Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχ. και Μηχανικών Υπολογιστών
Εργαστήριο Υπολογιστικών Συστημάτων

Δίκτυα Διασύνδεσης

Συστήματα Παράλληλης Επεξεργασίας

9ο Εξάμηνο

National Technical University of Athens
CSLab

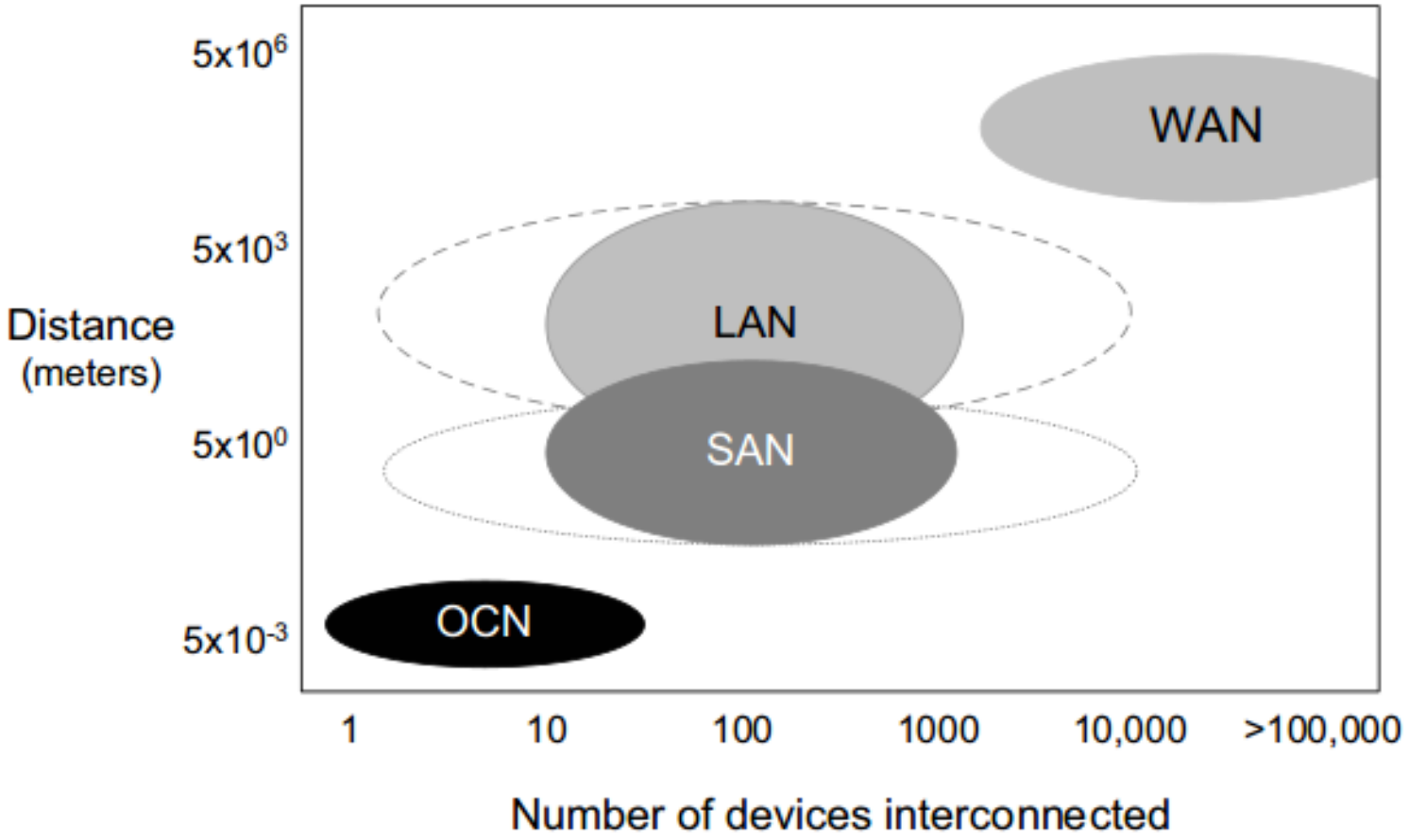
Δίκτυα διασύνδεσης

- Διασυνδέουν δομικές μονάδες ενός σύνθετου συστήματος
- **On-Chip Network (OCN) or Network-on-Chip (NoC):**
  - Caches
  - Processing cores
  - CMPs.
- **System/Storage Area Networks (SAN):**
  - Επεξεργαστές με μονάδες μνήμης
  - Υπολογιστές μεταξύ τους
  - Υπολογιστές με συσκευές αποθήκευσης
- **Local Area Networks (LAN):**
  - Υπολογιστές σε ένα τοπικό δίκτυο
- **Wide Area Networks (WAN):**
  - Υπολογιστές σε οποιοδήποτε σημείο του πλανήτη

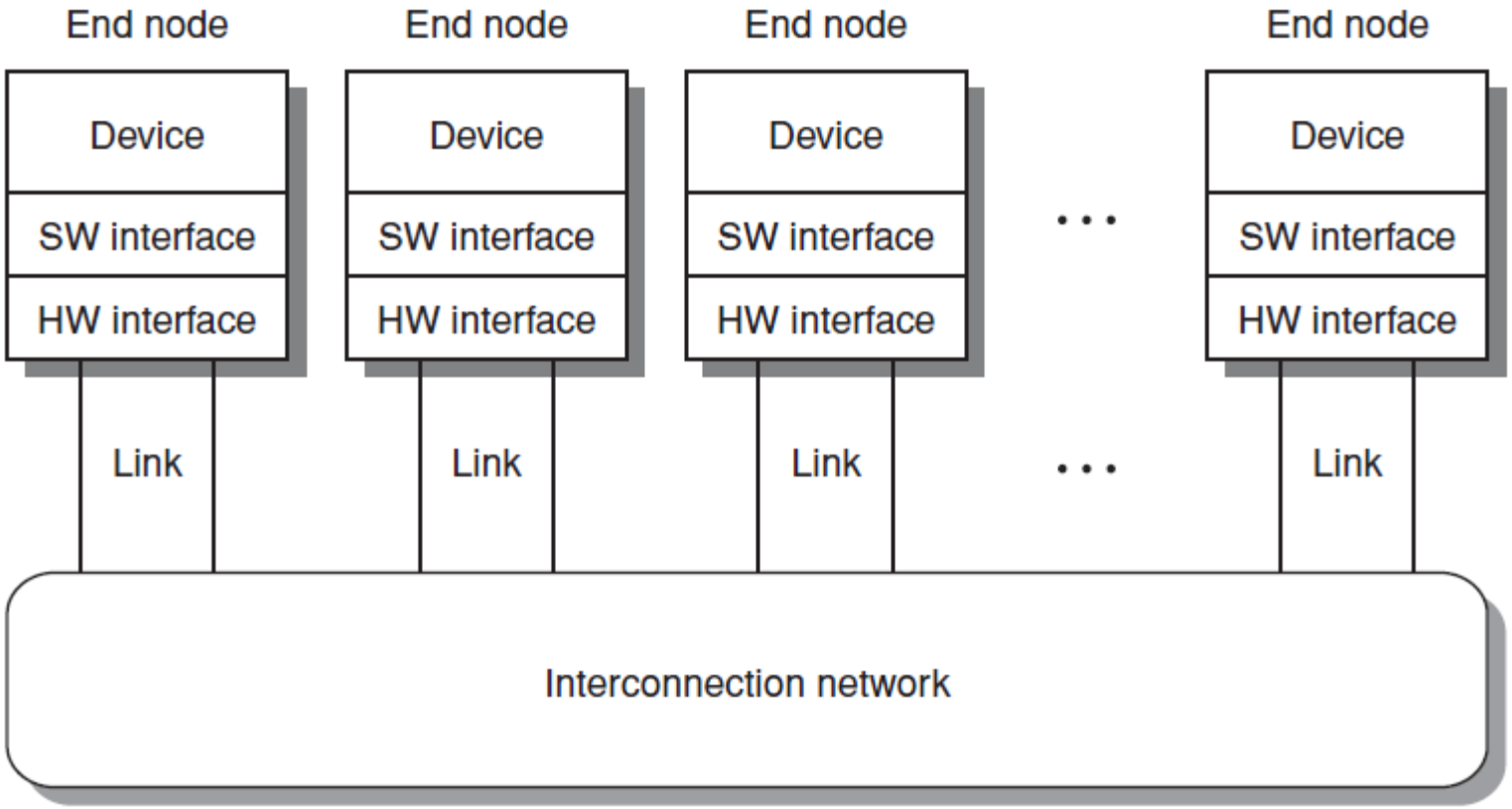
Δίκτυα διασύνδεσης

- Διασυνδέουν δομικές μονάδες ενός σύνθετου συστήματος
- **On-Chip Network (OCN) or Network-on-Chip (NoC):**
  - Caches
  - Processing cores
  - CMPs.
- **System/Storage Area Networks (SAN):**
  - Επεξεργαστές με μονάδες μνήμης
  - Υπολογιστές μεταξύ τους
  - Υπολογιστές με συσκευές αποθήκευσης
- **Local Area Networks (LAN):**
  - Υπολογιστές σε ένα τοπικό δίκτυο
- **Wide Area Networks (WAN):**
  - Υπολογιστές σε οποιοδήποτε σημείο του πλανήτη

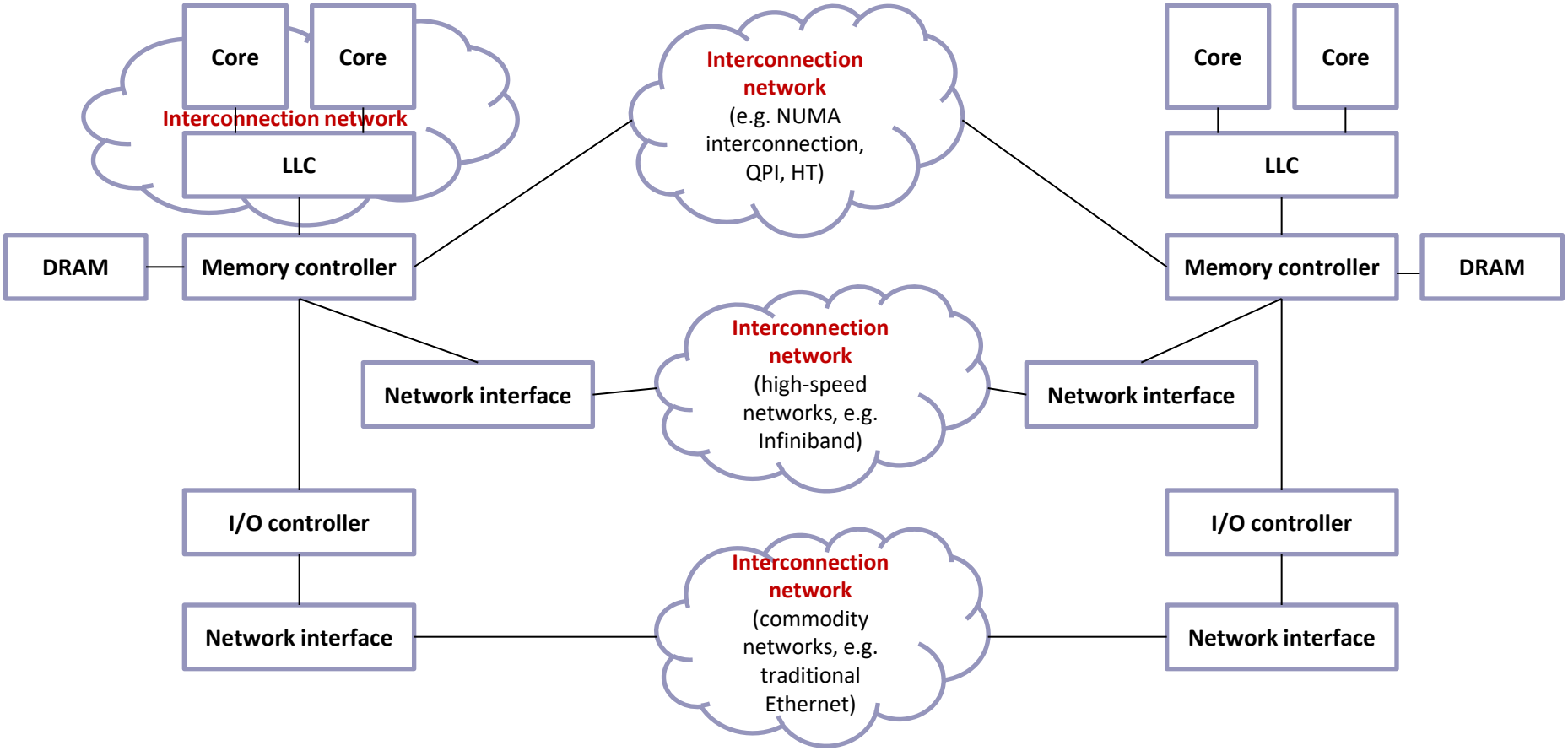
Δίκτυα διασύνδεσης



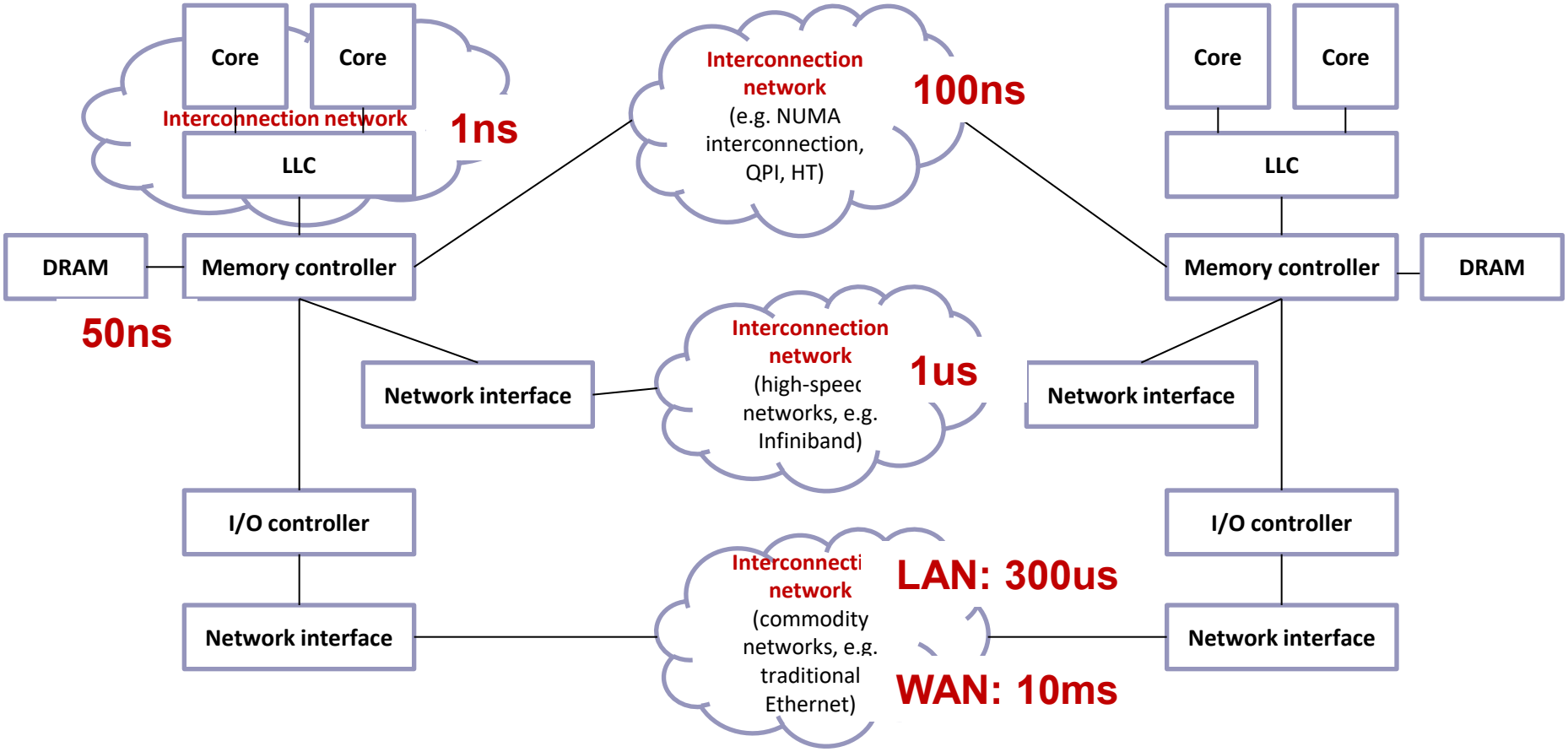
Δίκτυα διασύνδεσης (SAN, LAN και WAN)



Δίκτυα διασύνδεσης



Δίκτυα διασύνδεσης

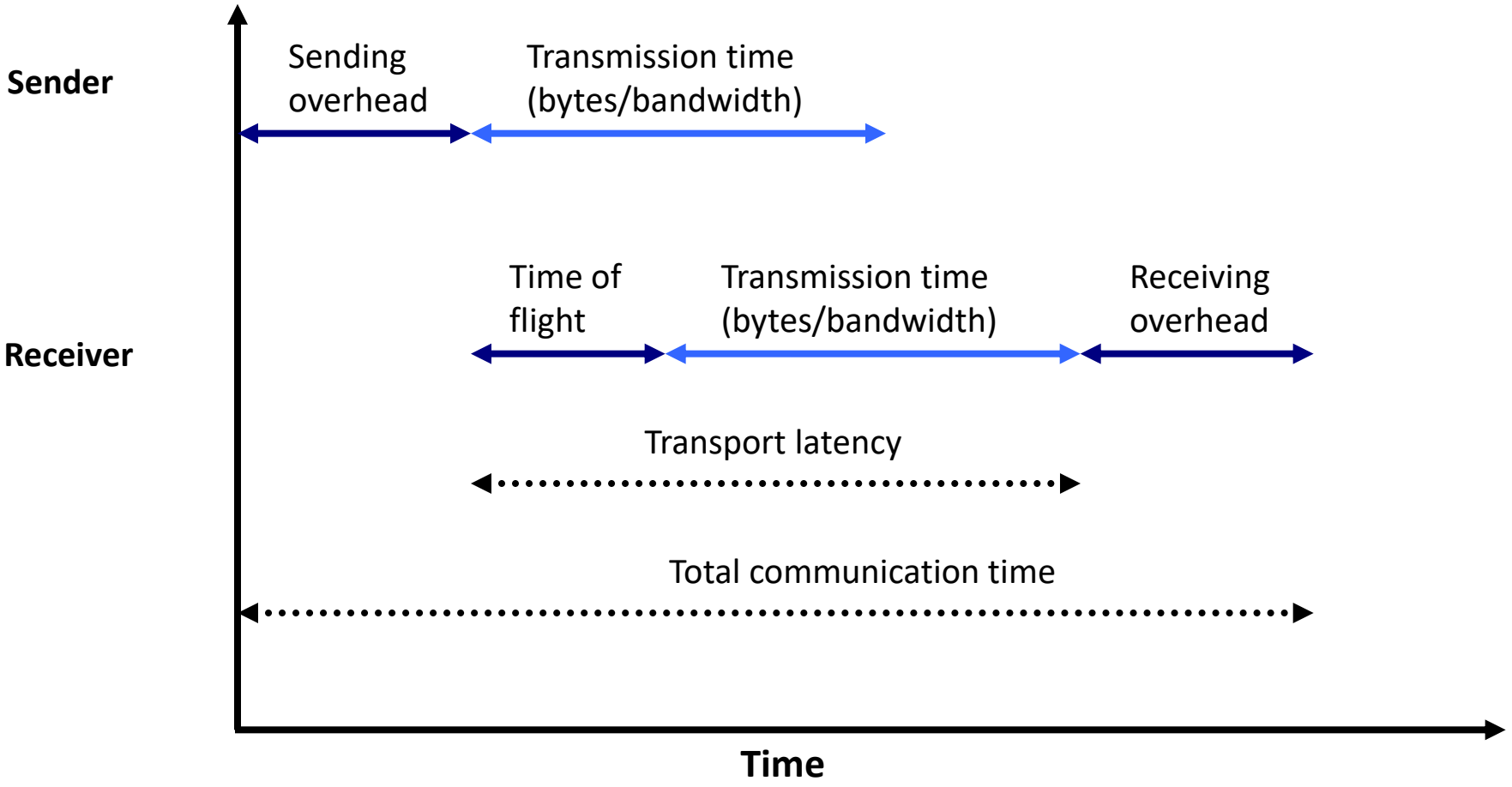


Κρίσιμες μετρικές για την αξιολόγηση ενός δικτύου διασύνδεσης

- **Επίδοση:**
  - **Latency**: Χρόνος που απαιτείται για να φτάσει το πρώτο byte πληροφορίας από τον αποστολέα στον παραλήπτη
  - **Bandwidth:** Ο ρυθμός με τον οποίο μεταδίδεται η πληροφορία
- **Κόστος:**
  - Αριθμός ports στα switches
  - Αριθμός switches
  - Αριθμός συνδέσεων
- **Επεκτασιμότητα (scalability):** Η δυνατότητα του δικτύου να υποστηρίξει επέκταση σε μεγαλύτερο αριθμό διασυνδεόμενων μονάδων

