

Συστήματα Peer To Peer (P2P Systems)

Γαλάνης Δημήτριος
Παπαδημητρίου Χριστίνα

Τα Peer-To-Peer προσελκύουν το ενδιαφέρον

- Ακαδημαϊκά Προσπάθειες
 - International Workshop on P2P Computing
 - Global and P2P Computing on Large Scale Distributed Systems
 - International Conference on P2P Computing
 - O'Reilly P2P and Web Services Conference
- Βιομηχανικές προσπάθειες
 - P2P Working Group: Intel, HP, Sony και άλλες μικρότερες
 - JXTA: μία πλατφόρμα από την Sun (open source)

Τι είναι Peer-To-Peer (P2P)

- Ο όρος αναφέρεται σε μία κατηγορία συστημάτων και εφαρμογών, τα οποία απασχολούν κατανεμημένους πόρους με σκοπό να εκτελέσουν μία αποφασιστικής σημασίας λειτουργία με κατανεμημένο τρόπο.
- Κατανεμημένοι πόροι:
 - η υπολογιστική ισχύς
 - τα δεδομένα (τόσο από την σκοπιά του μεγέθους τους όσο και του περιεχομένου τους)
 - και το εύρος ζώνης του δικτύου (bandwidth)

Τα Πλεονεκτήματα των P2P

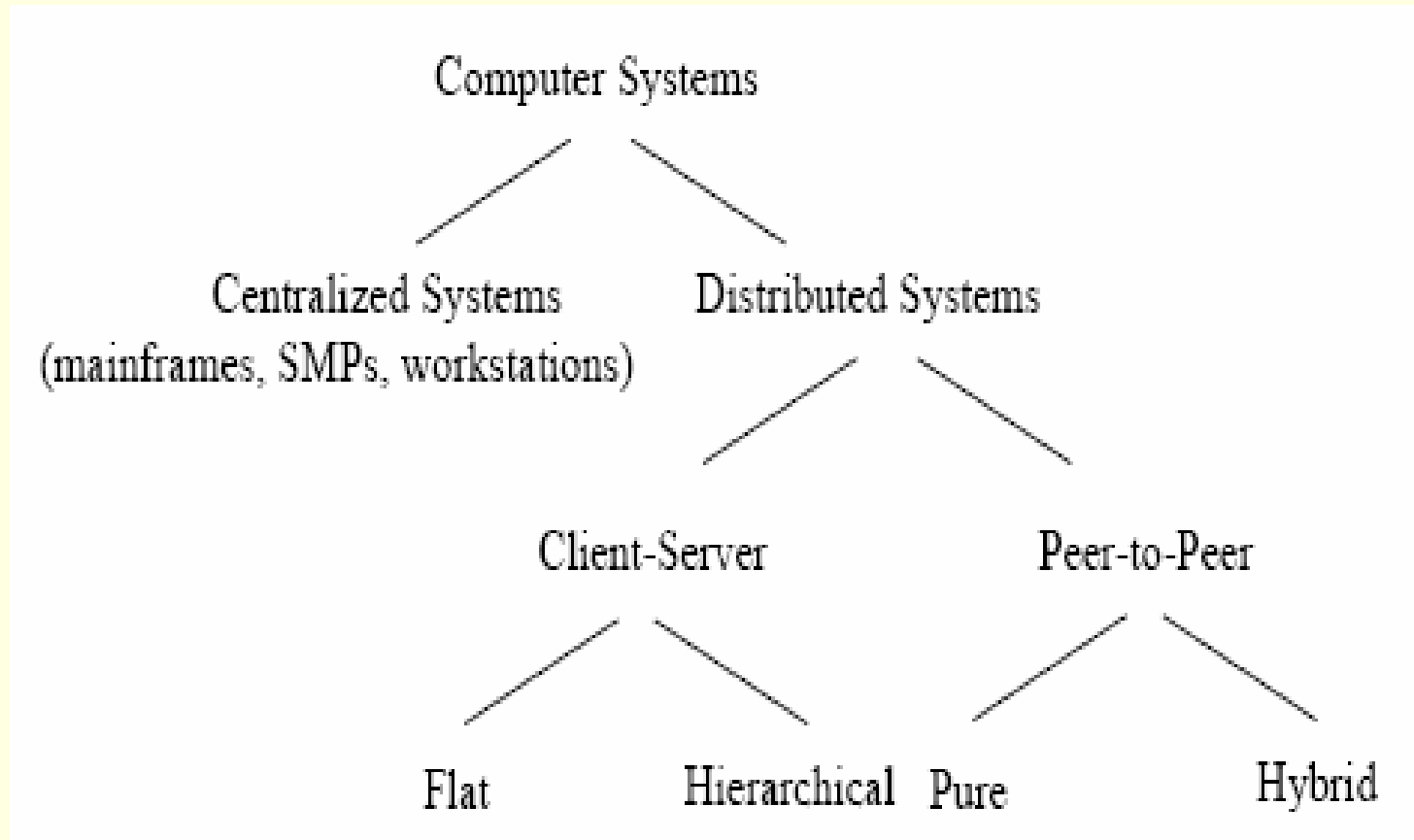
- Βελτίωση της κλιμάκωσης: μη ύπαρξη συγκεντρωτικών σημείων εξάρτησης
- Εξάλειψη της ανάγκης να υπάρχει μία δαπανηρή υποδομή καθιστώντας άμεση επικοινωνία μεταξύ των πελατών και
- Δυνατότητα συνάθροισης πόρων

- Ανοιχτό το θέμα της ασφάλειας

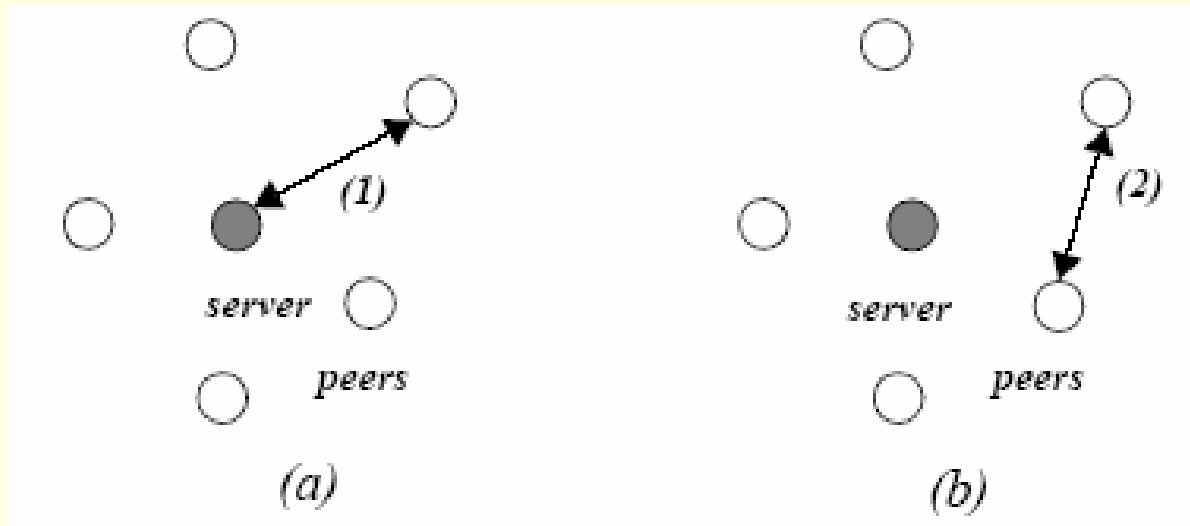
Στόχοι των P2P

- Μείωση του κόστους κατακερματισμού
- Βελτίωση της κλιμάκωσης και της αξιοπιστίας
- Συνάθροιση πόρων και διαλειτουργικότητα
- Αυξημένη αυτονομία
- Ανωνυμία και ιδιωτικότητα
- Δυναμισμός
- Ad-hoc επικοινωνία και συνεργασία

