACK + χαμένο + 6ACK) -> triple dupACK
14	1	4	40	38	40-38=2 πακέτα που έχουν χαθεί
15	2	4	42	40	Slow start
16	4	4	46	44	Τέλος Slow Start (cwnd = ssthresh)
17	5	4	51	49	Cong avoidance
18	6	4	57	55	Cong avoidance
19	7	4	64	62	Cong avoidance
20	8	4	72	70	Cong avoidance
21	9	4	81	79	Cong avoidance
22	10	4	91	87	Απώλεια 90 ^{ον} πακέτου (8ACK + χαμένο + 1ACK) -> Timeout
23-25	10	-	91	87	Timeout
26	1	5	92	89	Tahoe reset -> Slow start
27	2	5	93	90	Τέλος αρχείου

Θέμα 4 (5%)

Ποια από τα ακόλουθα εγγύεται το πρωτόκολλο UDP;

- επιβεβαιώσεις στον αποστολέα
- έλεγχο ροής
- αριθμούς σειράς σε κάθε δεδομένογραμμα
- τίποτα από τα παραπάνω**

Θέμα 5 (10%)

Ένας υπολογιστής έχει ένα αρχείο 2 Mbyte προς μετάδοση πάνω από μια ζεύξη Ethernet. Υποθέστε ότι τα δεδομένα μεταφέρονται με τεμάχια TCP.

Δεδομένα:

File size = 2 MB = 2,000,000 bytes

MTU = 576 bytes

Protocol: TCP over IP

IP header = 20 bytes

TCP header = 20 bytes

Packet #	TCP Payload	Total Size
1	536	536 + 40 = 576
2	536	576
3	536	576
...	...	...
3,731	536	576
3,732	184	184 + 40 = 224
Total	3,731*536 + 184 = 2,000,000	3,731*576 + 224 = 2,149,280

(α) Πόσα πακέτα IP απαιτούνται για την αποστολή του αρχείου εάν MTU (Μέγιστη Μονάδα Μεταφοράς) 576 Byte;

Maximum TCP payload per packet = MTU - IP header - TCP header = 576 - 20 - 20 = 536 bytes

Number of packets = [File size / TCP payload] = [2,000,000 / 536] = **3,732 packets**

(β) Ποιο είναι το συνολικό μέγεθος του τελευταίου πακέτου IP;

Data of last packet = 2,000,000 - 3,731*536 = 184 bytes

Total size of last packet = IP header + TCP header + Remaining data = 20 + 20 + 184 = **224 bytes**

(γ) Πόσα byte μεταδόθηκαν συνολικά στη ζεύξη;

Total transmitted = **3,731*576 + 224 = 2,149,280 bytes**

Θέμα 6 (5%)

Αν όλες οι ζεύξεις (links) στο Διαδίκτυο παρείχαν αξιόπιστη υπηρεσία παράδοσης θα ήταν άσκοπη η αξιόπιστη υπηρεσία παράδοσης TCP; Αιτιολογήστε την απάντησή σας.

Όχι, η αξιόπιστη υπηρεσία του TCP δεν θα ήταν άσκοπη.

- Η αξιοπιστία σε επίπεδο ζεύξης είναι τοπική (hop-by-hop), ενώ το TCP παρέχει end-to-end αξιοπιστία.
- Πακέτα μπορεί να χαθούν ή να αλλοιωθούν λόγω σφαλμάτων κόμβων, υπερχειλίσις buffers ή επανεκκινήσεων, ακόμη και με αξιόπιστα links.
- Το TCP προσφέρει επιπλέον **έλεγχο ροής, έλεγχο συμφόρησης και διατήρηση σειράς**, που δεν παρέχονται από τα links.
- Σύμφωνα με την end-to-end principle, η αξιοπιστία πρέπει να υλοποιείται στα άκρα.

Θέμα 7 (5%)

Ας υποθέσουμε ότι δύο κόμβοι αρχίζουν να μεταδίδουν ταυτόχρονα ένα αρχείο μήκους L μέσω ενός καναλιού εκπομπής (broadcast channel) ρυθμού R . Υποθέτοντας ότι η καθυστέρηση διάδοσης μεταξύ των κόμβων είναι d_{prop} , θα υπάρξει σύγκρουση αν $d_{prop} < L/R$;

Το L/R είναι ο χρόνος μετάδοσης του πλαισίου. Αν η καθυστέρηση διάδοσης d_{prop} είναι μικρότερη από τον χρόνο μετάδοσης, τότε κάθε κόμβος αρχίζει να εκπέμπει πριν προλάβει να λάβει το σήμα του άλλου, με αποτέλεσμα τα σήματα να επικαλυφθούν στο κανάλι και να προκληθεί **σύγκρουση**.

Θέμα 8 (5%)

Μια εφαρμογή UDP στέλνει πακέτο IP που θρυμματίζεται σε 4 θραύσματα. Υποθέστε ότι τα θραύσματα 1, 3 και 4 φθάνουν στον προορισμό και το 2 χάνεται. Μετά από λίγο το ξαναστέλνει με τον ίδιο ακριβώς θρυμματισμό. Αυτή τη φορά χάνεται το θραύσμα 1. Εάν το χρονόμετρο επανασυναρμολόγησης στον προορισμό για το πρώτο πακέτο δεν έχει λήξει όταν φθάνει το θραύσμα 2 της δεύτερης μετάδοσης, μπορεί ο δέκτης να επανασυναρμολογήσει το πακέτο IP; Δικαιολογήστε την απάντησή σας.

OXI

Η επανασυναρμολόγηση θραυσμάτων στο IPv4 γίνεται με βάση τα πεδία της επικεφαλίδας: Source IP, Destination IP, Protocol και Identification. Όλα τα θραύσματα που ανήκουν στο ίδιο αρχικό datagram έχουν την ίδια τιμή Identification. Όταν το UDP πακέτο επαναστέλνεται, δημιουργείται νέο IPv4 datagram, το οποίο (κανονικά) έχει διαφορετική τιμή Identification. Επομένως, το θραύσμα 2 της δεύτερης μετάδοσης δεν θεωρείται ότι ανήκει στο ίδιο datagram με τα θραύσματα 1, 3 και 4 της πρώτης μετάδοσης και δεν μπορεί να καλύψει το χαμένο θραύσμα 2 του πρώτου πακέτου. Έτσι, η επανασυναρμολόγηση αποτυγχάνει.

Θέμα 9 (10%)

Υποθέστε ότι ένας δρομολογητής IPv4 έχει τον δίπλα πίνακα δρομολόγησης. Ποιο είναι το επόμενο βήμα ή διεπαφή όταν ο δρομολογητής έχει να στείλει στους παρακάτω προορισμούς.

Προορισμός	Μάσκα Υποδικτύου	Επόμενο Βήμα/Διεπαφή		
67.16.120.0	255.255.252.0	R1	67.16.120.0/22	120.0 – 123.255
67.16.124.0	255.255.252.0	Eth0	67.16.124.0/22	124.0 – 127.255
67.16.123.64	255.255.255.192	Eth1	67.16.123.64/26	123.64 – 123.127
0.0.0.0	0.0.0.0	R2	default	

α) 67.16.123.156 -> Eth0

β) 67.16.123.105 -> Eth1

γ) 67.16.121.67 -> R1

δ) 67.16.127.36 -> Eth0

ε) 67.16.128.87 -> R2

Θέμα 10 (15%)

Υποθέστε ότι ένα αρχείο με $F = 20$ Gbits πρόκειται να διανεμηθεί σε $N = 15$ ομότιμους κόμβους. Ο εξυπηρετητής έχει ρυθμό ανεβάσματος (uploading) $u_s = 30$ Mbps και κάθε ομότιμος έχει ρυθμό κατεβάσματος (downloading) $d_i = 2$ Mbps και ρυθμό ανεβάσματος (uploading) $u_i = 300$ Kbps. Υπολογίστε τον ελάχιστο χρόνο διανομής για κάθε μία από τις ακόλουθες περιπτώσεις:

Δεδομένα:

File size: $F = 20$ Gbits

Number of peers: $N = 15$

Server upload rate: $u_s = 30$ Mbps

Peer download rate: $d_i = 2$ Mbps (each)

Peer upload rate: $u_i = 300$ Kbps = 0.3 Mbps (each)

(α) Διανομή πελάτη-εξυπηρετητή (client-server).

Formula: $D_{CS} = \max \{ NF/u_s, F/d_{\min} \}$

Υπολογισμοί:

1) Server upload time (to N peers): $NF/u_s = (15 \times 20,000) / 30 = 300,000 / 30 = 10,000$ seconds

2) Slowest peer download time: $F/d_{\min} = 20,000 / 2 = 10,000$ seconds

Minimum distribution time: $D_{CS} = \max \{ 10,000, 10,000 \} = 10,000$ seconds

(β) Διανομή ομότιμων (P2P).

Formula: $D_{P2P} = \max \{ F/u_s, F/d_{\min}, NF/(u_s + \sum u_i) \}$

Υπολογισμοί:

1) Server upload time (at least one copy): $F/u_s = 20,000 / 30 = 666.67$ seconds

2) Slowest peer download time: $F/d_{\min} = 20,000 / 2 = 10,000$ seconds

3) Total upload capacity constraint: Total upload = $u_s + N \times u_i = 30 + 15 \times 0.3 = 30 + 4.5 = 34.5$ Mbps

$NF/(u_s + \sum u_i) = (15 \times 20,000) / 34.5 = 300,000 / 34.5 = 8695.65$ seconds

Minimum distribution time: $D_{P2P} = \max \{ 666.67, 10,000, 8695.65 \} = 10,000$ seconds

Κανονική 2019

Θέμα 1 (15%)

Θεωρήστε ότι σας έχει εκχωρηθεί το μπλοκ διευθύνσεων IP 196.44.168.0/21.