Latency και Bandwidth

simplified



Δομή δικτύου και λειτουργίες

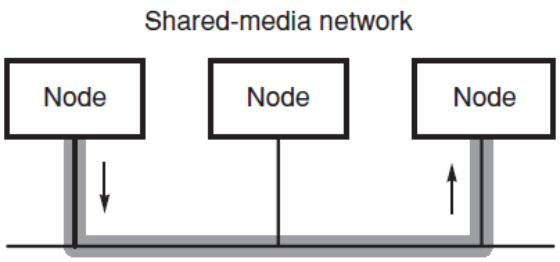
- **Τοπολογία (topology):** *Ποια μονοπάτια είναι δυνατά για την επικοινωνία; (Πώς διασυνδέονται φυσικά οι κόμβοι;)*
- **Δρομολόγηση (routing):** *Ποια από τα δυνατά μονοπάτια είναι επιτρεπτά (έγκυρα) για την επικοινωνία;*
- **Διαιτησία (arbitration):** *Πότε θα είναι διαθέσιμα τα μονοπάτια επικοινωνίας (σε συνθήκες διεκδίκησης ενός μονοπατιού από διαφορετικές λειτουργίες επικοινωνίας)*
- **Μεταγωγή (switching):** *Με ποιο τρόπο θα δοθεί το μονοπάτι σε μια λειτουργία επικοινωνίας;*

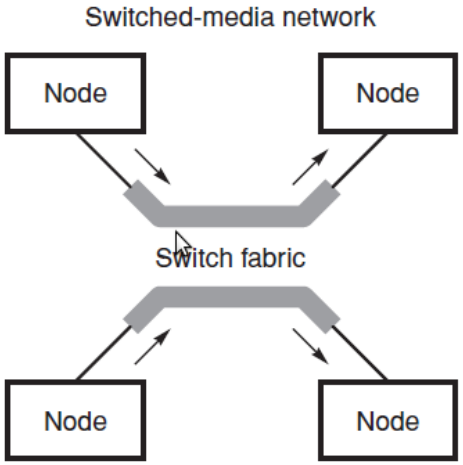
Χαρακτηριστικά τοπολογιών

- **Βαθμός κόμβου (node degree) d:** αριθμός συνδέσμων σε ένα κόμβο
  - Θέλουμε να είναι:
    - μικρός (λόγω κόστους)
    - σταθερός (για επεκτασιμότητα)
- **Διάμετρος δικτύου D:** μέγιστο ελάχιστο μονοπάτι μεταξύ δύο οποιονδήποτε κόμβων
  - Όσο μικρότερη, τόσο καλύτερη η χειρότερη περίπτωση επικοινωνίας
- **Εύρος τομής (bisection width) b:** ο ελάχιστος αριθμός ακμών που κόβουμε, χωρίζοντας το δίκτυο στα δύο
  - Αποτελεί ένα καλό δείκτη του μέγιστου εύρους ζώνης επικοινωνίας σε ένα δίκτυο

Κατηγορίες δικτύων

- **Shared-media networks:** Το μέσο είναι διαμοιραζόμενο από όλους τους κόμβους, π.χ.
  - Δίαυλος (bus) σε μονοεπεξεργαστικά και πολυεπεξεργαστικά συστήματα
  - Το παραδοσιακό Ethernet
- **Switched-media networks:** Υπάρχουν διακοπτόμενα μονοπάτια που μπορούν να υποστηρίξουν την ταυτόχρονη επικοινωνία ανάμεσα σε διαφορετικά ζεύγη κόμβων



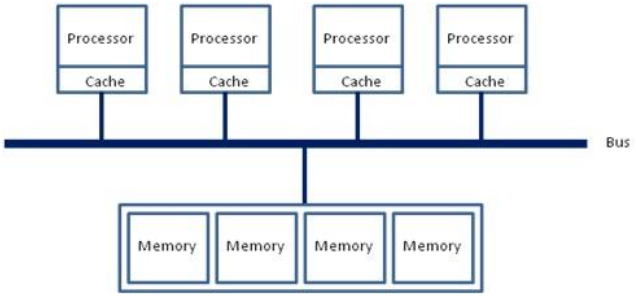


Κατηγορίες δικτύων

- **Shared-media networks:** Το μέσο είναι διαμοιραζόμενο από όλους τους κόμβους
  - Πλεονεκτήματα:
    - Εύκολο στην υλοποίηση
    - Χαμηλό κόστος
  - Μειονεκτήματα:
    - Χαμηλή κλιμάκωση (λόγω bandwidth, διαιτησίας, κλπ)
- **Switched-media networks:** Υπάρχουν διακοπτόμενα μονοπάτια που μπορούν να υποστηρίξουν την ταυτόχρονη επικοινωνία ανάμεσα σε διαφορετικά ζεύγη κόμβων
  - Centralized και distributed switched networks
  - Πλεονεκτήματα:
    - Καλή κλιμάκωση
    - Ευελιξία στο σχεδιασμό
    - Υψηλές επιδόσεις
  - Μειονεκτήματα:
    - Υψηλό κόστος

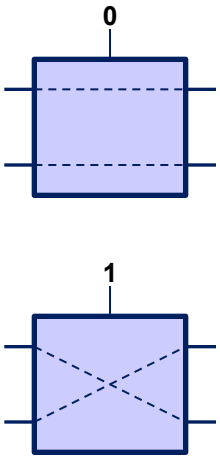
Shared-media networks:
Δίαυλος (bus)

- Παραδοσιακός τρόπος διασύνδεσης σε ένα NoC
- Απλή υλοποίηση με χαμηλό κόστος
  - Data, address, control buses
  - Διαιτησία (arbitration):
    - Κεντρική μέσω του control bus
    - Κατανεμημένη (CSMA/CD, Token Ring)
  - Μεταγωγή (switching)
    - Απλά η συσκευή συνδέεται στο μέσο
  - Δρομολόγηση (routing):
    - Σε όλους τους παραλήπτες (έλεγχος αν το πακέτο προορίζεται για εμένα)
    - Υποστηρίζει εύκολα broadcast και multicast



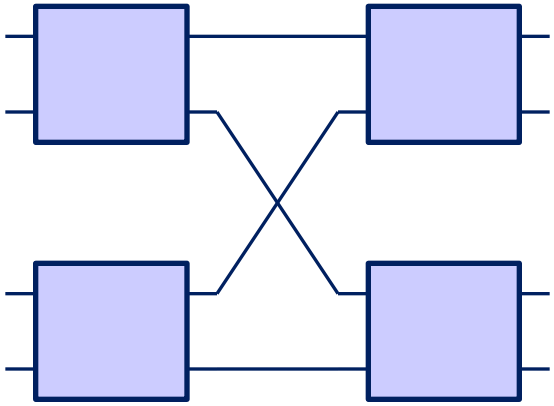
- Εύκολη υλοποίηση cache coherence με snooping
- Αλλά: δεν είναι επεκτάσιμος (τυπικά λίγες δεκάδες στοιχείων)
  - Περιορισμένο συνολικό bandwidth
  - Μεγάλο overhead στη διαιτησία για μεγάλο αριθμό κόμβων

Διακόπτες

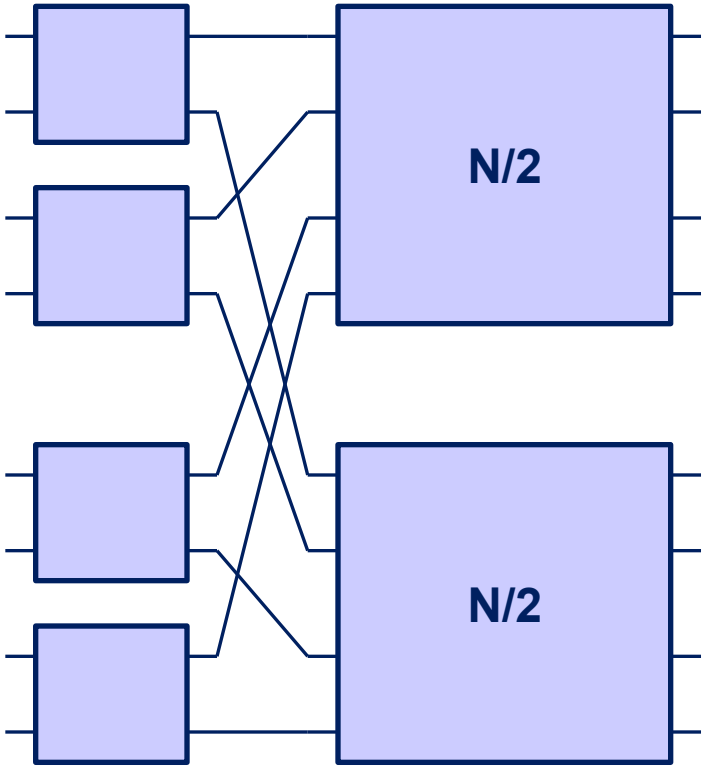


Βασικό building block
2 x 2 διακόπτης
(switching cell)

2 λειτουργίες:
“through” / “crossed”



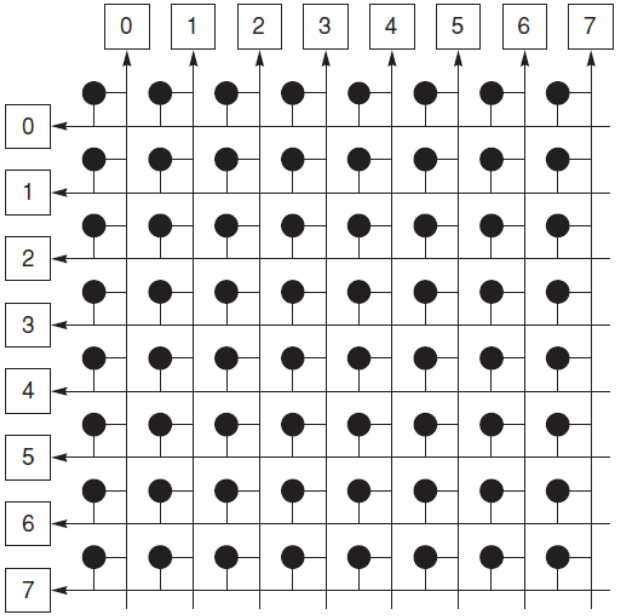
Κατασκευή 4 x 4
διακόπτη από 2 x 2



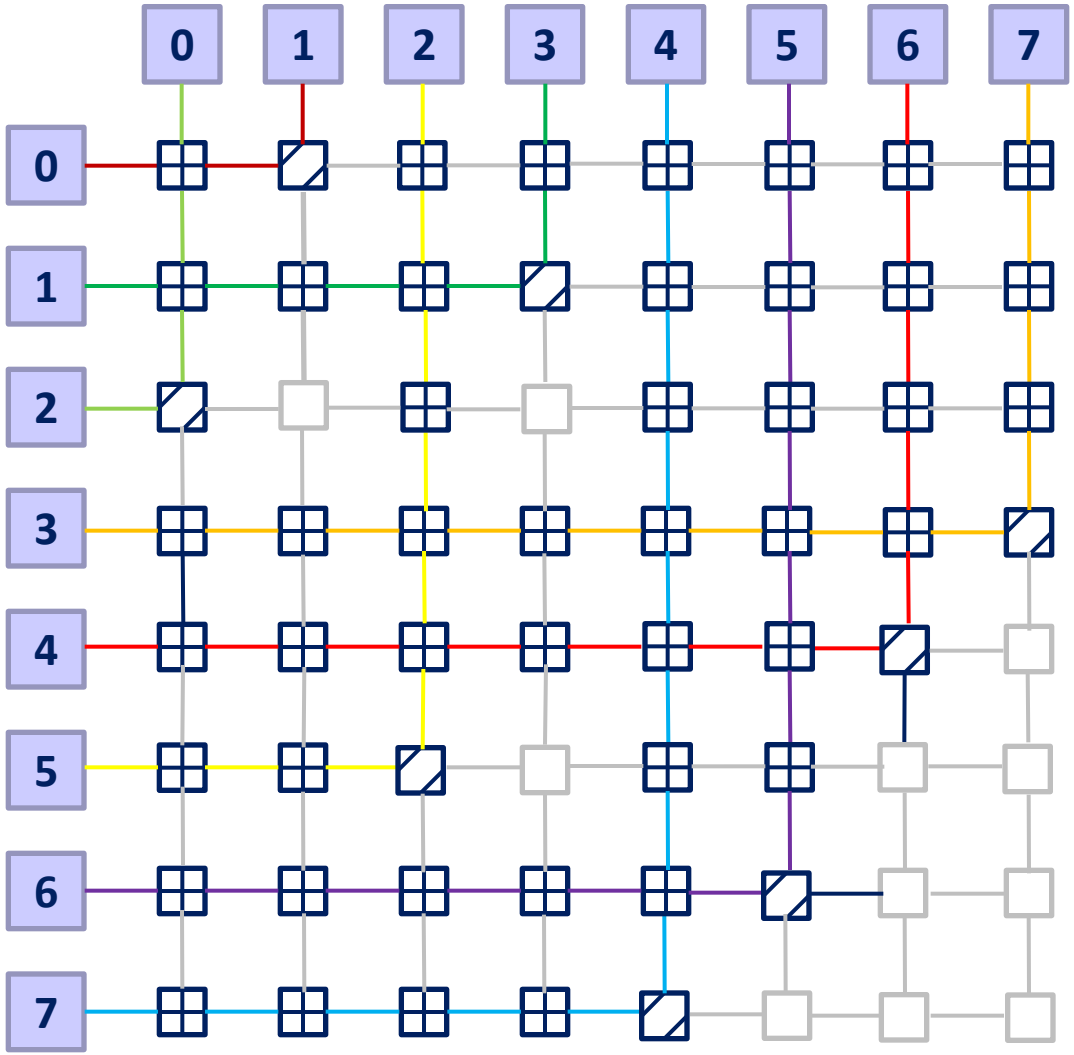
Γενίκευση: Αναδρομική κατασκευή N x N
διακόπτη από 2 N/2 x N/2 διακόπτες και
2 x 2 διακόπτες

Centralized switched networks:
Crossbar switch

- Απλούστερη, ταχύτερη αλλά και ακριβότερη λύση για τη διασύνδεση N στοιχείων
- Υποστηρίζει ταυτόχρονη επικοινωνία διαφορετικών ζευγών πηγής - προορισμού
- Απαιτεί $\mathbf{N^2}$ διακόπτες, δεν κλιμακώνει λόγω κόστους
- Χρησιμοποιείται σε NoC και routers (switches) για τη διασύνδεση λίγων δεκάδων στοιχείων



Centralized switched networks:
Crossbar switch



0 – 1
1 – 3
3 – 7
7 – 4
4 – 6
6 – 5
2 – 0
5 – 2

Γενική οργάνωση διακόπτη (router / switch)