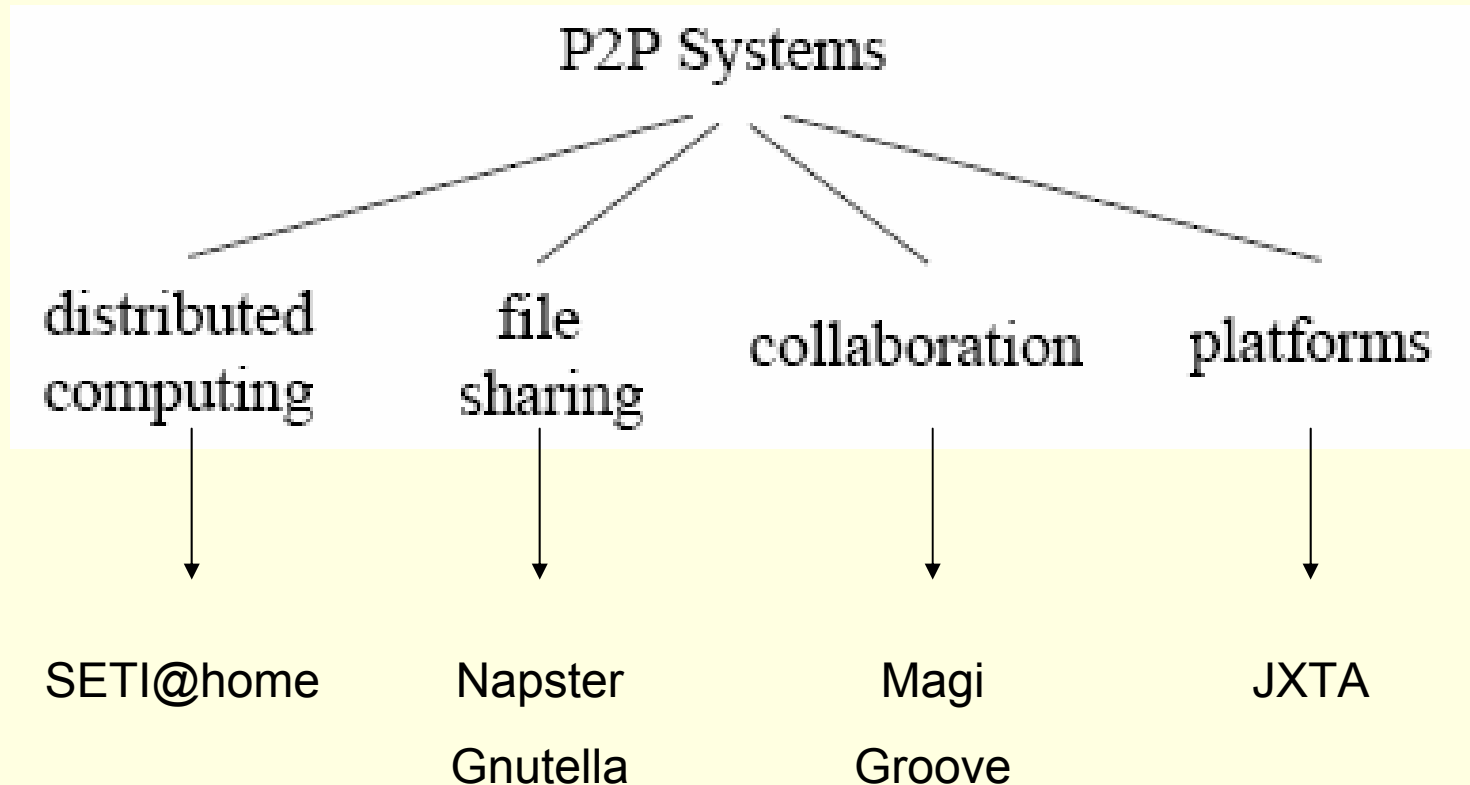
Ταξινόμηση Υπολογιστικών Συστημάτων



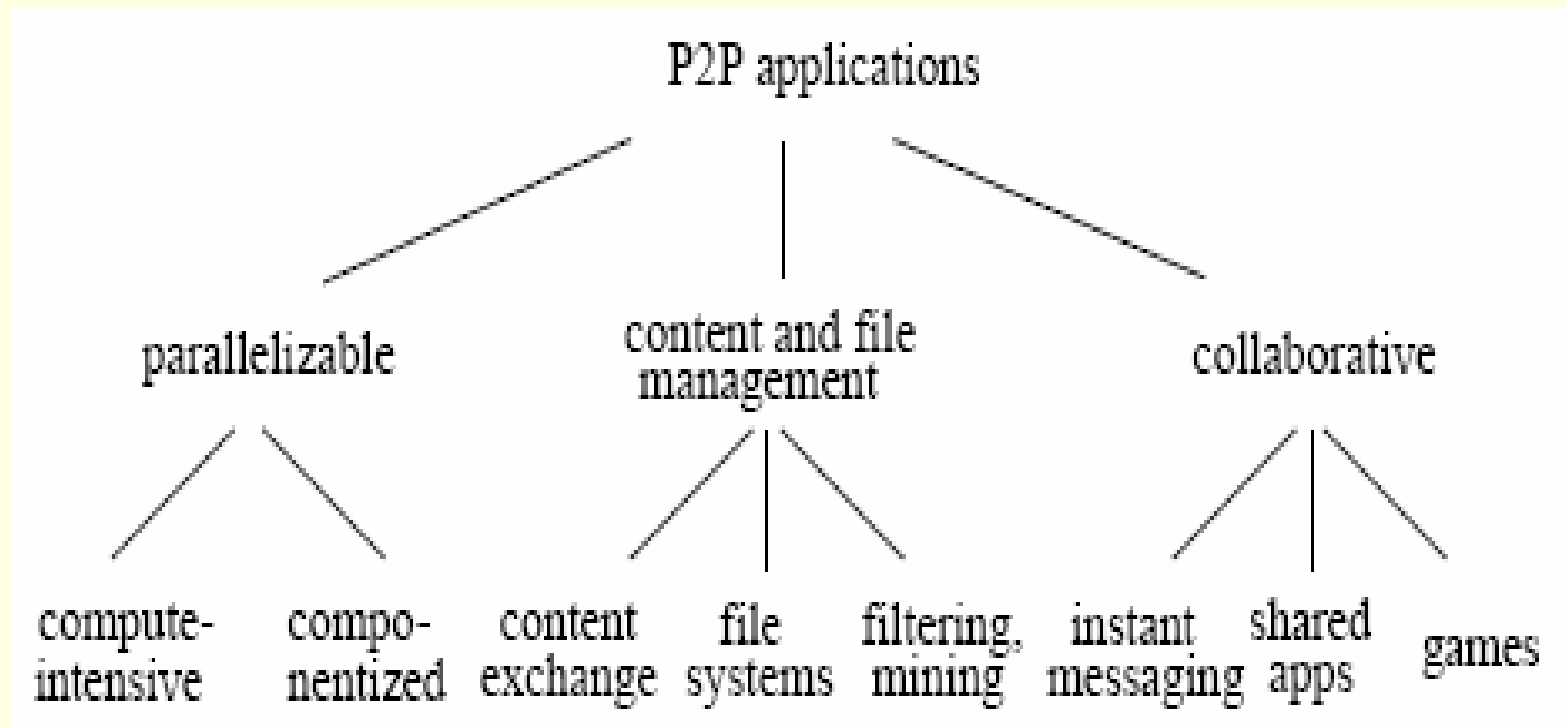
Υβριδικό P2P Μοντέλο



Ταξινόμηση των P2P Συστημάτων

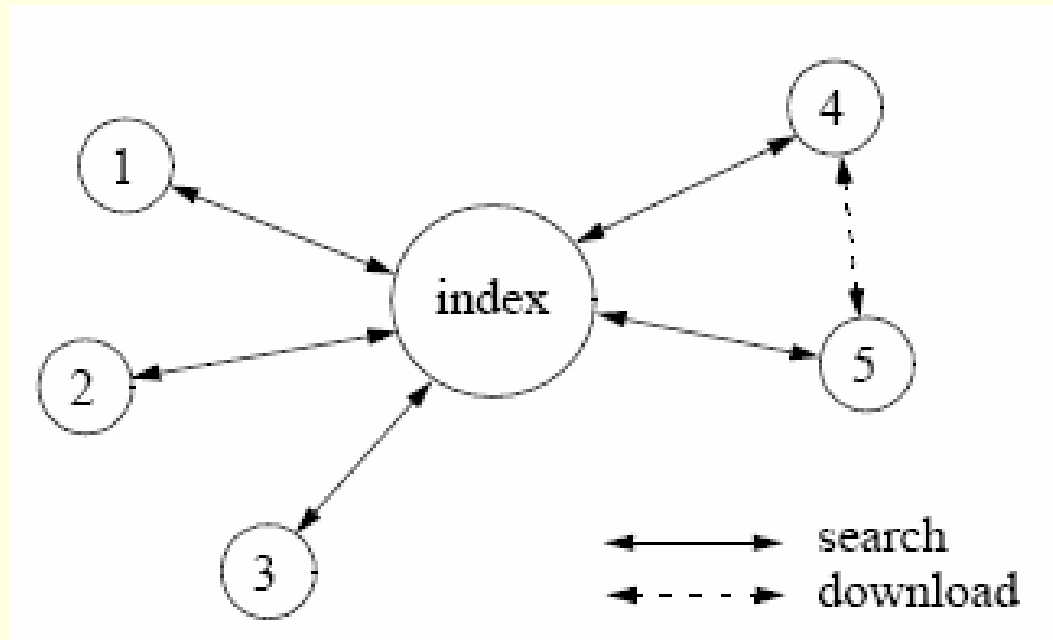


