# 2/5/2023
# Συστήματα κατανεμημένης μνήμης (I)

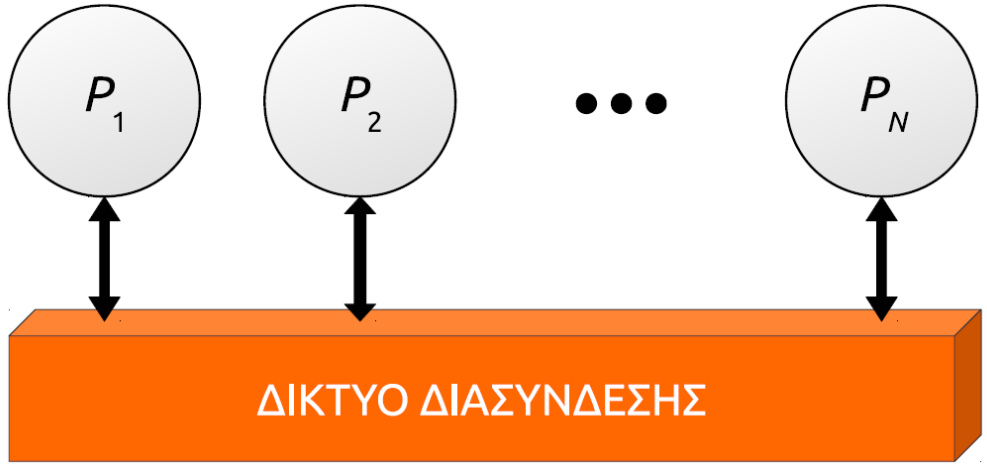
## Το δίκτυο και τα χαρακτηριστικά του

Λ8

Συστήματα & Λογισμικό Υψηλών Επιδόσεων

# Κεντρική ιδέα

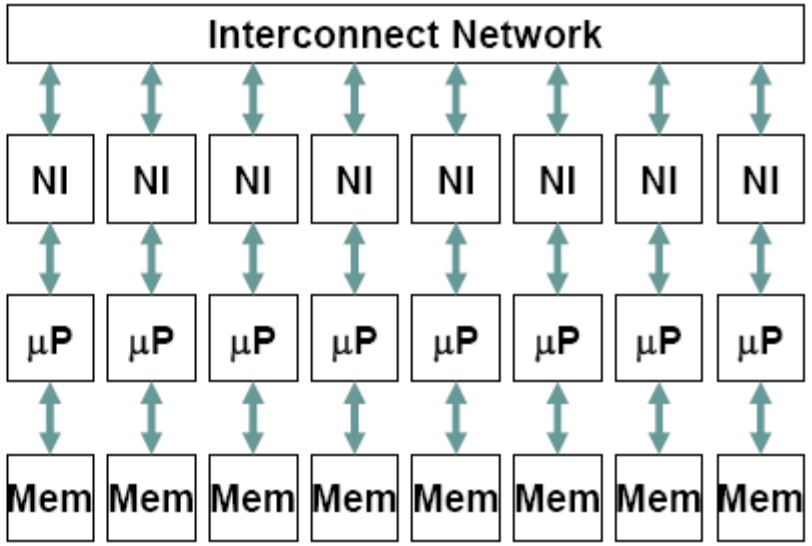
- Ανεξάρτητοι επεξεργαστές, ο καθένας με την ιδιωτική του μνήμη (κόμβος = CPU + μνήμη)



# Massively Parallel Processors (MPPs)

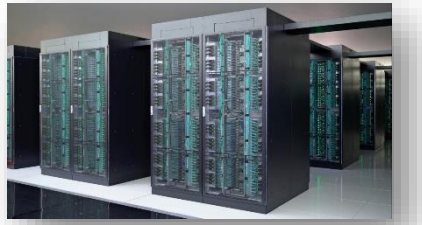
- Initial Research Projects
  - Caltech Cosmic Cube (early 1980s) using custom Mosaic processors
- Commercial Microprocessors including MPP Support
  - Transputer (1985)
  - nCube-1(1986) /nCube-2 (1990)
- Standard Microprocessors + Network Interfaces
  - Intel Paragon (i860)
  - TMC CM-5 (SPARC)
  - Meiko CS-2 (SPARC)
  - IBM SP-2 (RS/6000)
- MPP Vector Supers
  - Fujitsu VPP series

***Designs scale to 100s or 1000s of nodes***



# Clusters: παντού!

- Συλλογή από διασυνδεδεμένους «κόμβους»
  - Φτηνοί / ευρέως διαθέσιμοι επεξεργαστές (π.χ. Clusters από PCs)
  - Ο μόνος τρόπος να φτιάξουμε «οικονομικούς» υπερ-υπολογιστές (Teraflops)
  - Πολύ λίγοι έως πάρα πολλοί κόμβοι
- LUMI Finland (top500, #3)
  - 4,096 nodes (2x64-core AMD CPUs in 1536 nodes / 1x64-core AMD CPU + 4 x AMD GPUs in 2560 nodes)
  - Interconnect: Slingshot 11 (HPE/Cray)
  - Total # cores: 2,220,288
  - Κόστος: ~ $145.000.000
- RIKEN Center for Computational Science FUGAKU (top500, #2)
  - 158,976 nodes (Fujitsu A64FX 48C 2.2GHz)
  - Interconnect: TufuD (6D torus)
  - Total # cores: 7,630,848
  - Κόστος: ~ $1.000.000.000
- Oak Ridge National Laboratory FRONTIER (top500, #1)
  - 9,472 nodes (1 × 64-core AMD CPU + 4 x AMD GPUs )
  - Interconnect: Slingshot 11 (HPE/Cray)
  - Total # cores: 8,730,112
  - Κόστος: ~ $600.000.000

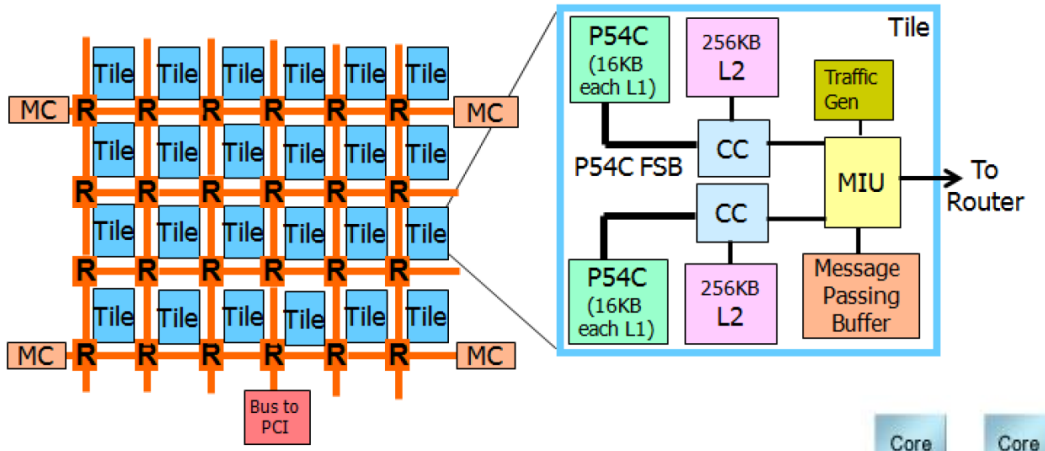
# Στο τμήμα

- Στο τμήμα μας:
  - Γενικότερο δίκτυο σταθμών εργασίας (100Mbps-1Gbps ethernet, αργό με πολύ κίνηση)
  - Αυτόνομο cluster [ παλιό ]
    - 16 κόμβοι, κάθε κόμβος 2 CPUS, κάθε CPU διπύρηνη (64 cores)
    - gigabit Ethernet
  - Αυτόνομο cluster [ νέο ]
    - 12 κόμβοι, κάθε κόμβος 1 (από 2) CPUS, κάθε CPU 8πύρηνη (96 cores)
    - gigabit Ethernet

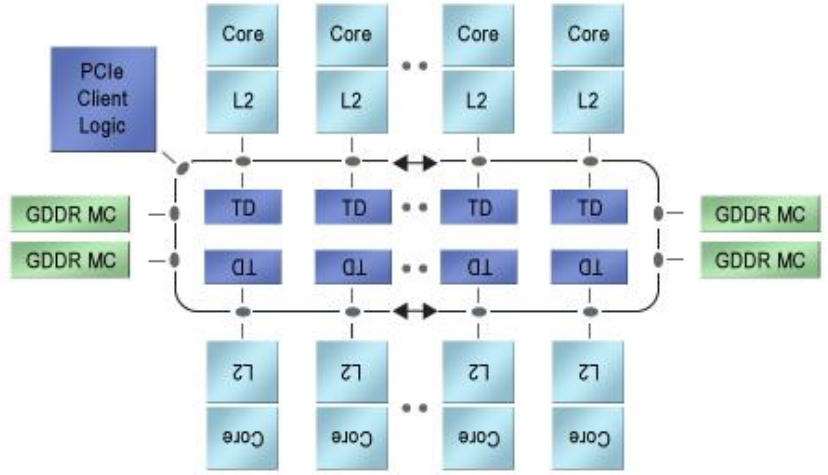

  - http://gatepc73.cse.uoi.gr:8880/ganglia/

# Manycore και NoCs

- Πάρα πολλοί πυρήνες
  - Π.χ. Intel TeraScale I (80-cores), TeraScale II (SCC, 48-cores / 24 tiles– see below)
    - Mesh network (NoC)



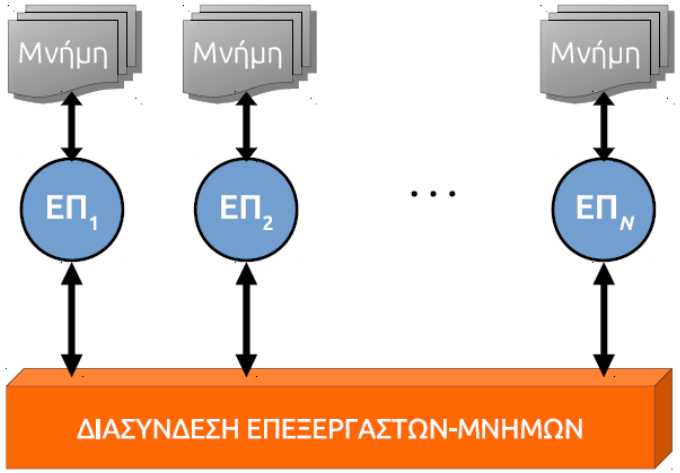

  - Intel Xeon Phi 1st gen (up to 61 cores)
    - Bidirectional Ring network



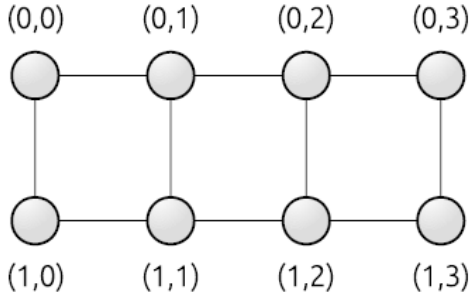
# Το δίκτυο διασύνδεσης

# Πολυεπεξεργαστές κατανεμημένης μνήμης

- Ανεξάρτητοι επεξεργαστές, ο καθένας με την ιδιωτική του μνήμη (κόμβος = CPU + μνήμη)