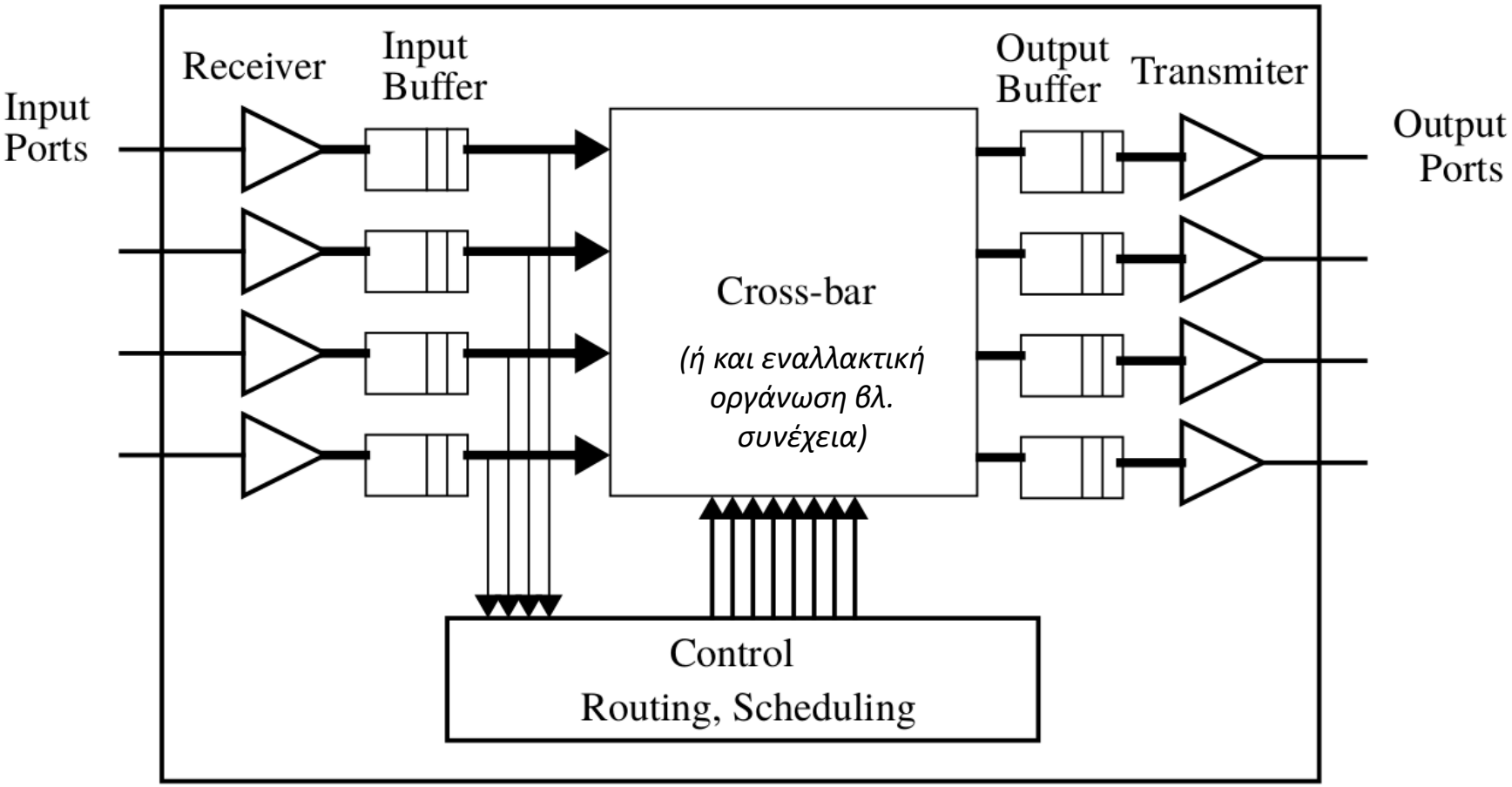
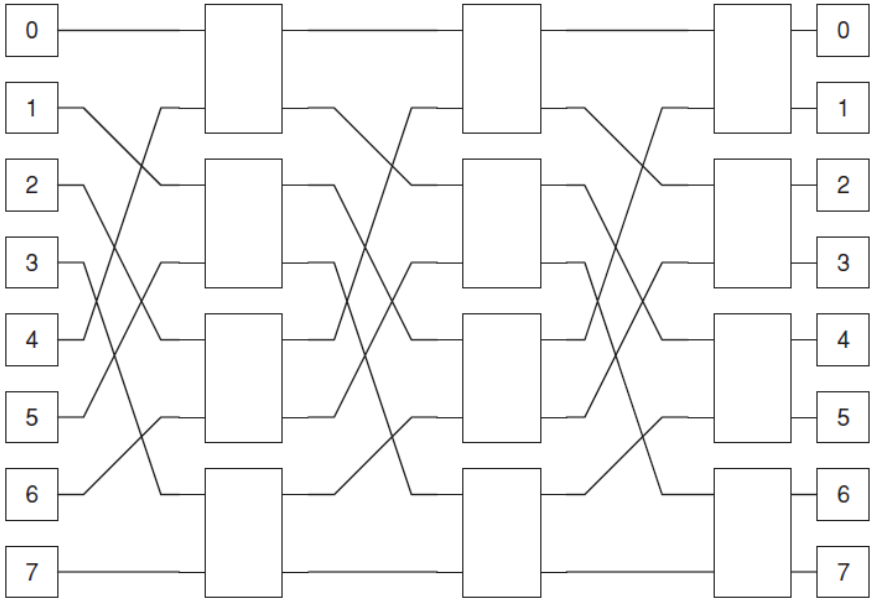


Image taken from: Parallel Computer Architecture, D. Culler, J.P. Singh

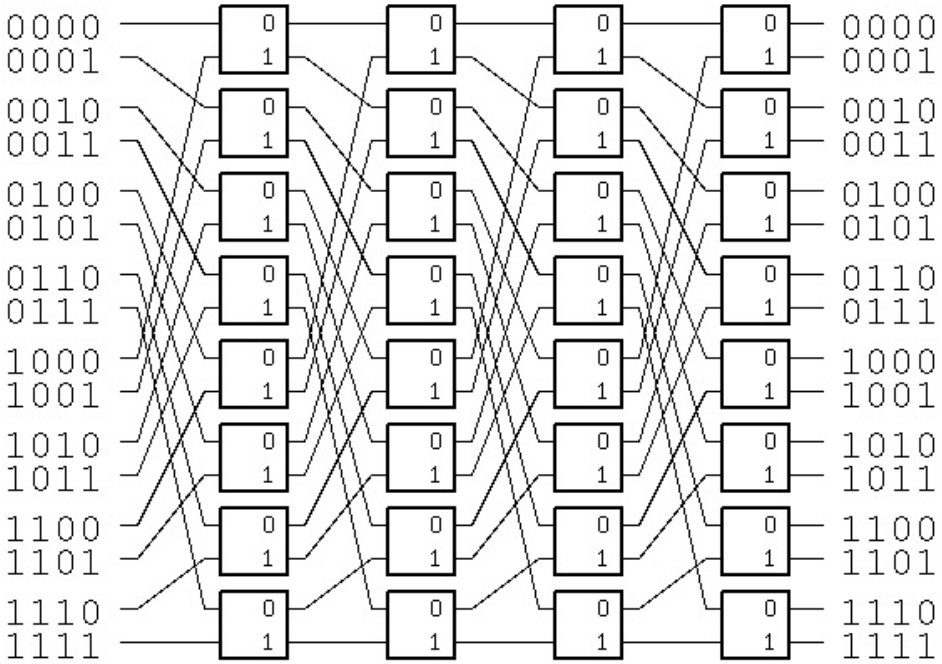
Centralized switched networks:
Multistage Interconnection Networks

- Διασυνδέουν N στοιχεία με τη χρήση πολυεπίπεδων διακοπτών
- Αν χρησιμοποιηθούν k * k διακόπτες, χρειάζονται $\log_k N$ στάδια με N/k διακόπτες ανά στάδιο (σύνολο **$N/k \log_k N$** διακόπτες)
- Ανάλογα με τη διασύνδεση των διακοπτών έχουν προκύψει διαφορετικά δίκτυα που ανταποκρίνονται σε διαφορετικά patterns επικοινωνίας



Centralized switched networks:
Δίκτυο Omega

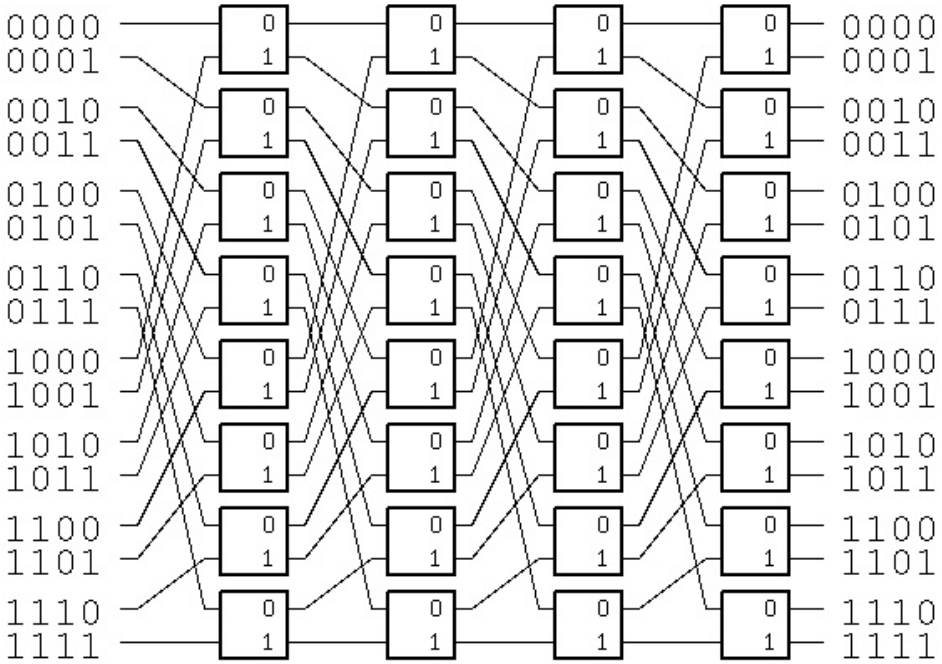
- Ονομάζεται και Perfect Shuffle (οι διασυνδέσεις σε κάθε επίπεδο προκύπτουν σαν ανακάτεμα τράπουλας)
- Destination-tag και xor-tag routing
- Είναι blocking (πολλά μονοπάτια επικαλύπτονται)



Centralized switched networks:
Δίκτυο Omega

Destination-tag routing

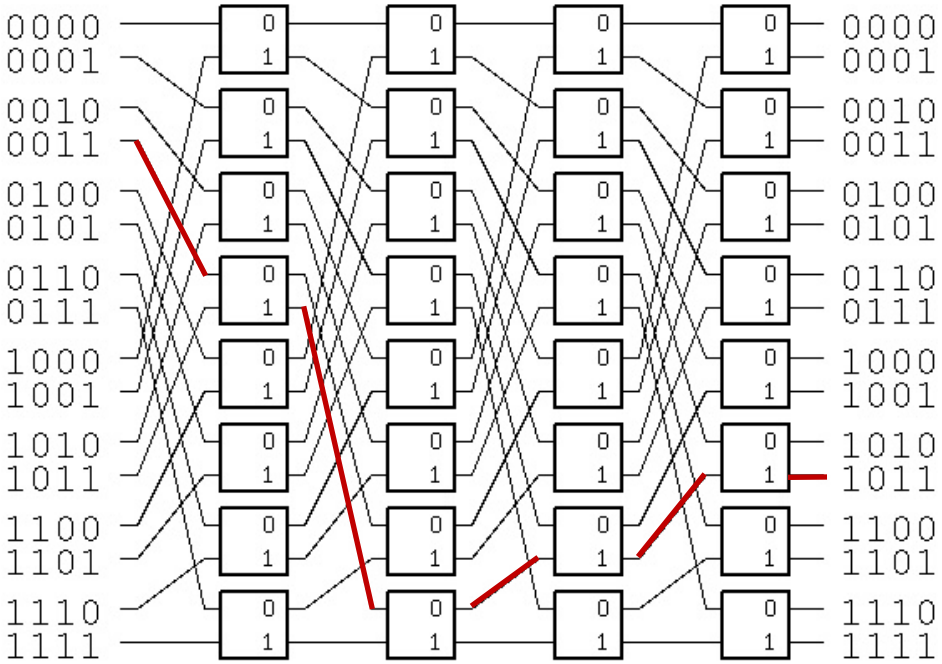
- Λαμβάνεται υπόψη μόνο ο προορισμός
- Π.χ. από οποιαδήποτε πηγή, για να φτάσω στον προορισμό 1011 θα πάρω διαδοχικά τις εξόδους «κάτω», «πάνω», «κάτω», «κάτω»



Centralized switched networks:
Δίκτυο Omega

Destination-tag routing

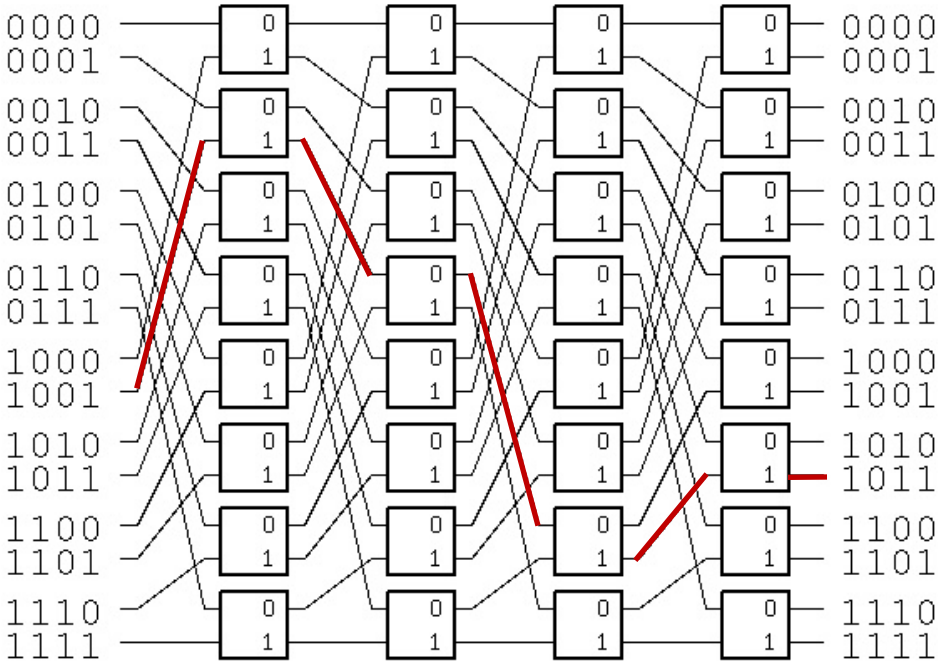
- Λαμβάνεται υπόψη μόνο ο προορισμός
- Π.χ. από οποιαδήποτε πηγή, για να φτάσω στον προορισμό 1011 θα πάρω διαδοχικά τις εξόδους «κάτω», «πάνω», «κάτω», «κάτω»



Centralized switched networks:
Δίκτυο Omega

Destination-tag routing

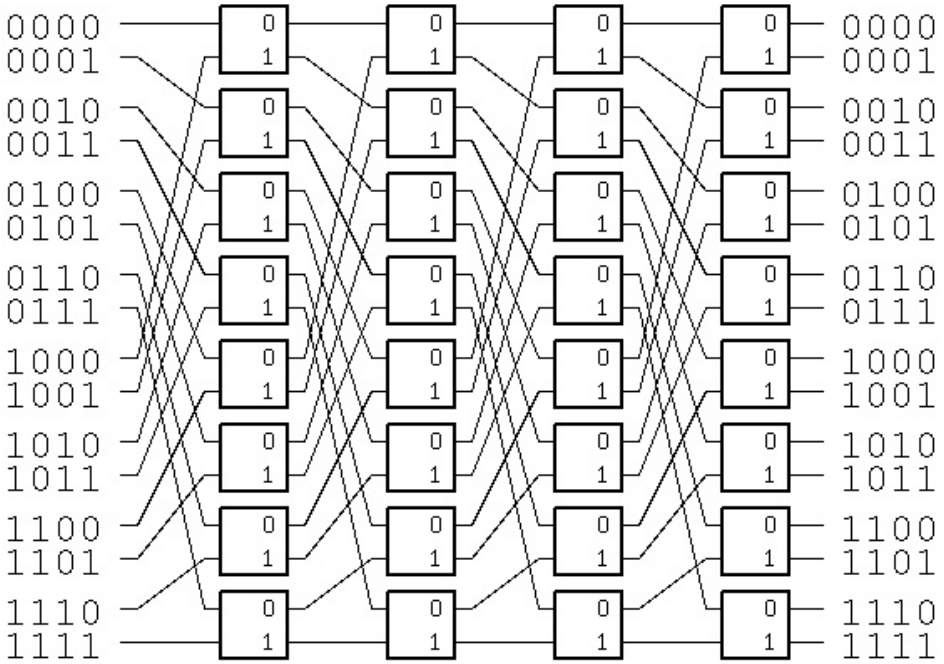
- Λαμβάνεται υπόψη μόνο ο προορισμός
- Π.χ. από οποιαδήποτε πηγή, για να φτάσω στον προορισμό 1011 θα πάρω διαδοχικά τις εξόδους «κάτω», «πάνω», «κάτω», «κάτω»



Centralized switched networks:
Δίκτυο Omega

XOR-tag routing

- Source xor Destination
- Αν το αποτέλεσμα είναι 0, ο αντίστοιχος διακόπτης περνιέται through, αν είναι 1 περνιέται crossed

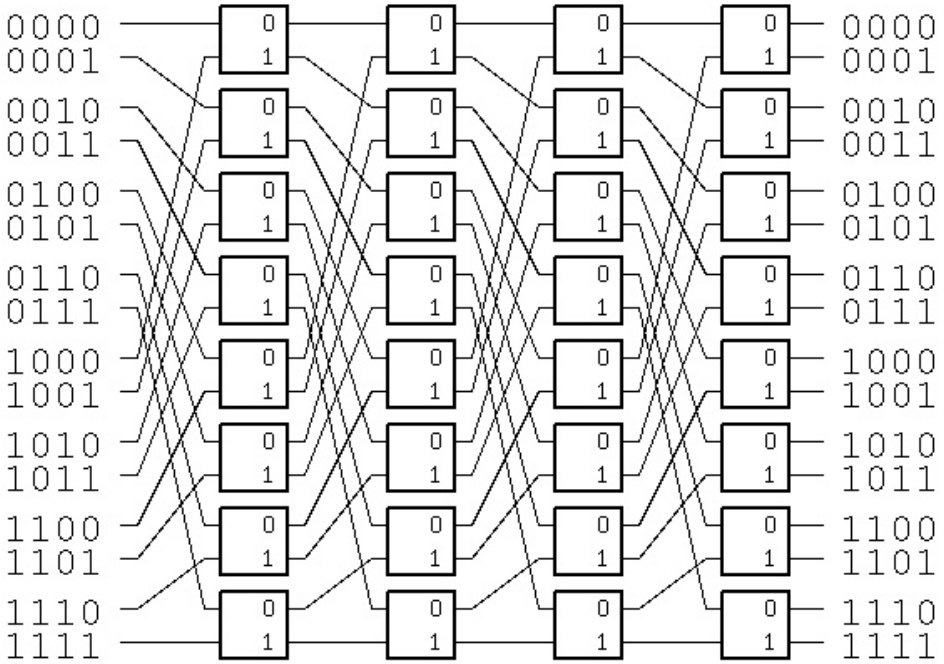


Centralized switched networks:
Δίκτυο Omega

XOR-tag routing

- Source xor Destination
- Αν το αποτέλεσμα είναι 0, ο αντίστοιχος διακόπτης περνιέται through, αν είναι 1 περνιέται crossed