Ταξινόμηση των Εφαρμογών P2P



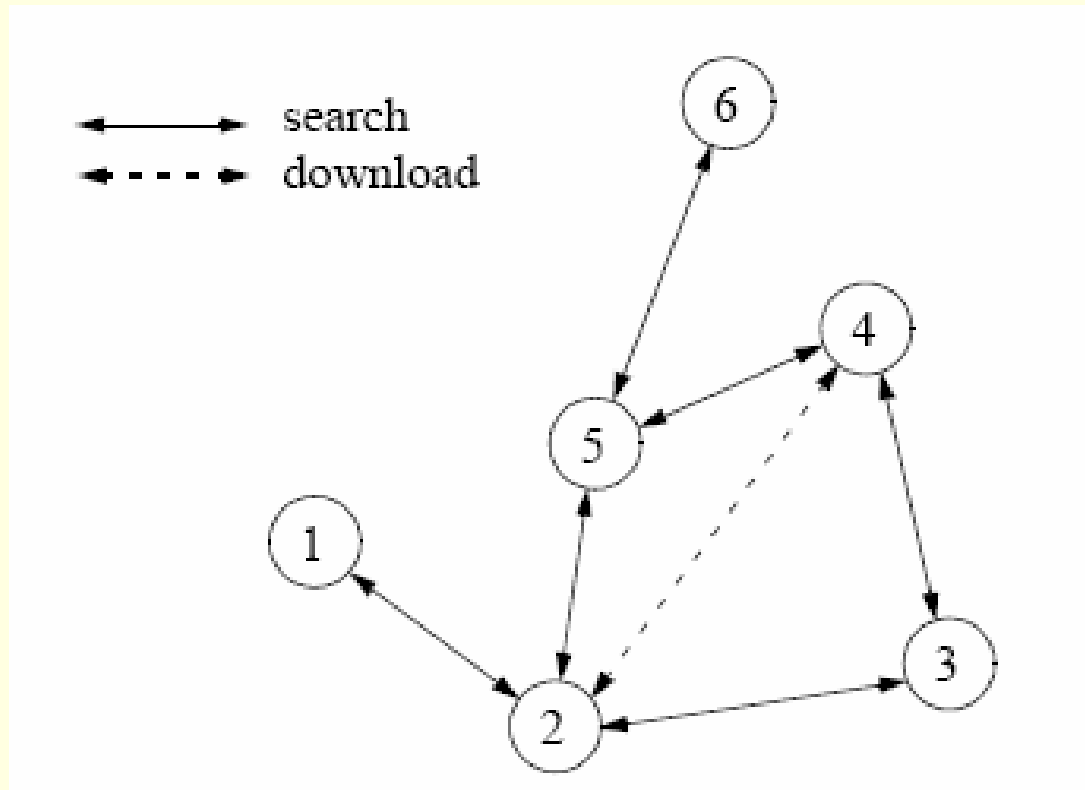
Κατηγορίες Αλγορίθμων P2P (1/3)

- Μοντέλο συγκεντρωτικού ευρετηρίου (Centralized directory model)



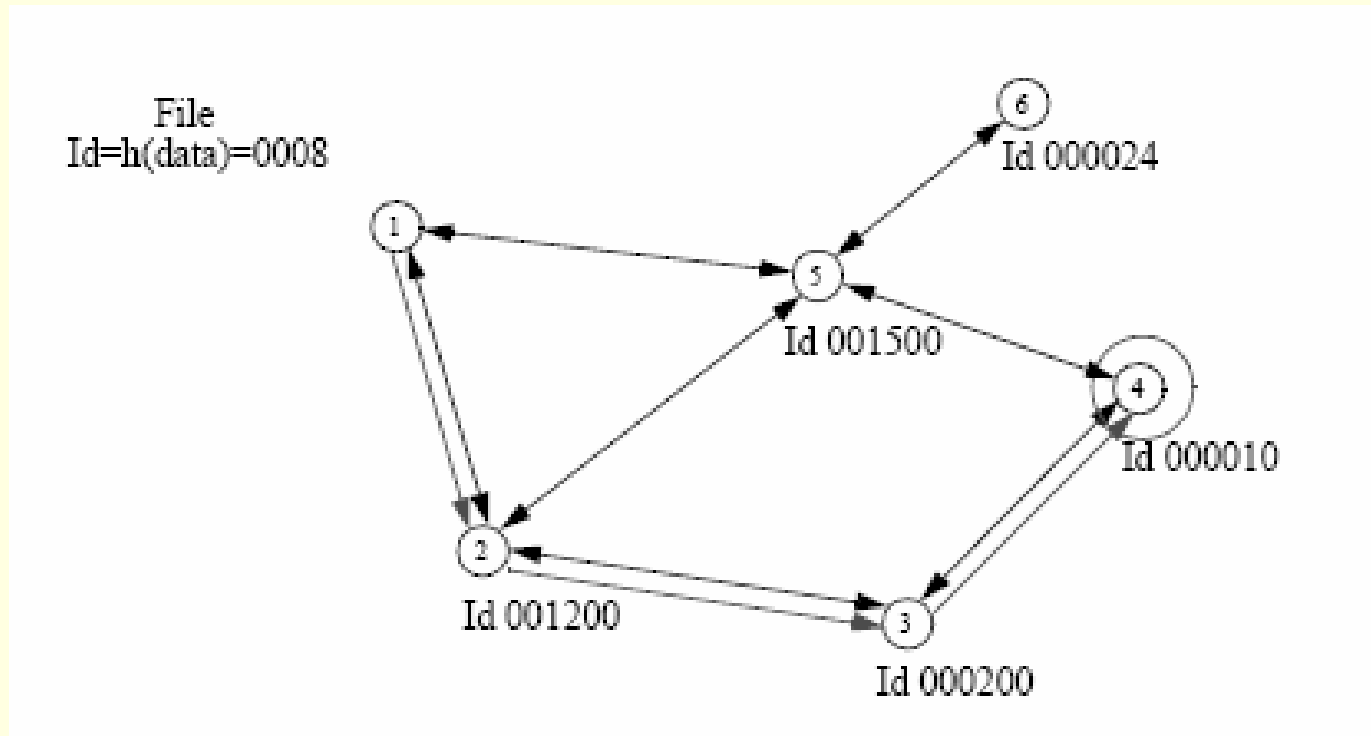
Κατηγορίες Αλγορίθμων P2P (2/3)