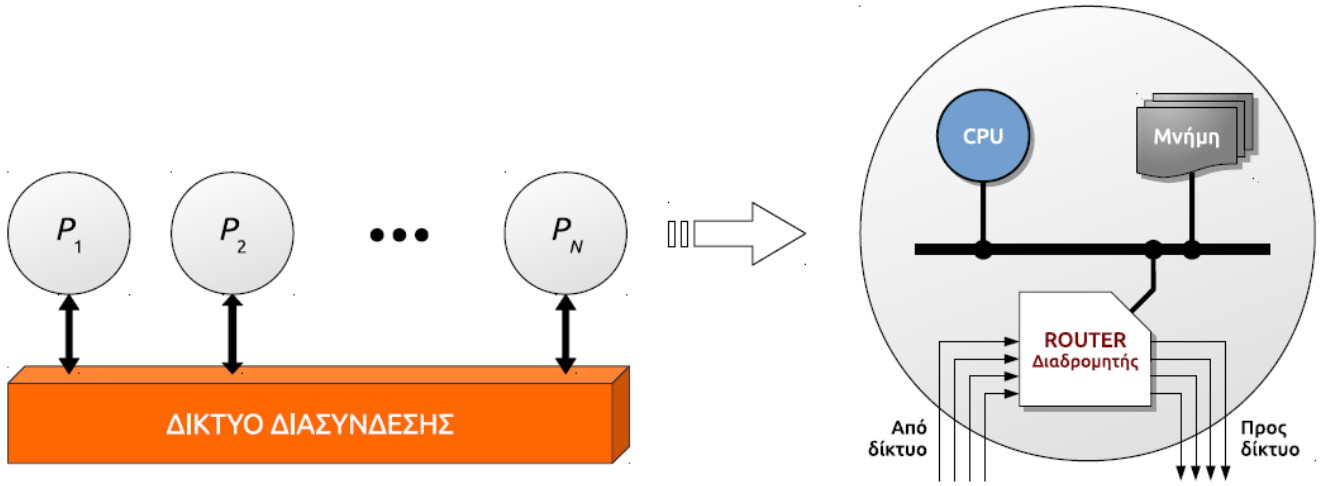



- Δίκτυο διασύνδεσης επεξεργαστών (interconnection network)
  - δίαυλος
  - δίκτυο διακοπτών
  - point-to-point, στατικό, άμεσο δίκτυο

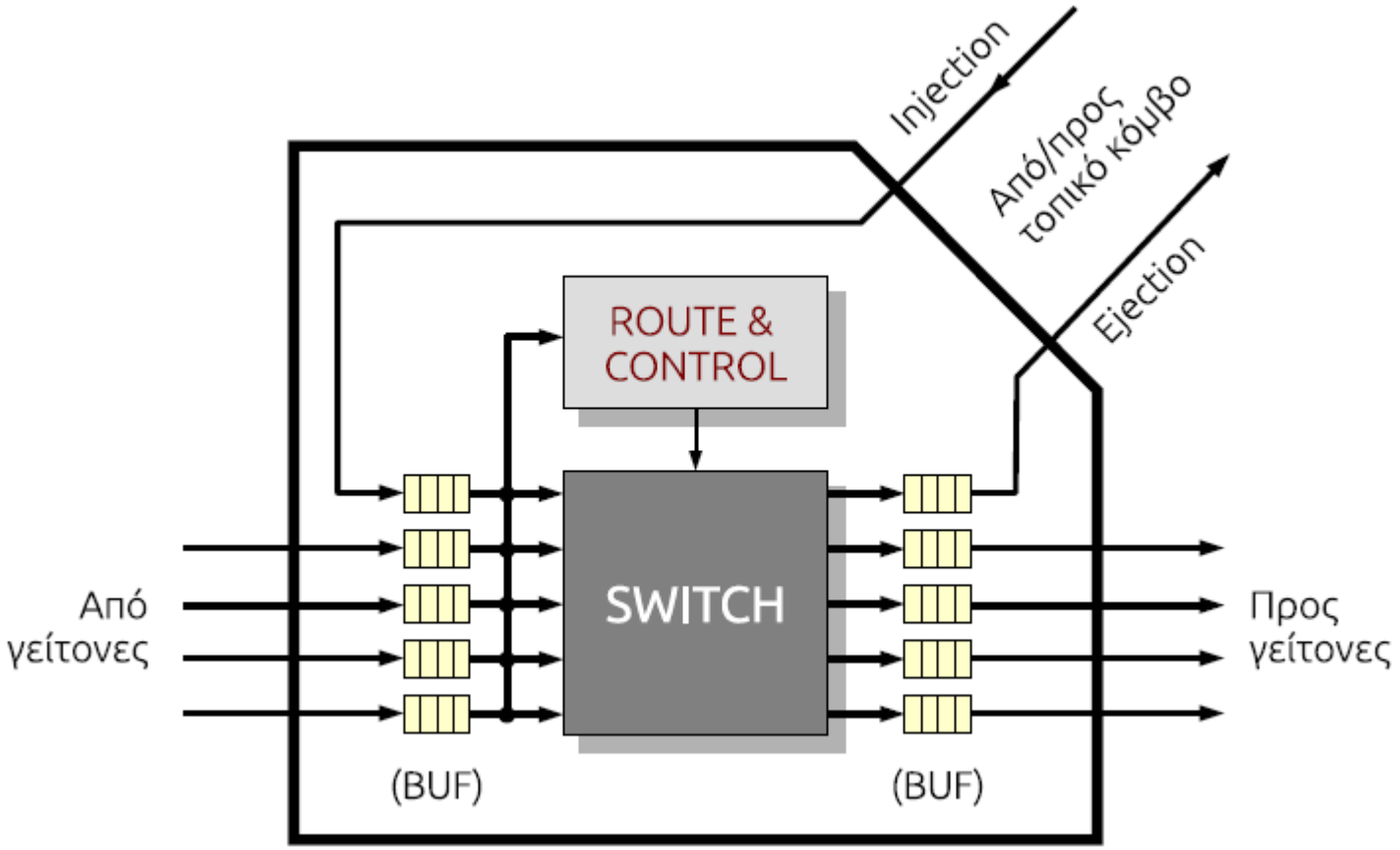


# Βασική οργάνωση

- Επικοινωνία επεξεργαστών μέσω ανταλλαγής μηνυμάτων, επάνω από το δίκτυο διασύνδεσης
- Ο διαδρομητής (router) συνδέει τον κόμβο με το δίκτυο
  - Κανάλια από προς γείτονες / τοπική μνήμη



# Τυπική δομή διαδρομητή



# Πολυϋπολογιστές

- Λόγω του ότι κάθε κόμβος είναι ουσιαστικά ένας (σχεδόν) ολοκληρωμένος και αυτόνομος υπολογιστής, οι ΠΚΜ είναι γνωστοί και ως *πολυϋπολογιστές* (multicomputers)

- Η οργάνωση μοιάζει με δίκτυο υπολογιστων
  - Διαφορές:
    - ταχύτητα
    - τοπολογία
    - λειτουργικό σύστημα
    - ...

# Ένα δίκτυο διασύνδεσης χαρακτηρίζεται από:

- Την *τοπολογία* του
  - Ποίος κόμβος συνδέεται με ποιον – χωρική διάταξη
- Τη *διαδρόμησή* του (routing)
  - Ποιο από όλα τα δυνατά μονοπάτια θα επιλεχθεί
    - Πολλές επιλογές πολιτικών
- Τον *έλεγχο ροής* του (flow control)
  - Πώς διανέμονται οι πόροι του δικτύου (κανάλια, buffers κλπ), τι συμβαίνει σε περίπτωση συγκρούσεων
    - Αρχιτεκτονική του διαδρομητή
- Τη *μεταγωγή* του (switching)
  - Πώς μεταφέρεται εσωτερικά σε έναν διαδρομητή το μήνυμα από μία είσοδο σε μία έξοδό του
    - Κυκλώματος (circuit switching)
    - Πακέτου / μηνύματος / SAF (Store-and-Forward)
    - Virtual Cut-Through (VCT)
    - Wormhole, …

# Βιβλιογραφία

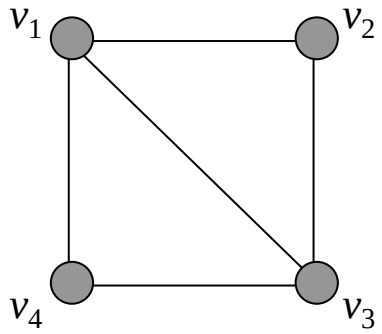

- *Interconnection networks: an engineering approach*, Duato, Yalamanchili, Ni
- *Principles and practices of interconnection networks*, W. Dally, Towles
- *Topological structure and analysis of interconnection networks*, J. Xu

# Η τοπολογία του δικτύου διασύνδεσης

# Τοπολογία

- Διάταξη των κόμβων στον χώρο και συνδεσμολογία μεταξύ τους
- Γράφοι ως φυσική αναπαράσταση του δικτύου
  - κόμβοι = κορυφές
  - συνδέσεις = ακμές
- Αναδρομή στην ορολογία και τις ιδιότητες των γράφων

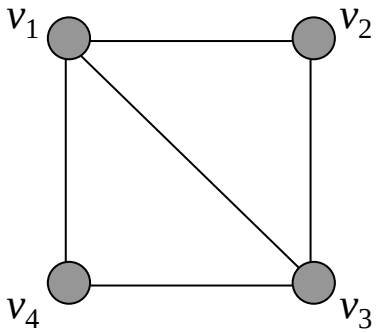
# Ορολογία και ιδιότητες των γράφων




- Σύνολο κορυφών και σύνολο ακμών: G = (V,E)
  - μη κατευθυνόμενοι
- Αν e = vu ∈ E, τότε οι v, u είναι *γειτονικές (neighbors, adjacent)*
- Η ακμή είναι *προσκείμενη (incident)* στις κορυφές

- Αν η v έχει $\beta(v)$ γείτονες, τότε έχει *βαθμό* $\beta(v)$
- Αν όλες οι κορυφές έχουν τον ίδιο βαθμό $\beta$, τότε *β-regular (τακτικός)*
- Δ(G), δ(G): μέγιστος, ελάχιστος βαθμός

# Ορολογία και ιδιότητες των γράφων

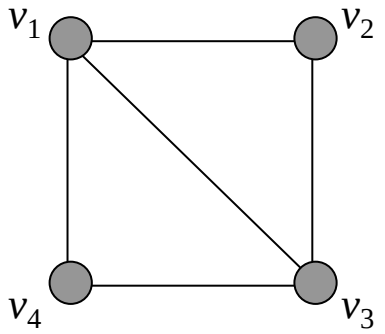



- *Περίπατος* (walk)
  - ακολουθία γειτονικών κορυφών (π.χ. από v1 στη v4: v1, v2, v3, v2, v1, v4)
- *ίχνος* (trail)
  - περίπατος χωρίς επαναλαμβανόμενες ακμές (π.χ. v1, v2, v3, v1, v4)
- *μονοπάτι* (path)
  - ίχνος χωρίς επαναλαμβανόμενες κορυφές (π.χ. v1, v2, v3, v4)

- Συνδεδεμένοι γράφοι
- *Κύκλος, Hamiltonicity*

# Αποστάσεις

- Μήκος μονοπατιού
  - ο αριθμός ακμών που περιέχει



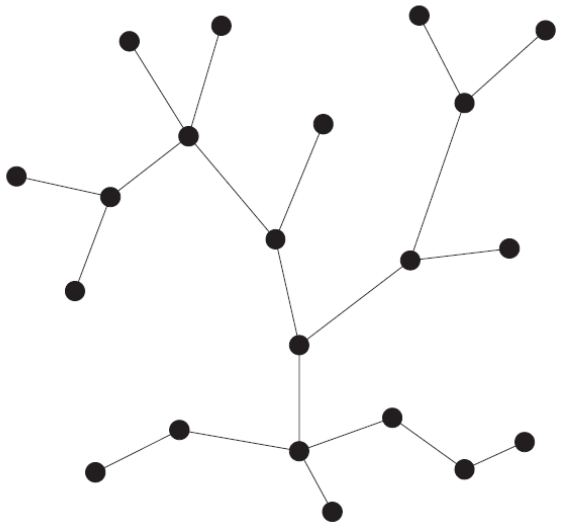

- *Απόσταση* dist(v,u)
  - το μικρότερο μήκος από όλα τα μονοπάτια v-u.