Π.χ. 0010 -> 1110
0010 xor 1110 = 1100
crossed
crossed
through
through

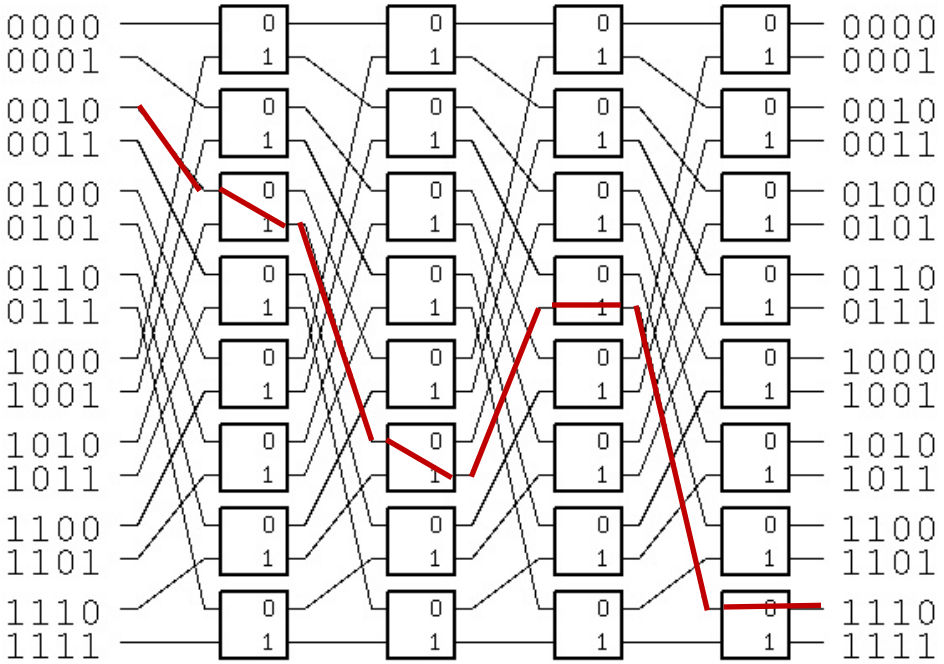


Centralized switched networks:
Δίκτυο Omega

XOR-tag routing

- Source xor Destination
- Αν το αποτέλεσμα είναι 0, ο αντίστοιχος διακόπτης περνιέται through, αν είναι 1 περνιέται crossed

Π.χ. 0010 -> 1110
0010 xor 1110 = 1100
crossed
crossed
through
through



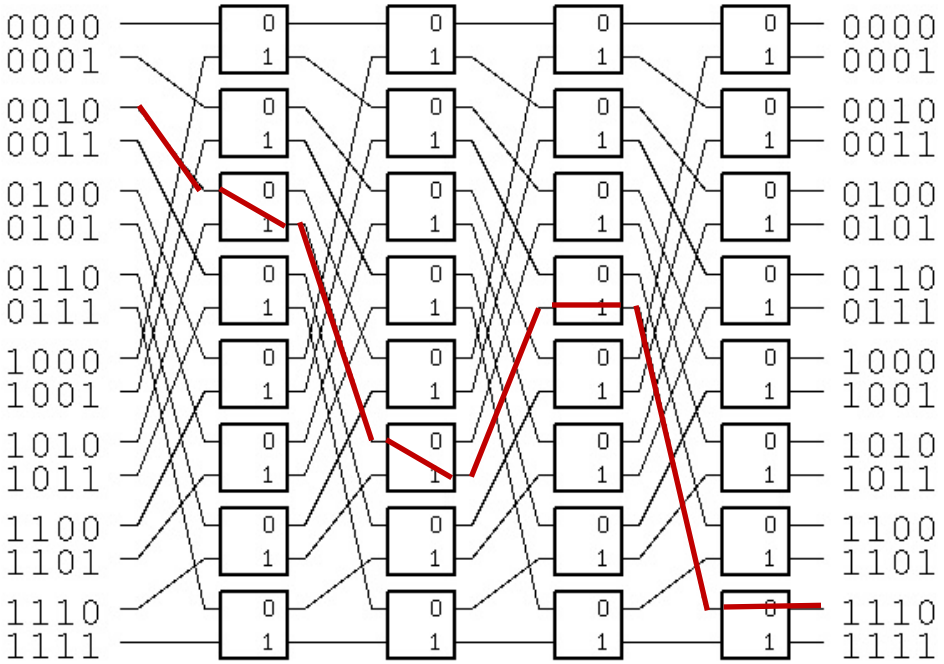
Centralized switched networks:
Δίκτυο Omega

XOR-tag routing

- Source xor Destination
- Αν το αποτέλεσμα είναι 0, ο αντίστοιχος διακόπτης περνιέται through, αν είναι 1 περνιέται crossed

Π.χ. 0010 -> 1110
0010 xor 1110 = 1100
crossed
crossed
through
through

1010-> 1011
1010 xor 1011 = 0001
through
through
through
crossed



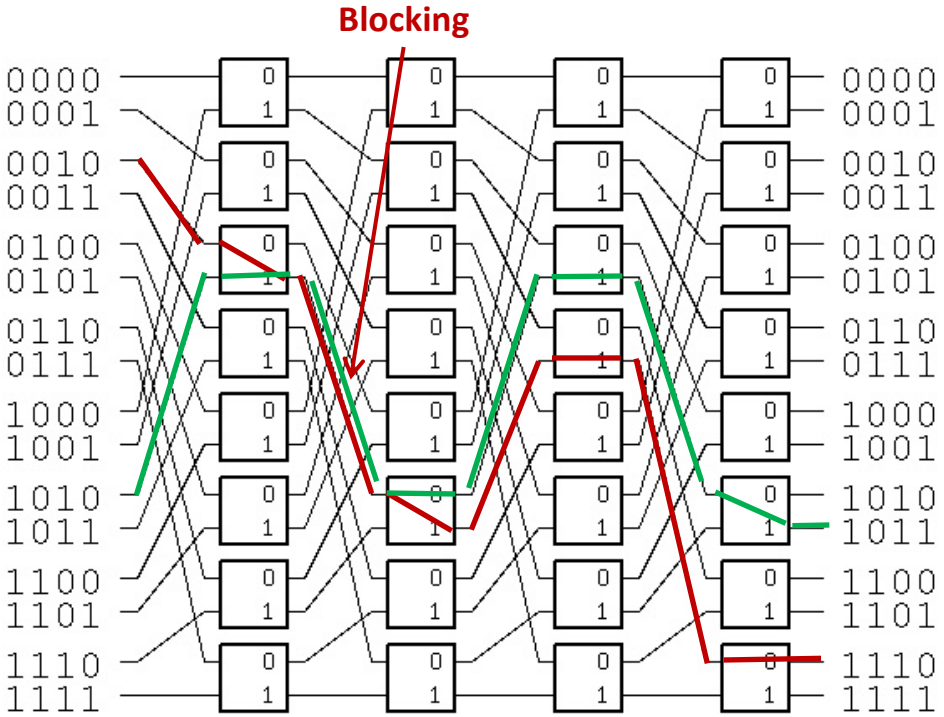
Centralized switched networks:
Δίκτυο Omega

XOR-tag routing

- Source xor Destination
- Αν το αποτέλεσμα είναι 0, ο αντίστοιχος διακόπτης περνιέται through, αν είναι 1 περνιέται crossed

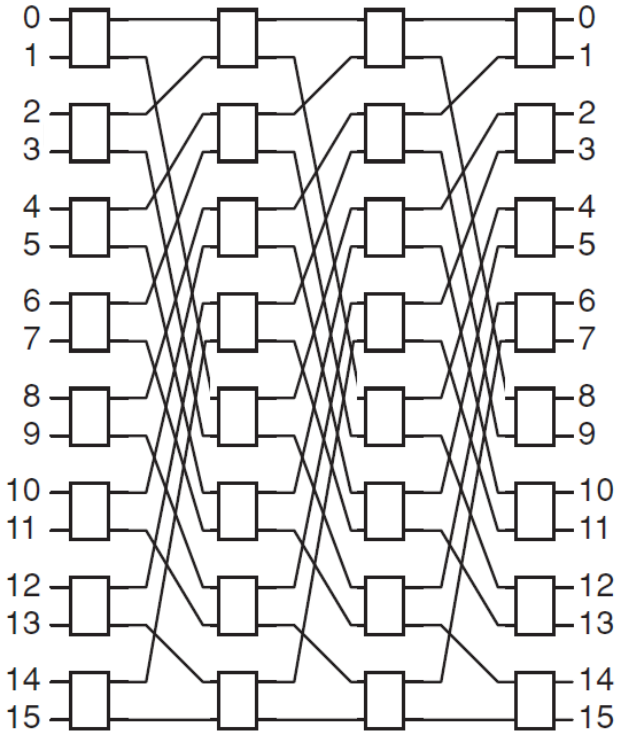
Π.χ. 0010 -> 1110
0010 xor 1110 = 1100
crossed
crossed
through
through

1010-> 1011
1010 xor 1011 = 0001
through
through
through
crossed



Centralized switched networks:
Δίκτυο Benes

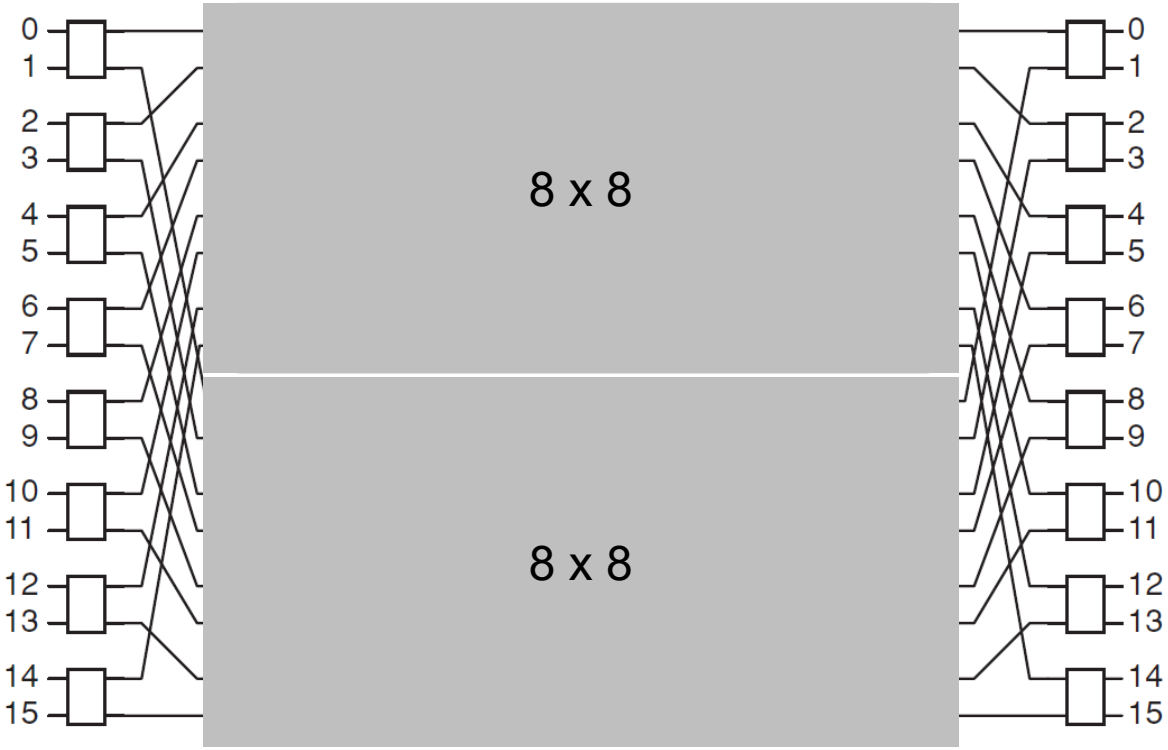
- **Στόχος:** Μείωση συμφόρησης (contention) λόγω διεκδίκησης κοινών διαδρομών
- **Προσέγγιση:** Χρήση επιπλέον διακοπτών
  - Περισσότερα επίπεδα
  - Μεγαλύτερους διακόπτες



Omega

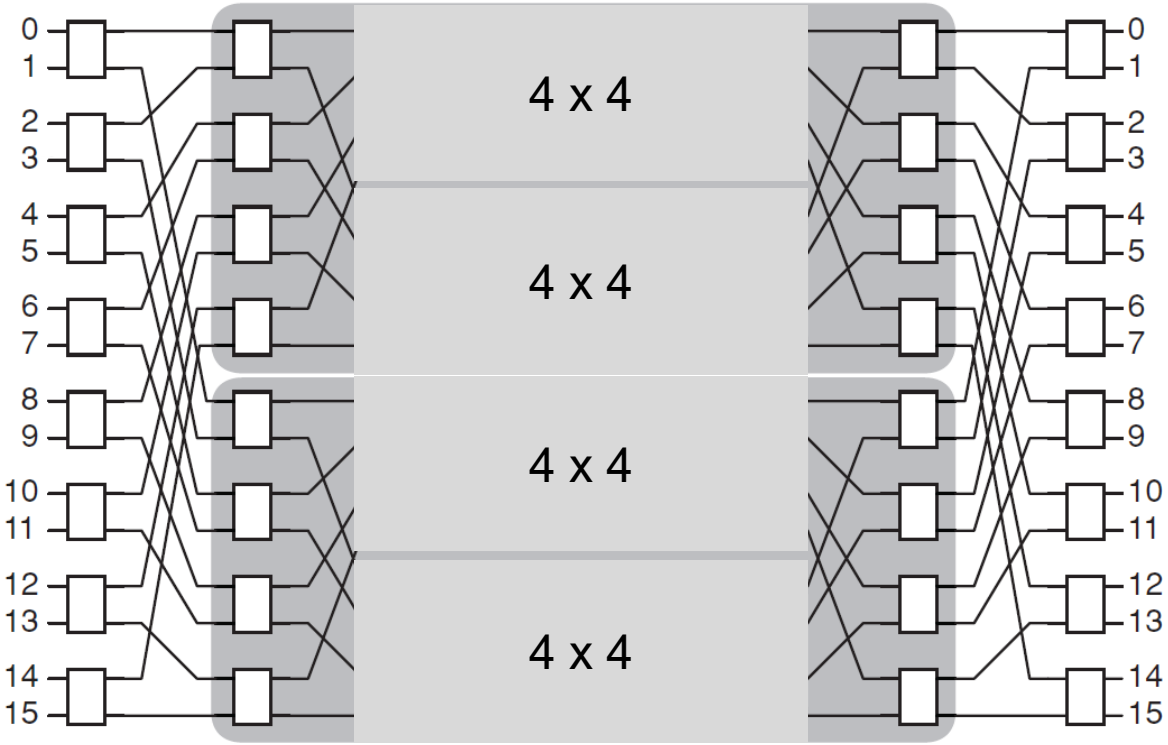
Centralized switched networks:
Δίκτυο Benes

- **Στόχος:** Μείωση συμφόρησης (contention) λόγω διεκδίκησης κοινών διαδρομών
- **Προσέγγιση:** Χρήση επιπλέον διακοπτών
  - Περισσότερα επίπεδα
  - Μεγαλύτερους διακόπτες



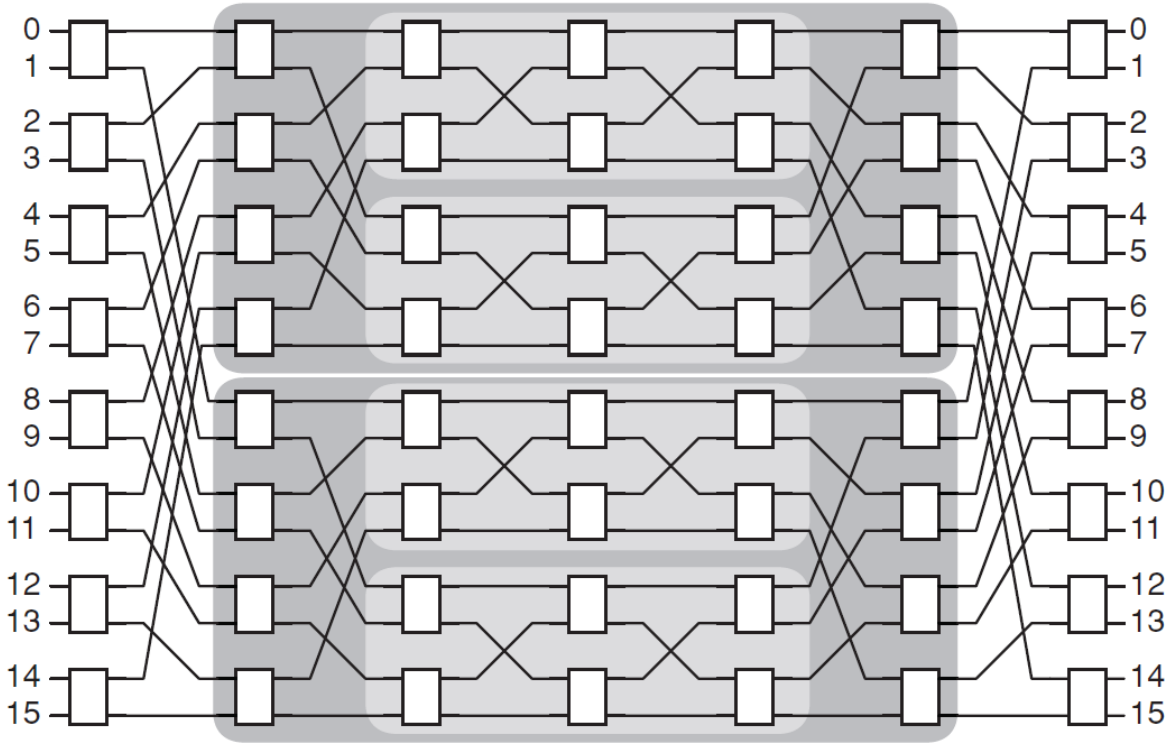
Centralized switched networks:
Δίκτυο Benes

- **Στόχος:** Μείωση συμφόρησης (contention) λόγω διεκδίκησης κοινών διαδρομών
- **Προσέγγιση:** Χρήση επιπλέον διακοπτών
  - Περισσότερα επίπεδα
  - Μεγαλύτερους διακόπτες



Centralized switched networks:
Δίκτυο Benes

- **Στόχος:** Μείωση συμφόρησης (contention) λόγω διεκδίκησης κοινών διαδρομών
- **Προσέγγιση:** Χρήση επιπλέον διακοπτών
  - Περισσότερα επίπεδα
  - Μεγαλύτερους διακόπτες

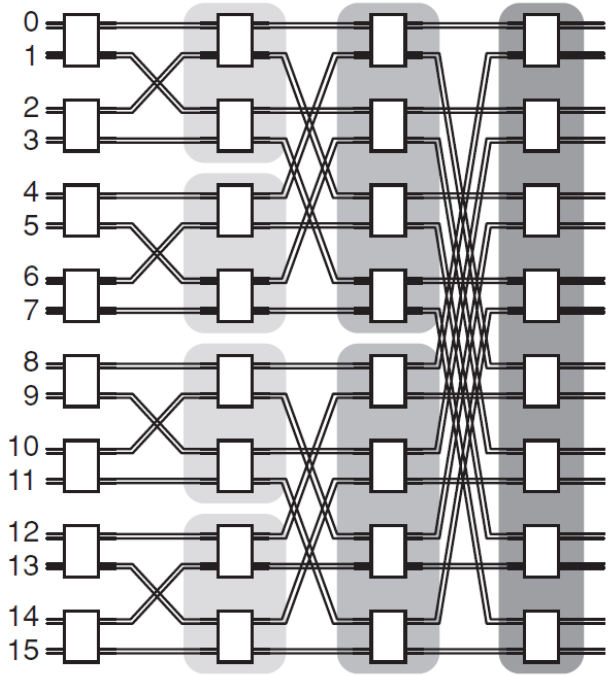


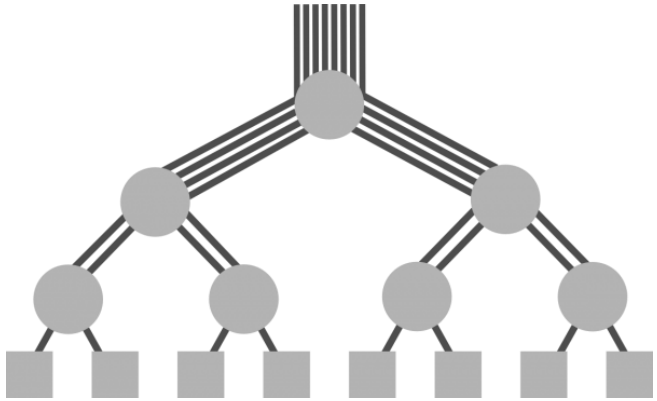
Δίκτυο Benes
16-port Clos topology

Centralized switched networks:
Fat tree

- Τα φύλλα του δέντρου είναι τα στοιχεία που διασυνδέονται
- Οι εσωτερικοί κόμβοι είναι διακόπτες
- Χρησιμοποιείται κατά κόρον σε SANs και Supercomputers (π.χ. Infiniband, κλπ)
- Ιδιότητες του fat tree:
  - Στα ενδιάμεσα επίπεδα **uplinks = downlinks**
  - Στο υψηλότερο επίπεδο **uplinks = 0**
  - Σε όλα τα επίπεδα: **downlinks = nodes**