(α) Καθορίστε το μήκος προθέματος που επιτρέπει την ύπαρξη **400** host σε κάθε υποδίκτυο.

Usable hosts = $2^n - 2$ (network + broadcast excluded) $\rightarrow$ prefix = $\lceil \log_2(400+2) \rceil$

$2^9 - 2 = 510 > 400 \rightarrow$ prefix = /23

(β) Ποιός είναι ο μέγιστος αριθμός υποδικτύων που μπορεί να οριστεί;

$23 - 21 = 2$ bit for subnets $\Rightarrow 2^2 = 4$ subnets

(γ) Για κάθε υποδίκτυο καταγράψτε τη διεύθυνση IP αυτού (με συμβολισμό CIDR) καθώς και την αντίστοιχη διεύθυνση εκπομπής.

Υποδίκτυο	Network Addresses	Εύρος διευθύνσεων	IP Addresses	Broadcast Addresses
A	196.44.168.0/23	168.0 – 169.255	196.44.168.0	196.44.169.255
B	196.44.170.0/23	170.0 – 171.255	196.44.170.0	196.44.171.255
C	196.44.172.0/23	172.0 – 173.255	196.44.172.0	196.44.173.255
D	196.44.174.0/23	174.0 – 175.255	196.44.174.0	196.44.175.255

Θέμα 2 (10%)

(α) **Οντότητα UDP** στέλνει σε ομότιμη της δεδομένογραμμα με 2038 byte δεδομένων. Οι κόμβοι εκπομπής και λήψης συνδέονται άμεσα σε Ethernet με MTU = 1500 byte. Πόσα δεδομένογραμματα θα φτάσουν στην οντότητα UDP του δέκτη ως αποτέλεσμα της μετάδοσης;

Δεδομένα:

UDP data = 2038 bytes

MTU = 1500 bytes

Direct Ethernet connection

Υπολογισμοί:

Total UDP datagram size = UDP data + UDP header + IP header = 2038 + 8 + 20 = 2066 bytes

Maximum data per fragment = MTU - IP header = 1500 - 20 = 1480 bytes

Fragment offset alignment (multiple of 8): Max data (aligned) = $\lfloor 1480/8 \rfloor \times 8 = 185 \times 8 = 1480$ bytes

$\Rightarrow$ **Maximum data per fragment = 1480 bytes**

Total payload to fragment = UDP header + UDP data = 8 + 2038 = 2046 bytes

2 IP fragments: 1480 + 566 = 2046 bytes

Στο επίπεδο UDP, ο δέκτης επανασυναρμολογεί τα IP fragments πριν παραδώσει στο UDP layer. $\Rightarrow$ **1 UDP datagram**

(β) Υποθέστε ότι μια εφαρμογή χρησιμοποιεί UDP για να στείλει 38 bytes δεδομένων σε έναν εξυπηρετητή. Ποιά θα είναι η απόδοση της μετάδοσης στο στρώμα IP; Υποθέστε ότι δεν υπάρχουν επιλογές στην επικεφαλίδα IP.

Δεδομένα:

Application data = 38 bytes

UDP header = 8 bytes

IP header = 20 bytes (no options)

Υπολογισμοί:

Total datagram size: Total = IP header + UDP header + Data = 20 + 8 + 38 = 66 bytes

Overhead = IP header + UDP header = 20 + 8 = 28 bytes

Efficiency (at IP layer) = Data / Total = 38 / 66 = 0.5758 = **57.58%**

Θέμα 3 (15%)

(α) Τη χρονική στιγμή 0, αποστολέας TCP Reno ξεκινάει μια σύνδεση με διαφημιζόμενο παράθυρο 64 KB και MSS = 1 KB. Μόλις εγκατασταθεί η σύνδεση, ο αποστολέας TCP αρχίζει να στέλνει δεδομένα. Εάν το RTT είναι 250 ms και υποθέτοντας ότι δεν χάνεται κανένα τεμάχιο ή ότι δεν εμφανίζονται λήξεις χρονομέτρου ποια χρονική στιγμή το παράθυρο συμφόρησης θα είναι ίσο με 32 KB;

RTT	cwnd	ssthresh	Σχόλια
0	1	64	Έναρξη – Slow Start
1	2	64	Slow Start (διπλασιασμός)
2	4	64	Slow Start
3	8	64	Slow Start
4	16	64	Slow Start
5	32	64	Slow Start

Cwnd(5) = 32: $t = 5 * RTT = 5 * 0.25 = 1.25$ sec

(β) Αμέσως μετά εμφανίζεται λήξη **χρονομέτρου**. Πόσο χρόνο μετά την εκπνοή του χρονομέτρου θα γίνει το παράθυρο συμφόρησης ίσο με 20 KB, εάν ο αποστολέας συνεχίζει να στέλνει δεδομένα στην εγκατεστημένη σύνδεση και δεν υπάρχουν επιπλέον απώλειες ή λήξεις χρονομέτρων;

Μετά το timeout:

RTT	cwnd	ssthresh	Σχόλια
0	1	16	Slow Start
1	2	16	Slow Start
2	4	16	Slow Start
3	8	16	Slow Start
4	16	16	Τέλος Slow Start (cwnd = ssthresh)
5	17	16	Congestion Avoidance
6	18	16	Congestion Avoidance
7	19	16	Congestion Avoidance
8	20	16	Congestion Avoidance

Cwnd(8)=20: $t = 8 * RTT = 2 \text{ sec}$

(γ) Ενώ το παράθυρο συμφόρησης παραμένει στην προηγούμενη τιμή, ο αποστολέας λαμβάνει τρεις ταυτόσημες επαληθεύσεις. Πόσο χρόνο μετά τη λήψη της τρίτης επαλήθευσης θα γίνει το παράθυρο συμφόρησης ίσο με 12 KB, εάν ο αποστολέας συνεχίζει να στέλνει δεδομένα στην εγκατεστημένη σύνδεση και δεν υπάρχουν επιπλέον απώλειες ή λήξεις χρονομέτρων;

Μετά από triple duplicate ACK:

RTT	cwnd	ssthresh	Σχόλια
0	13	10	Fast Retransmit / Fast Recovery
1	10	10	Congestion Avoidance
2	11	10	Congestion Avoidance
3	12	10	Congestion Avoidance

Cwnd(3)=12: $t = 3 * RTT = 0.75 \text{ sec}$

Θέμα 4 (10%)

Θεωρήστε τοπικό δίκτυο Ethernet που απαρτίζεται από 6 κόμβους συνδεδεμένους σε μεταγωγέα (switch). Ο κόμβος A θέλει να επικοινωνήσει με τον κόμβο B, του ίδιου τοπικού δικτύου, αλλά δεν γνωρίζει την IP διεύθυνσή του. Στο ίδιο τοπικό δίκτυο λειτουργεί εξυπηρετητής DNS του οποίου η διεύθυνση IP είναι γνωστή στον A. Υποθέστε ότι αυτή είναι η μοναδική κίνηση στο δίκτυο, ο πίνακας προώθησης του διακόπτη και οι πίνακες ARP των κόμβων είναι αρχικά άδειοι και δεν υπάρχουν λάθη ή απώλειες.

(α) Ποιά πακέτα ποιών πρωτοκόλλων θα ανταλλαχθούν μέχρι την ολοκλήρωση εγκατάστασης σύνδεσης TCP από τον A στον B;

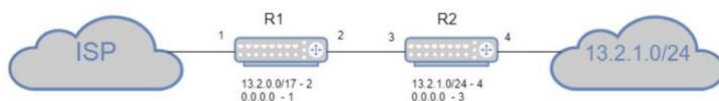
Βήμα	Πρωτόκολλο	Είδος πακέτου	Πηγή → Προορισμός	Σκοπός	Frames
1	ARP	ARP Request (broadcast)	A → LAN	Εύρεση MAC του DNS	6
2	ARP	ARP Reply (unicast)	DNS → A	DNS IP → DNS MAC	2
3	DNS / UDP / IP	DNS Query	A → DNS	Ερώτημα για IP του B	2
4	DNS / UDP / IP	DNS Response	DNS → A	Επιστροφή IP B	2
5	ARP	ARP Request (broadcast)	A → LAN	Εύρεση MAC του B	6
6	ARP	ARP Reply (unicast)	B → A	B IP → B MAC	2
7	TCP / IP	SYN	A → B	Έναρξη TCP σύνδεσης	2
8	TCP / IP	SYN-ACK	B → A	Αποδοχή σύνδεσης	2
9	TCP / IP	ACK	A → B	Ολοκλήρωση 3-way handshake	2

(β) Πόσα πλαίσια Ethernet θα μεταδοθούν συνολικά σε **όλο** το τοπικό δίκτυο;

Frames = 8 (ARP για DNS) + 4 (DNS) + 8 (ARP για B) + 6 (TCP handshake) = 26 frames

Θέμα 5 (15%)

Σε μικρή εταιρεία εκχωρείται ένα μεγάλο μπλοκ διευθύνσεων IP 13.2.0.0/17, αλλά χρησιμοποιείται μόνο ένα μέρος αυτών (13.2.1.0/24) για τους υπολογιστές της. Η εταιρεία έχει έναν μόνο ISP για πρόσβαση στο υπόλοιπο Internet. Στο σχήμα φαίνονται οι πίνακες προώθησης στο δρομολογητή του ISP (αριστερά) και του ακραίου δρομολογητή της εταιρείας (δεξιά).