- Κορυφή u *εκκεντρική* ως προς την v: $dist(v,u) = \max_w\{v, w\}$
  - οπότε *εκκεντρικότητα* e(v) = dist(v,u)

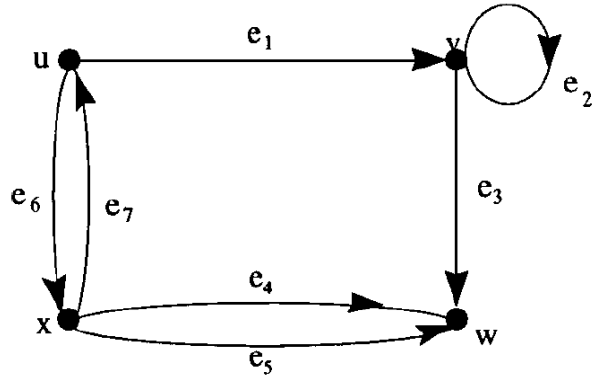
- *Διάμετρος* = η μεγαλύτερη εκκεντρικότητα, D(G) (diameter)
- *Ακτίνα* = η μικρότερη εκκεντρικότητα, R(G) (radius)

# Δέντρα

- Δέντρα (trees)
  - όχι κύκλοι
  - μοναδικά μονοπάτια
  - συνδεδεμένα, n κόμβοι, n–1 ακμές



- Κατευθυνόμενοι γράφοι
  - Οι ακμές έχουν κατεύθυνση και άρα vu ≠uv
  - out-degree (d+), in-degree (d-), balanced
  - τα άλλα όπως στους μη κατευθυνόμενους
  - ασθενώς / ισχυρά συνδεδεμένοι



# Άλλα χαρακτηριστικά των γράφων

- *vertex-disjoint paths* (ξένα ως προς τις κορυφές)
- *edge-disjoint paths* (ξένα ως προς τις ακμές)
- *vertex connectivity*, κ(G) (συνδεσμικότητα κορυφών)
- *edge connectivity*, λ(G) (συνδεσμικότητα ακμών)

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

# Τι θέλουμε από έναν δίκτυο διασύνδεσης …

- Το δίκτυο διασύνδεσης θα πρέπει να μεταφέρει *όσο το δυνατόν περισσότερα* μηνύματα, *όσο το δυνατόν γρηγορότερα* με *ελάχιστο κόστος* και *μέγιστη αξιοπιστία*. Αυτά είναι αλληλοσυγκρουόμενα, όμως.

**Τοπολογία**:

- Μικρή διάμετρος, μικρή μέση απόσταση

  *μικρή καθυστέρηση σε packet-switching, μικρή contention σε wormhole switching*

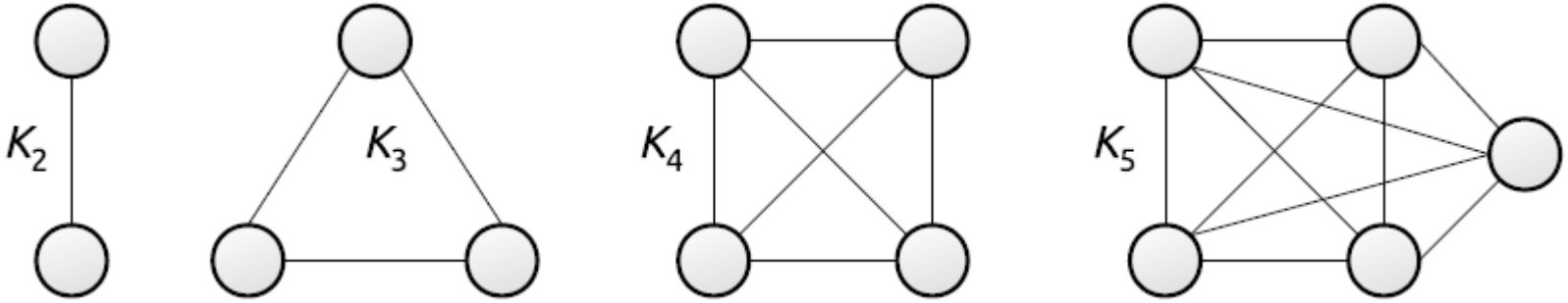
- Μικρός και σταθερός βαθμός

  *απλοί και οικονομικοί routers, μικρότερη και σταθερή καλωδίωση, χαμηλότερη connectivity, μεγαλύτερες αποστάσεις*

- Υψηλό connectivity
- Συμμετρία
- Εύκολη **ενσωμάτωση** άλλων γράφων και σε άλλους γράφους

  - Διότι αν ένα δίκτυο Α εμπεριέχεται σε ένα άλλο Β, τότε το δεύτερο θα έχει, εκτός των άλλων, και τις ιδιότητες του πρώτου
  - Διότι πολλές φορές έχουμε σχεδιάσει έναν αλγόριθμο για ένα δίκτυο Α (π.χ. υπάρχουν εξαιρετικοί αλγόριθμοι πολλαπλασιασμού πινάκων για tori) αλλά η παράλληλη μηχανή μας διαθέτει διασυνδετικό δίκτυο Β (π.χ. ο helios.cc.uoi.gr είναι υπερκύβος).

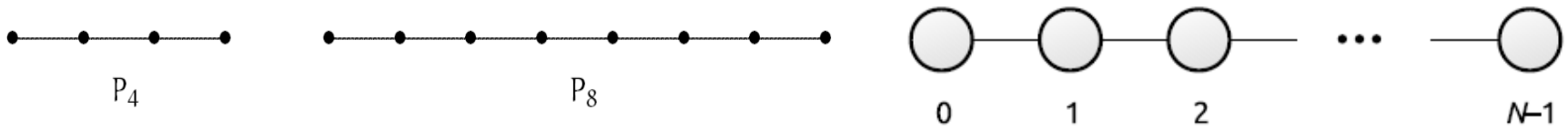
# Βασικοί γράφοι: Πλήρης γράφος (complete graph)




- Όλες οι κορυφές συνδέονται με όλες
- Ο $K_N$ έχει N κορυφές
- N(N-1)/2 ακμές
- (N-1)-regular
- $D(K_N) = 1$
- $\kappa(K_N) = N-1$
- Εμπεριέχει όλους τους γράφους με ≤ N κορυφές

- Πρακτικός μόνο για μικρό N

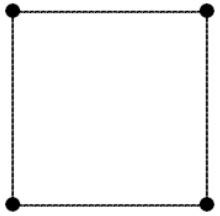
# Βασικοί γράφοι: Γραμμικός γράφος (linear array)

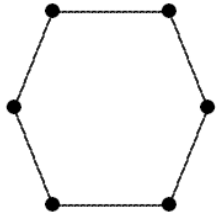


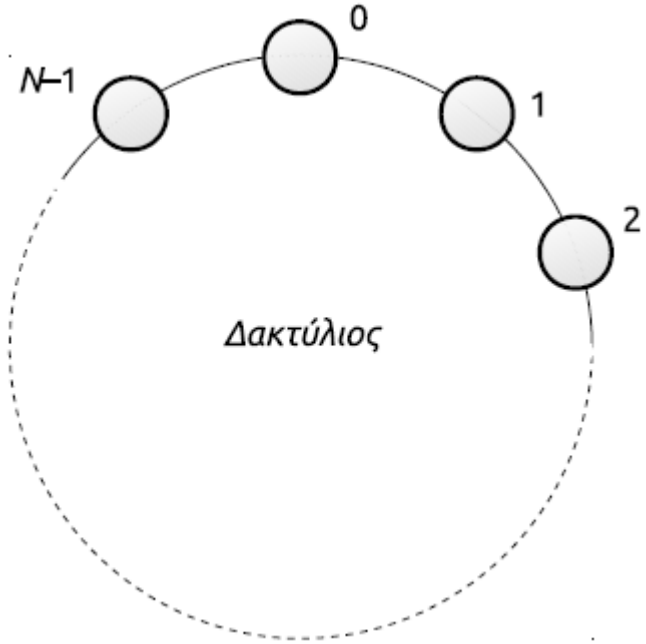

- Απλό μονοπάτι
- Ο $P_N$ έχει N κορυφές
- N-1 ακμές
- Μη τακτικός (βαθμοί 1 και 2)
- $D(P_N) = N-1$
- $\kappa(P_N) = 1$

- Μη πρακτικός – μόνο για εξειδικευμένες αρχιτεκτονικές (π.χ. συστολικές διατάξεις)

# Βασικοί γράφοι: Δακτύλιος (ring)








- Απλός κύκλος
- Ο $R_N$ έχει N κορυφές
- N ακμές
- 2-regular, συμμετρικός
- $D(R_N) = floor(N/2)$
- $\kappa(R_N) = 2$

- Μεγάλη διάμετρος, πολύ βασικός γράφος για κατασκευή άλλων τοπολογιών

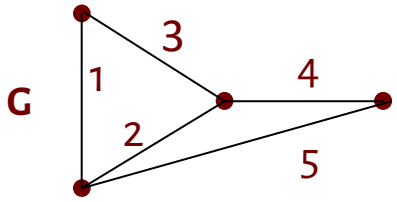
# Σχεδιασμός με βάση κριτήρια

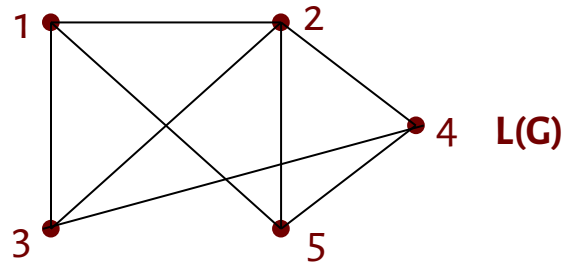
- Συνήθως θέλουμε να βρούμε κάποιο δίκτυο που πληροί κάποια κριτήρια, π.χ. να έχει συγκεκριμένο βαθμό ή συγκεκριμένη διάμετρο με συγκεκριμένο # κόμβων / ακμών
- Θέλουμε ένα μεθοδικό τρόπο να παράγουμε τέτοια δίκτυα (π.χ. ξεκινώντας από πιο απλά δίκτυα)
- Μερικές βασικές τεχνικές
  - *Γράφοι ακμών*
  - *Καρτεσιανό γινόμενο*
  - *Η μέθοδος Cayley*

» line graphs

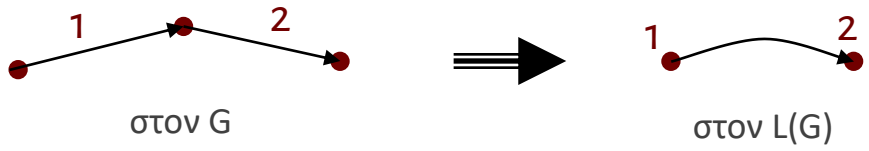
# (Δι)Γράφος ακμών (line (di)graph)

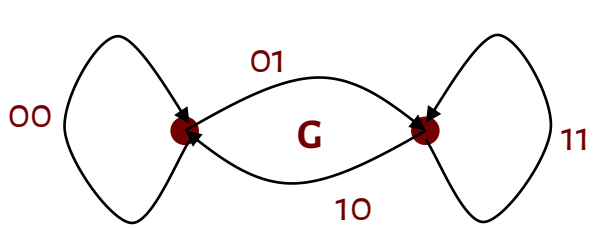
- Ξεκινώντας από έναν γράφο G, ο γράφος ακμών του, L(G), παράγεται ως εξής:
  - Κάθε κορυφή του L(G) αντιστοιχεί σε μία ακμή του G
  - Δύο κορυφές του L(G) γειτνιάζουν αν οι αντίστοιχες ακμές στον G προσπίπτουν στην ίδια κορυφή