Folded Benes network

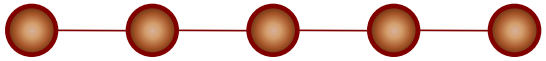




Distributed switched networks

- Οι διακόπτες του δικτύου κατανέμονται στους κόμβους του συστήματος
  - Ή όλοι οι διακόπτες περιλαμβάνουν τερματικούς κόμβους (compute nodes)
- Σε πολλές περιπτώσεις μεγάλος αριθμός (ίσος με τον αριθμό των κόμβων) από μικρούς διακόπτες
- Συχνά οι διακόπτες ολοκληρώνονται μαζί με τον επεξεργαστή
- Κρίσιμες μετρικές:
  - Αριθμός συνδέσμων (κόστος)
  - Βαθμός κόμβου (κόστος και επεκτασιμότητα)
  - Διάμετρος (επίδοση)
  - Εύρος τομής (επίδοση)

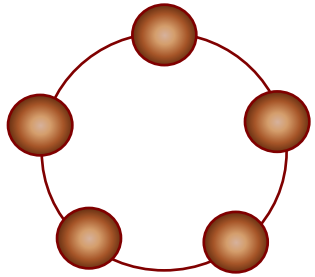
Distributed switched networks:
Γραμμικό



- N κόμβοι
- N-1 σύνδεσμοι
- Βαθμός d = 2 για τους εσωτερικούς κόμβους
- Διάμετρος D = N-1
- Εύρος τομής b = 1
- Δεν είναι συμμετρικό
- Επεκτάσιμο
- Διαφορά από το διάδρομο: διαφορετικά κανάλια-σύνδεσμοι μπορούν να χρησιμοποιούνται ταυτόχρονα

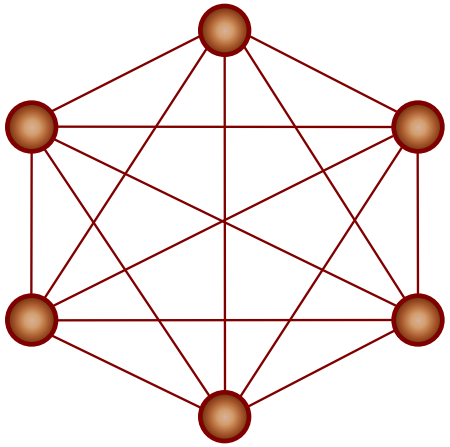
Distributed switched networks:
Δακτύλιος

- N κόμβοι
- N σύνδεσμοι
- Βαθμός κόμβων d = 2
- Διάμετρος: D = floor(N/2)
- Εύρος τομής b = 2
- Είναι συμμετρικό



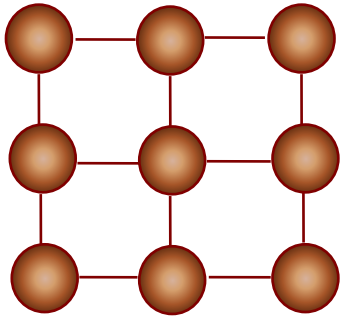
Distributed switched networks:
Πλήρες

- N κόμβοι
- N(N-1)/2 σύνδεσμοι
- Βαθμός κόμβου d = N-1
- Διάμετρος D = 1
- Εύρος τομής $b = (N/2)^2$
- Είναι συμμετρικό



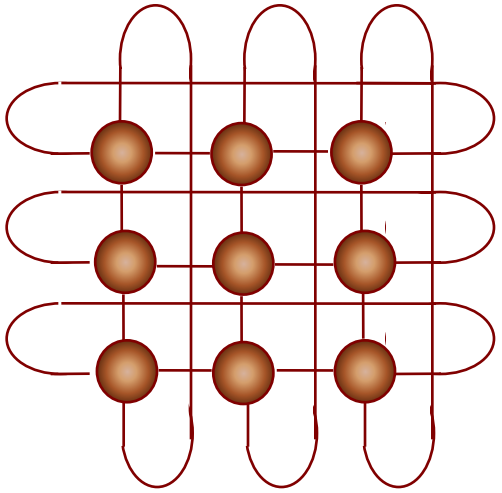
Distributed switched networks:
Mesh

- $N=n^k$ κόμβοι
- k-διάστατο mesh με n κόμβους ανά διεύθυνση
- βαθμός κόμβου d = 2k
- διάμετρος δικτύου D = k(n-1)
- Για ένα 2-διάστατο mesh:
  - $N=n^2$ κόμβοι
  - $2N-2n=2n^2-2n$ σύνδεσμοι
  - Βαθμός εσωτερικών κόμβων d=4
  - Διάμετρος D=2(n-1)
  - Εύρος τομής b=n
  - Δεν είναι συμμετρικό



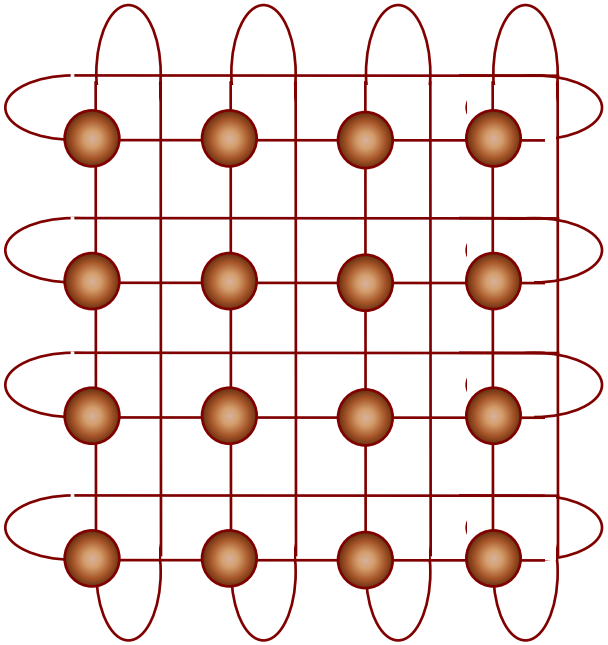
Distributed switched networks:
Torrus

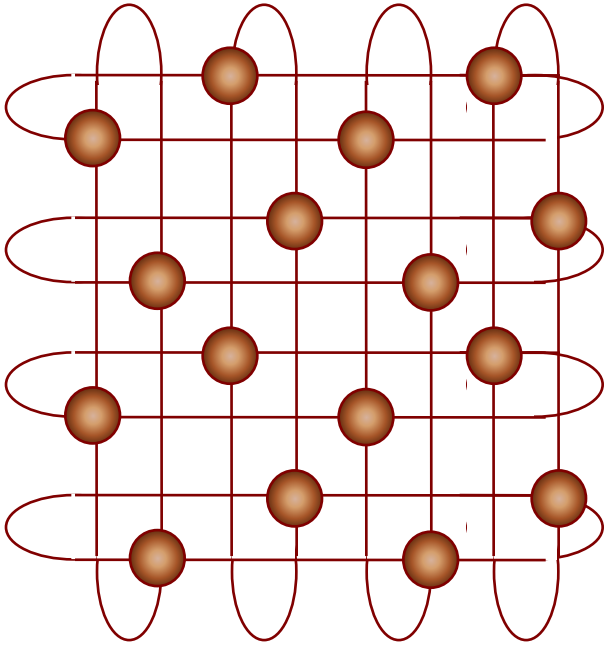
- Υποδιπλασιάζεται η διάμετρος
- για έναν n×n δυαδικό torus (k=2):
  - $N=n^2$ κόμβοι
  - 2N σύνδεσμοι
  - βαθμός κόμβου d=4
  - Διάμετρος D = 2 floor(N/2)
  - Εύρος τομής 2n
  - Είναι συμμετρικό



Distributed switched networks:
Iliac mesh

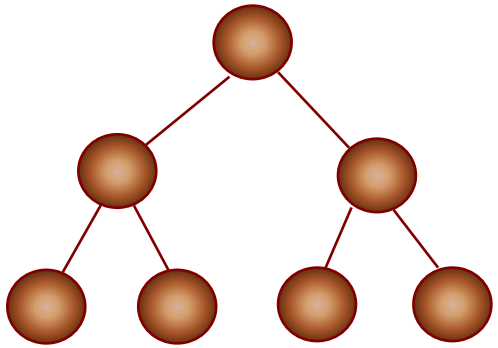
Αναδίπλωση συνδέσεων για την εξισορρόπηση του μήκους των καλωδίων





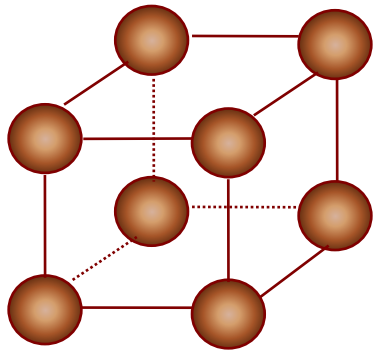
Distributed switched networks:
Δέντρο

- $N = 2^k-1$ κόμβοι
- N-1 σύνδεσμοι
- Βαθμός κόμβου d = 3 (επεκτάσιμο)
- Διάμετρος: D = 2(k-1)
- Εύρος τομής b = 1 (bottleneck)
- Δεν είναι συμμετρικό

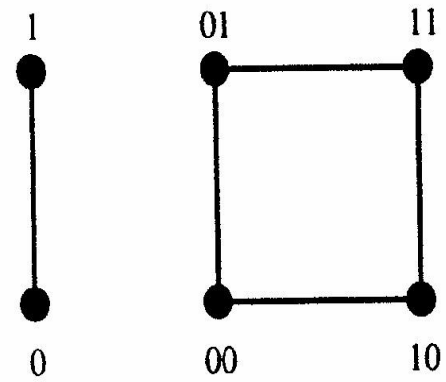


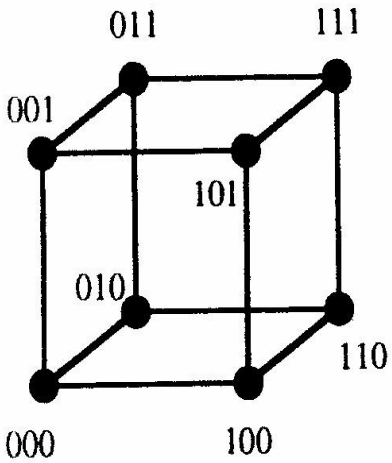
Distributed switched networks:
Υπερκύβος (hypercube)

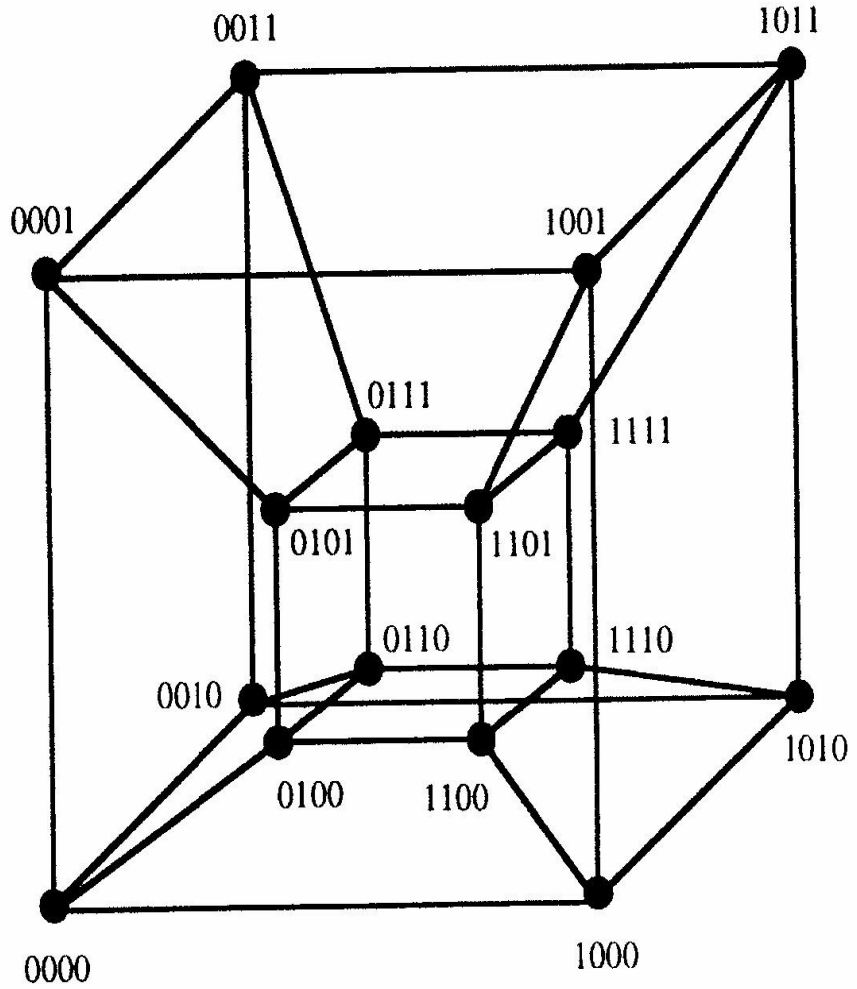
- $N=2^n$ κόμβοι
- $nN/2$ σύνδεσμοι
- Βαθμός κόμβου $d=n$
- Διάμετρος $D=n$
- Εύρος τομής $b=N/2$
- Είναι συμμετρικό
- Άμεσος προσδιορισμός διαδρομής



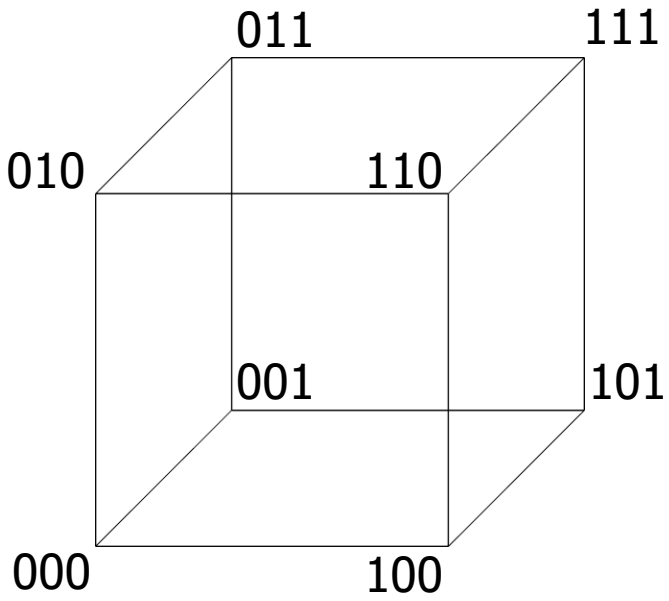
Αναδρομική Κατασκευή Υπερκύβου





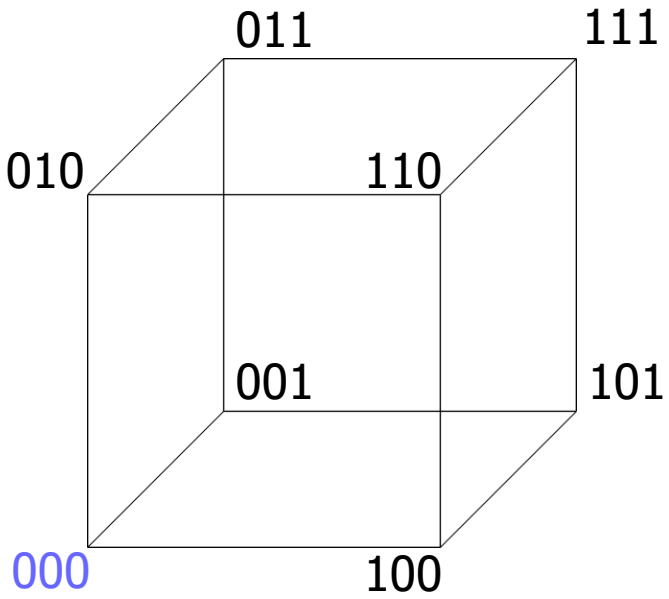


Hypercube Routing



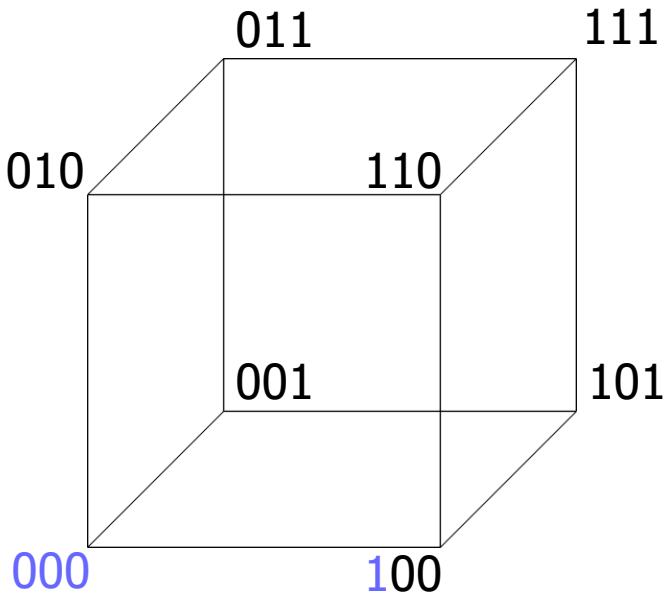
Οι διευθύνσεις γειτονικών κόμβων διαφέρουν κατά 1 bit