(α) Πόσες διευθύνσεις έχει διαθέσιμες οι εταιρεία;

Prefix = /17 => $32-17=15 \text{ bits} \Rightarrow 2^{15}=32,768 \text{ addresses}$

(β) Ποιά είναι η μικρότερη και ποια η μεγαλύτερη διεύθυνση IP (από το μέρος 13.2.1.0/24), ασχέτως αν χρησιμοποιείται ή όχι;

Prefix = /24 => $32-24=8 \text{ bits} \Rightarrow 2^8=256 \text{ addresses}$

13.2.1.0/24 → 13.2.1.0-13.2.1.255

(γ) Υποθέστε ότι ο δρομολογητής του ISP λαμβάνει πακέτο με διεύθυνση προορισμού 13.2.1.1. Ποιά διαδρομή θα ακολουθήσει αυτό το πακέτο;

R1: το 13.2.1.1 ταιριάζει στο 13.2.0.0/17, άρα προώθηση από θύρα 2 προς R2.

R2: το 13.2.1.1 ταιριάζει στο 13.2.1.0/24, άρα προώθηση από θύρα 4 προς το LAN (13.2.1.0/24) και φτάνει στον host.

Διαδρομή: R1 → (2) → R2 → (4) → 13.2.1.0/24 → 13.2.1.1

(δ) Τι τελικά θα συμβεί στο πακέτο με διεύθυνση προορισμού 13.2.20.1;

R1: το 13.2.20.1 ταιριάζει στο 13.2.0.0/17, άρα προώθηση από θύρα 2 προς R2.

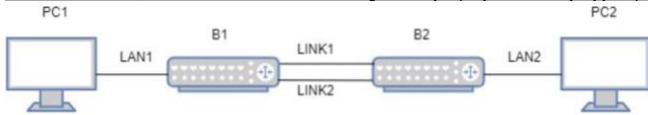
R2: δεν έχει ειδική διαδρομή για 13.2.20.0/..., άρα χρησιμοποιεί default και το στέλνει πίσω στον R1 (θύρα 3).

R1: ξανά το βλέπει ως /17 και το ξαναστέλνει στον R2...

➡ Δημιουργείται **routing loop μεταξύ R1 και R2** μέχρι να μηδενιστεί το TTL, οπότε το πακέτο **απορρίπτεται** (συνήθως με **ICMP Time Exceeded** προς την πηγή).

Θέμα 6 (10%)

Τα B1 και B2 στο σχήμα είναι γέφυρες (μεταγωγείς Ethernet) με τρεις θύρες, αλλά δεν έχουν ενεργοποιημένο το πρωτόκολλο επικαλύπτοντος δέντρου (spanning tree protocol) για την αποφυγή σχηματισμού βρόχων. Αρχικά οι πίνακες ARP των υπολογιστών είναι άδειοι, το ίδιο και οι πίνακες των γεφυρών. Περιγράψτε τι θα συμβεί εάν το PC2 κάνει **ping** στο PC1.



Για να εκτελέσει το ping, το PC2 στέλνει αρχικά **ARP Request (broadcast)** για να βρει τη MAC του PC1. Οι γέφυρες B2 και B1, έχοντας άδειους πίνακες και χωρίς ενεργοποιημένο STP, προωθούν το broadcast μέσω flooding και από τους δύο παράλληλους συνδέσμους (LINK1, LINK2).

Λόγω της ύπαρξης βρόχου επιπέδου ζεύξης, το ARP Request ανακυκλώνεται επ' αόριστον, δημιουργώντας **broadcast storm** και προκαλώντας αστάθεια στους πίνακες προώθησης (**MAC flapping**) των γεφυρών.

Ως αποτέλεσμα, ακόμη και αν το PC1 απαντήσει με ARP Reply και στη συνέχεια στα ICMP Echo, τα πλαίσια είτε προωθούνται λανθασμένα είτε χάνονται μέσα στη συμφόρηση.

Συμπέρασμα: Χωρίς STP, το ping από το PC2 στο PC1 είναι αναξιόπιστο ή πρακτικά αδύνατο, λόγω βρόχου στο επίπεδο ζεύξης.

Θέμα 7 (5%)

Προσδιορίστε τι πρωτόκολλο μεταφοράς θα χρησιμοποιούσατε αν η εφαρμογή σας είχε μια από τις παρακάτω ιδιότητες και δικαιολογήστε την επιλογή σας.

1. Αξιοπιστία

TCP. Το TCP παρέχει αξιόπιστη μεταφορά δεδομένων με μηχανισμούς επιβεβαιώσεων (ACKs), επαναμεταδόσεων και διατήρησης σωστής σειράς πακέτων, άρα εξασφαλίζει ότι όλα τα δεδομένα θα φτάσουν σωστά.

2. Μεγάλου μήκους μηνύματα

TCP. Το TCP είναι κατάλληλο για μεγάλα μηνύματα, καθώς τα τεμαχίζει σε τμήματα (segmentation), τα μεταδίδει αξιόπιστα και τα επανασυνθέτει σωστά στον παραλήπτη.

3. Ταχύτητα

UDP. Το UDP δεν έχει overhead σύνδεσης, επιβεβαιώσεων ή επαναμεταδόσεων, με αποτέλεσμα μικρότερη καθυστέρηση και μεγαλύτερη ταχύτητα, ειδικά για εφαρμογές πραγματικού χρόνου.

4. Αλληλεπιδράσεις μικρής διάρκειας

UDP. Το UDP δεν απαιτεί εγκατάσταση σύνδεσης (handshake), άρα είναι πιο αποδοτικό για σύντομες ανταλλαγές δεδομένων, όπου ο χρόνος εγκατάστασης TCP θα ήταν δυσανάλογος.

5. Έλεγχος συμφόρησης

TCP. Το TCP ενσωματώνει μηχανισμούς ελέγχου συμφόρησης (π.χ. slow start, congestion avoidance), προσαρμόζοντας τον ρυθμό μετάδοσης ώστε να αποφεύγεται η υπερφόρτωση του δικτύου.

Θέμα 8 (10%)

(α) Υποθέστε ότι οι διευθύνσεις MAC δύο κόμβων σε δεκαεξαδική μορφή είναι: 78:08:BF:01:99:33 και 78:08:BF:01:88:55. Μπορείτε να συμπεράνετε ότι οι δύο κόμβοι βρίσκονται στο ίδιο τοπικό δίκτυο;

Όχι, **δεν μπορούμε** να συμπεράνουμε ότι βρίσκονται στο ίδιο τοπικό δίκτυο μόνο από τις MAC διευθύνσεις.

Οι MAC είναι **διευθύνσεις επιπέδου ζεύξης (L2)** και το κοινό πρόθεμα (π.χ. τα πρώτα bytes/OUI) απλώς δείχνει ότι πιθανόν οι κάρτες δικτύου είναι από **τον ίδιο κατασκευαστή/σειρά**, όχι ότι οι κόμβοι είναι στο ίδιο Ethernet LAN/ίδιο broadcast domain (θα μπορούσαν να είναι σε τελείως διαφορετικά LANs).

(β) Μπορεί ένα πακέτο IP να κυκλοφορεί αενάως εντός του διαδικτύου εάν λόγω κάποιου λάθους οι πίνακες δρομολόγησης σχηματίζουν τοπολογία που περιέχει κυκλική διαδρομή;

Όχι, ένα IP πακέτο **δεν μπορεί** να κυκλοφορεί αενάως στο Internet, ακόμη κι αν υπάρχει δρομολογιακός βρόχος.

Κάθε IP πακέτο έχει πεδίο TTL (Time To Live) που μειώνεται κατά 1 σε κάθε router· όταν φτάσει στο 0, το πακέτο **απορρίπτεται** (και συνήθως επιστρέφεται ICMP Time Exceeded).

Θέμα 9 (10%)

Τεμάχιο TCP μεγέθους 1074 byte (συμπεριλαμβανομένων των επικεφαλίδων 20 byte του στρώματος δικτύου και 20 byte του στρώματος μεταφοράς) μεταφέρεται μέσα από ζεύξη που έχει μέγιστη μονάδα μεταφοράς (MTU) 384 byte. Προσδιορίστε όλα τα θραύσματα (πακέτα IP) που θα παραχθούν καθώς και τις ακόλουθες παραμέτρους για καθένα από αυτά: Ταυτότητα, Σημείες (DF, MF, flags), θέση θραύσματος (Fragment Offset), συνολικό μέγεθος θραύσματος, μέγεθος δεδομένων θραύσματος. Θεωρείστε ότι το αρχικό πακέτο IP έχει αριθμό ταυτότητας 623.

Δεδομένα:

Total TCP segment size = 1074 bytes

IP header = 20 bytes

TCP header = 20 bytes

Data payload = 1074 - 20 - 20 = 1034 bytes

MTU = 384 bytes

Initial IP ID = 623

Υπολογισμοί:

Maximum data per fragment = MTU - IP header = 384 - 20 = 364 bytes

&& Max data (aligned to 8) = $\lfloor 364/8 \rfloor \times 8 = 45 \times 8 = 360$ bytes

=> Max data = 360 bytes

Data payload = 360 + 360 + 314 = 1034 bytes

Θραύσμα	Συνολικό μέγεθος θραύσματος (byte)	Μέγεθος δεδομένων θραύσματος (byte)	Identification	MF flag	DF flag	Fragment offset (δεκαδική μορφή)
1	360 + 20 = 380	360	623	1	0	0
2	380	360	623	1	0	45
3	314+20 = 334	1034 - 2*360 = 314	623	0	0	90
4						

Θέμα 1 (15%)

Θεωρήστε **πακέτο IP** μήκους 1094 byte, με επικεφαλίδα 20 byte και αριθμό ταυτότητας 675, το οποίο πρόκειται να μεταδοθεί σε δίκτυο όπου η MTU είναι 384 byte.

(α) Να καθορίσετε **τον αριθμό των θραυσμάτων** που θα προκύψουν.