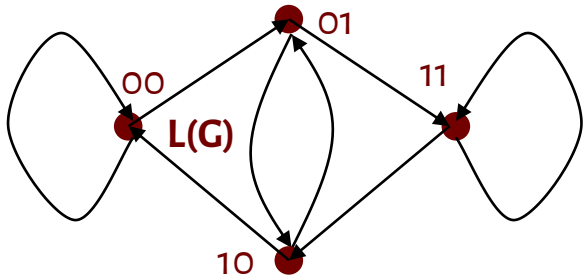



- Για κατευθυνόμενους, αντίστοιχα:
- Παράδειγμα:



# Βασικά χαρακτηριστικά των line graphs

- Έστω $L = L(G)$
- $|V(L)| = |E(G)|$
- Ακμές
  - (γράφος) $|E(L)| = \frac{1}{2} \sum_{u \in V(G)} (d(u))^2 - |E(G)|$
  - (διγράφος) $|E(L)| = \sum_{u \in V(G)} d_{in}(u) d_{out}(u)$
- Αν $e = vu \in E(G)$,
  - $d(e) = d_G(v) + d_G(u) - 2$ (γράφος)
  - $d_{in}(e) = d_{in}(v)$ και $d_{out}(e) = d_{out}(u)$ (διγράφος)
- Connectivity
  - $\kappa(G) \leq \lambda(G) \leq \kappa(L) \leq \lambda(L)$
- Διάμετρος
  - $D(G) \leq D(L) \leq D(G)+1$ (γράφος)
  - $D(L) = D(G)+1$, εκτός αν G είναι κύκλος οπότε $D(L) = D(G)$ (διγράφος)

*Περισσότερες κορυφές, μεγαλύτερος βαθμός, παρόμοια διάμετρος, κλπ.*

# Επαναλαμβανό-μενοι line digraphs

- $L^0(G) = G,\ L^1(G) = L(G)$ και

  $L^n(G) = L\left(L^{n-1}(G)\right) = \underbrace{L\left(L(\cdots L(G))\right)}_{n}$

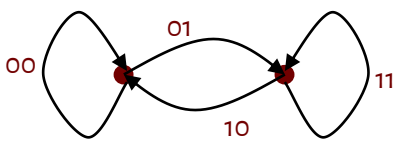
- Ιδιότητες (για ισχυρά συνδεδεμένους):
  - Αν ο G είναι k-regular, ο $L^n(G)$ είναι k-regular με $k^n|V(G)|$ κορυφές
  - $D(L^n(G)) = n + D(G)$, αν ο G δεν είναι κύκλος
  - Περίπου ίδιο connectivity
  - Κάθε κορυφή του $L^n(G)$ αντιπροσωπεύει έναν κατευθυνόμενο περίπατο στον G, μήκους n. Μας βοηθάει να σχεδιάσουμε αλγόριθμους διαδρόμησης.

# Παραδείγματα

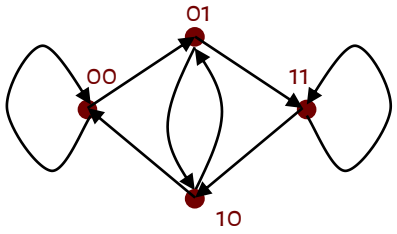
- Έστω $K^+_d$ ($d \geq 2$) ο πλήρης κατευθυνόμενος γράφος όπου σε κάθε κορυφή έχουμε βάλει και ένα loop.

- Ο διγράφος ***de Bruijn*** ορίζεται ως η (n-1)-οστή επανάληψη του γράφου ακμών του $K^+_d$:

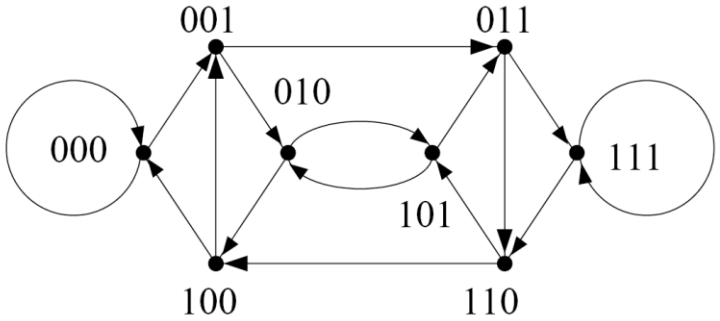
$$B(d,n) = L^{n-1}(K^+_d).$$




$B(2,1) = K^+_2$




$B(2,2) = L(B(2,1))$




$B(2,3) = L(B(2,2))$

# Καρτεσιανό γινόμενο

# Καρτεσιανό γινόμενο 2 γράφων

- Ορισμός:

$$G = (V, E) = G_1 \times G_2$$

  - Ο $G_1$ ($G_2$) ονομάζεται 1η (2η) ***διάσταση*** *(dimension)*

$$V = \{(v_1, v_2)\}$$

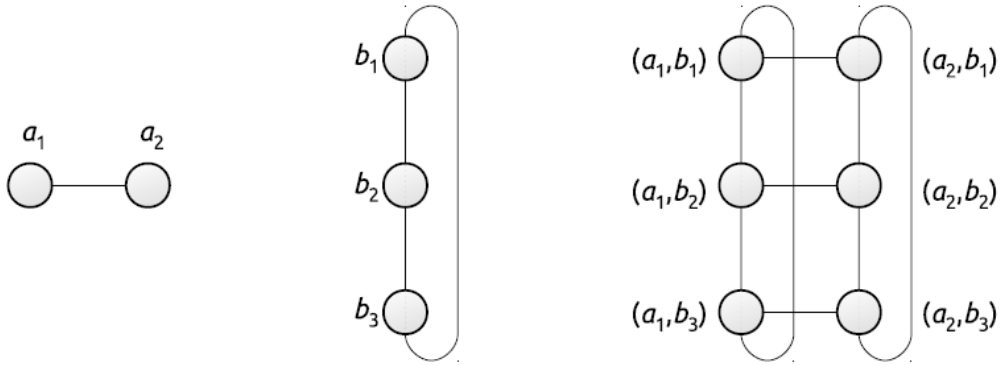
- Οι κορυφές είναι το καρτεσιανό γινόμενο των κορυφών των επιμέρους γράφων και άρα έχουν ως ετικέτα/διεύθυνση ένα ζεύγος.
  - Το πρώτο 1ο (2ο) στοιχείο του ζεύγους ονομάζεται 1η (2η) ***συντεταγμένη*** *(coordinate)*

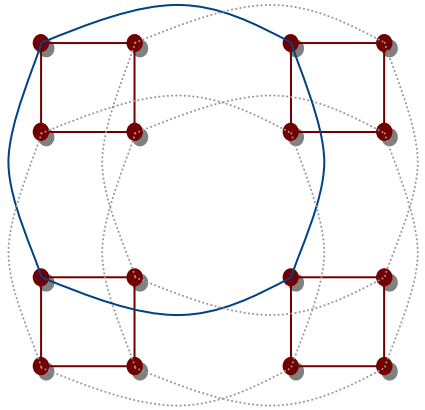
$$E = \{(v_1, v_2)(u_1, u_2) \mid v_1 = u_1 \text{ and } v_2u_2 \in E_2 \text{ OR } v_2 = u_2 \text{ and } v_1u_1 \in E_1\}$$

- Δύο κορυφές στον $G$ γειτονεύουν μόνο αν έχουν στη μία διάσταση έχουν ίδια συντεταγμένη ενώ στην άλλη έχουν γειτονικές συντεταγμένες.

# Παραδείγματα

$$E = \{(v_1, v_2)(u_1, u_2) \mid v_1 = u_1 \text{ and } v_2 u_2 \in E_2 \text{ OR } v_2 = u_2 \text{ and } v_1 u_1 \in E_1\}$$



# Γενίκευση σε *k* διαστάσεις

$$G_i = (V_i, E_i), \qquad i = 1,2, \dots, k$$

$$G = (V, E) = G_1 \times G_2 \times \cdots \times G_k = (G_1 \times G_2 \times \cdots \times G_{k-1}) \times G_k = G' \times G_k$$

$$V = V_1 \times V_2 \times \cdots \times V_k = \{(v_1, v_2, \dots, v_k) | v_i \in V_i\}$$

$$E = \{(v_1, v_2, \dots, v_k)(u_1, u_2, \dots, u_k) | \exists j : v_j u_j \in E_j \text{ and } \forall i \neq j : v_i = u_i\}$$

- Δύο κορυφές είναι γειτονικές αν και μόνο αν έχουν τις αντίστοιχες συντεταγμένες τους ίσες, *εκτός από μία* και στη συγκεκριμένη διάσταση οι συντεταγμένες τους είναι γειτονικές ($v_j u_j \in E(G_j)$)

# Μερικές ιδιότητες

$$G_1 \times G_2 = G_2 \times G_1$$

$$|V| = |V_1| \times |V_2|$$

$$v = (v_1, v_2)$$

$$\beta(v) = \beta(v_1) + \beta(v_2)$$

$$dist(v,u) = dist_1(v_1,u_1) + dist_2(v_2,u_2)$$

$$e(v) = e_1(v_1) + e_2(v_2)$$

$$D(G) = D(G_1) + D(G_2)$$

Για $k$ διαστάσεις:

$$|V| = |V_1| \times |V_2| \times \cdots \times |V_k|$$

$$v = (v_1, v_2, \ldots, v_k)$$

$$\beta(v) = \sum_{i=1}^{k} \beta(v_i)$$

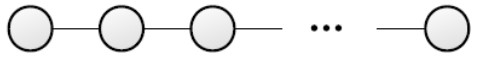
$$dist(v,u) = \sum_{i=1}^{k} dist_i(v_i,u_i)$$

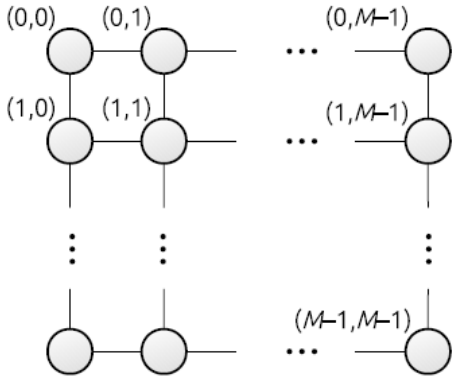
$$e(v) = \sum_{i=1}^{k} e_i(v_i)$$

$$D(G) = \sum_{i=1}^{k} D(G_i)$$

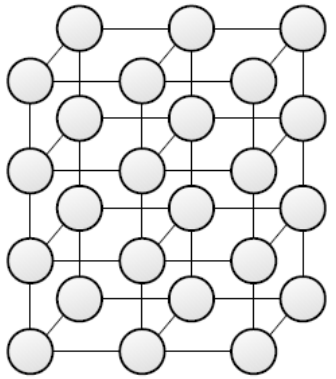
# Γνωστά δίκτυα ως καρτεσιανά γινόμενα: Πλέγματα

- Πλέγματα: γινόμενα γραμμικών γράφων






*2D πλέγμα M x M*

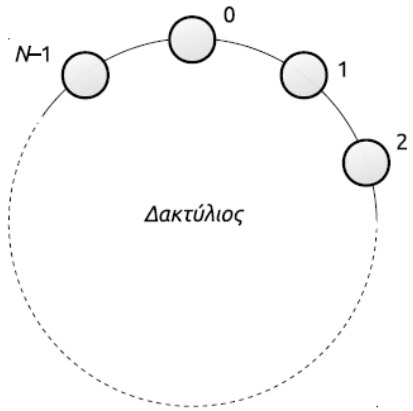


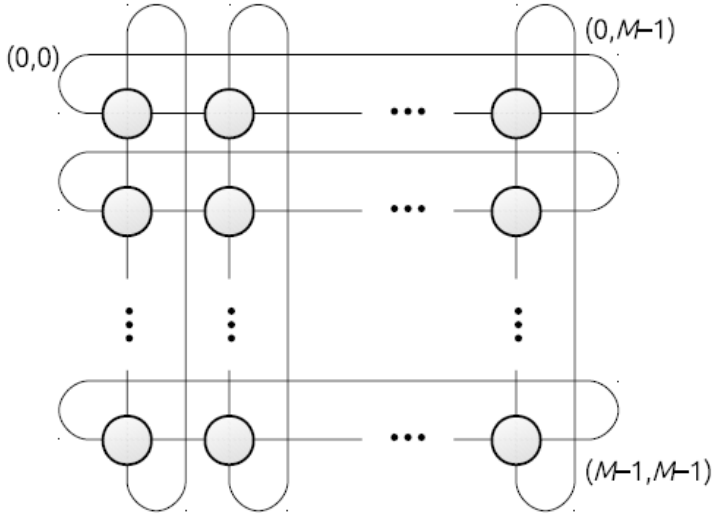
*3D πλέγμα 4 x 3 x 2*

- Σε τετραγωνικό πλέγμα 2D, $N = M$ x $M$:
  - Βαθμός: 2 (γωνίες), 3 (ακμές), 4 (όλοι οι άλλοι κόμβοι)
  - Διάμετρος: $2(M-1) \approx 2(N)^{1/2}$

# Γνωστά δίκτυα ως καρτεσιανά γινόμενα: Tori

- Tori: γινόμενα δακτυλίων




*2D torus M x M*

- Δύο κόμβοι $(x_1, x_2, ..., x_d)$ και $(y_1, y_2, ..., y_d)$ συνδέονται μόνο εφόσον

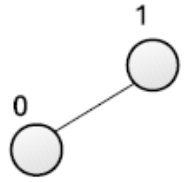
$$\sum_{i=1}^{d} |x_i - y_i| = 1$$

όπου η αφαίρεση είναι mod $|V_i|$.