Hypercube Routing



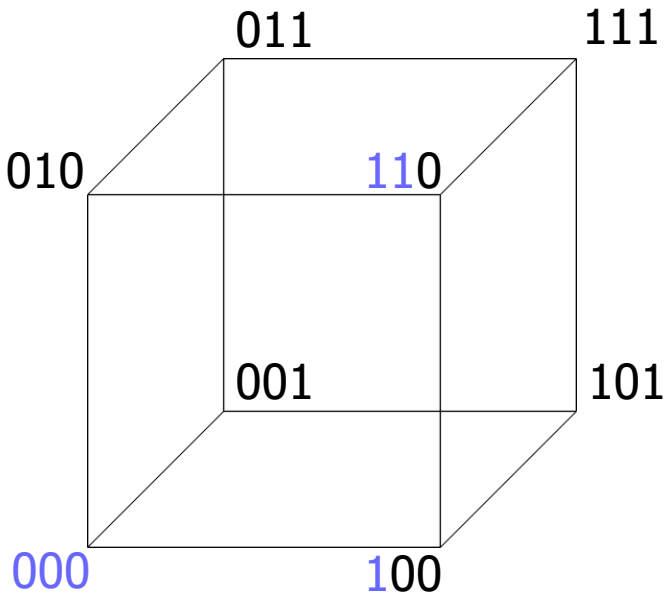
000 → 111

Hypercube Routing



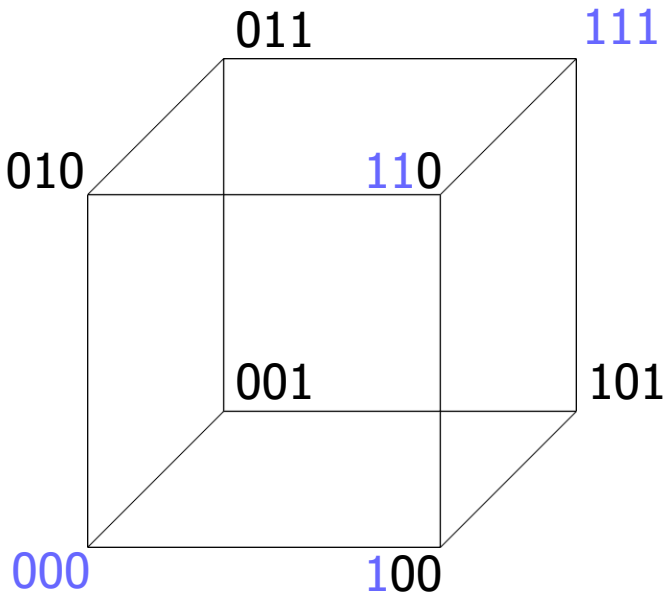
000 → 111

Hypercube Routing



000 → 111

Hypercube Routing



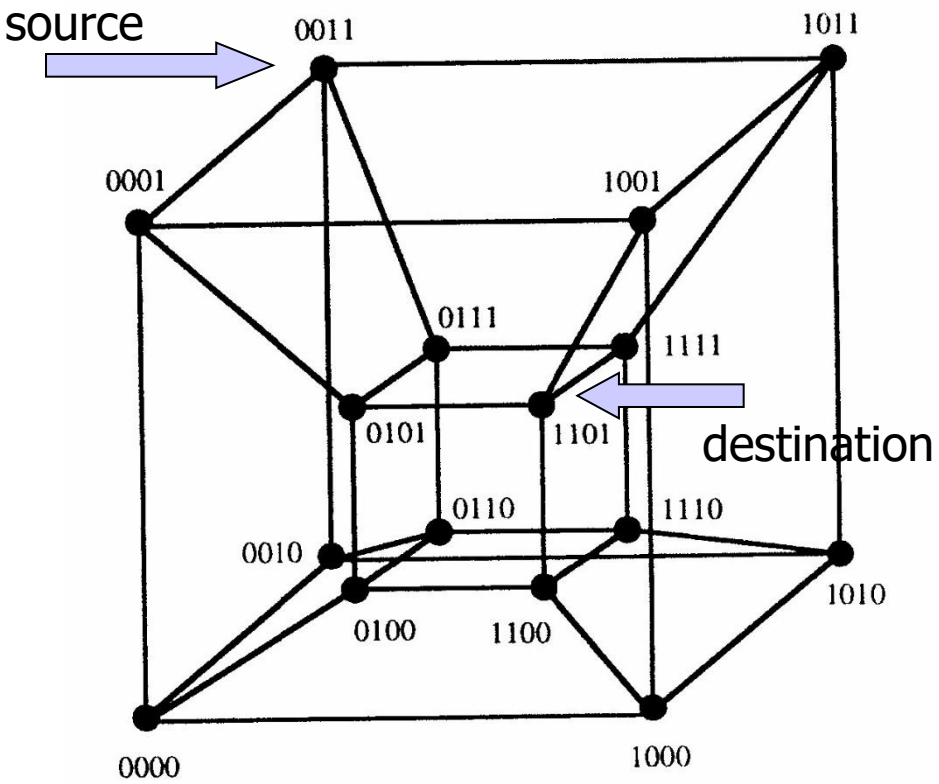
000 → 111

Παράδειγμα Προσδιορισμού Διαδρομής

$$0011 \rightarrow 1101$$

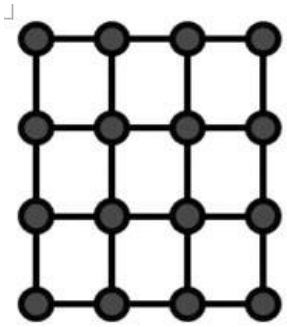
$$0011 \oplus 1101 = 1110$$

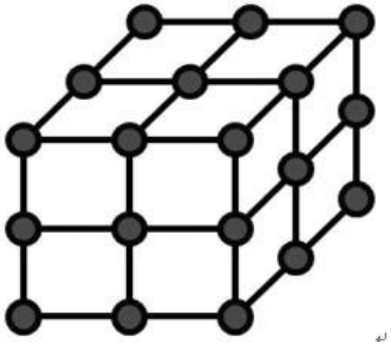
$$0011 \rightarrow 1011 \rightarrow 1111 \rightarrow 1101$$



Distributed switched networks:
Γενίκευση: k-δικός n-κύβος

- $N = k^n$ κόμβοι
- nN σύνδεσμοι
- Βαθμός κόμβου $d = 2n$
- Διάμετρος: $D = n \, floor(k/2)$
- Εύρος τομής $b = 2k^{n-1}$
- Είναι συμμετρικό





Χαρακτηριστικά συνδεσμολογιών

Τύπος Δικτύου	Κόμβοι	Σύνδεσμοι	Βαθμός κόμβου	Διάμετρος δικτύου	Εύρος τομής	Συμμετρία
Γραμμικό	N	N-1	2	N-1	1	Όχι
Δακτύλιος	N	N	2	$\lfloor N/2 \rfloor$	2	Ναι
Πλήρες	N	N(N-1)/2	N-1	1	$(N/2)^2$	Ναι
Δυαδικό δένδρο	$N=2^k-1$	N-1	3	2(k-1)	1	Όχι
Αστεροειδές	N	N-1	N-1	2	$\lfloor N/2 \rfloor$	Όχι
2D-Mesh	$N=n^2$	2N-2n	4	2(n-1)	n	Όχι
Iliac Mesh	$N=n^2$	2N	4	N-1	2n	Όχι
2D-Torus	$N=n^2$	2N	4	$2\lfloor n/2 \rfloor$	2n	Ναι
Υπερκύβος	$N=2^n$	nN/2	n	n	N/2	Ναι
k-δικός n-κύβος	$N=k^n$	nN	2n	$2k-1+\lfloor k/2 \rfloor$ $n\lfloor k/2 \rfloor$	$2k^{n-1}$	Ναι

Τάσεις στα δίκτυα εμπορικών συστημάτων

- Στόχοι:
  - Μείωση του κόστους του δικτύου (αριθμός και μέγεθος διακοπτών, αριθμός συνδέσμων)
  - Διατήρηση μεγάλου εύρους ζώνης/εύρους τομής
  - Μείωση του latency του δικτύου
- Πώς μπορώ να μειώσω το κόστος και το latency;
  - Κρατώντας μικρή τη διάμετρο του δικτύου
- Τι σημαίνει μικρή διάμετρος;
  - Λιγότεροι διακόπτες σε ένα μονοπάτι από έναν κόμβο σε έναν άλλο
  - Μικρότερο μέσο μήκος μονοπατιού (latency)
- Αυτή η τάση εμφανίζεται στα δίκτυα των σύγχρονων υπερυπολογιστών
  - Επικρατούσες τοπολογίες: **οι ιεραρχικές τοπολογίες**

Δίκτυα πραγματικών συστημάτων

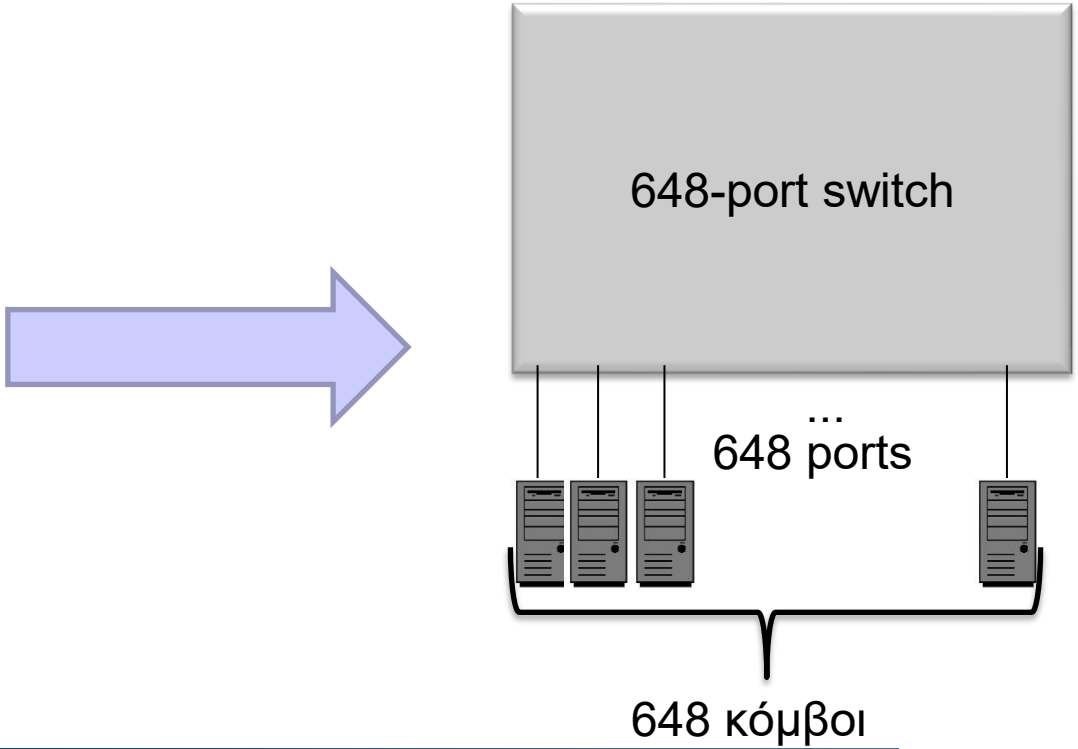
- Slingshot: **dragonfly** (βλ. συνέχεια)
- Fugaku (Tofu interconnect) : **6D torus**
- InfiniBand configuration: **fat tree**
- Historical note (1987): Connection Machine CM-2, 8192 nodes, hypercube

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	
5	**HPC6** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9, HPE Eni S.p.A. Italy	3,143,520	477.90	606.97	8,461
6	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899

Ένα παράδειγμα fat tree

- Ο ελληνικός υπερυπολογιστής ARIS χρησιμοποιεί την τεχνολογία InfiniBand FDR και την τοπολογία fat tree

- Χρησιμοποιεί το 648-port Mellanox switch SX-6536

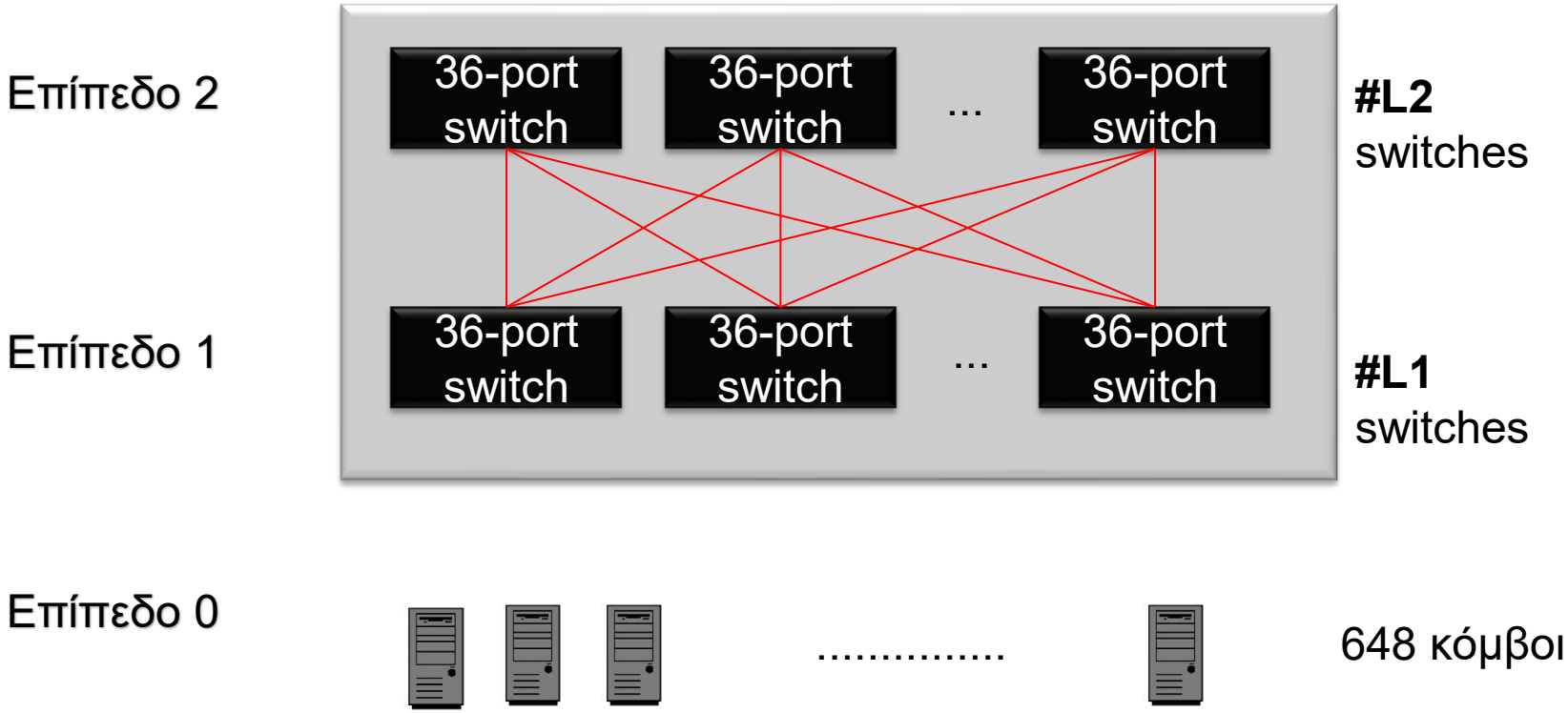


Ένα παράδειγμα fat tree

- Το 648-port switch αποτελείται από πολλά 36-port switches σε τοπολογία fat tree **δύο επιπέδων**
- Γενικά, σε ένα **port** του switch μπορούμε να συνδέσουμε:
  - Έναν κόμβο
  - Ένα άλλο switch

Ένα παράδειγμα fat tree

- Τα 36-port switches συνδέονται σε τοπολογία δύο επιπέδων
  - Πόσα switches χρειαζόμαστε σε κάθε επίπεδο;

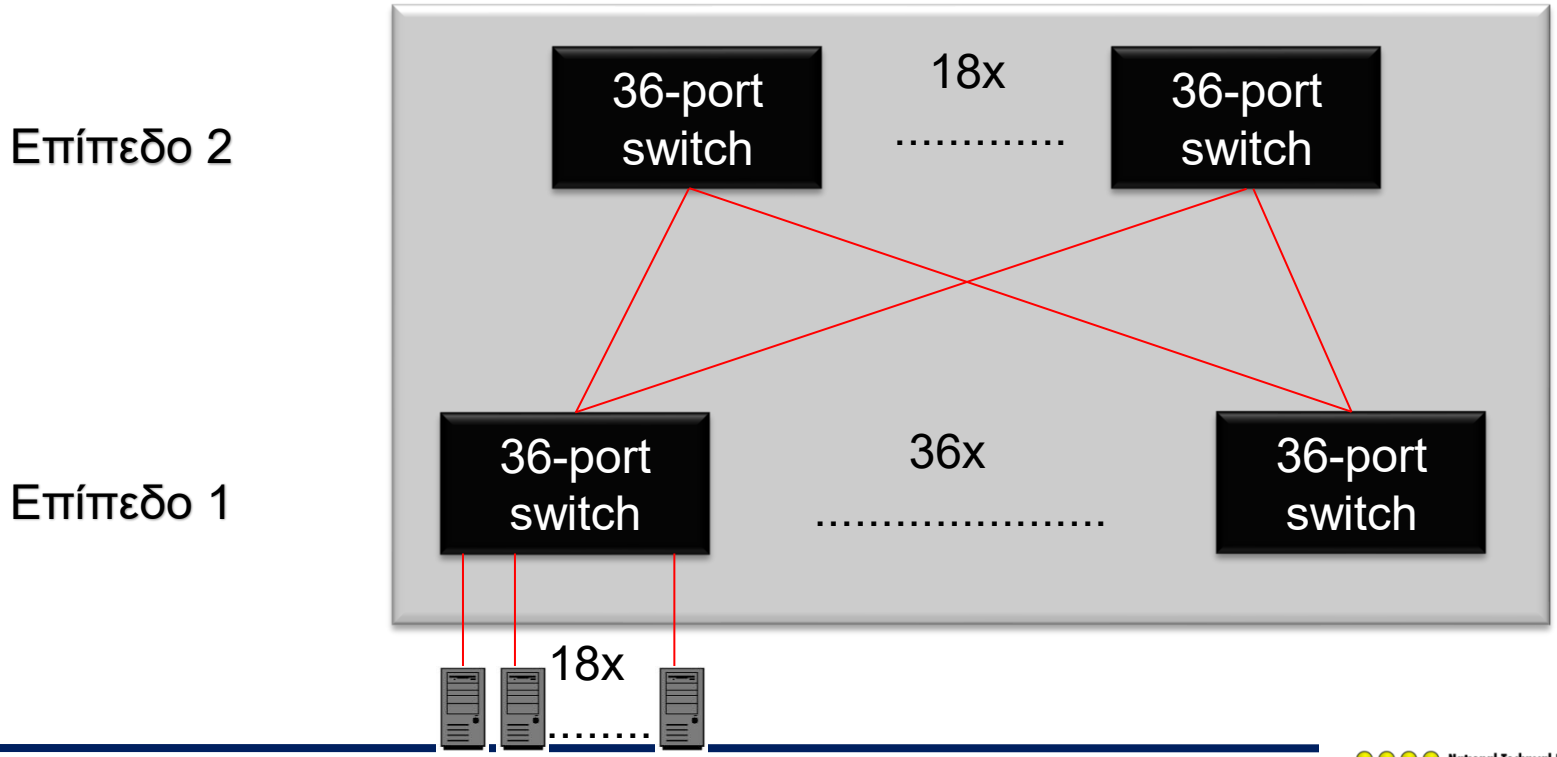


Ένα παράδειγμα fat tree

- Ιδιότητες του fat tree:
  - Στα ενδιάμεσα επίπεδα **uplinks = downlinks**
  - Στο υψηλότερο επίπεδο **uplinks = 0**
  - Σε όλα τα επίπεδα **downlinks = nodes**
- Διαθέσιμα ports για συνδέσεις:
  - #L1 * 36 ports στο επίπεδο 1
  - #L2 * 36 ports στο επίπεδο 2
- **Στο επίπεδο 1** (ενδιάμεσο επίπεδο)**:**
  - downlinks = uplinks = (36 ports / 2) * #L1 = 18 * #L1
  - downlinks = κόμβοι = 648
  - 648 = 18 * #L1 => **#L1 = 36**
- **Στο επίπεδο 2** (υψηλότερο επίπεδο):
  - downlinks = 648 = 36 * #L2 => **#L2= 18**