- Μοντέλο πλημμύρας αιτήσεων
(Flooded requests model)



Κατηγορίες Αλγορίθμων P2P (3/3)

- Μοντέλο δρομολόγησης εγγράφων (Document routing model)



Αλγόριθμοι P2P

- Οι Chord, CAN, Tapestry και Pastry δίνουν έμφαση:
 - στην εύκολη κλιμάκωση όταν μεγάλος αριθμός κόμβων εισέρχεται στο σύστημα,
 - στην εύρεση των εγγράφων με μικρή καθυστέρηση,
 - στον εύκολο χειρισμό των αφίξεων και αναχωρήσεων των peers,
 - στη διατήρηση μικρής ποσότητας πληροφορίας δρομολόγησης και
 - στην κατανομή του φόρτου ισομερώς ανάμεσα στους κόμβους

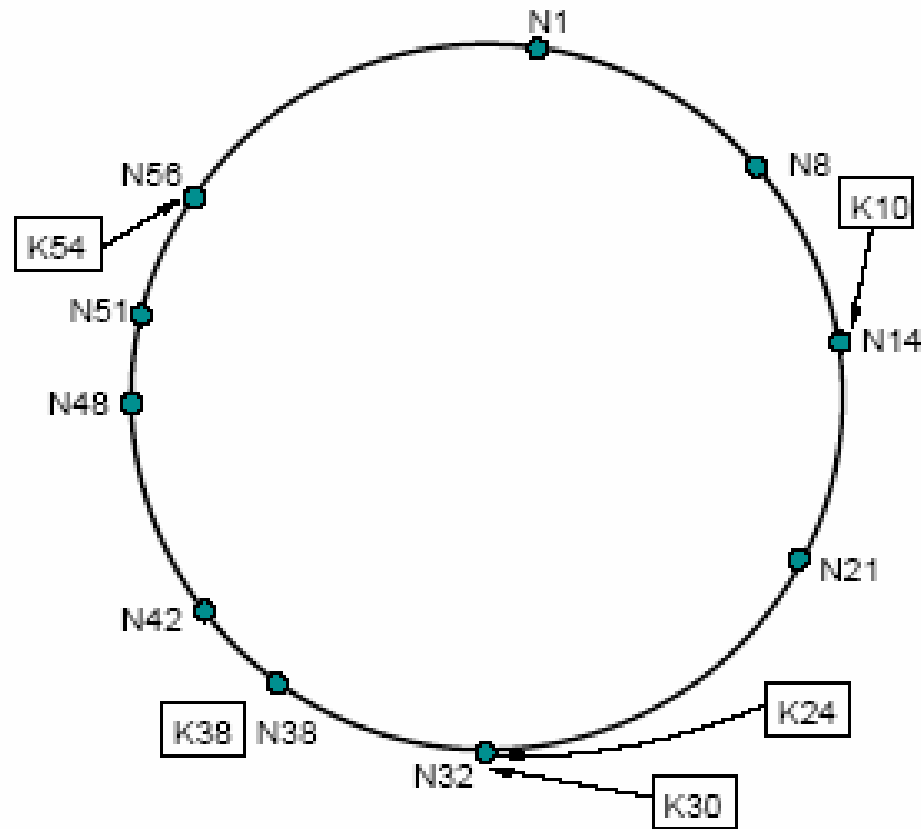
Κατανεμημένος πίνακας κατακερματισμού (DHT)

- Το μοντέλο δρομολόγησης εγγράφων στηρίζεται για την υλοποίηση του στον DHT.
- **Κύρια ιδέα:**
 - τα δεδομένα αντιπροσωπεύονται από μοναδικά κλειδιά
 - οι κόμβοι του συστήματος (peers) είναι πρόθυμοι να αποθηκεύσουν κλειδιά για λογαριασμό άλλων
- Ένας DHT υποστηρίζει μόνο μία λειτουργία την `lookup(key)`.
 - Την αναζήτηση και εύρεση της ταυτότητας (π.χ. IP διεύθυνση) του κόμβου που είναι υπεύθυνος για ένα συγκεκριμένο κλειδί (key).

Αλγόριθμος Chord

- Προσφέρει ένα γρήγορο κατανεμημένο υπολογισμό των απεικονίσεων των κλειδιών σε κόμβους υπεύθυνους για αυτά.
- Χρησιμοποιεί m -bit αναγνωριστικά για τους κόμβους και τα αντικείμενα τα οποία παράγονται χρησιμοποιώντας μια συνάρτηση κατακερματισμού.
- Τα αναγνωριστικά στο πρωτόκολλο Chord διατάσσονται πάνω σε έναν δακτύλιο σε αύξουσα σειρά από 0 έως $2^m - 1$.
- Ένα κλειδί ανατίθεται στον κόμβο που έχει το ίδιο αναγνωριστικό με αυτό ή στον πρώτο κόμβο που έπεται αυτού στο δακτύλιο
- Ο κόμβος που είναι υπεύθυνος για ένα κλειδί ονομάζεται successor κόμβος του k και συμβολίζεται με $\text{successor}(k)$.