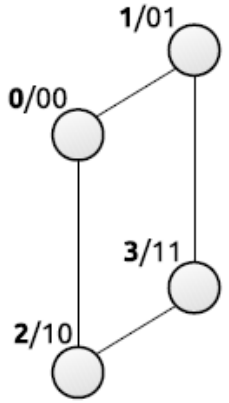
- Σε τετραγωνικό torus 2D, $N = M \times M$:
  - Τακτικός, συμμετρικός γράφος, βαθμός: 4
  - Διάμετρος: 2 x floor($M$/2) ≈ $N^{1/2}$

# Γνωστά δίκτυα ως καρτεσιανά γινόμενα: Υπερκύβοι

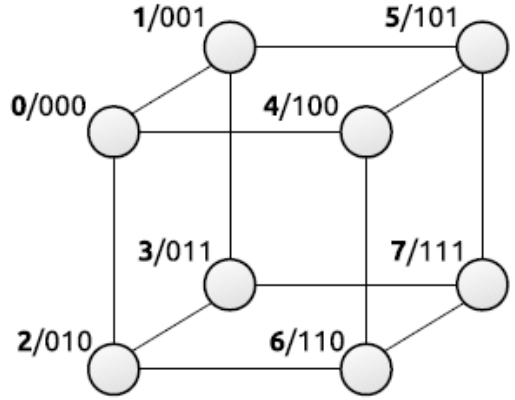
- Καρτεσιανό γινόμενο από γράφους 2 κόμβων
  - $K_2 = L_2 = R_2 = \text{whatever}_2$




*μονοδιάστατος κύβος* – $Q_1$




*διδιάστατος κύβος* – $Q_2$




*τριδιάστατος κύβος* – $Q_3$

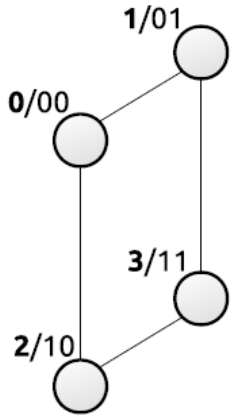
# Υπερκύβος: κι άλλοι ορισμοί

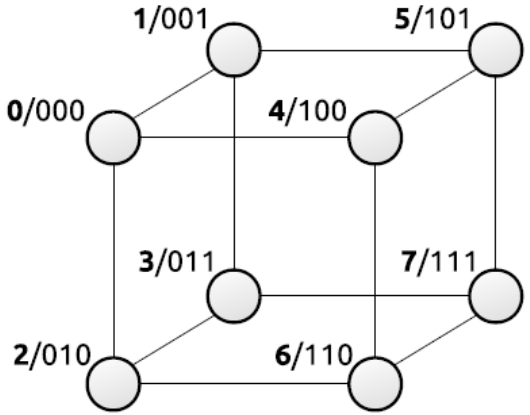
- Καρτεσιανό γινόμενο από $P_2$, $R_2$, $K_2$ ...

- Επίσης, είναι ιεραρχικά αναδρομικός (καρτεσιανό γινόμενο από μικρότερους κύβους)

$$Q_1=K_2,\ Q_d=Q_{d-1}\times Q_1=Q_k\times Q_{d-k}$$

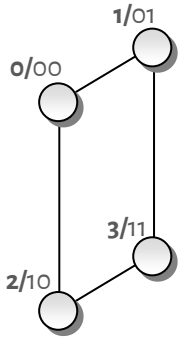
- Ισοδύναμος ορισμός, ενίοτε βολικότερος:
  - $N = 2^d$ κόμβοι με ετικέτες $d$-ψήφιους δυαδικούς αριθμούς.
  - *Δύο κόμβοι γειτνιάζουν αν και μόνο αν οι ετικέτες τους διαφέρουν σε 1 bit.*



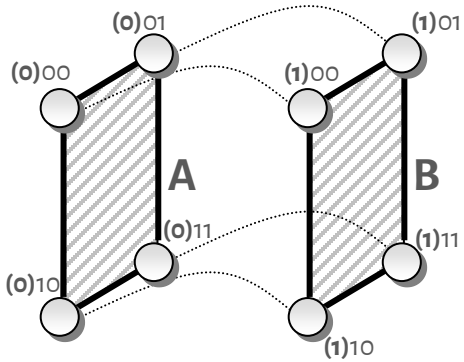



# Κατασκευή με βάση τον τελευταίο ορισμό

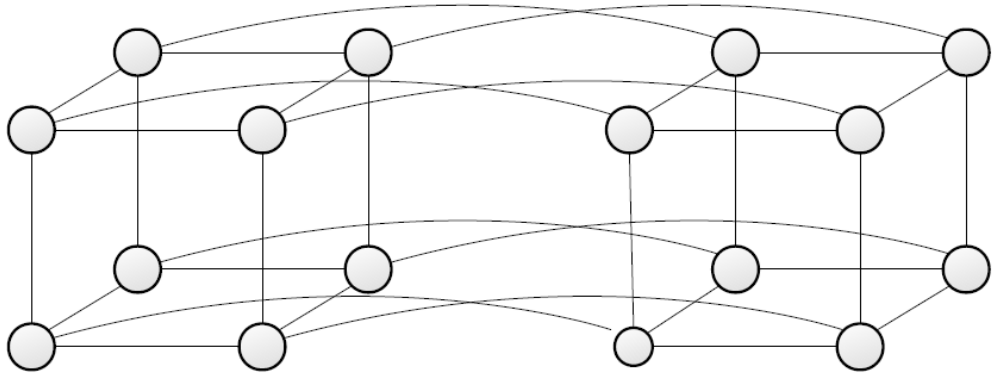
- Δύο ίδια αντίγραφα του αμέσως μικρότερου κύβου
- Συνδέω τις κορυφές με ίδια ετικέτα
- Στον πρώτο βάζω το 0 μπροστά από κάθε ετικέτα, στο δεύτερο βάζω το 1.




Διδιάστατος κύβος ($Q_2$)




Κατασκευή από δύο διδιάστατους κύβους



*τετραδιάστατος κύβος* – $Q_4$

# Μερικές ιδιότητες

- Τακτικός, βαθμού $d$ ( $= \log_2 N$)
- Διάμετρος $D(Q_d) = d$
- Hamiltonian, vertex symmetric, edge symmetric
- Βέλτιστη συνδεσμικότητα ($d$) (τόσα παράλληλα μονοπάτια)
- Γενικώς, βέλτιστες είναι πάρα πολλές από τις ιδιότητές του, βέλτιστα συμπεριφέρονται πάρα πολλοί αλγόριθμοι.
  - Γι' αυτό και ήταν από τα δημοφιλέστερα / πιο μελετημένα δίκτυα

- Σημαντικότερα μειονεκτήματα:
  - Καλή, αλλά όχι και τέλεια διάμετρος, αλλά κυρίως:
  - Μεγάλος, μη σταθερός βαθμός
  - Γι' αυτό και (σε συνδυασμό με άλλες εξελίξεις) υπερσκελίστηκε από άλλα δίκτυα χαμηλού και σταθερού βαθμού (π.χ. πλέγματα).

# Υπερκύβος

- Δίκτυο που μονοπωλούσε το ενδιαφέρον παλαιότερα. Τώρα το ενδιαφέρον μοιράζεται και σε άλλα δίκτυα με έμφαση στον χαμηλό βαθμό.
- Μερικά cube-like δίκτυα (για μείωση διαμέτρου ή/και μείωση βαθμού):
  - Folded cubes
  - Crossed cubes
  - Reduced cubes
  - Hierarchical cubes
  - Twisted cubes
  - Dual cubes
  - …

# Συγκριτικός πίνακας

| | Κόμβοι | Βαθμός | Διάμετρος | Παραδείγματα συστημάτων |
|---|---|---|---|---|
| **Πλήρης γράφος** | $N$ | $N-1$ | 1 | |
| **Γραμμ. γράφος** | $N$ | $1, 2$ | $N-1$ | |
| **Δακτύλιος** | $N$ | 2 | $\lfloor N/2 \rfloor$ | Sequent Symmetry, Intel Xeon Phi |
| **Πλέγμα 2D** | $N = M \times M$ | $2, 3, 4$ | $2M - 2 \approx 2\sqrt{N}$ | Intel Paragon Adapt. Epiphany |
| **Πλέγμα 3D** | $N = M \times M \times M$ | $2, 3, \ldots, 6$ | $3M - 3 \approx 3\sqrt[3]{N}$ | MIT J Machine, Sandia Red Storm |
| **Torus 2D** | $N = M \times M$ | 4 | $2\lfloor M/2 \rfloor \approx \sqrt{N}$ | Cray X1E |
| **Torus 3D** | $N = M \times M \times M$ | 6 | $3\lfloor M/2 \rfloor \approx \sqrt[3]{N}$ | Cray XT3/4/5/6 Cray XE6 |
| **Υπερκύβος** | $N = 2^d$ | $d = \log_2 N$ | $d = \log_2 N$ | Intel iPSC-1/2, nCUBE-1/2/3, SGI Origin |

# Μεταγωγή (switching)

# Μεταγωγή

- Ενώ ο έλεγχος ροής φυσικού μέσου μεταφέρει bits μεταξύ δύο διαδρομητών, η *μεταγωγή (switching) ενώνει εσωτερικά σε έναν διαδρομητή το κανάλι εισόδου με το επιλεγμένο κανάλι εξόδου και μεταφέρει δεδομένα*.
  - Καθορίζει το *πώς* και το *πότε* θα γίνει η σύνδεση αυτή
  - Μπορεί να γίνει στιγμιαία, για μικρό ή για μεγάλο χρονικό διάστημα
  - Μπορεί να γίνει αφού αποφασιστεί το κανάλι εξόδου, δηλαδή αφού ολοκληρωθεί η λειτουργία της διαδρόμησης στον router, κατά τη διάρκειά της ή ακόμα και πριν (!)
  - Γενικά είναι ο μηχανισμός που εσωτερικά στον διαδρομητή προωθεί τα bits από μία είσοδο σε μία προκαθορισμένη έξοδο (η επιλογή της εξόδου δεν είναι αρμοδιότητα της μεταγωγής αλλά της λειτουργίας της διαδρόμησης).
  - Υψηλές επιδόσεις: *χρονικά, αν γίνεται, να υπάρχει επικάλυψη με τις υπόλοιπες λειτουργίες του διαδρομητή*.