Ένα παράδειγμα fat tree

- Το δίκτυο του ARIS - ένα fat-tree ως 648-port switch
  - 18 switches στο επίπεδο 2, 36 switches στο επίπεδο 1
  - 18 downlinks * 36 switches του επιπέδου 1 = 648 ports προς κόμβους



Κλιμάκωση για fat tree

- Στο παράδειγμα του ARIS έχω fat tree δύο επιπέδων (ειδική περίπτωση) όπου ο αριθμός των ports είναι ίδιος στα switches των δύο επιπέδων
- Στη γενική περίπτωση και για να κλιμακώσει το δίκτυο στον επιθυμητό αριθμό κόμβων θέλουμε να αυξάνουμε τον αριθμό των ports όσο ανεβαίνουμε επίπεδο
- Βλ. διαφάνεια 19 για την κατασκευή N-port switches από k-port switches
  - Προσοχή στην επίδραση στο κόστος και το latency

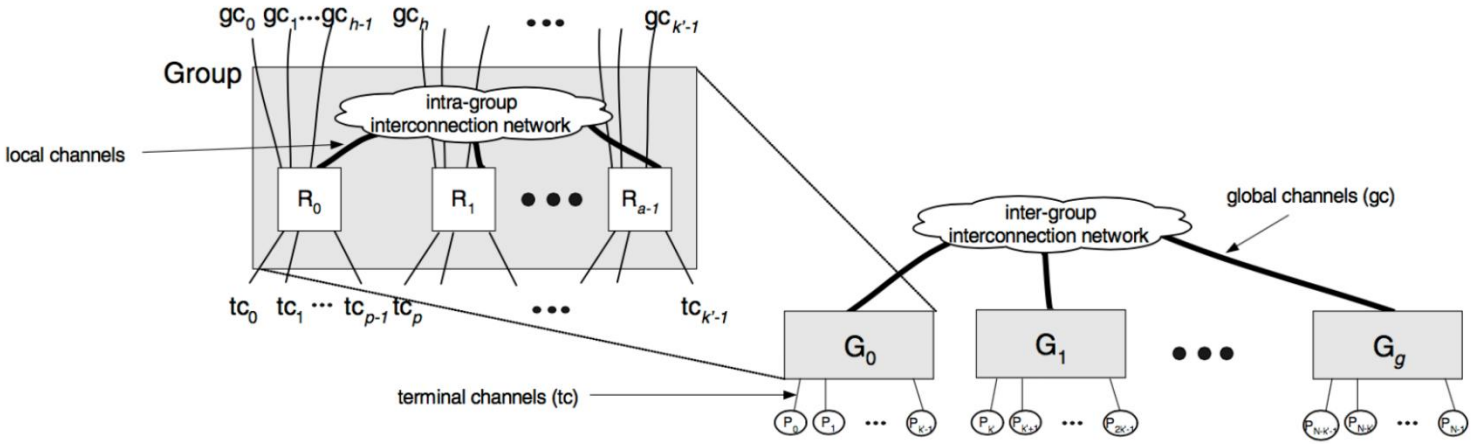
Η τοπολογία Dragonfly

- Ιεραρχική τοπολογία
  - Αποτελείται από πολλά groups διακοπτών
- Τα groups είναι συνδεδεμένα σε πλήρη τοπολογία
- Στο εσωτερικό ενός group, οι διακόπτες μπορούν να συνδεθούν σε οποιαδήποτε τοπολογία
  - Canonical dragonfly: πλήρης τοπολογία εντός group

- Πλεονεκτήματα: χαμηλή διάμετρος
- Μειονεκτήματα: απαιτεί προσαρμοστική δρομολόγηση

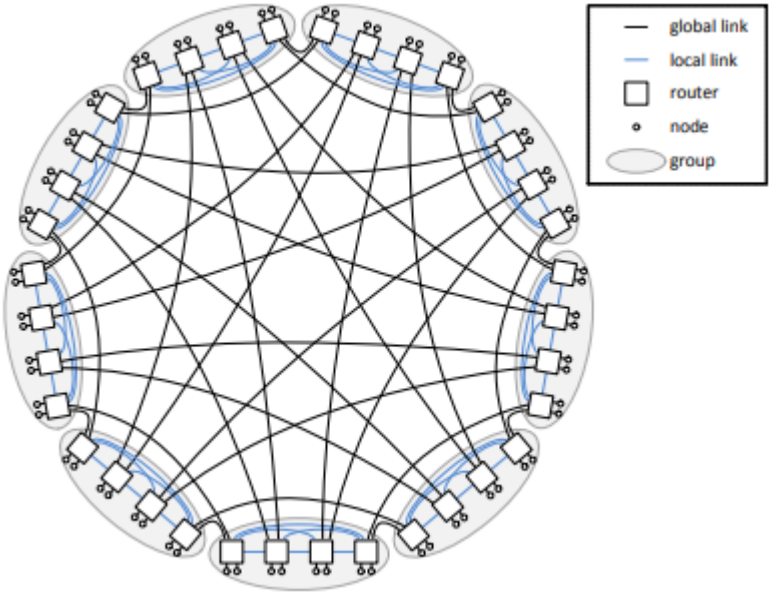
Η «κανονική» τοπολογία Dragonfly

- g: πλήθος groups
- a: πλήθος διακοπτών/routers ανά group
  - Σύνολο διακοπτών S = g * a
- p: πλήθος ports ανά διακόπτη για σύνδεση με τερματικούς (υπολογιστικούς) κόμβους
  - Σύνολο κόμβων N = g * a * p
- h: πλήθος ports ανά διακόπτη για σύνδεση με τα υπόλοιπα groups
- Μέγεθος διακοπτών (radix): k = p + h + (a – 1) ports



Η «κανονική» τοπολογία Dragonfly

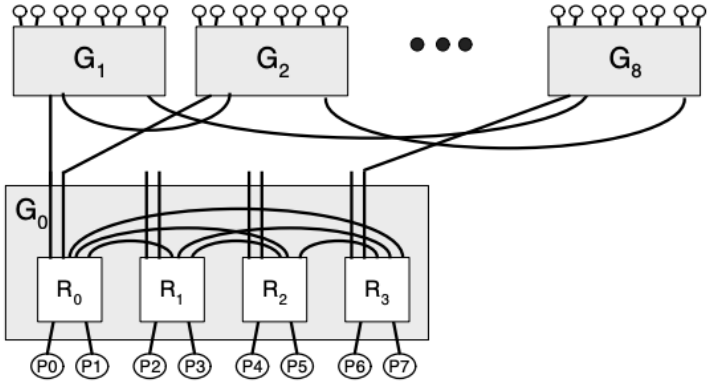
- Υπάρχουν πολλά διαφορετικά configurations ακόμα και του κανονικού dragonfly που υπαγορεύονται από την σχεδιαστική επιλογή μέγεθος group vs αριθμός groups
- Οι επιλογές μπορεί να υπαγορεύονται και από φυσικούς περιορισμούς (π.χ. ένα group να χωρά σε ένα ικρίωμα – rack)
- Η καθυστέρηση (latency) για επικοινωνία εντός / εκτός group μπορεί να είναι διαφορετική.



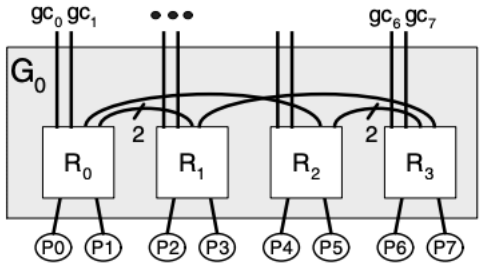
h = p = 2
a = 4

Σχεδιάζοντας την τοπολογία Dragonfly

- Κανονική τοπολογία
  - k = 7 (έστω 7-port switches)
  - Αν a = 4, h = p = 2
  - g = 9
  - **N =** 9 (group) * 4 (διακόπτες / group) * 2 (κόμβοι / διακόπτη) = **72**
  - Τοπολογία εντός group: πλήρης

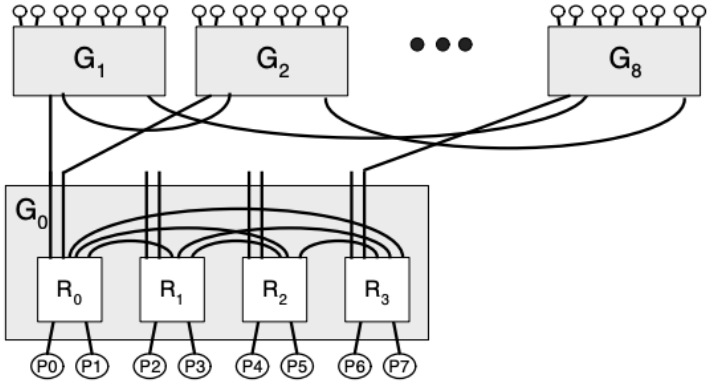


- Εναλλακτική τοπολογία #1
  - k = 7
  - **N = 72**
  - Τοπολογία εντός group: 2D-flattened butterfly
  - Εντός του group, οι κόμβοι είναι πιο κοντά, άρα μπορώ να μειώσω κάποιες απευθείας συνδέσεις για να αυξήσω το εύρος ζώνης
    - **Latency/Bandwidth trade-off**

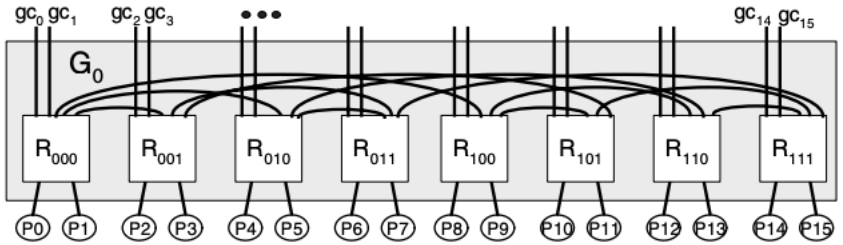


Σχεδιάζοντας την τοπολογία Dragonfly

- Κανονική τοπολογία
  - k = 7 (έστω 7-port switches)
  - Αν a = 4, h = p = 2
  - g = 9
  - **N =** 9 (group) * 4 (διακόπτες / group) * 2 (κόμβοι / διακόπτη) = **72**
  - Τοπολογία εντός group: πλήρης



- Εναλλακτική τοπολογία #2
  - k = 7
  - Τοπολογία εντός group: 3D-flattened butterfly
    - άρα k <> h + p + (a – 1)
  - a = 8, h = p = 2
  - g = 17
  - **N = 17 * 8 * 2 = 272**
  - Περισσότερα hops εντός του group (**μεγαλύτερο latency**) αλλά καλύτερη **κλιμακωσιμότητα**



Ζητήματα δρομολόγησης (routing)

- Εφαρμόζεται σε κάθε διακόπτη ανεξάρτητα από την τοπολογία
- Ορίζει τα επιτρεπόμενα μονοπάτια και κατευθύνει τα πακέτα μέσα στο δίκτυο
- *Ιδανικά:* Παρέχει τόσες επιλογές δρομολόγησης όσα και τα φυσικά μονοπάτια που παρέχει η τοπολογία, και κατανέμει ομοιόμορφα το φορτίο στο δίκτυο
- Απαιτούνται απλές και γρήγορες τεχνικές

Ζητήματα δρομολόγησης (routing)

- **Μηχανισμοί** δρομολόγησης:
  - **Αριθμητικοί:** ο υπολογισμός της διαδρομής γίνεται με απλές πράξεις λαμβάνοντας υπόψη π.χ. την πηγή ή/και τον προορισμό (βλ. destination/xor-tag routing στο δίκτυο omega)
  - **Υπολογισμός στην πηγή:** Ο αποστολέας υπολογίζει και ενσωματώνει στην κεφαλίδα του μηνύματος τη ρύθμιση κάθε ενδιάμεσου διακόπτη.
    - + Απλοποιεί τη σχεδίαση των διακοπτών
    - - Μεγαλώνει την κεφαλίδα
    - - Δεν υποστηρίζει εύκολα προσαρμοστική δρομολόγηση (βλ. συνέχεια)
  - **Αναζήτηση σε πίνακα δρομολόγησης:** Γενική προσέγγιση, όπου κάθε διακόπτης τηρεί έναν πίνακα δρομολόγησης.
    - + Μικρό μέγεθος κεφαλίδας
    - - Κόστος αποθήκευσης πίνακα δρομολόγησης
    - - Επικοινωνία μεταξύ διακοπτών για την ενημέρωση των πινάκων
    - Γενικά εφαρμόζεται σε LAN και WAN
- **Πολιτικές** δρομολόγησης: ντετερμινιστική, oblivious, προσαρμοστική
  - Tradeoff ανάμεσα σε απλότητα και ανοχή σε σφάλματα / αποφυγή συμφόρησης

Αλγόριθμοι (πολιτικές) δρομολόγησης

- Υπάρχουν τρεις τύποι αλγορίθμων δρομολόγησης
  - **Deterministic:** Για ένα συγκεκριμένο ζεύγος αφετηρίας-προορισμού, επιλέγεται πάντα το ίδιο μονοπάτι
  - **Oblivious:** Για ένα συγκεκριμένο ζεύγος αφετηρίας-προορισμού, επιλέγονται διαφορετικά μονοπάτια, άσχετα με την κατάσταση του δικτύου
  - **Adaptive:** Για ένα συγκεκριμένο ζεύγος αφετηρίας-προορισμού, επιλέγονται διαφορετικά μονοπάτια, ανάλογα με την κατάσταση του δικτύου
    - Για την προσαρμογή στην κατάσταση του δικτύου απαιτείται τροφοδότηση από το δίκτυο (τοπικά ή συνολικά)
    - Τα διαφορετικά μονοπάτια μπορεί να είναι ελάχιστα ή μη ελάχιστα

Deterministic routing

- Στη ντετερμινιστική δρομολόγηση, όλα τα πακέτα από μία συγκεκριμένη αφετηρία σε έναν συγκεκριμένο προορισμό ακολουθούν την ίδια διαδρομή
- **Dimension-order routing:** Αλγόριθμος ντετερμινιστικής δρομολόγησης
  - Διάσχιση του δικτύου ανά διάσταση
  - Π.χ. σε ένα 2D-mesh, πρώτα διάσχιση κατά X, μετά διάσχιση κατά Y
- Πλεονεκτήματα
  - Απλός αλγόριθμος
  - Δεν δημιουργεί deadlocks (βλ. συνέχεια) στις περισσότερες τοπολογίες
- Μειονεκτήματα
  - Μπορεί να δημιουργήσει φαινόμενα ανταγωνισμού
  - Δεν αξιοποιεί τα διαφορετικά μονοπάτια στο δίκτυο

Oblivious routing

- Στην oblivious δρομολόγηση, τα πακέτα μπορούν να κινηθούν από διαφορετικά μονοπάτια από την αφετηρία στον προορισμό, χωρίς να λαμβάνουν υπόψη την κατάσταση του δικτύου
- **Αλγόριθμος Valiant:** Αλγόριθμος για oblivious δρομολόγηση
  1. Τυχαία επιλογή ενός ενδιάμεσου προορισμού
  2. Δρομολόγηση από την αφετηρία ως τον ενδιάμεσο προορισμό
  3. Δρομολόγηση από τον ενδιάμεσο προορισμό στον τελικό προορισμό
     - Η ενδιάμεση δρομολόγηση μπορεί να είναι διαφορετική - π.χ. dimension-order
- Πλεονεκτήματα
  - Η επιλογή τυχαίων προορισμών κατανέμει περισσότερο ομοιόμορφα το φορτίο στο δίκτυο
    - Random pattern -> Uniform traffic
- Μειονεκτήματα
  - Τα μονοπάτια που επιλέγονται δεν είναι ελάχιστα
- Εναλλακτικά
  - Αξίζει να χρησιμοποιηθεί όταν το φορτίο στο δίκτυο είναι υψηλό

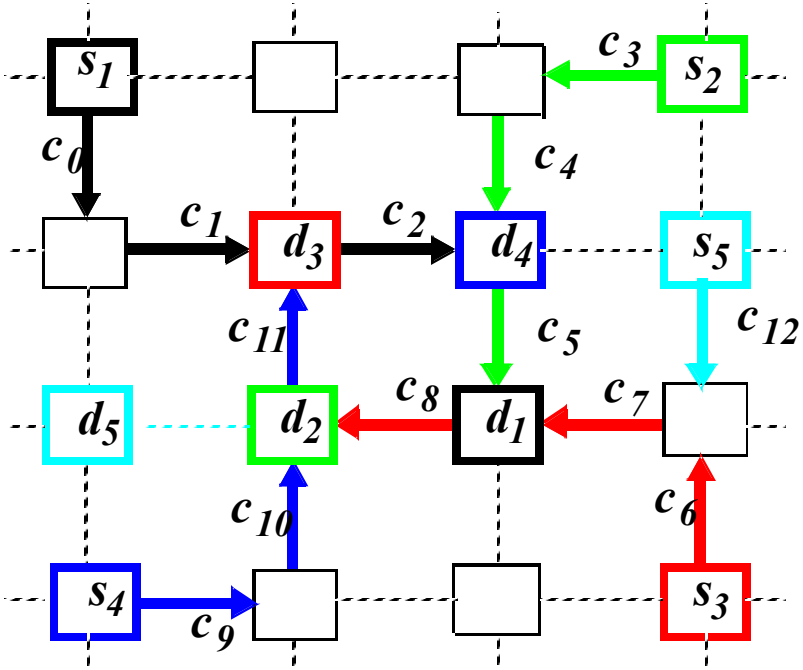
Adaptive routing

- Στην προσαρμοστική δρομολόγηση, τα πακέτα μπορούν να κινηθούν από διαφορετικά μονοπάτια από την αφετηρία στον προορισμό, λαμβάνοντας υπόψη την τρέχουσα κατάσταση του δικτύου
  - **Ελάχιστη προσαρμογή:** Αλγόριθμος για προσαρμοστική δρομολόγηση
    - Ο δρομολογητής κάνει τις ελάχιστες δυνατές προσαρμογές σε σχέση με την προκαθορισμένη δρομολόγηση
    - Επιλέγει μόνο ανάμεσα σε ελάχιστα μονοπάτια
    - **Πλεονεκτήματα**
      - Έχει επίγνωση των τοπικών φαινομένων ανταγωνισμού
    - **Μειονεκτήματα**
      - Η επιλογή ελάχιστου μονοπατιού μειώνει τη δυνατότητα εξισορρόπησης του φορτίου
  - **Μη-ελάχιστη προσαρμογή:** Αλγόριθμος για προσαρμοστική δρομολόγηση
    - Ο δρομολογητής στέλνει πακέτα σε κάποιο σημείο στο δίκτυο, άσχετα με την απόστασή του από τον προορισμό
    - **Πλεονεκτήματα**
      - Μπορεί να επιτύχει εξισορρόπηση φορτίου και καλύτερη χρήση του δικτύου
    - **Μειονεκτήματα**
      - Πρέπει να εξασφαλίζει ότι δεν θα εμφανιστεί livelock