(β) Να ορίσετε την τιμή των παρακάτω πεδίων για κάθε θραύσμα:

Δεδομένα:

Total IP packet length = 1094 bytes

IP header = 20 bytes

Data payload = 1094 - 20 = 1074 bytes

Identification = 675

MTU = 384 bytes

Υπολογισμοί:

Maximum data per fragment = MTU - IP header = 384 - 20 = 364 bytes

&& Max data (aligned to 8) = $\lfloor 364/8 \rfloor \times 8 = 45 \times 8 = 360$ bytes

=> Max data = 360 bytes

Number of fragments = $\lceil 1074 / 360 \rceil = \lceil 2.98 \rceil = 3$ fragments

Data payload = 360 + 360 + 354 = 1074 bytes

Θραύσμα	Total length (σε byte)	Identification	MF flag	Fragment offset (σε δεκαδική μορφή)
1	360+20 = 380	675	1	0
2	380	675	1	45
3	354+20 = 374	675	0	90
4				

Θέμα 2 (15%)

Υποθέστε ότι ένας δρομολογητής IPv4 έχει τον παρακάτω πίνακα δρομολόγησης:

Προορισμός	Μάσκα υποδικτύου	Επόμενο βήμα / διεπαφή		
67.16.160.0	255.255.252.0	R1	67.16.160.0/22	160.0 – 163.255
67.16.164.0	255.255.252.0	Eth0	67.16.164.0/22	164.0 – 167.255
67.16.163.64	255.255.255.192	Eth1	67.16.163.64/26	163.64 – 163.127
0.0.0.0	0.0.0.0	R2	default	

Να προσδιορίσετε το επόμενο βήμα ή τη διεπαφή εξόδου όταν ο δρομολογητής πρόκειται να στείλει πακέτα προς τους παρακάτω προορισμούς:

(α) 67.16.163.152 -> R1

(β) 67.16.163.103 -> Eth1

(γ) 67.16.161.65 -> R1

(δ) 67.16.167.34 -> Eth0

Θέμα 3 (10%)

Θεωρείστε ότι σας έχει εκχωρηθεί το μπλοκ διευθύνσεων IP **198.42.160.0/20**.

(α) Καθορίστε το μήκος προθέματος δικτύου που επιτρέπει την ύπαρξη **900 host** σε κάθε υποδίκτυο.

Usable hosts = $2^n - 2$ (network + broadcast excluded) → prefix = $\lceil \log_2(900+2) \rceil = 10$

$2^{10} - 2 = 1022 > 900 \rightarrow$ prefix = /22

(β) Ποιος είναι ο μέγιστος αριθμός υποδικτύων που μπορεί να οριστεί;

/22 - /20 = 2 bit for subnets => $2^2 = 4$ subnets

Υποδίκτυο	Network Addresses	Εύρος διευθύνσεων
A	198.42.160.0/22	160.0 – 163.255
B	198.42.164.0/22	164.0 – 167.255
C	198.42.168.0/22	168.0 – 171.255
D	198.42.172.0/22	172.0 – 175.255

(γ) Για το υποδίκτυο με τις μικρότερες αριθμητικά διευθύνσεις καταγράψτε τη διεύθυνση IP αυτού (με συμβολισμό **CIDR**).

198.42.160.0

(δ) Για το υποδίκτυο με τις μεγαλύτερες αριθμητικά διευθύνσεις καταγράψτε τη διεύθυνση εκπομπής αυτού (με συμβολισμό **CIDR**).

198.42.175.255

Θέμα 4 (15%)

Υποθέστε ότι ο υπολογιστής A εκπέμπει ένα ARP request με σκοπό να εντοπίσει τον υπολογιστή B. Ωστόσο, στο ARP request που αποστέλλει, ο A δηλώνει ως MAC διεύθυνση αποστολέα τη MAC διεύθυνση του υπολογιστή C, αντί για τη δική του.

α) Θα λάβει ο υπολογιστής A απάντηση (ARP reply) από τον B; Να αιτιολογήσετε την απάντησή σας.

OXI

	ARP Request (A → broadcast)	ARP Reply (B → C)
Source MAC	MAC_C (spoofed)	MAC_B
Source IP	IP_A	IP_B
Target MAC	ff:ff:ff:ff:ff:ff	MAC_C
Target IP	IP_B	IP_A

Όταν ο B λάβει το ARP request: Ενημερώνει τον ARP table του: IP_A → MAC_C
Ο A δεν θα λάβει την απάντηση γιατί το frame προορίζεται για MAC_C.

β) Αν οι πίνακες ARP των υπολογιστών A, B και C ήταν αρχικά κενοί πριν από την αποστολή του ARP request από τον A, να αναφέρετε το περιεχόμενο των πινάκων ARP μετά την ολοκλήρωση της ανταλλαγής ARP μηνυμάτων.

ARP tables:

A	IP	MAC	B	IP	MAC	C	IP	MAC
				IP_A	MAC_C		IP_B	MAC_B

Θέμα 5 (15%)

Υποθέστε ότι ένα αρχείο μεγέθους $F = 20 \text{ Gbits}$ πρόκειται να διανεμηθεί σε $N = 10$ ομότιμους κόμβους. Ο εξυπηρετητής έχει ρυθμό ανεβάσματος (uploading) $u_s = 30 \text{ Mbps}$, και κάθε ομότιμος κόμβος έχει ρυθμό κατεβάσματος (downloading) $d_i = 2 \text{ Mbps}$ και ρυθμό ανεβάσματος (uploading) $u_i = 300 \text{ Kbps}$. Να υπολογιστεί ο **ελάχιστος χρόνος διανομής** για καθεμία από τις ακόλουθες περιπτώσεις:

Δεδομένα:

File size: $F = 20 \text{ Gbits} = 20,000 \text{ bits}$

Number of peers: $N = 10$

Server upload rate: $u_s = 30 \text{ Mbps}$

Peer download rate: $d_i = 2 \text{ Mbps}$ (each)

Peer upload rate: $u_i = 300 \text{ Kbps} = 0.3 \text{ Mbps}$ (each)

(α) Διανομή πελάτη-εξυπηρετητή (**client-server**).

Formula: $D_{CS} = \max \{ NF/u_s, F/d_{\min} \}$

Υπολογισμοί:

1) Server upload time (to N peers): $NF/u_s = (10 \times 20,000) / 30 = 200,000 / 30 = 6,666.67 \text{ seconds}$

2) Slowest peer download time: $F/d_{\min} = 20,000 / 2 = 10,000 \text{ seconds}$

Minimum distribution time: $D_{CS} = \max \{ 6,666.67, 10,000 \} = 10,000 \text{ seconds}$

(β) Διανομή ομότιμων (**P2P**).

Formula: $D_{P2P} = \max \{ F/u_s, F/d_{\min}, NF/(u_s + \sum u_i) \}$

Υπολογισμοί:

1) Server upload time (at least one copy): $F/u_s = 20,000 / 30 = 666.67 \text{ seconds}$

2) Slowest peer download time: $F/d_{\min} = 20,000 / 2 = 10,000 \text{ seconds}$

3) Total upload capacity constraint: Total upload = $u_s + N \times u_i = 30 + 10 \times 0.3 = 30 + 3 = 33 \text{ Mbps}$

$NF/(u_s + \sum u_i) = (10 \times 20,000) / 33 = 200,000 / 33 = 6,060.61 \text{ seconds}$

Minimum distribution time: $D_{P2P} = \max \{ 666.67, 10,000, 6,060.61 \} = 10,000 \text{ seconds}$

Σε αυτό το σενάριο, το **peer download rate (2 Mbps)** είναι το bottleneck και για τις δύο περιπτώσεις, άρα ο χρόνος είναι ίδιος. Το P2P όφελος δεν φαίνεται επειδή το $d_{\min}$ είναι πολύ χαμηλό.

Θέμα 6 (15%)

Θέλετε να κατεβάσετε στον υπολογιστή σας μία ιστοσελίδα μεγέθους 100 kB, η οποία περιέχει 10 ενσωματωμένες εικόνες, καθεμία από τις οποίες έχει επίσης μέγεθος 100 kB. Η σελίδα και οι εικόνες βρίσκονται στον ίδιο εξυπηρετητή. Η διαδρομή δικτύου μεταξύ του πλοηγού (browser) και του εξυπηρετητή ιστού θεωρείται ως μία ζεύξη ταχύτητας 10 Mbps με RTT ίσο με 100 msec. Ο χρόνος μετάδοσης του μηνύματος HTTP GET στη ζεύξη θεωρείται μηδενικός. Να υπολογιστεί ο χρόνος που παρεμβάλλεται από τη στιγμή που αποστέλλεται η αίτηση URL μέχρι να ληφθεί ολόκληρη η σελίδα μαζί με τις ενσωματωμένες εικόνες, για τις ακόλουθες περιπτώσεις:

Δεδομένα:

Base HTML page: 100 kB

10 embedded images: 100 kB each

Link speed: 10 Mbps

RTT = 100 ms

HTTP GET transmission time ≈ 0

Transmission time per object: $T_{\text{trans}} = (100 \text{ kB} \times 8) / 10 \text{ Mbps} = 800 \text{ kb} / 10 \text{ Mbps} = 0.08 \text{ sec} = 80 \text{ ms}$

(α) Non-persistent HTTP χωρίς παράλληλες συνδέσεις μεταξύ πλοηγού και εξυπηρετητή.

Process:

- Κάθε object απαιτεί νέα TCP connection

- TCP handshake (1 RTT) + Request/Response (1 RTT) + Transmission time

Time per object: $T_{\text{object}} = 1 \text{ RTT (TCP setup)} + 1 \text{ RTT (HTTP GET + start response)} + T_{\text{trans}} = 100 + 100 + 80 = 280 \text{ ms}$

Total objects: 1 HTML + 10 images = 11 objects

$T_{\text{total}} = 11 \times 280 = 3,080 \text{ ms} = 3.08 \text{ seconds}$

(β) Non-persistent HTTP με παράλληλες συνδέσεις, οι οποίες μοιράζονται εξίσου τη ζεύξη.