# Τυπική δομή διαδρομητή

Control/Arbitration

Τμήμα που αποφασίζει τι θα συμβεί στην περίπτωση συγκρούσεων (π.χ. δύο πακέτα εισόδου πρέπει να πάνε στο ίδιο κανάλι εξόδου)

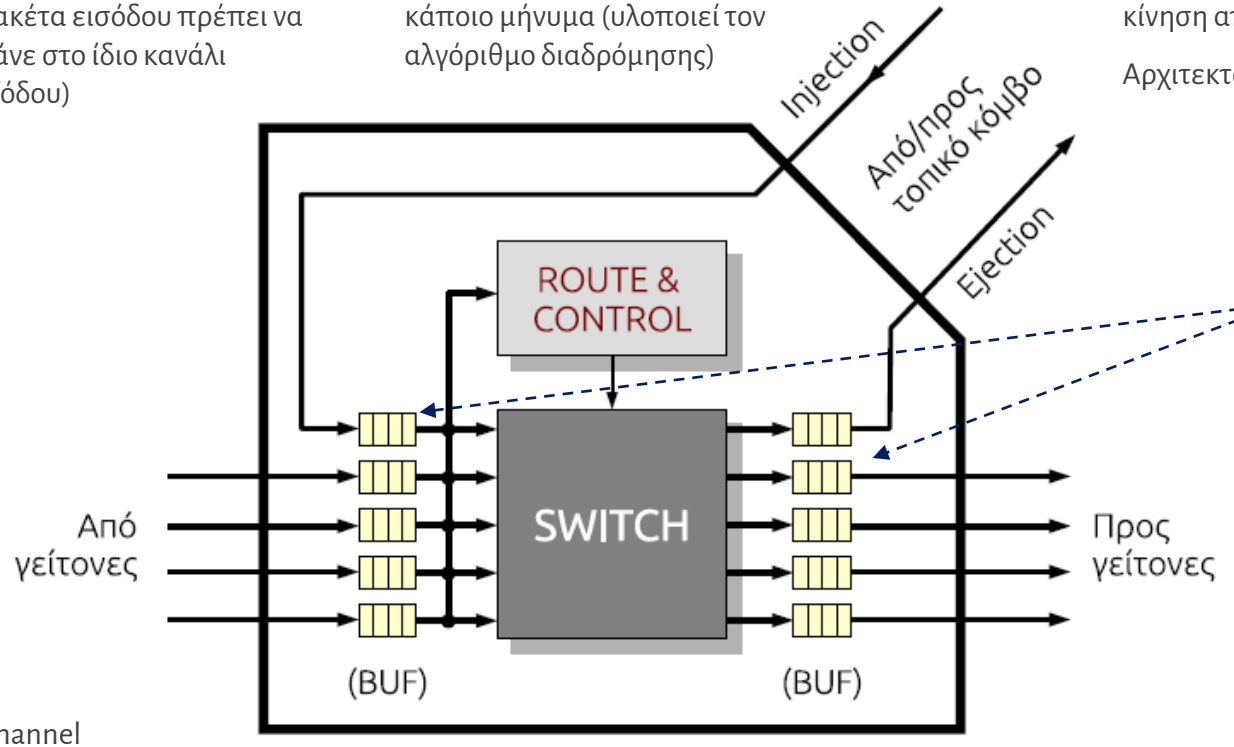
Routing

Τμήμα που αποφασίζει το κανάλι εξόδου στο οποίο θα συνδεθεί ένα κανάλι εισόδου για να προωθηθεί κάποιο μήνυμα (υλοποιεί τον αλγόριθμο διαδρόμησης)

Injection / Ejection

Κανάλια που μεταφέρουν μηνύματα από / προς τον τοπικό κόμβο (δηλαδή εισάγουν κίνηση στο / αφαιρούν κίνηση από το δίκτυο).

Αρχιτεκτονικές 1-port, k-port, all-port




Buffers (Αν υπάρχουν!)

Χρησιμεύουν για την προσωρινή αποθήκευση των πακέτων, πριν προχωρήσουν στον επόμενο διαδρομητή. Μπορεί να μην υπάρχουν καθόλου (BUFFERLESS), να υπάρχουν μόνο στις εξόδους (OUTPUT BUFFERED/QUEUED), μόνο στις εισόδους (INPUT BUFFERED/QUEUED) ή και στις δύο μεριές, όπως εδώ, αλλά μπορεί και να υπάρχουν κοινόχρηστοι buffers (SHARED BUFFERS).

Channel

Αποτελείται από το μέσο (π.χ. καλώδιο), buffers και επιπλέον κυκλώματα ελέγχου (link controllers)

Switch

Συνδέει τα κανάλια εισόδου με τα κανάλια εξόδου. Συνήθως είναι διακόπτης crossbar εφόσον τα κανάλια δεν είναι πάρα πολλά. Μπορεί όμως να αποτελείται από σύνολο διακοπτών, όπως π.χ. multistage δίκτυο ή ακόμα και point-to-point δίκτυο από διακόπτες.

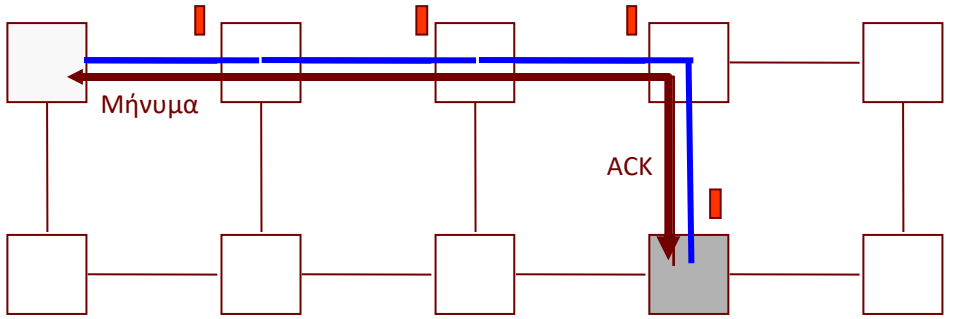
# Μεταγωγή

- Πώς μεταφέρονται μηνύματα από ένα κανάλι εισόδου σε ένα κανάλι εξόδου στον ίδιο κόμβο.

- Μερικές τεχνικές μεταγωγής:
  - Κυκλώματος (circuit switching)
  - Πακέτου / μηνύματος / SAF (Store-and-Forward)
  - Virtual Cut-Through (VCT)
  - Wormhole
  - Virtual channels
  - Pipelined circuit switching
  - …

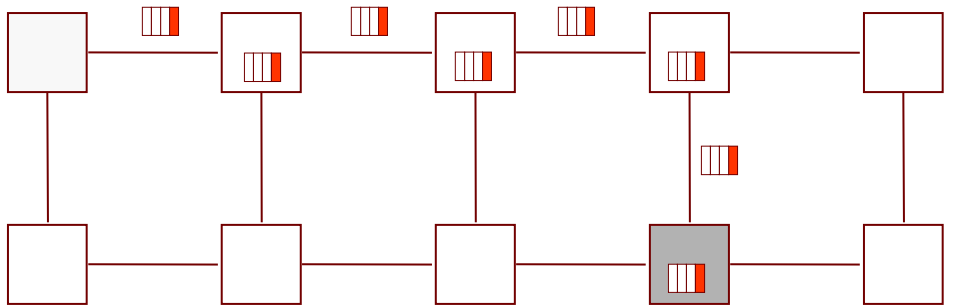
# Μεταγωγή κυκλώματος

- Τρεις φάσεις:
  - σχηματισμός (και δέσμευση) του μονοπατιού από το probe
  - μεταφορά του μηνύματος
  - αποδέσμευση του μονοπατιού



# Μεταγωγή SAF

- Πακέτου / μηνύματος / SAF (Store-and-Forward)
  - Το μήνυμα χωρίζεται σε πακέτα σταθερού μήκους
  - Κάθε πακέτο προωθείται ανεξάρτητα.
  - Οι κόμβοι
    - (α) το λαμβάνουν και το αποθηκεύουν σε buffer και
    - (β) το προωθούν στον επόμενο κόμβο

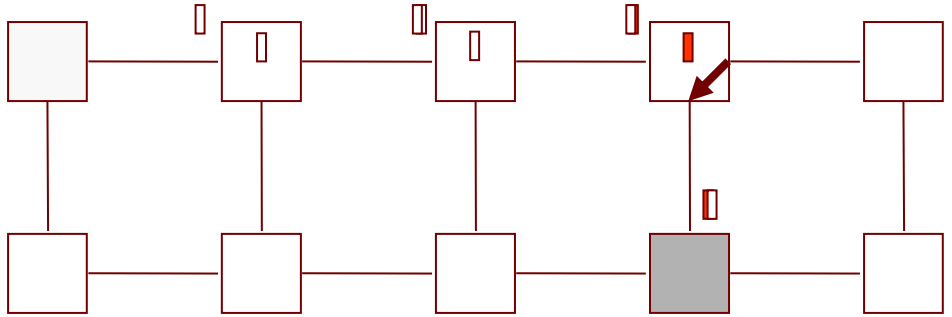


# Μεταγωγή VCT

- Σαν το SAF αλλά:
  - Αν το κανάλι εξόδου είναι ελεύθερο, καθώς λαμβάνονται τα bits της επικεφαλίδας, αποφασίζεται το κανάλι εξόδου και όλο το μήνυμα διοχετεύεται κατευθείαν εκεί (άρα ελάχιστη καθυστέρηση).
  - Αν όχι, buffering όπως στο SAF.
  - Ταχύτητα αν δεν υπάρξει εμπόδιο
  - Όμως, δεν εξαλείφεται η ανάγκη για buffers που έχει και το SAF.

# Μεταγωγή wormhole

- Ανάμεσα σε VCT και circuit switching. Το μήνυμα χωρίζεται σε ΠΟΛΥ μικρά πακέτα, τα *flits* (1-4 bytes). Το πρώτο αποτελεί την επικεφαλίδα
- Η επικεφαλίδα προχωρά με VCT αλλά τα υπόλοιπα flits ακολουθούν (και δεσμεύουν) τους προηγούμενους κόμβους, χωρίς κενά, σαν σε pipeline.
- Αν η επικεφαλίδα μπλοκάρει κάπου, τα flits αποθηκεύονται *εκεί που βρίσκονται* (άρα πολύ μικροί buffers απαιτούνται), εμποδίζοντας με τη σειρά τους άλλα flits να περάσουν.