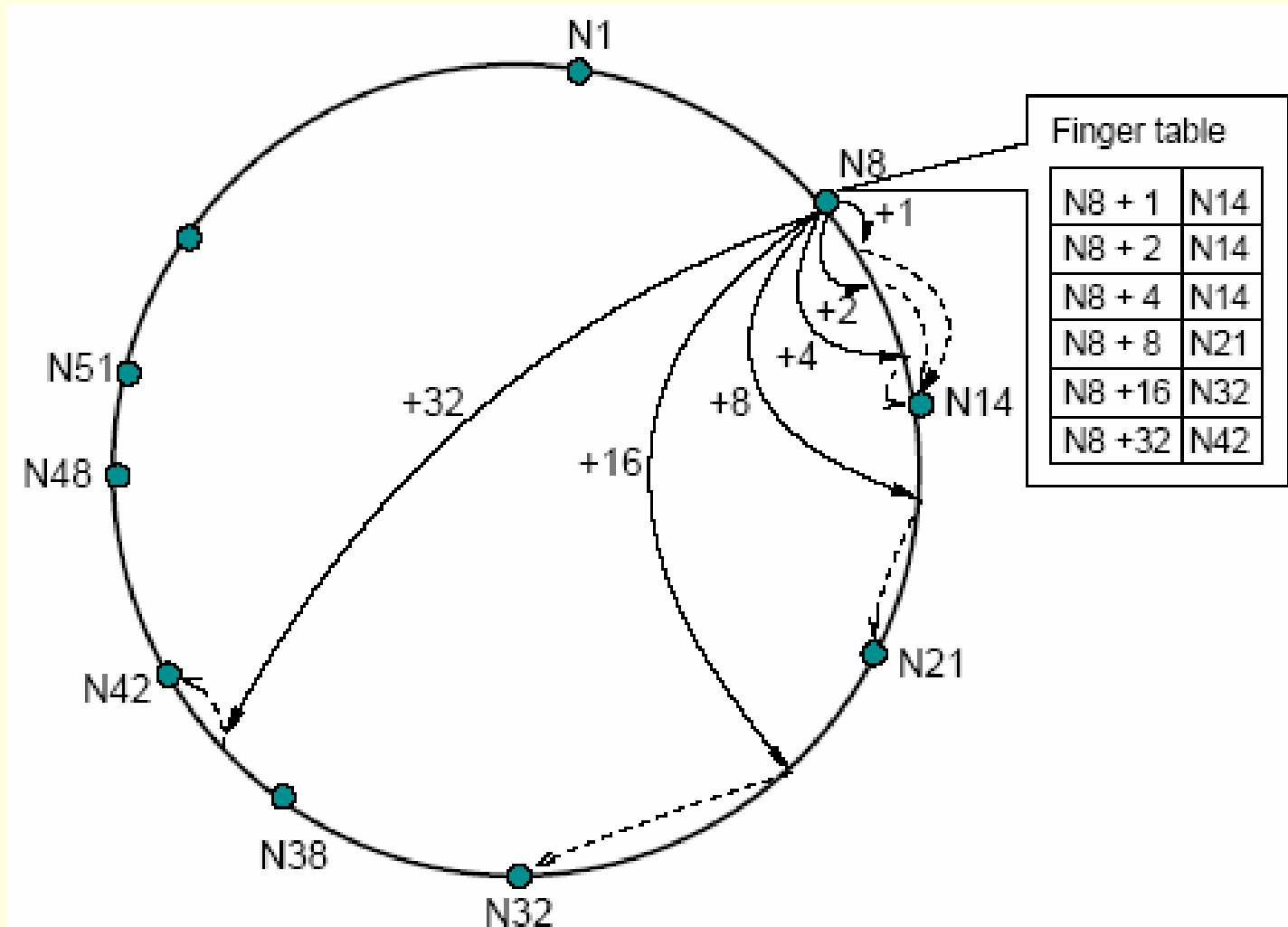
Ο δακτύλιος του Chord



Η αναζήτηση στο Chord

- Προώθηση της αναζήτησης στον επόμενο κόμβο έως ότου φτάσει σε αυτόν που είναι υπεύθυνος για το κλειδί της αναζήτησης
 - Η καθυστέρηση μπορεί να είναι πολύ μεγάλη.
- Βελτίωση: Χρησιμοποίηση περισσότερης ποσότητας πληροφορίας δρομολόγησης αποθηκευμένη σε πίνακες (finger tables).
- Η i -οστή εγγραφή ενός τέτοιου πίνακα σε έναν κόμβο n περιέχει το αναγνωριστικό του πρώτου κόμβου s που ακολουθεί τον n κατά το λιγότερο 2^{i-1} στον δακτύλιο ($s = \text{successor}(n + 2^{i-1})$) όπου $1 \leq i \leq m$

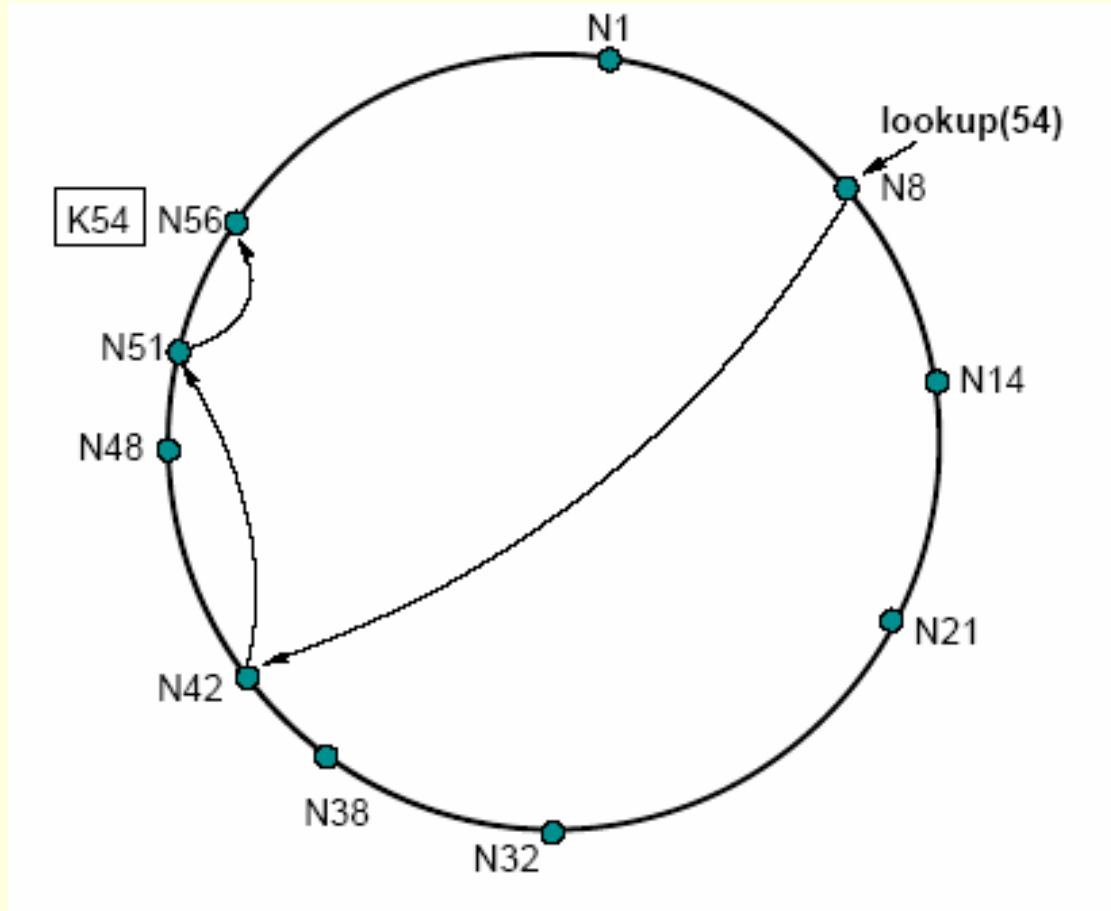
Παράδειγμα πίνακα δρομολόγησης



Κανόνας αναζήτησης

- Σε κάθε κόμβο n όταν φτάσει μια αίτηση αναζήτησης για έναν κλειδί k εξετάζεται αν αυτό βρίσκεται μεταξύ του κόμβου και του επόμενου ενεργού κόμβου.
 - Αν αυτό συμβαίνει, τότε ο κόμβος n στέλνει στον αρχικό κόμβο που ξεκίνησε την αναζήτηση το αναγνωριστικό του επόμενου του (successor) και η διαδικασία τερματίζεται.
 - Σε διαφορετική περίπτωση ο κόμβος n βρίσκει από τον finger table τον κόμβο n' που προηγείται του κλειδιού πιο άμεσα και προωθεί την αναζήτηση σε αυτόν

Παράδειγμα αναζήτησης



Λειτουργίες του Chord

- **Σύνδεση ενός κόμβου η στο δακτύλιο:** απαιτείται αυτός να γνωρίζει κάποιον κόμβο η' ο οποίος είναι ήδη συνδεδεμένος από τον οποίο μαθαίνει τον επόμενο του στον δακτύλιο.
- **Αποσύνδεση:** ειδοποιεί τον προηγούμενο και τον επόμενο του για την νέα τοπολογία του δακτυλίου
- **Σταθεροποίηση:** Περιοδικά κάθε κόμβος ανανεώνει αν χρειάζεται τις εγγραφές του πίνακα δρομολόγησης και ελέγχει αν ο επόμενος τους και ο προηγούμενος τους παραμένουν ίδιοι και διανέμει εκ νέου τα κλειδιά.
- **Αποφυγή αποτυχίας:** Κάθε κόμβος αποθηκεύει μία λίστα από successors.
- Η ορθότητα των αναζητήσεων διασφαλίζεται από τη λειτουργία σταθεροποίησης