Process:

- HTML page: 1 connection (sequential)
- 10 images: parallel connections (simultaneous)
- Bandwidth shared equally: $10 \text{ Mbps} / 10 = 1 \text{ Mbps per connection}$

HTML page download: $T_{\text{html}} = 1 \text{ RTT (TCP)} + 1 \text{ RTT (HTTP)} + T_{\text{trans}} = 100 + 100 + 80 = 280 \text{ ms}$

Images download (parallel):

Transmission time per image (1 Mbps shared): $T_{\text{trans_shared}} = 800 \text{ kb} / 1 \text{ Mbps} = 800 \text{ ms}$

$T_{\text{images}} = 1 \text{ RTT (TCP)} + 1 \text{ RTT (HTTP)} + T_{\text{trans_shared}} = 100 + 100 + 800 = 1,000 \text{ ms}$

$T_{\text{total}} = T_{\text{html}} + T_{\text{images}} = 280 + 1,000 = 1,280 \text{ ms} = 1.28 \text{ seconds}$

(γ) Persistent HTTP χωρίς συνεχή παροχή (pipelining).

Process:

- Single TCP connection (1 setup)
- Each request waits for previous response (sequential)
- No need for repeated TCP handshakes

HTML page: $T_{\text{html}} = 1 \text{ RTT (TCP setup)} + 1 \text{ RTT (HTTP GET/response)} + T_{\text{trans}} = 100 + 100 + 80 = 280 \text{ ms}$

Each image (over persistent connection): $T_{\text{image}} = 1 \text{ RTT (HTTP GET/response)} + T_{\text{trans}} = 100 + 80 = 180 \text{ ms}$

$T_{\text{total}} = T_{\text{html}} + 10 \times T_{\text{image}} = 280 + 10 \times 180 = 280 + 1,800 = 2,080 \text{ ms} = 2.08 \text{ seconds}$

Θέμα 7 (15%)

1. Ο αλγόριθμος αργής εκκίνησης αυξάνει το ρυθμό αποστολής του πομπού ταχύτερα από την προσθετική αύξηση.

ΣΩΣΤΟ.

Στην αργή εκκίνηση το cwnd αυξάνεται εκθετικά (διπλασιάζεται ανά RTT), ενώ στην προσθετική αύξηση αυξάνεται γραμμικά.

2. Αν το μέγεθος διαφημιζόμενου παραθύρου μιας σύνδεσης TCP (μετρούμενο σε sec) είναι μικρότερο από RTT, η ζεύξη θα λειτουργεί πολύ αποδοτικά.

ΛΑΘΟΣ.

Αν το παράθυρο είναι μικρότερο από το RTT (σε sec), ο αποστολέας δεν μπορεί να κρατήσει τη ζεύξη πλήρως απασχολημένη.

3. Αν η τιμή της λήξης χρόνου επαναμετάδοσης (RTO) είναι πολύ μικρή, μπορεί να έχουμε άσκοπες επαναμεταδόσεις.

ΣΩΣΤΟ.

4. Ένας δέκτης περιορίζει το διαφημιζόμενο μέγεθος παραθύρου σε απάντηση της συμφόρησης που εμφανίζεται στους δρομολογητές κατά μήκος της διαδρομής.

ΛΑΘΟΣ.

Το διαφημιζόμενο παράθυρο ελέγχεται από τον δέκτη και σχετίζεται με έλεγχο ροής, όχι με συμφόρηση στο δίκτυο.

5. Δεν υπάρχει βελτίωση της επίδοσης, αν το μέγεθος διαφημιζόμενου παραθύρου μιας σύνδεσης TCP (μετρούμενο σε sec) είναι μεγαλύτερο από RTT.

ΣΩΣΤΟ.

Όταν το παράθυρο ξεπερνά το RTT (σε sec), η ζεύξη είναι ήδη πλήρως αξιοποιημένη και δεν βελτιώνεται η απόδοση.

6. Οι αλγόριθμοι ελέγχου συμφόρησης στο TCP θεωρούν ότι τεμάχια TCP χάνονται μόνο όταν οι ουρές στους δρομολογητές υπερχειλίζουν.

ΣΩΣΤΟ.

Το TCP θεωρεί την απώλεια πακέτων ως ένδειξη συμφόρησης λόγω υπερχειλίσσης ουρών στους δρομολογητές.

7. Η τιμή της λήξης χρόνου επαναμετάδοσης (RTO) ενός πομπού τίθεται πάντοτε ίση με την τιμή του RTT.

ΛΑΘΟΣ.

Το RTO δεν είναι ίσο με το RTT, αλλά μεγαλύτερο ώστε να λαμβάνει υπόψη τη μεταβλητότητα του RTT.

8. Η τιμή της λήξης χρόνου επαναμετάδοσης (RTO) ενός πομπού αυξάνει με τη μεταβλητότητα των μετρούμενων τιμών RTT.

ΣΩΣΤΟ.

Το RTO αυξάνεται όταν αυξάνεται η διακύμανση των μετρήσεων RTT για αποφυγή ψευδών timeouts.

$RTO = \text{Estimated RTT} + 4 \times \text{DevRTT}$

Κανονική 2024

Θέμα 1 (20%)

Θεωρήστε ένα δίκτυο IP με έναν δρομολογητή και τέσσερα υποδίκτυα A, B, C και D, τα οποία πρέπει να υποστηρίζουν 50, 50, 100 και 200 hosts, αντίστοιχα. Σας δίνεται για διάθεση το σύνολο διευθύνσεων 100.5.32.0/23. Να αναθέσετε, χρησιμοποιώντας CIDR (Classless Inter-Domain Routing), διευθύνσεις δικτύου στα υποδίκτυα A, B, C και D, έτσι ώστε:

- να καλύπτονται οι απαιτήσεις σε αριθμό hosts κάθε υποδικτύου και
- να χρησιμοποιούνται οι διαθέσιμες διευθύνσεις με τον πιο αποδοτικό τρόπο.

100.5.32.0/23 → 32-23 = 9 bits for hosts → $2^9 = 512$ total addresses

Range: 100.5.32.0 - 100.5.33.255

Usable hosts = $2^n - 2$ (network + broadcast excluded) → prefix = $/(32-n)$

Υποδίκτυο	Απαιτούμενοι hosts	Υπολογισμός	Prefix	Network Addresses	Εύρος διευθύνσεων
D	≥ 200	$2^8 - 2 = 254$	/24	100.5.32.0/24	32.0 – 32.255
C	≥ 100	$2^7 - 2 = 126$	/25	100.5.33.0/25	33.0 – 33.127
B	≥ 50	$2^6 - 2 = 62$	/26	100.5.255.128/26	33.128 – 33.191
A	≥ 50	$2^6 - 2 = 62$	/26	100.5.255.192/26	33.192 – 33.255

Επιπλέον, υποθέστε ότι πρέπει να δημιουργηθεί και ένα επιπλέον υποδίκτυο E. Ποιο είναι το μέγιστο πλήθος hosts που μπορεί να υποστηρίξει το υποδίκτυο E και ποια είναι η αντίστοιχη διεύθυνση υποδικτύου;

Μετά την παραπάνω βέλτιστη κατανομή δεν απομένει καθόλου ελεύθερος χώρος στο /23. Άρα δεν μπορεί να δημιουργηθεί επιπλέον υποδίκτυο E.

Θέμα 2 (15%)

Υποθέστε ότι ένας δρομολογητής IPv4 διαθέτει τον παρακάτω πίνακα δρομολόγησης:

Προορισμός	Μάσκα υποδικτύου	Επόμενο βήμα / Διεπαφή		
76.12.100.0	255.255.252.0	R1	76.12.100.0/22	100.0 – 103.255
76.12.104.0	255.255.252.0	Eth0	76.12.104.0/22	104.0 – 107.255
76.12.103.64	255.255.255.192	Eth1	76.12.103.64/26	103.64 – 103.127
0.0.0.0	0.0.0.0	R2	Default	

Να βρεθεί το επόμενο βήμα (next hop) ή η διεπαφή εξόδου που θα επιλέξει ο δρομολογητής (με βάση τον κανόνα του longest prefix match) για να προωθήσει πακέτα προς τους παρακάτω προορισμούς:

- α) 76.12.103.152 → R1
β) 76.12.107.34 → Eth0
γ) 76.12.101.65 → R1
δ) 76.12.108.89 → R2
ε) 76.12.103.103 → Eth1

Θέμα 3 (20%)

Τεμάχιο TCP μεγέθους 2084 byte (συμπεριλαμβανομένων των επικεφαλίδων 20 byte του στρώματος δικτύου και 20 byte του στρώματος μεταφοράς) μεταφέρεται μέσα από ζεύξη που έχει Μέγιστη Μονάδα Μεταφοράς (MTU) 1000 byte. Συμπληρώστε τον παρακάτω πίνακα για τα θραύσματα που θα παραχθούν. Θεωρείστε ότι το αρχικό πακέτο IP έχει αριθμό ταυτότητας 800.