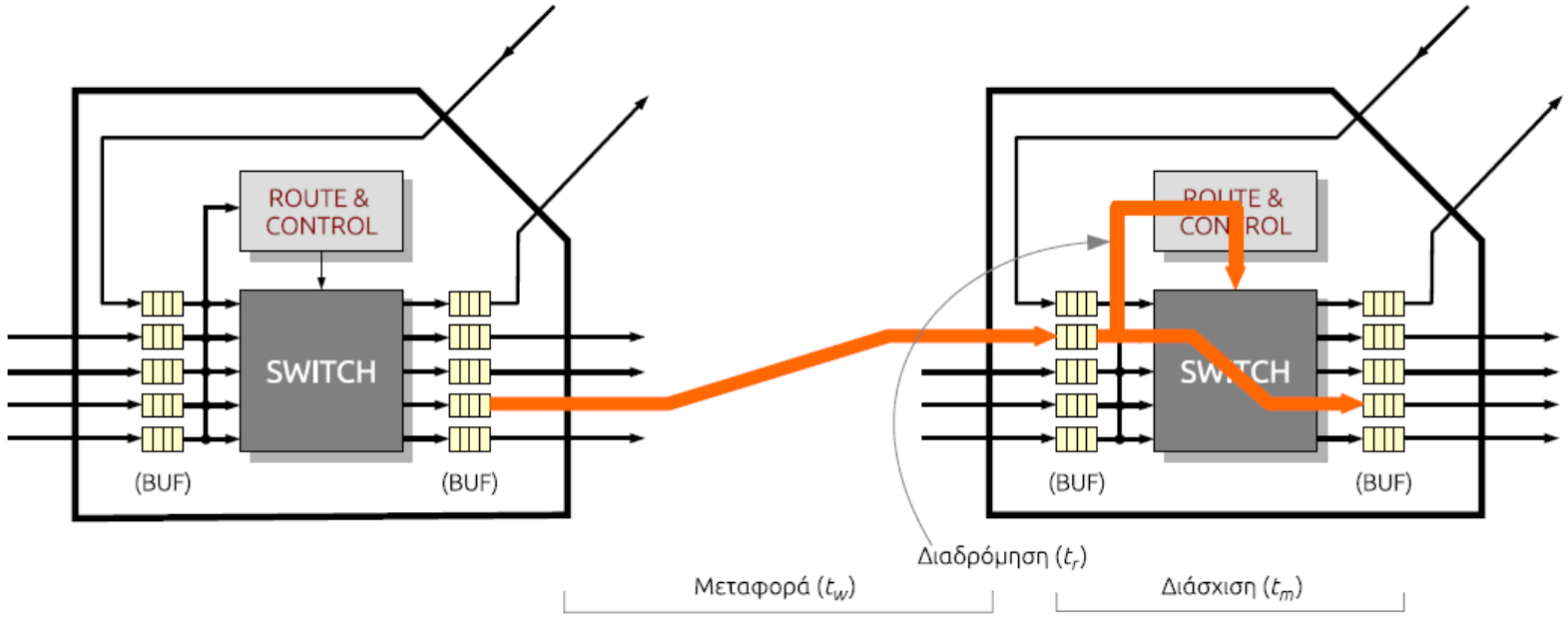


# Ας υποθέσουμε ότι …

- Κάθε πακέτο αποτελείται από **1 flit** επικεφαλίδας και **M flits** δεδομένων
  - Σύνολο: **M+1 flits** για το πλήρες πακέτο
- Το μήνυμα πρέπει να διανύσει μονοπάτι μήκους **D** για να φτάσει στον προορισμό του
- *Δεν συναντάει κανένα εμπόδιο (δηλαδή αναμονή λόγω κατειλημμένων καναλιών) στον δρόμο του*

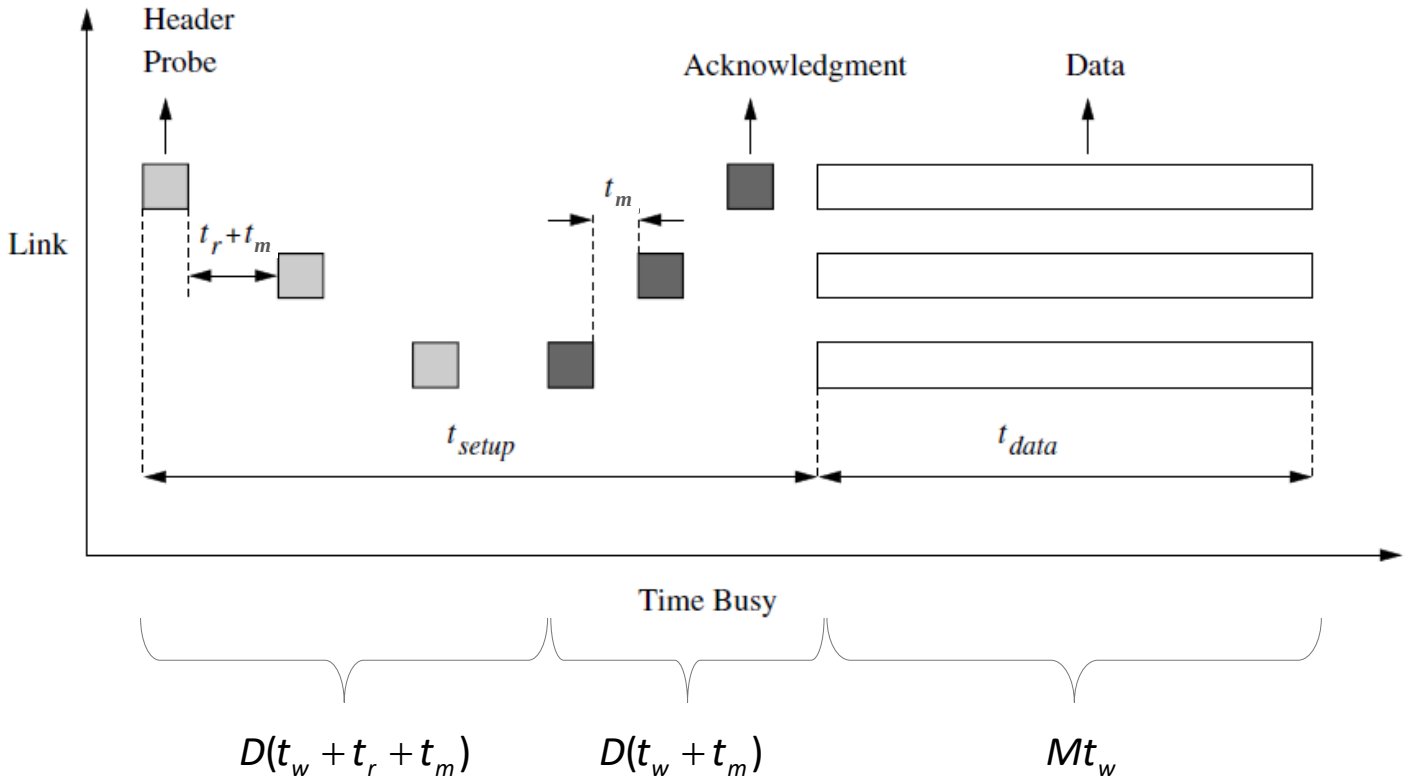
- Κανάλια με συχνότητα **B** Hz
- Το κανάλι έχει *πλάτος* 1 *phit* (= #bits που μεταφέρει ταυτόχρονα σε 1 κύκλο)
  - Υποθέτουμε 1 phit = 1 flit
  - Δηλαδή χωρητικότητα / ρυθμός μεταφοράς: **B flits / sec**.
  - Χρόνος ***μεταφοράς***, για να διασχίσει ένα flit το κανάλι: $\mathbf{t_w = 1/B}$ **sec**.

# Χρόνοι



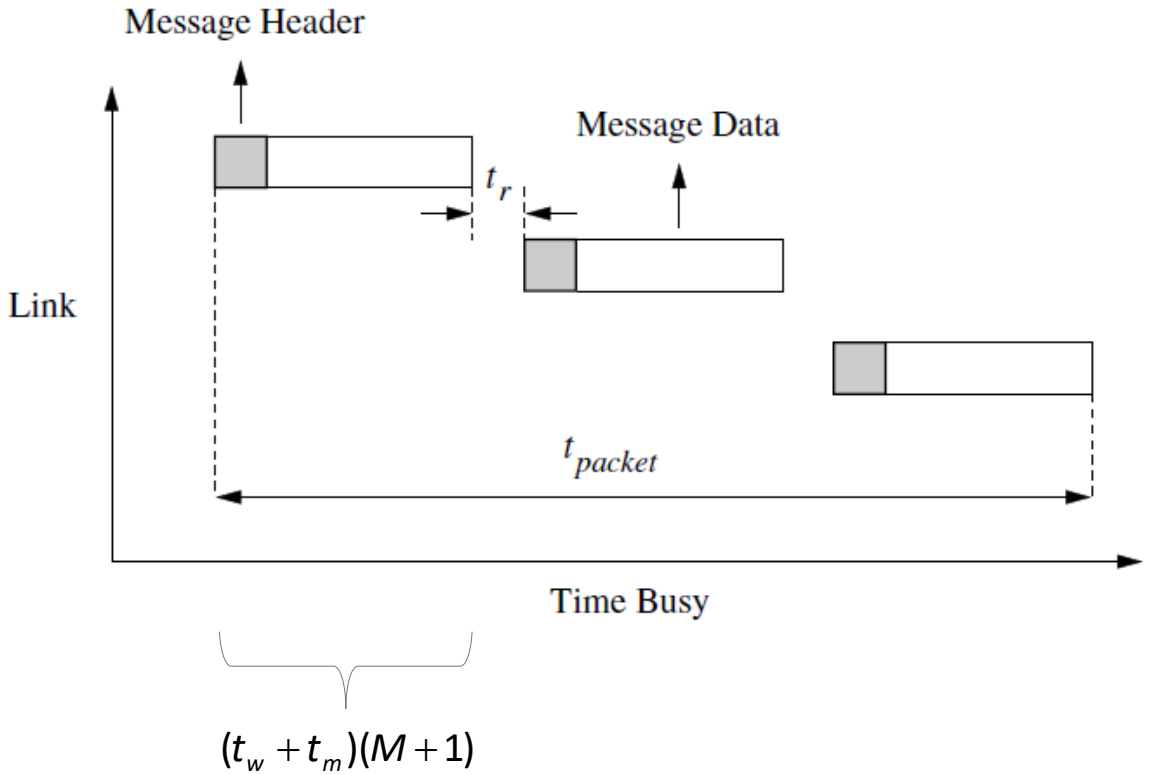

- Για να αποφασιστεί το κανάλι εξόδου (Route & control) σε ενδιάμεσο κόμβο, απαιτείται χρόνος $\mathbf{t_r}$
  - Χρόνος *διαδρόμησης*
- Για να μεταφερθεί από την είσοδο στην έξοδο σε ένα ενδιάμεσο κόμβο χρειάζεται χρόνος $\mathbf{t_m}$
  - Περιλαμβάνει όλες τις καθυστερήσεις (buffering, πέρασμα από switch κλπ) για να περάσει από κανάλι εισόδου σε κανάλι εξόδου
  - Χρόνος *διάσχισης*

# Χρόνος με μεταγωγή κυκλώματος



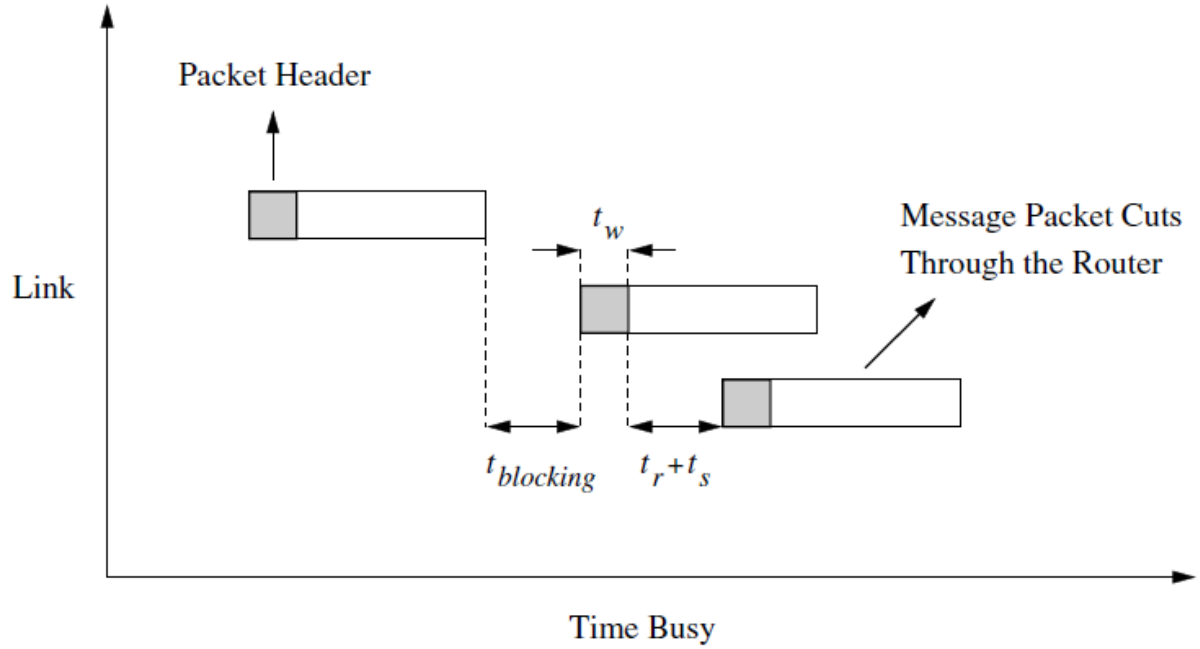

$$T_{\text{circuit switching}} = D\left(t_r + 2(t_m + t_w)\right) + Mt_w$$

# Χρόνος με μεταγωγή SaF




$$T_{\text{packet switching}} = D\left(t_r + (t_w + t_m)(M+1)\right)$$
$$= D(t_r + t_w + t_m) + DM(t_w + t_m)$$