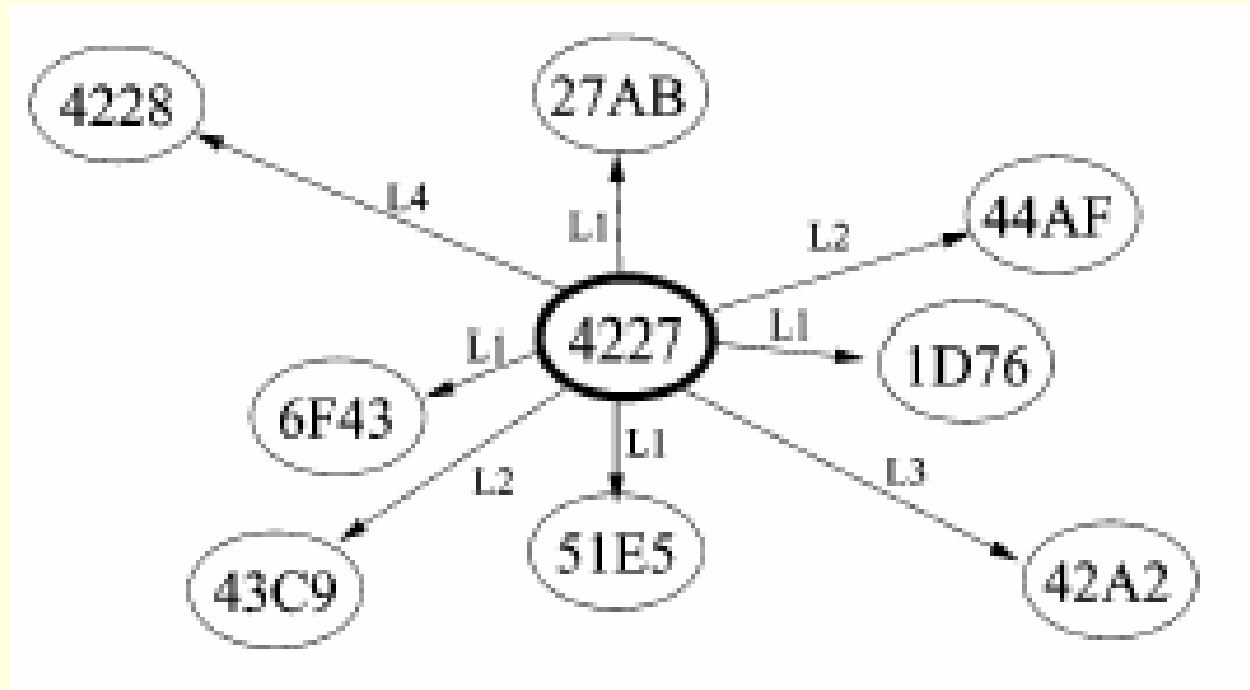
Τι είναι το Tapestry

- Μία επεκτάσιμη υποδομή, η οποία παρέχει λειτουργίες εντοπισμού και δρομολόγησης για αποκεντριοποιημένα αντικείμενα (Decentralized Object Location and Routing – DOLR).
 - Η διεπαφή DOLR επικεντρώνεται στη δρομολόγηση μηνυμάτων σε τελικά σημεία, όπως κόμβοι ή αντίγραφα αντικειμένων.
- Είναι ένα P2P δίκτυο επικάλυψης (overlay network), το οποίο παρέχει υψηλή απόδοση, κλιμάκωση και εξαρτώμενη από τη θέση δρομολόγηση μηνυμάτων σε κοντινά τελικά σημεία χρησιμοποιώντας μόνο τοπικούς πόρους.

Η Δομή του Tapestry

- Οι κόμβοι του έχουν ένα *nodeID* ομοιόμορφα και τυχαία επιλεγμένο από έναν μεγάλο χώρο αναγνωριστικών.
- Ένας φυσικός ξένιος (host) μπορεί να φιλοξενεί περισσότερους από ένα κόμβους.
- Στα τελικά σημεία συγκεκριμένων εφαρμογών αναθέτουμε σφαιρικά μοναδικά αναγνωριστικά (GUIDs), τα οποία επιλέγονται από τον ίδιο χώρο αναγνωριστικών.
- Χώρος αναγνωριστικών με τιμές 160-bit .
- Γενικά, λέμε ότι ο κόμβος N έχει *nodeID* N_{id} και ένα αντικείμενο O έχει GUID O_G .
- Κάθε μήνυμα περιέχει ένα αναγνωριστικό A_{id} .

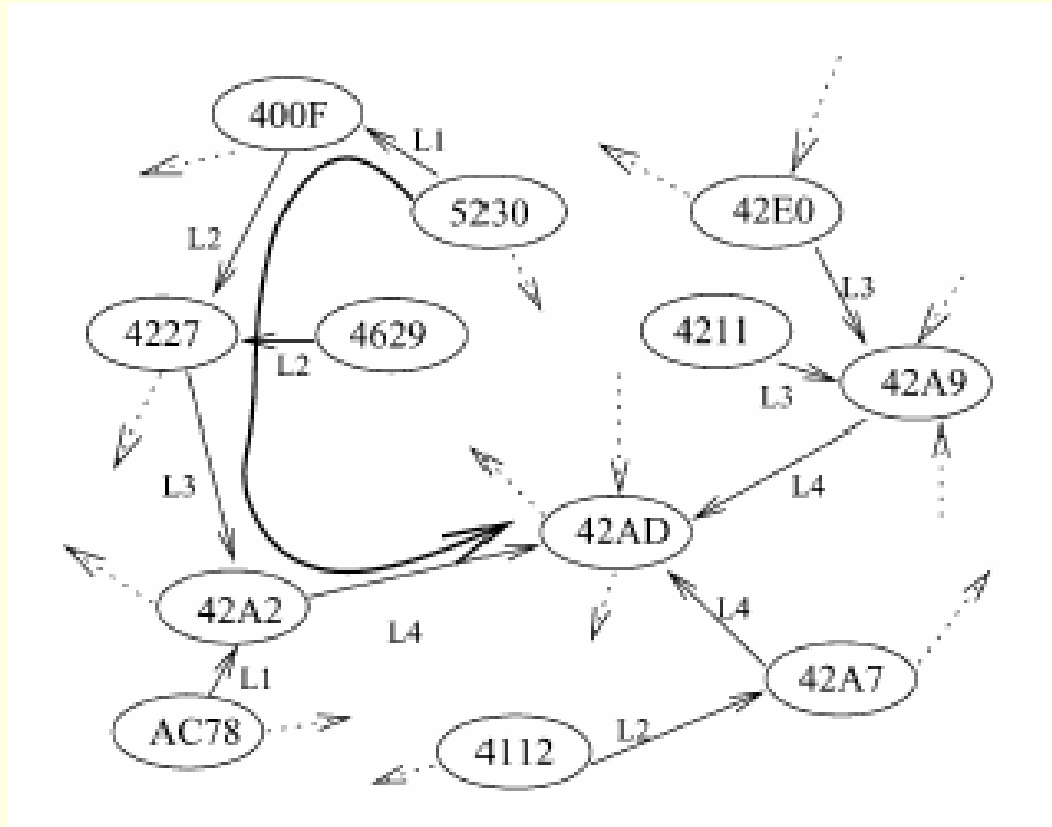
Πλέγμα Δρομολόγησης



Για παράδειγμα, $4^{***} \Rightarrow 42^{**} \Rightarrow 42A^* \Rightarrow 42AD$,

όπου το * μπορεί να είναι οποιοδήποτε επιτρεπτό σύμβολο.