Δεδομένα:

Total TCP segment size = 2084 bytes

IP header = 20 bytes

TCP header = 20 bytes

Data payload = 2084 - 20 - 20 = 2044 bytes

MTU = 1000 bytes

Initial IP ID = 800

Υπολογισμοί:

Maximum data per fragment = MTU - IP header = 1000 - 20 = 980 bytes

&& Max data (aligned to 8) = $\lceil 980/8 \rceil \times 8 = 122 \times 8 = 976$ bytes

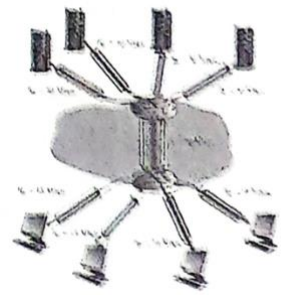
=> **Max data = 976 bytes**

Data payload = 976 + 976 + 92 = 2044 bytes

Θραύσμα	Συνολικό μέγεθος θραύσματος (byte)	Μέγεθος δεδομένων θραύσματος (byte)	Identification	MF flag	DF flag	Fragment offset (δεκαδική μορφή)
1	976 + 20 = 996	976	800	1	0	0
2	996	976	800	1	0	122
3	112	2044 - 2*976 = 92	800	0	0	244
4						

Θέμα 4 (15%)

Θεωρείστε το διπλανό σενάριο, με τέσσερις διαφορετικούς εξυπηρετητές συνδεδεμένους σε τέσσερις διαφορετικούς χρήστες, σε τέσσερις διαδρομές τριών βημάτων (hops) η καθεμία. Τα τέσσερα ζευγάρια μοιράζονται ένα κοινό μεσαίο βήμα (hop) με χωρητικότητα μετάδοσης $R = 80$ Mbps. Οι τέσσερις συνδέσεις από τους εξυπηρετητές προς τον κοινόχρηστο σύνδεσμο έχουν χωρητικότητα μετάδοσης $R_s = 50$ Mbps. Κάθε ένας από τους τέσσερις συνδέσμους από τον κοινόχρηστο μεσαίο σύνδεσμο σε έναν χρήστη έχει χωρητικότητα μετάδοσης $R_c = 15$ Mbps.



(α) Ποια είναι η τιμή ρυθμού αποστολής δεδομένων (σε Mbps) από τους εξυπηρετητές για καθένα από τα τέσσερα ζεύγη χρήστη-εξυπηρετητή χωρίς να εμφανίζεται συμφόρηση στη διαδρομή, υποθέτοντας ότι η μεσαία ζεύξη είναι δίκαιη (fair) και κατανέμει τον ρυθμό μετάδοσης εξίσου μεταξύ των τεσσάρων ζευγών; Εξηγήστε την απάντησή σας.

Bottleneck analysis:

Για κάθε connection, εξετάζουμε τα 3 links:

1) Server link: $R_s = 50$ Mbps

2) Shared link (fair sharing): $R/4 = 80/4 = 20$ Mbps per flow

3) Client link: $R_c = 15$ Mbps

Μέγιστος ρυθμός χωρίς συμφόρηση: $x = \min(R_s, R/4, R_c) = \min(50, 20, 15) = 15$ Mbps $\rightarrow$ Bottleneck: Client link

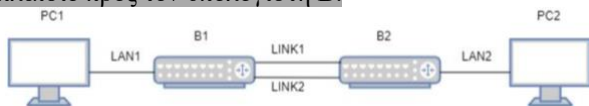
Ο ρυθμός αποστολής για κάθε ζεύγος = **15 Mbps**. Το client link περιορίζει τον throughput. Αν οι servers στέλνουν με >15 Mbps, θα δημιουργηθεί queue/congestion στο client link. Για να αποφευχθεί συμφόρηση, κάθε server πρέπει να στέλνει maximum 15 Mbps.

(β) Αν υποθέσουμε ότι οι αποστολείς (εξυπηρετητές) στέλνουν με τον μέγιστο δυνατό ρυθμό, ποιοι είναι οι βαθμοί χρησιμοποίησης των συνδέσμων αποστολέα-μεσαίου συνδέσμου (R_s), του μεσαίου συνδέσμου (R) και των συνδέσμων μεσαίου συνδέσμου-χρηστών (R_c);

Τύπος συνδέσμου	Χωρητικότητα (Mbps)	Ρυθμός που μεταφέρεται (Mbps)	Βαθμός χρησιμοποίησης
R_s (server $\rightarrow$ μεσαίος)	50	15	$(15/50 = 0.30) \rightarrow 30\%$
R (κοινός μεσαίος)	80	60 ($4 \cdot 15$)	$(60/80 = 0.75) \rightarrow 75\%$
R_c (μεσαίος $\rightarrow$ χρήστης)	15	15	$(15/15 = 1.00) \rightarrow 100\%$

Θέμα 5 (10%)

Τα R1 και R2 στο σχήμα είναι γέφυρες (μεταγωγείς επιπέδου ζεύξης), αλλά δεν έχουν ενεργοποιημένο το πρωτόκολλο επικαλύπτοντος δένδρου (spanning tree protocol) για την αποφυγή σχηματισμού βρόχων. Αρχικά, οι πίνακες ARP των υπολογιστών είναι κενές, όπως και οι πίνακες προώθησης των γεφυρών. Περιγράψτε τη ροή των πλαισίων όταν ο υπολογιστής A στέλνει ένα πλαίσιο προς τον υπολογιστή B.



- Αρχικά, οι πίνακες ARP των υπολογιστών και οι πίνακες προώθησης των γεφυρών R1 και R2 είναι κενές. Ο υπολογιστής A, μη γνωρίζοντας τη MAC διεύθυνση του B, εκπέμπει ARP Request ως broadcast πλαίσιο.
- Οι γέφυρες R1 και R2, έχοντας άδειους πίνακες και χωρίς ενεργοποιημένο το Spanning Tree Protocol, προωθούν το broadcast μέσω flooding από όλες τις θύρες εκτός της εισερχόμενης. Λόγω της ύπαρξης φυσικού βρόχου στην τοπολογία, το πλαίσιο κυκλοφορεί επανειλημμένα στο δίκτυο, προκαλώντας broadcast storm. Παράλληλα, οι γέφυρες ενημερώνουν τους πίνακες προώθησης τους με ασυνεπή τρόπο (MAC table flapping), καθώς η MAC διεύθυνση του A «μαθαίνεται» από διαφορετικές θύρες.
- Ο υπολογιστής B ενδέχεται να λάβει το ARP Request και να απαντήσει με ARP Reply (unicast) προς τον A. Ωστόσο, λόγω της αστάθειας των πινάκων των γεφυρών και της συνεχούς ανακυκλοφορίας πλαισίων, το πλαίσιο απάντησης είτε θα προωθηθεί λανθασμένα είτε θα χαθεί μέσα στη συμφόρηση.
- Συνεπώς, η απουσία του Spanning Tree οδηγεί σε βρόχο επιπέδου ζεύξης, καταιγισμό πλαισίων και αποσταθεροποίηση των πινάκων προώθησης, καθιστώντας την επικοινωνία από τον A προς τον B αναξιόπιστη ή πρακτικά αδύνατη.

Θέμα 6 (20%)

Τη χρονική στιγμή 0, αποστολέας TCP Reno ξεκινάει μια σύνδεση με διαφημιζόμενο παράθυρο 64 KB και MSS = 1 KB. Μόλις εγκατασταθεί η σύνδεση, ο αποστολέας TCP αρχίζει να στέλνει δεδομένα.

(α) Εάν το RTT είναι 200 ms και υποθέτοντας ότι δεν χάνεται κανένα τεμάχιο ή ότι δεν εμφανίζονται λήξεις χρονομέτρου, ποια χρονική στιγμή το παράθυρο συμφόρησης θα είναι ίσο με 16 KB;

RTT	cwnd	ssthresh	Σχόλια
0	1	64	Έναρξη – Slow Start
1	2	64	Slow Start (διπλασιασμός)

2	4	64	Slow Start
3	8	64	Slow Start
4	16	64	Slow Start

$Cwnd(4)=16$: $t = 4 * RTT = 4 * 0.2 = 0.8 \text{ sec}$

(β) Αμέσως μετά εμφανίζεται **λήξη χρονομέτρου**. Πόσο χρόνο μετά την εκπνοή του χρονομέτρου θα γίνει το παράθυρο συμφόρησης ίσο με 14 KB, εάν ο αποστολέας συνεχίζει να στέλνει δεδομένα στην εγκατεστημένη σύνδεση και δεν υπάρχουν επιπλέον απώλειες ή λήξεις χρονομέτρων;

Μετά το timeout:

RTT	cwnd	ssthresh	Σχόλια
0	1	8	Slow Start
1	2	8	Slow Start
2	4	8	Slow Start
3	8	8	Τέλος Slow Start ($cwnd = ssthresh$)
4	9	8	Congestion Avoidance
5	10	8	Congestion Avoidance
6	11	8	Congestion Avoidance
7	12	8	Congestion Avoidance
8	13	8	Congestion Avoidance
9	14	8	Congestion Avoidance

$Cwnd(9)=14$: $t = 9 * RTT = 1.8 \text{ sec}$

(γ) Ενώ το παράθυρο συμφόρησης παραμένει στην προηγούμενη τιμή, ο αποστολέας λαμβάνει τρεις ταυτόσημες επαληθεύσεις. Πόσο χρόνο μετά τη λήψη της τρίτης επαλήθευσης θα γίνει το παράθυρο συμφόρησης ίσο με 12 KB, εάν ο αποστολέας συνεχίζει να στέλνει δεδομένα στην εγκατεστημένη σύνδεση και δεν υπάρχουν επιπλέον απώλειες ή λήξεις χρονομέτρων;

Μετά από triple duplicate ACK:

RTT	cwnd	ssthresh	Σχόλια
0	10	7	Fast Retransmit / Fast Recovery
1	7	7	Congestion Avoidance
2	8	7	Congestion Avoidance
3	9	7	Congestion Avoidance
4	10	7	Congestion Avoidance
5	11	7	Congestion Avoidance
6	12	7	Congestion Avoidance

$Cwnd(6)=12$: $t = 6 * RTT = 1.2 \text{ sec}$

Κανονική 2025

Θέμα 1 (10%)

Ας υποθέσουμε ότι μεταξύ ενός υπολογιστή A και ενός υπολογιστή B τα **πακέτα IP** περιορίζονται στα 1500 bytes (συμπεριλαμβανομένης της επικεφαλίδας). Υποθέτοντας επικεφαλίδα IP 20 bytes και TCP 20 bytes, πόσα πακέτα θα απαιτηθούν για την αποστολή ενός αρχείου MP3 μεγέθους 5 εκατομμυρίων bytes; Ποιο θα είναι το μήκος (σε bytes) κάθε πακέτου;