# Χρόνος με VCT (πάντα, χωρίς αναμονές)




$$T_{VCT} = D(t_r + t_m + t_w) + Mt_w$$

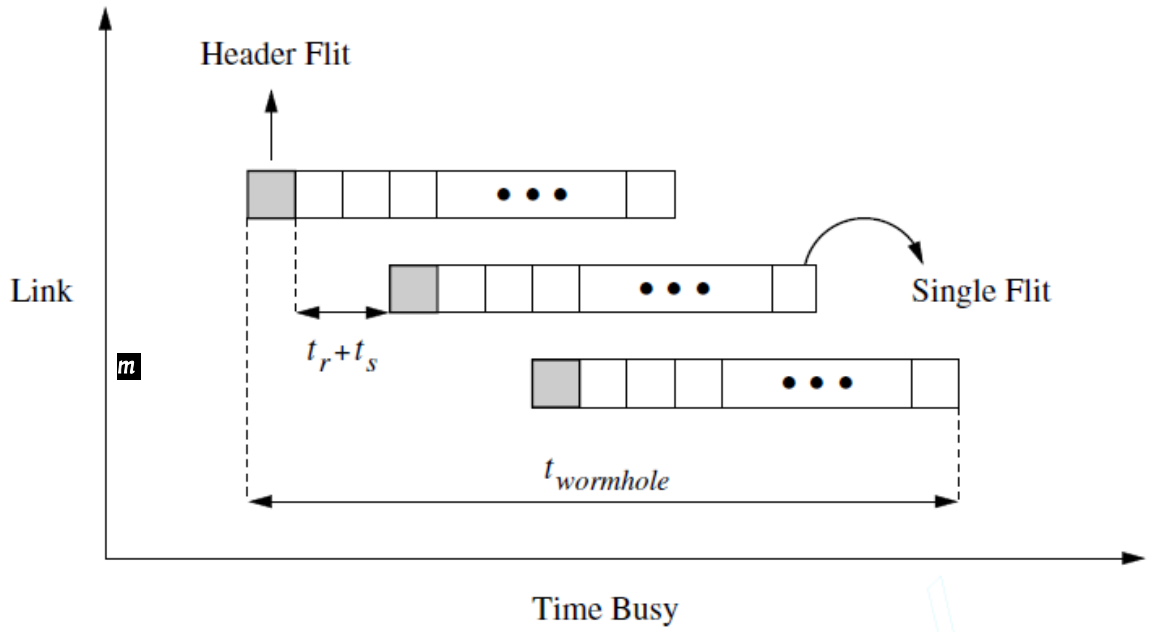
Ο χρόνος για να φτάσει το header flit

Τα άλλα flit ακολουθούν από πίσω και καταφθάνουν το ένα μετά το άλλο.

*Αν ο χρόνος να διασχισθεί ένας διαδρομητής ($t_m$) είναι μεγαλύτερος από τον χρόνο μετάδοσης στο κανάλι ($t_w$), τότε ο όρος πρέπει να είναι $Mt_m$.*

*Αν ο διαδρομητής ήταν μόνο input-buffered?*

# Χρόνος με wormhole switching




$$T_{\text{wormhole routing}} = D(t_r + t_m + t_w) + Mt_w$$

(Ίδιος με το VCT χωρίς αναμονές.)

# Όλοι οι χρονισμοί

- $T_{circuit\ switching} = D(t_w+2(t_r+t_m)) + Mt_w$
- $T_{SAF} = D(t_w+t_r+t_m) + DM(t_w+t_r)$
- $T_{VCT} = T_{WR} = D(t_w+t_r+t_m) + Mt_w$

# Σύγκριση

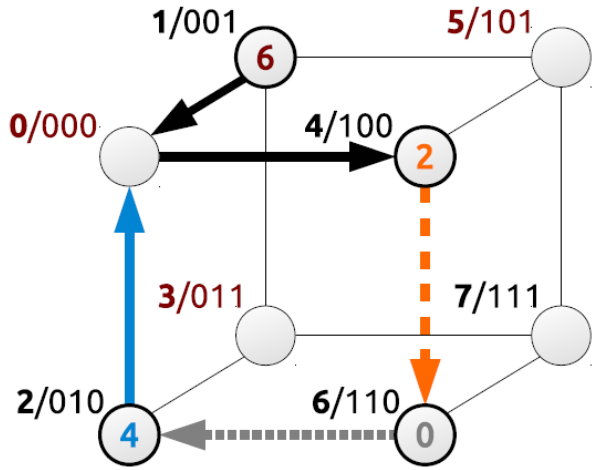
- Υπεραπλουστεύοντας, ας υποθέσουμε ότι $\mathbf{t_w \approx t_r \approx t_m = 1}$ χρονική μονάδα. Οι εκφράσεις μας απλοποιούνται ως εξής:
  - $T_{\text{circuit switching}} = T_{VCT} = T_{WR} = \Theta(D+M)$
  - $T_{SAF} = \Theta(DM)$

- Αν τα μηνύματα δεν είναι πάρα πολύ μικρά, $(M = O(D))$, τότε
  - $T_{\text{circuit switching}} = T_{VCT} = T_{WR} = \Theta(M)$
  - Επομένως οι μεταγωγές κυκλώματος, VCT και wormhole εξαρτώνται σχεδόν αποκλειστικά από το M και άρα είναι ***ανεξάρτητες της απόστασης (distance insensitive)***.

- Όλα αυτά, βέβαια, με την προϋπόθεση ότι δεν υπάρχουν συγκρούσεις / αναμονές στο μονοπάτι του μηνύματος.

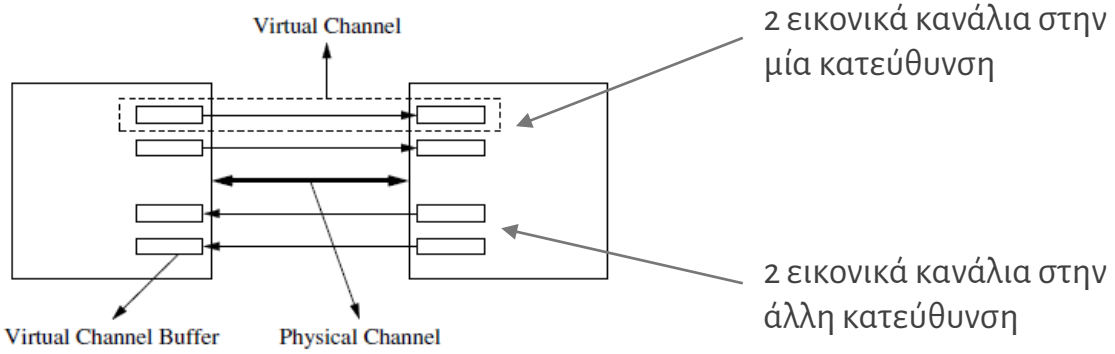
# Wormhole switching

- Το wormhole switching έχει επικρατήσει διότι
  - Ταχύτητα ακόμα και σε δίκτυα μεγάλης διαμέτρου
  - Ελάχιστο buffering
  - Οπότε γίνεται δυνατή η *υλοποίηση routers σε ανεξάρτητο chip και όχι μέσω της μνήμης του κόμβου*
    - *High-speed (low latency) routers and networks*
- Σε υψηλή κίνηση είναι ιδιαίτερα επιρρεπές σε *deadlocks* αφού δεσμεύει πολλούς πόρους (κανάλια) στην πορεία του.



# Τεχνική: virtual channels (εικονικά κανάλια)

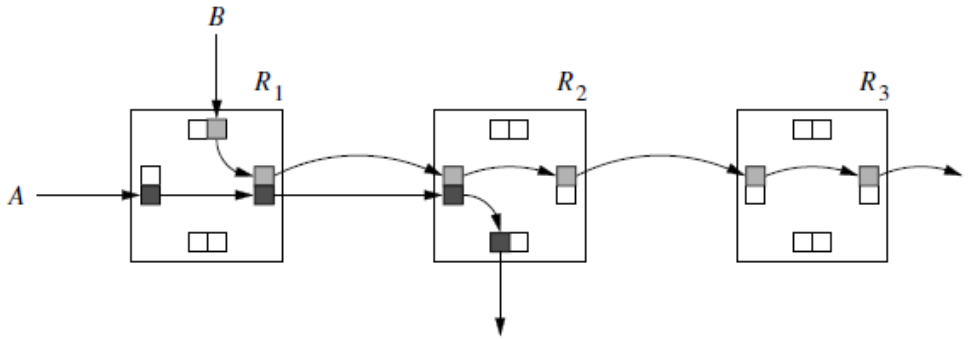
- Συνήθως οι buffers στα κανάλια είναι ουρές FIFO
  - Επομένως, αν το header flit μπλοκάρει, όλα τα προηγούμενα κανάλια *δεσμεύονται (σαν το circuit switching)*
  - *Κανένα άλλο μήνυμα δεν μπορεί να προχωρήσει*
- Τεχνική για βελτίωση της κατάστασης: virtual channels
  - Κάθε φυσικό κανάλι «υποδιαιρείται» σε πολλά εικονικά («λογικά») κανάλια.
  - Τα εικονικά κανάλια *πολυπλέκονται στο φυσικό κανάλι χρονικά*
  - Κάθε εικονικό κανάλι ορίζεται ουσιαστικά από ζεύγος buffers σε δύο γειτονικούς routers



# Virtual channels

- Αν πολυπλέκονται χρονικά κ εικονικά κανάλια πάνω σε 1 φυσικό κανάλι B bits/sec, είναι σαν να έχω κ διαφορετικά φυσικά κανάλια, το καθένα (B/κ) bits/sec, δηλαδή πιο πολλά αλλά πιο αργά κανάλια.
- Αρχικά χρησιμοποιήθηκαν για το πρόβλημα του deadlock
- Όμως, μπορούν να βελτιώσουν και τις επιδόσεις μιας και πλέον το φυσικό κανάλι δεν δεσμεύεται εξ ολοκλήρου από κάποιο μπλοκαρισμένο μήνυμα
- Μπορούν έτσι να προχωρούν μαζί παραπάνω από ένα μηνύματα στο κανάλι

# Virtual channels




- Εδώ, αν το Α είχε μπλοκάρει στον R2, θα περίμενε αναγκαστικά και το Β (άρα 2 μηνύματα σε αναμονή, ενώ τώρα κανένα)
- Επίσης, αν το Α ήταν τεράστιο, το Β θα περίμενε για πολύ ώρα ενώ τώρα όχι.
- Όμως σίγουρα χάνουμε σε ταχύτητα και επίσης αυξάνει και η πολυπλοκότητα του διαδρομητή
- Επομένως καλό είναι να μην είναι πολλά τα virtual channels