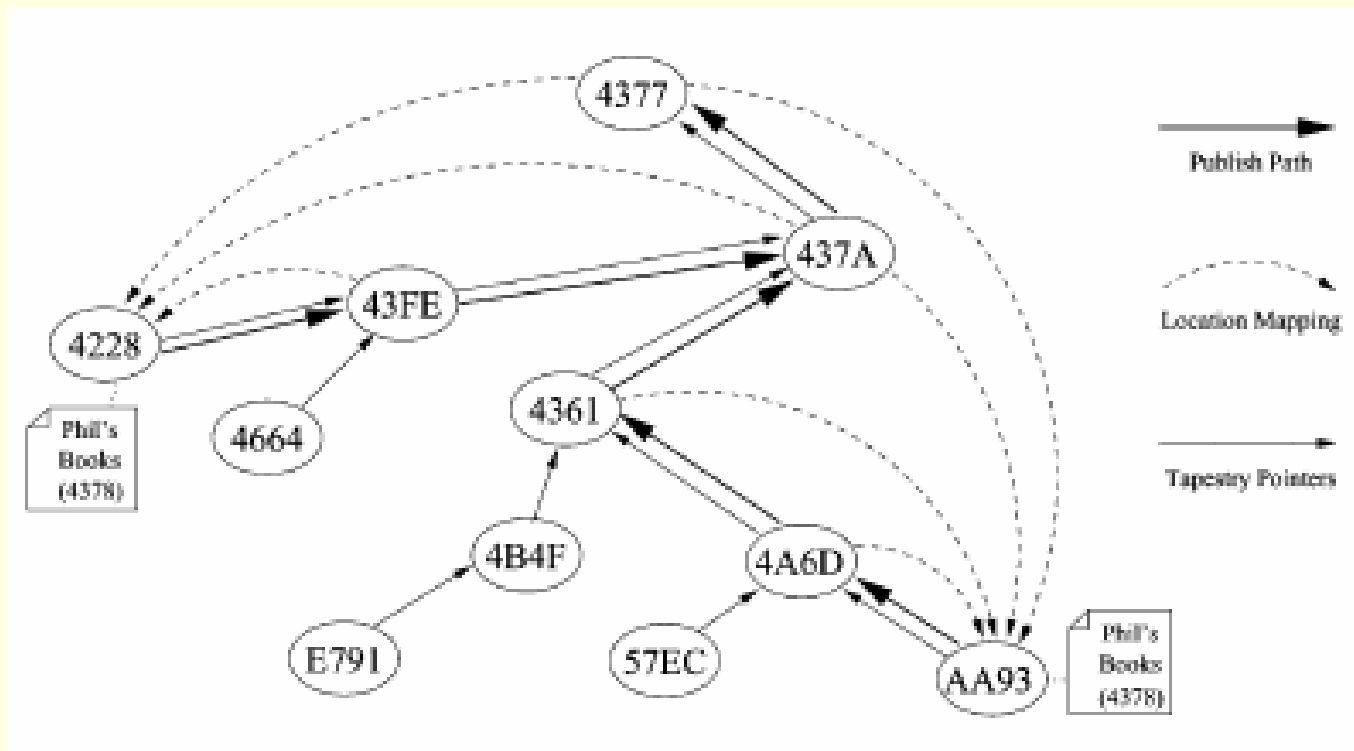
Παράδειγμα Μηνύματος



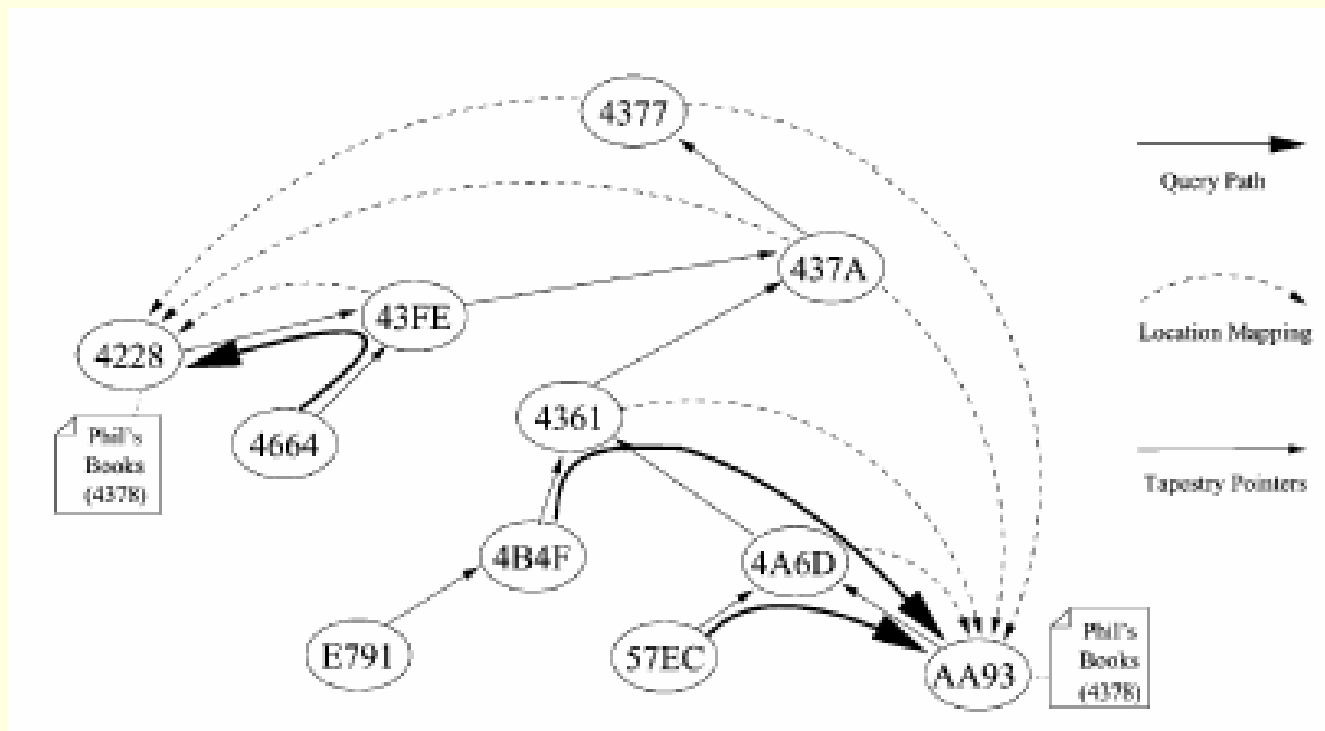
Παράδειγμα Μηνύματος

- Ο δρομολογητής για τον n -οστο hop μοιράζεται ένα πρόθεμα μήκους $\geq n$ με τον προορισμό ID.
- Για τη δρομολόγηση το Tapestry κοιτάζει στο χάρτη του $(n+1)$ -οστου επιπέδου για καταχώρηση, η οποία συμπίπτει με το επόμενο ψηφίο στο ID προορισμού.
- Η μέθοδος αυτή εγγυάται ότι οποιοσδήποτε κόμβος του συστήματος θα είναι προσβάσιμος το πολύ σε λογικά hops, σε ένα σύστημα με χώρο ονομάτων μεγέθους N , IDs βάσης και υποθέτοντας συνεπείς χάρτες γειτόνων.

Παράδειγμα Δημοσίευσης Αντικειμένου στο Tapestry



Παράδειγμα Δρομολόγησης Αντικειμένου στο Tapestry



Εισαγωγή Κόμβου στο Tapestry

- 4 συστατικά μέρη για την εισαγωγή ενός νέου κόμβου σε ένα δίκτυο Tapestry.
 - Κόμβοι που χρειάζεται να γνωρίζουν ειδοποιούνται από τον N, διότι ο N συμπληρώνει μία κενή καταχώρηση στους πίνακες δρομολόγησης τους.
 - Ο N μπορεί να γίνει το νέο αντικείμενο ρίζα για υπάρχοντες κόμβους. Οι αναφορές σε αυτά τα αντικείμενα πρέπει να μετακινηθούν στον N για να διατηρεί τη διαθεσιμότητα των αντικειμένων.
 - Οι αλγόριθμοι πρέπει να κατασκευάσουν ένα σχεδόν βέλτιστο πίνακα δρομολόγησης για τον N.
 - Ειδοποιούνται οι κόμβοι κοντά στον N και αυτοί μπορεί να θεωρήσουν τη χρήση του N στους δικούς τους πίνακες δρομολόγησης.