Δεδομένα:

MTU (Maximum Transmission Unit) = 1500 bytes

IP header = 20 bytes

TCP header = 20 bytes

File size = 5,000,000 bytes

Υπολογισμοί:

Payload per packet = MTU - IP header - TCP header = 1500 - 20 - 20 = 1460 bytes

Number of packets = [File size / Payload] = [5,000,000 / 1460] = **3425 packets**

3424 πρώτα πακέτα: Total length = IP header + TCP header + Payload per packet = 20 + 20 + 1460 = 1500 bytes

Τελευταίο πακέτο: Total length = 20 + 20 + (5,000,000 - 3424*1460) = 20 + 20 + 960 = 1000 bytes

Θέμα 2 (20%)

Δίνεται η τοπολογία του σχήματος. Να γίνει ανάθεση διευθύνσεων IP στα έξι υποδίκτυα, με τους ακόλουθους περιορισμούς:

- Όλες οι διευθύνσεις IP πρέπει να ανήκουν στο δίκτυο 214.97.254.0/23.
- Το υποδίκτυο A πρέπει να διαθέτει επαρκή αριθμό διευθύνσεων ώστε να υποστηρίζει τουλάχιστον 250 διεπαφές.
- Το υποδίκτυο B πρέπει να διαθέτει επαρκή αριθμό διευθύνσεων ώστε να υποστηρίζει τουλάχιστον 120 διεπαφές.
- Το υποδίκτυο C πρέπει να διαθέτει επαρκή αριθμό διευθύνσεων ώστε να υποστηρίζει τουλάχιστον 60 διεπαφές.
- Τα υποδίκτυα D, E και F πρέπει να μπορούν να υποστηρίξουν τουλάχιστον δύο διεπαφές το καθένα.

Ανάλυση

214.97.254.0/23 → 32-23 = 9 bits for hosts → $2^9 = 512$ total addresses

Range: 214.97.254.0 - 214.97.255.255

Usable hosts = $2^n - 2$ (network + broadcast excluded) → prefix = /(32-n)

Υποδίκτυο	Απαιτούμενοι hosts	Υπολογισμός	Prefix	Network Addresses	Εύρος διευθύνσεων
A	≥ 250	$2^8 - 2 = 254$	/24	214.97.254.0/24	254.0 – 254.255
B	≥ 120	$2^7 - 2 = 126$	/25	214.97.255.0/25	255.0 – 255.127
C	≥ 60	$2^6 - 2 = 62$	/26	214.97.255.128/26	255.128 – 255.191
D	≥ 2	$2^2 - 2 = 2$	/30	214.97.255.192/30	255.192 – 255.195
E	≥ 2	$2^2 - 2 = 2$	/30	214.97.255.196/30	255.196 – 255.199
F	≥ 2	$2^2 - 2 = 2$	/30	214.97.255.200/30	255.200 – 255.203

Θέμα 3 (10%)

Υποθέστε ότι ο υπολογιστής A εκπέμπει ένα ARP request με σκοπό να εντοπίσει τον υπολογιστή B. Ωστόσο, στο ARP request που αποστέλλει, ο A δηλώνει ως MAC διεύθυνση αποστολέα τη MAC διεύθυνση του υπολογιστή C, αντί για τη δική του.

α) Θα λάβει ο υπολογιστής A απάντηση (ARP reply) από τον B;

OXI

	ARP Request (A → broadcast)	ARP Reply (B → C)
Source MAC	MAC C (spoofed)	MAC B
Source IP	IP A	IP B
Target MAC	ff:ff:ff:ff:ff:ff	MAC C
Target IP	IP B	IP A

Όταν ο B λάβει το ARP request: Ενημερώνει τον ARP table του: IP_A → MAC_C

Ο A δεν θα λάβει την απάντηση γιατί το frame προορίζεται για MAC_C.

β) Αν οι πίνακες ARP των υπολογιστών A, B και C ήταν αρχικά κενοί πριν από την αποστολή του ARP request από τον A, να αναφέρετε το περιεχόμενο των πινάκων ARP μετά την ολοκλήρωση της ανταλλαγής ARP μηνυμάτων.

ARP tables:

A	IP	MAC	B	IP	MAC	C	IP	MAC
				IP_A	MAC_C		IP_B	MAC_B

Θέμα 4 (20%)

(α) Στο πρωτόκολλο CSMA/CD, μετά την 5η σύγκρουση, ποια είναι η πιθανότητα ένας κόμβος να επιλέξει τιμή K = 4 στο δυαδικό εκθετικό backoff; Αν τελικά επιλέξει K = 4, ποια είναι η καθυστέρηση επαναμετάδοσης του πλαισίου σε δίκτυο Ethernet 10 Mbps;

Στο binary exponential backoff:

Μετά από n συγκρούσεις: $K \in \{0, 1, 2, \dots, 2^n - 1\}$

Για $n = 5$: $K \in \{0, 1, 2, \dots, 2^5 - 1\} = \{0, 1, \dots, 31\} \rightarrow P(K=4) = 1/32 = 0.03125 = 3.125\%$

Retransmission delay για $K=4$: Delay = $K \times 512 = 2048$ bit times, Bit time (10 Mbps) = $1 / 10^7 \text{ s} = 0.1 \mu\text{s}$

Delay = $2048 \times 0.1 \mu\text{s} = 204.8 \mu\text{s}$

(β) Έστω ότι δύο κόμβοι ξεκινούν να μεταδίδουν ταυτόχρονα ένα πλαίσιο μήκους L σε ένα κοινό κανάλι εκπομπής με ρυθμό μετάδοσης R . Έστω d η καθυστέρηση διάδοσης του καναλιού. Θα υπάρξει σύγκρουση αν ισχύει $d < L / R$;

ΝΑΙ

Αφού οι δύο κόμβοι ξεκινούν να μεταδίδουν ταυτόχρονα, τα σήματά τους διαδίδονται στο μέσο και επικαλύπτονται. Εφόσον ισχύει $d < L / R$ ο χρόνος διάδοσης είναι μικρότερος από τον χρόνο μετάδοσης του πλαισίου, άρα κάθε κόμβος θα δει το σήμα του άλλου ενώ ακόμη μεταδίδει. Συνεπώς η σύγκρουση όχι μόνο συμβαίνει, αλλά και ανιχνεύεται.

(γ) Μια εφαρμογή UDP στέλνει ένα πακέτο IPv4, το οποίο θρυμματίζεται σε 4 θραύσματα. Υποθέστε ότι τα θραύσματα 1, 3 και 4 φτάνουν στον προορισμό, ενώ το θραύσμα 2 χάνεται. Μετά από λίγο, ο αποστολέας επαναμεταδίδει το πακέτο με ακριβώς τον ίδιο θρυμματισμό. Αυτή τη φορά χάνεται το θραύσμα 1.

Αν ο χρονομετρητής επανασυναρμολόγησης του πρώτου πακέτου στον προορισμό δεν έχει λήξει όταν φτάσει το θραύσμα 2 της δεύτερης μετάδοσης, μπορεί ο δέκτης να επανασυναρμολογήσει το πακέτο IPv4;

ΟΧΙ

Η επανασυναρμολόγηση θραυσμάτων στο IPv4 γίνεται με βάση τα πεδία της επικεφαλίδας: Source IP, Destination IP, Protocol και Identification. Όλα τα θραύσματα που ανήκουν στο ίδιο αρχικό datagram έχουν την ίδια τιμή Identification. Όταν το UDP πακέτο επαναστέλνεται, δημιουργείται νέο IPv4 datagram, το οποίο (κανονικά) έχει διαφορετική τιμή Identification. Επομένως, το θραύσμα 2 της δεύτερης μετάδοσης δεν θεωρείται ότι ανήκει στο ίδιο datagram με τα θραύσματα 1, 3 και 4 της πρώτης μετάδοσης και δεν μπορεί να καλύψει το χαμένο θραύσμα 2 του πρώτου πακέτου. Έτσι, η επανασυναρμολόγηση αποτυγχάνει.

(δ) Ένας δρομολογητής BGP επιλέγει πάντα τη διαδρομή χωρίς βρόχο που έχει τον μικρότερο αριθμό αυτόνομων συστημάτων (AS);

ΟΧΙ

Το BGP χρησιμοποιεί route selection algorithm με συγκεκριμένη σειρά προτεραιότητας:

1. Highest Local Preference (local policy)
2. Shortest AS-PATH
3. Πλησιέστερος next hop router

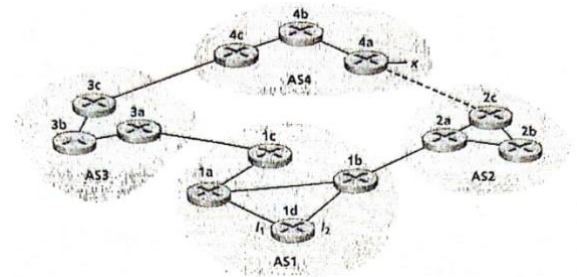
Το AS-PATH length είναι 2ος κανόνας, όχι πρώτος. Το Local Preference έχει υψηλότερη προτεραιότητα.

Θέμα 5 (15%)

Θεωρήστε το δίκτυο που φαίνεται στο παρακάτω σχήμα. Υποθέστε ότι:

- Τα αυτόνομα συστήματα AS3 και AS2 χρησιμοποιούν το OSPF ως intra-AS πρωτόκολλο δρομολόγησης.
- Τα αυτόνομα συστήματα AS1 και AS4 χρησιμοποιούν το RIP ως intra-AS πρωτόκολλο δρομολόγησης.
- Το κόστος διαδρομής στο εσωτερικό ενός AS υπολογίζεται με βάση τον αριθμό των hops που απαρτίζουν τη διαδρομή.
- Τα πρωτόκολλα eBGP και iBGP χρησιμοποιούνται ως inter-AS πρωτόκολλα δρομολόγησης.