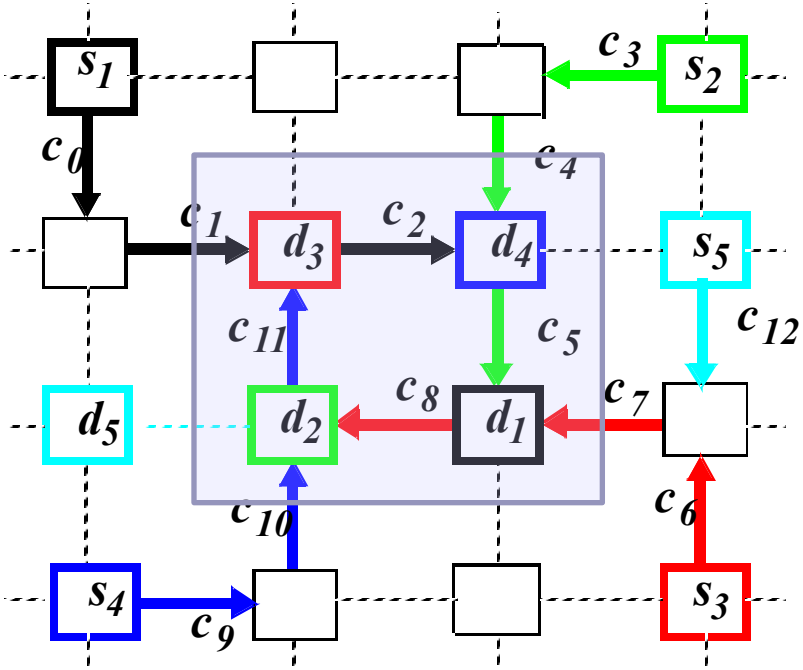
Livelock και deadlock

- *Προβλήματα:* Καταστάσεις κατά τις οποίες ένα πακέτο δεν φτάνει ποτέ στον προορισμό του:
  - ***Livelock***
    - Προκύπτει όταν υπάρχει άπειρος επιτρεπόμενος αριθμός από ενδιάμεσους κόμβους
    - *Λύση:* Περιορισμός των ενδιάμεσων κόμβων που θα περάσει ένα πακέτο
  - ***Deadlock***
    - Προκύπτει όταν ένα σύνολο από πακέτα μπλοκάρουν περιμένοντας πόρους του δικτύου (π.χ. συνδέσεις, buffers) να απελευθερωθούν
    - Η πιθανότητα αυξάνει σε καταστάσεις συμφόρησης

Deadlock κατά τη δρομολόγηση σε 2-διάστατο mesh



Deadlock κατά τη δρομολόγηση σε 2-διάστατο mesh



Υπάρχει κυκλική εξάρτηση στην αίτηση για πόρους του δικτύου

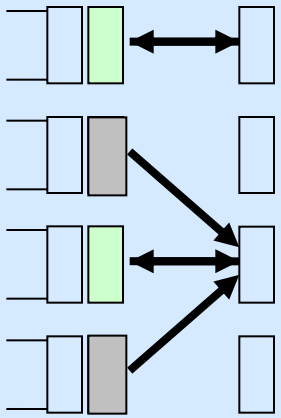
Στρατηγικές χειρισμού deadlocks

- Αποφυγή deadlock:
  - Π.χ. **DOR** (dimension-order routing) σε meshes και hypercubes (εφαρμόζει global ordering στους πόρους), **Up*/Down* routing**
- Ανάνηψη από deadlock: επιτρέπει την εμφάνιση deadlock αλλά επεμβαίνει και επιλύει την κυκλική εξάρτηση
  - Απαιτείται μηχανισμός εντοπισμού (πιθανότητας) αδιεξόδου
  - Ανάκαμψη με οπισθοδρόμηση (regressive recovery - abort-and-retry): Αφαιρεί πακέτα από την κυκλική εξάρτηση και αναμεταδίδει μετά από κάποια καθυστέρηση
  - Ανάκαμψη με πρόοδο (progressive recovery - preemptive): Αφαιρεί πακέτα από την κυκλική εξάρτηση και αναζητά εναλλακτικό δρόμο που δεν οδηγεί σε αδιέξοδο

Διαιτησία

- Εφαρμόζεται σε κάθε διακόπτη ανεξάρτητα από την τοπολογία
- Καθορίζει το *πότε* θα είναι διαθέσιμη η χρήση των μονοπατιών και απαιτείται για την επίλυση συγκρούσεων για κοινούς πόρους
- Ιδανικά:
  - Βελτιστοποίηση των συνταιριασμάτων ανάμεσα στους διαθέσιμους πόρους και τα πακέτα που τους διεκδικούν
  - Σε επίπεδο διακόπτη οι διαιτητές μεγιστοποιούν το συνταίριασμα ανάμεσα στις πόρτες εξόδου και στα πακέτα που βρίσκονται στην είσοδο
- *Προβλήματα:*
  - ***Starvation***
    - Προκύπτει όταν δεν παρέχονται ποτέ πόροι σε κάποιο πακέτο
    - *Λύση:* Απόδοση πόρων με δικαιοσύνη
- Απλές προσεγγίσεις διαιτησίας σε διακόπτες
  - Two-phased arbiters, three-phased arbiters και iterative arbiters

Διαιτησία: Two-phased vs. Three-phased arbiter

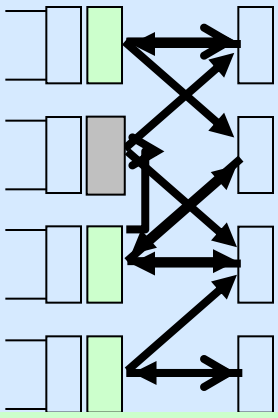


request phase

grant phase

Only two matches out of four requests (*50%* matching)

Two-phased arbiter



request phase

grant phase

accept phase

Now, three matches out of four requests (*75%* matching)

Three-phased arbiter

Μεταγωγή (switching)

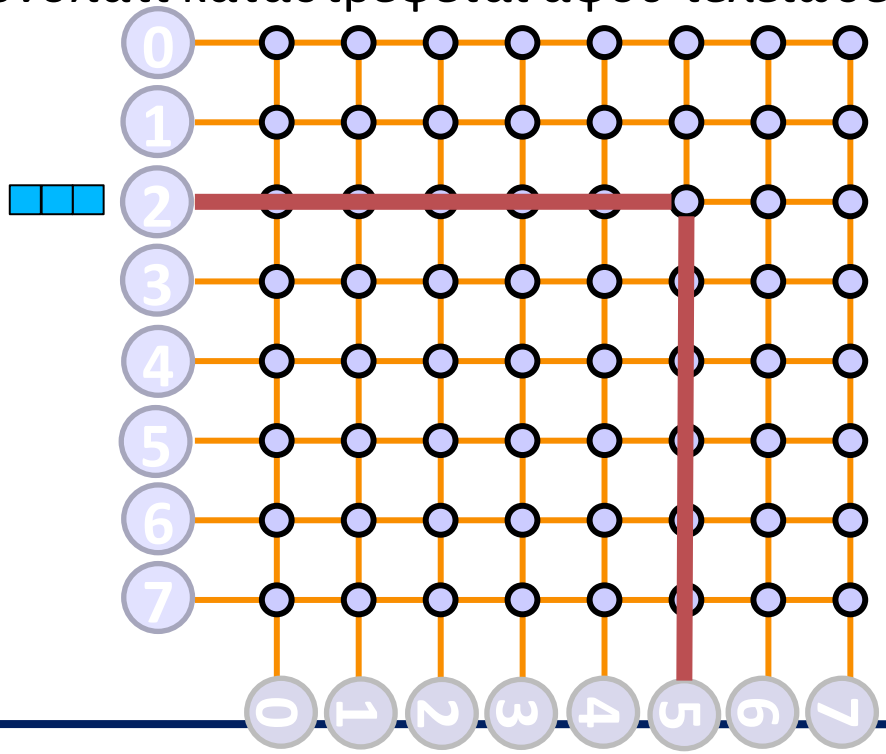
- Εφαρμόζεται σε κάθε διακόπτη ανεξάρτητα από την τοπολογία
- Εγκαθιστά τη σύνδεση των μονοπατιών για τα πακέτα και χρειάζεται για να αυξηθεί η χρησιμοποίηση των μοιραζόμενων πόρων
- Ιδανικά:
  - Εγκατάσταση σύνδεσης ανάμεσα στους πόρους του δικτύου για ακριβώς το χρονικό διάστημα που αυτοί είναι απαραίτητοι
  - Επιτρέπεται αποδοτική χρήση του bandwidth από ανταγωνιστικές ροές
- *Τεχνικές μεταγωγής:*
  - Circuit switching
    - Pipelined circuit switching
  - Packet switching
    - Store-and-forward switching
    - Cut-through switching: virtual cut-through και wormhole

Circuit switching

- Ένα μονοπάτι «κύκλωμα» δημιουργείται εξαρχής και καταστρέφεται μετά τη χρήση
- Υπάρχει η δυνατότητα μετάδοσης πολλών πακέτων μετά την εγκατάσταση της επικοινωνίας
  - *pipelined circuit switching*
- Η δρομολόγηση, η διαιτησία και η μεταγωγή πραγματοποιείται μία φορά για όλη τη σειρά των πακέτων
  - Δεν απαιτείται πληροφορία δρομολόγησης σε κάθε επικεφαλίδα πακέτου
  - Μειώνει το latency και την κατανάλωση bandwidth
- Μπορεί να σπαταλά πολύτιμο bandwidth δικτύου
  - Κατά τη δημιουργία του κυκλώματος
  - Αν δεν αποσταλούν πολλά μηνύματα μετά την εγκατάσταση του κυκλώματος

Circuit switching

- Έστω ότι θέλω να στείλω ένα μήνυμα από το 2 στο 5
- Με circuit switching:
  - Το μονοπάτι κατασκευάζεται πριν ξεκινήσει η αποστολή
    - Δημιουργείται ένα κύκλωμα από το 2 στο 5
  - Το μονοπάτι καταστρέφεται αφού τελειώσει η αποστολή

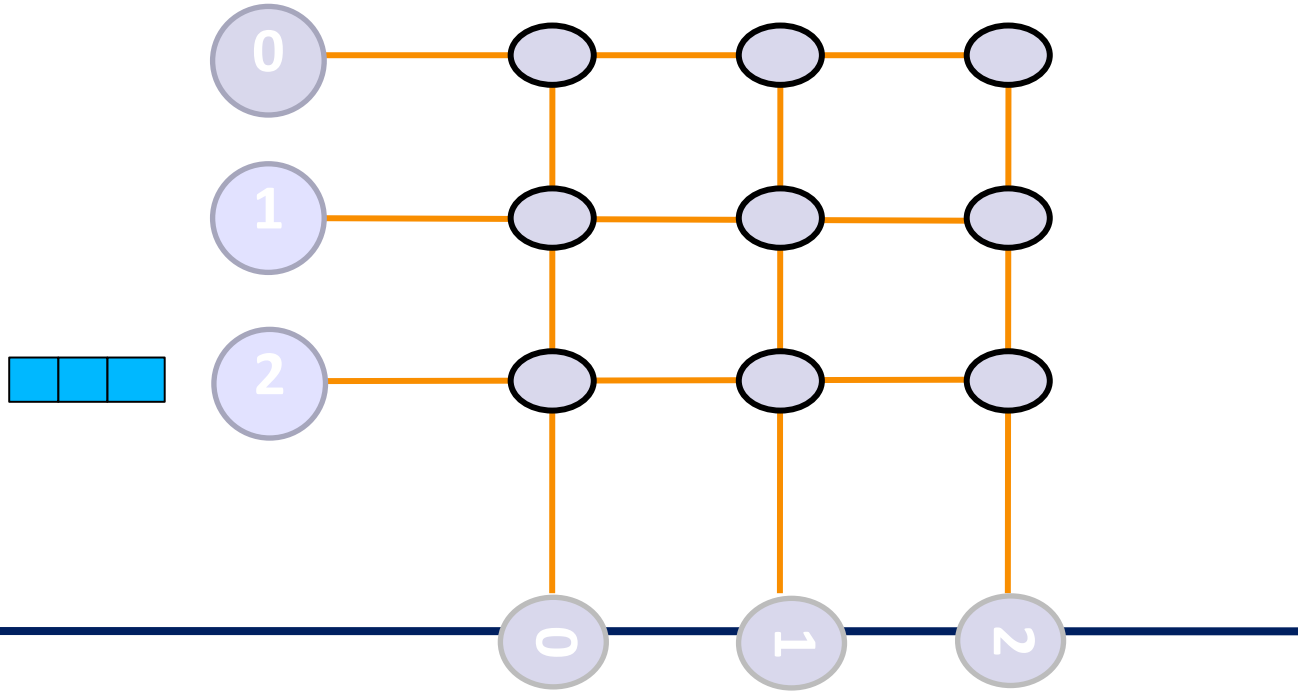


Packet switching

- Η δρομολόγηση, η διαιτησία και η μεταγωγή πραγματοποιείται για κάθε πακέτο
- Πιο αποδοτικός διαμοιρασμός των πόρων του δικτύου
- ***Store-and-forward* switching**
  - Όλα τα bits ενός πακέτου μεταδίδονται μόνο όταν όλο το πακέτο είναι έτοιμο
  - Ο χρόνος μετάδοσης πολλαπλασιάζεται με τον αριθμό των ενδιάμεσων κόμβων
- ***Cut-through* switching**
  - Bits ενός πακέτου μπορούν να προωθηθούν όταν έχει ληφθεί ολόκληρη η κεφαλίδα
  - Ο χρόνος μετάδοσης είναι αθροιστικός σε σχέση με τον αριθμό των ενδιάμεσων κόμβων
  - ***Virtual cut-through*:** έλεγχος ροής σε επίπεδο πακέτου
  - ***Wormhole***: έλεγχος ροής σε επίπεδο *fl*ow un*it* (*flit)* που είναι μικρότερη του πακέτου

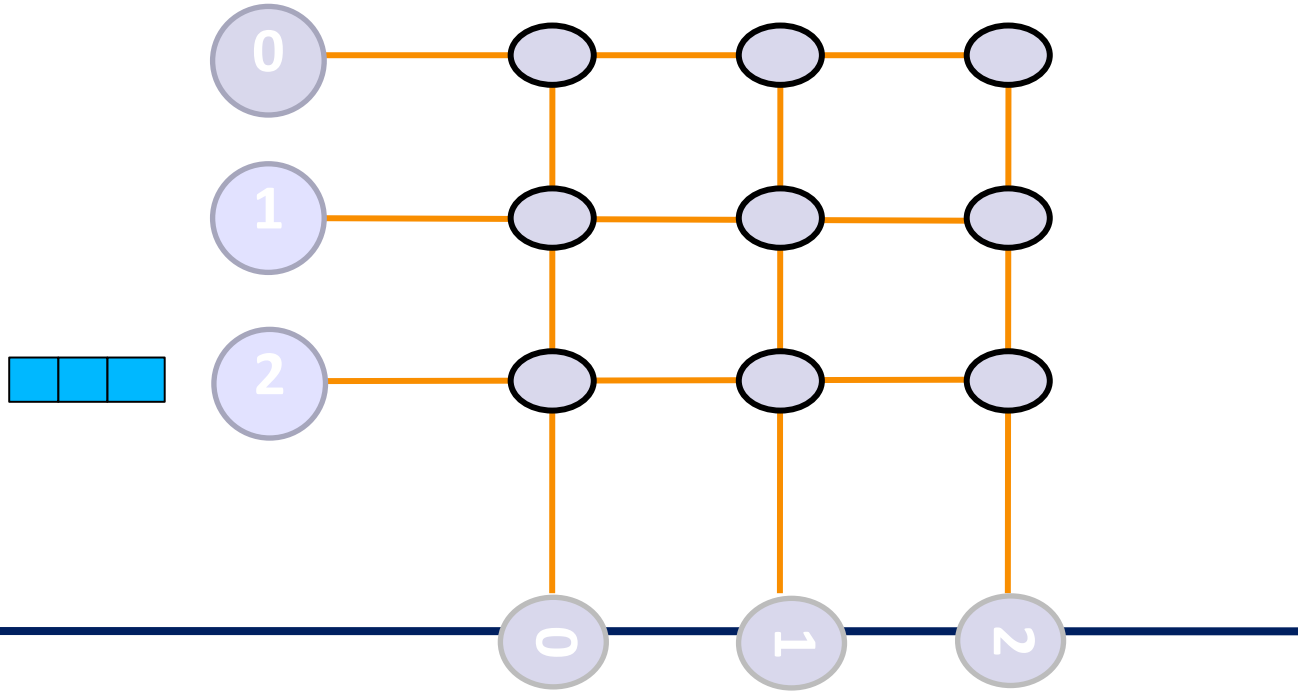
Store-and-forward switching

- Έστω ότι θέλω να στείλω ένα μήνυμα από το 2 στο 1
- Με store-and-forward switching:
  - Ένα πακέτο δρομολογείται από την αφετηρία σε έναν κόμβο του δικτύου
  - Το πακέτο αποθηκεύεται ολόκληρο σ'αυτόν τον κόμβο
  - Το πακέτο δρομολογείται στον επόμενο κόμβο
  - Η διαδικασία επαναλαμβάνεται μέχρι να φτάσει στον προορισμό



Virtual cut-through switching

- Έστω ότι θέλω να στείλω ένα μήνυμα από το 2 στο 1
- Με virtual cut-through switching:
  - Αν υπάρχει αποθηκευτικός χώρος στην έξοδο, το πρώτο πακέτο (header) δρομολογείται από την αφετηρία προς τον προορισμό
  - Τα υπόλοιπα πακέτα ακολουθούν με pipelined τρόπο



Επιπλέον διάβασμα

Computer Architecture: A Quantitative Approach, D. Patterson

Appendix E: Interconnection Networks