Αρχικά, υποθέστε ότι δεν υπάρχει φυσική σύνδεση μεταξύ των αυτόνομων συστημάτων AS2 και AS4.



(α) Μέσω ποιου πρωτοκόλλου (OSPF, RIP, eBGP ή iBGP) μαθαίνουν οι δρομολογητές 3c, 3a, 1c και 1d το πρόθεμα x;

Δεδομένα

AS3, AS2: OSPF (intra-AS)

AS1, AS4: RIP (intra-AS)

Cost metric: hop count

BGP (eBGP + iBGP): inter-AS

Prefix x originates at AS4

Router 3c: AS4 → AS3 (μέσω 4c-3c link), Πρωτόκολλο: eBGP (από 4c προς 3c)

Router 3a: Μαθαίνει από 3c (εντός του AS3), Πρωτόκολλο: iBGP (internal BGP session 3c → 3a)

Router 1c: AS4 → AS3 → AS1, Μαθαίνει από 3a (inter-AS connection), Πρωτόκολλο: eBGP (από 3a προς 1c)

Router 1d: Μαθαίνει από 1c (εντός του AS1), Πρωτόκολλο: iBGP (internal BGP session 1c → 1d)

(β) Όταν ο δρομολογητής 1d μάθει το πρόθεμα x, εισάγει στον πίνακα προώθησής του μία εγγραφή της μορφής (x, I). Θα είναι το I ίσο με I₁ ή με I₂;

Το 1d μαθαίνει το X με iBGP μέσω του border router 1c. Η επιλογή της πραγματικής εξόδου από το AS1 γίνεται με βάση το intra-AS πρωτόκολλο δρομολόγησης, το οποίο στο AS1 είναι το RIP.

I₁: 1d → 1a → 1c ⇒ 2 hops

I₂: 1d → 1b → 1a → 1c ⇒ 3 hops

Το **I = I₁**, επειδή το intra-AS πρωτόκολλο (RIP) επιλέγει τη διαδρομή με το μικρότερο κόστος προς το BGP next-hop.

(γ) Τώρα υποθέστε ότι υπάρχει φυσική σύνδεση μεταξύ των αυτόνομων συστημάτων AS2 και AS4. Ο δρομολογητής 1d μαθαίνει ότι το πρόθεμα x είναι προσβάσιμο τόσο μέσω του AS2 όσο και μέσω του AS3. Σε αυτή την περίπτωση, θα είναι το I ίσο με I₁ ή με I₂;

Τώρα το AS1 μαθαίνει το πρόθεμα x από δύο εξωτερικές πύλες:

- μέσω AS3: AS1 ←(eBGP)– 3a ... AS3 ... AS4 → x (είσοδος στο AS1 από τον 1c)
- μέσω AS2: AS1 ←(eBGP)– 2a ... AS2 ... AS4 → x (είσοδος στο AS1 από τον 1b)

BGP επιλογή (στο AS1):

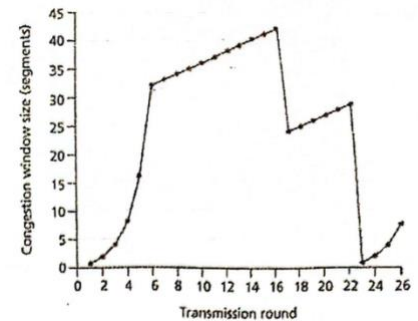
Αφού δεν δίνονται πολιτικές (LOCAL_PREF κτλ.), θεωρούμε ίσες. Τότε:

το AS-PATH και για τις δύο διαδρομές έχει ίδιο μήκος (AS2–AS4 vs AS3–AS4) ⇒ ισοπαλία,

άρα εφαρμόζεται το **hot-potato routing**: επιλέγεται η έξοδος από το AS που είναι πιο κοντά (μικρότερο intra-AS κόστος / hops) προς το BGP next-hop.

Intra-AS κόστος (RIP στο AS1):

- προς 1b (έξοδος προς AS2): 1d → 1b = 1 hop ⇒ interface I₂
 - προς 1c (έξοδος προς AS3): 1d → 1a → 1c = 2 hops ⇒ interface I₁ Επειδή 1 hop
- Άρα στο (x, I) ισχύει: I=I₂ (έξοδος προς 1b/AS2).



Θέμα 6 (25%)

Υποθέστε ότι το TCP Reno συμπεριφέρεται όπως φαίνεται στο διάγραμμα, για μία συγκεκριμένη χρονική περίοδο. Με βάση το διάγραμμα, να απαντήσετε με κατάλληλη αιτιολόγηση στα παρακάτω ερωτήματα:

TCP Reno

RTT	cwnd	ssthresh	Σχόλια
1	1	32	Έναρξη – Slow Start
2	2	32	Slow Start (διπλασιασμός)
3	4	32	Slow Start
4	8	32	Slow Start
5	16	32	Slow Start
6	32	32	Τέλος Slow Start (cwnd = ssthresh)
7	33	32	Congestion Avoidance
8	34	32	Congestion Avoidance
9	35	32	Congestion Avoidance
10	36	32	Congestion Avoidance
11	37	32	Congestion Avoidance
12	38	32	Congestion Avoidance
13	39	32	Congestion Avoidance
14	40	32	Congestion Avoidance
15	41	32	Congestion Avoidance
16	42	32	Απώλεια με triple dupACK → Fast Retransmit / Fast Recovery
17	24	21	Congestion Avoidance
18	25	21	Congestion Avoidance
19	26	21	Congestion Avoidance
20	27	21	Congestion Avoidance
21	28	21	Congestion Avoidance
22	29	21	Απώλεια με timeout → επιστροφή σε Slow Start
23	1	14	Slow Start
24	2	14	Slow Start
25	4	14	Slow Start
26	8	14	Slow Start
27	14	14	Τέλος Slow Start (cwnd > ssthresh)
28	15	14	Congestion Avoidance
29	16	14	Congestion Avoidance
30	17	14	Congestion Avoidance

(α) Σε ποια χρονικά διαστήματα, εφόσον υπάρχουν, το TCP Reno λειτουργεί στη φάση της αργής εκκίνησης (slow start); Σε ποια χρονικά διαστήματα, εφόσον υπάρχουν, λειτουργεί στη φάση της αποφυγής συμφόρησης (congestion avoidance);

Slow Start: 1-6, 23-27

Congestion Avoidance: 7-16, 17-28, 28-30

(β) Μετά τον 16ο γύρο μετάδοσης (transmission round) ανιχνεύεται απώλεια πακέτου. Η απώλεια αυτή ανιχνεύεται λόγω λήψης τριπλού ταυτόσημου ACK ή λόγω timeout; Αντίστοιχα, πώς ανιχνεύεται η απώλεια πακέτου μετά τον 22ο γύρο μετάδοσης;

RTT = 16: ssthresh ← cwnd/2=21, cwnd ← ssthresh+3=24 → triple dupACK (Fast Retransmit/ Fast Recovery)

RTT = 22: ssthresh ← cwnd/2=14, cwnd ← 1 → timeout

(γ) Ποια είναι η τιμή του ssthresh: κατά την αρχική μετάδοση, και κατά τον 20ό γύρο μετάδοσης;

RTT = 0: ssthresh = 32

RTT = 20: ssthresh = 21

Αλλαγή σε TCP Tahoe

TCP Tahoe

RTT	cwnd	ssthresh	Σχόλια
1	1	32	Έναρξη – Slow Start
2	2	32	Slow Start
3	4	32	Slow Start
4	8	32	Slow Start
5	16	32	Slow Start
6	32	32	Τέλος Slow Start
7	33	32	Congestion Avoidance
8	34	32	Congestion Avoidance
9	35	32	Congestion Avoidance
10	36	32	Congestion Avoidance
11	37	32	Congestion Avoidance
12	38	32	Congestion Avoidance
13	39	32	Congestion Avoidance
14	40	32	Congestion Avoidance
15	41	32	Congestion Avoidance
16	42	32	Triple dup ACK → Tahoe reset
17	1	21	Slow Start
18	2	21	Slow Start
19	4	21	Slow Start
20	8	21	Slow Start
21	16	21	Slow Start
22	21	21	Τέλος Slow Start (cwnd > ssthresh), Timeout → reset
23	1	10	Slow Start
24	2	10	Slow Start
25	4	10	Slow Start
26	8	10	Slow Start
27	10	10	Τέλος Slow Start (cwnd > ssthresh)
28	11	10	Congestion Avoidance
29	12	10	Congestion Avoidance
30	13	10	Congestion Avoidance

(δ) Ας υποθέσουμε τώρα ότι αντί για TCP Reno χρησιμοποιείται TCP Tahoe, και ότι στον 16ο γύρο μετάδοσης λαμβάνεται τριπλό ταυτόσημο ACK. Ποιες θα είναι οι τιμές του ssthresh και του παραθύρου συμφόρησης (cwnd) στον 17ο γύρο μετάδοσης;

RTT = 16: ssthresh ← cwnd/2=21, cwnd ← 1 → triple dupACK (Tahoe Reset)

(ε) Ας υποθέσουμε ξανά ότι χρησιμοποιείται TCP Tahoe και, σε συνέχεια του προηγούμενου ερωτήματος, συμβαίνει ένα γεγονός timeout στον 22ο γύρο μετάδοσης, αφού στον 16ο γύρο έχει ληφθεί τριπλό ταυτόσημο ACK. Πόσα πακέτα έχουν σταλεί από τον 17ο έως και τον 22ο γύρο μετάδοσης, συμπεριλαμβανομένων και των γύρων 17 και 22;

RTT=17-22: packets = Σcwnd(i) = 1+2+4+8+16+21 = 52