Εκούσια Διαγραφή Κόμβου στο Tapestry

- Ενημερώνει το σύνολο κόμβων D με πίσω δείκτες στον N για την πρόθεση του
- Λέει έναν κόμβο αντικαταστάτη για κάθε επίπεδο δρομολόγησης από το δικό του πίνακα δρομολόγησης
- Κάθε ειδοποιημένος κόμβος στέλνει επαναδημοσίευση της κίνησης του αντικειμένου τόσο στον N όσο και στον αντικαταστάτη του.
- Ο N δρομολογεί τις αναφορές τοπικών αντικείμενων στις νέες τους ρίζες και ειδοποιεί τους κόμβους του D όταν τελειώσει.

Ακούσια Διαγραφή Κόμβου στο Tapestry

- Δημιουργεί πλεονασμούς μέσα στους πίνακες δρομολόγησης και στις αναφορές θέσεις ενός αντικειμένου (π.χ. οι c-1 εφεδρικοί δείκτες προώθησης για κάθε καταχώρηση του πίνακα δρομολόγησης).
- Οι κόμβοι χρησιμοποιούν περιοδικά beacons για να ανιχνεύσουν αποτυχίες εξερχόμενων συνδέσμων και κόμβων .
 - Πυροδοτούν την επιδιόρθωση του πλέγματος δρομολόγησης και ξεκινούν την ανακατανομή και τη δημιουργία αντιγράφων των αναφορών θέσης των αντικειμένων.
- Διατηρεί σχεδόν το 100% του ποσοστού επιτυχίας όσον αφορά τη δρομολόγηση μηνυμάτων σε κόμβους και αντικείμενα.

Αλγόριθμος Pastry

- Σε κάθε κόμβο ανατίθεται ένα τυχαίο αναγνωριστικό το οποίο υποδεικνύει την θέση του σε ένα δακτύλιο από αναγνωριστικά
- Τα ψηφία του αναγνωριστικού κάθε κόμβου διαλέγονται από μια βάση 2^b
- 128-bit αναγνωριστικά
- Το Pastry δρομολογεί τα κλειδιά στον ενεργό κόμβο με το αναγνωριστικό που είναι αριθμητικά κοντινότερο με το κλειδί.
- Κάθε κόμβος n διατηρεί ένα σύνολο φύλλων L , που αποτελείται από τους $|L|/2$ κοντινότερους κόμβους στο n αλλά μεγαλύτερους από n και από τους $|L|/2$ κοντινότερους κόμβους στο n αλλά μικρότερους από n .

Παράδειγμα κόμβου Pastry

NodeId 10233102			
Leaf set	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

- $b=2$
- $|L| = 2^b = 4$
- 16-bit αναγνωριστικά

Pastry

- Η ορθότητα του συνόλου των φύλλων είναι το μόνο που απαιτείται για την σωστή προώθηση των μηνυμάτων που ανταλλάσσονται.
- Η μόνη περίπτωση στην οποία δεν γίνεται σωστή προώθηση των μηνυμάτων είναι όταν $|L|/2$ κόμβοι με γειτονικά αναγνωριστικά πέσουν ταυτόχρονα.
- Βελτιστοποίηση της λειτουργίας προώθησης μηνυμάτων
 - Χρήση πινάκων δρομολόγησης με δείκτες προς διασκορπισμένους κόμβους του συστήματος
 - Απαιτούνται $\lceil \log_{2^b} N \rceil$ γραμμές με $2^b - 1$ εγγραφές

Pastry

- Κάθε εγγραφή στην γραμμή i στον πίνακα δρομολόγησης ενός κόμβου n είναι ένας δείκτης σε έναν άλλο κόμβο που μοιράζεται τα ίδια πρώτα i bits με τον n και το $i + 1$ bit διαφέρει (προφανώς υπάρχουν $2^b - 1$ πιθανοί τέτοιοι κόμβοι)
- Αν το κλειδί καλύπτεται από κάποιο κόμβο του συνόλου φύλλων τότε η αναζήτηση προωθείται σε αυτόν
- Αλλιώς η αναζήτηση θα προωθείτε σε κάποιο κόμβο του πίνακα δρομολόγησης που έχει το μεγαλύτερο κοινό πρόθεμα με το κλειδί

Pastry

- Εφόσον οι πίνακες δρομολόγησης και τα σύνολα φύλλων είναι σωστά ο αναμενόμενος αριθμός βημάτων που εκτελεί το Pastry για την δρομολόγηση ενός κλειδιού στον σωστό κόμβο είναι το πολύ $\left\lceil \log_{2^b} N \right\rceil$
- Σε κάθε βήμα που γίνεται χρησιμοποιώντας τον πίνακα δρομολόγησης ο αριθμός των κοινών ψηφίων μεταξύ του κλειδιού και του κόμβου αυξάνονται κατά 1 και μειώνεται το σύνολο των κόμβων κατά 2^b που τα αναγνωριστικά τους έχουν το ίδιο μεγαλύτερο πρόθεμα με το κλειδί

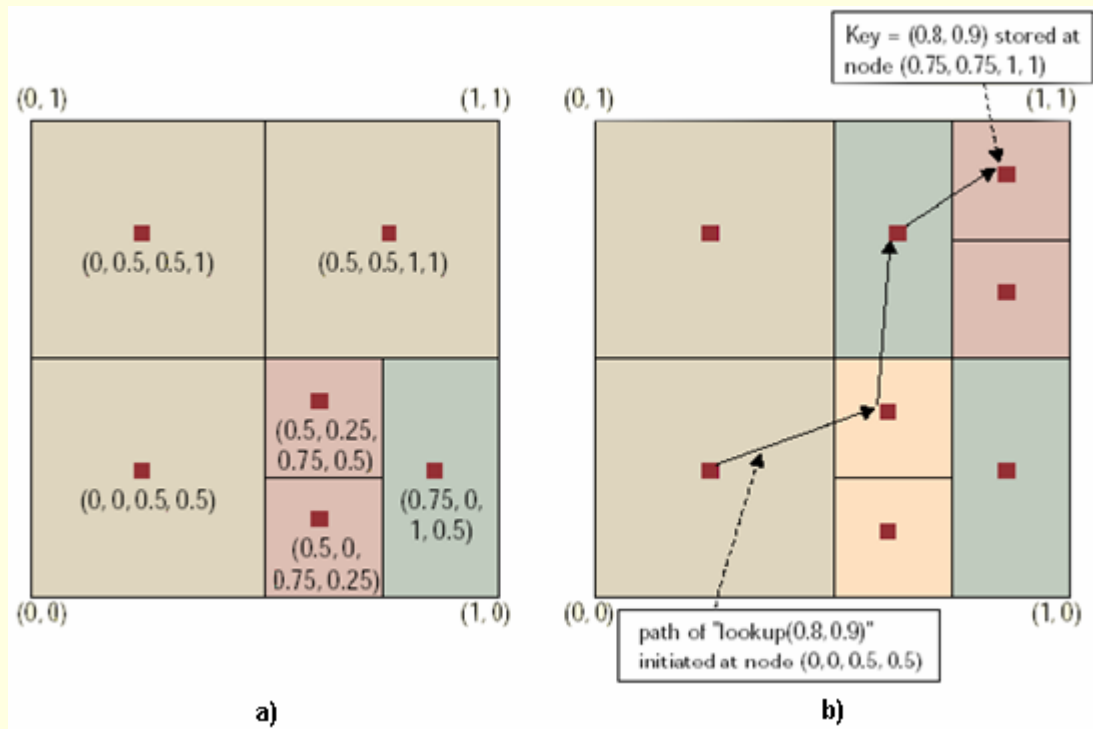
Pastry - Σύνδεση

- Όταν ένας κόμβος X συνδέεται υποθέτουμε ότι γνωρίζει έναν κόμβο A κοντινό σε αυτόν που ήδη είναι συνδεδεμένος στο σύστημα και στον οποίο στέλνει ένα πακέτο σύνδεσης και περιέχει ένα κλειδί ίσο με το X .
- Το Pastry δρομολογεί το πακέτο στον κόμβο Z που το αναγνωριστικό του συμφωνεί περισσότερο με το αναγνωριστικό του X
- Κόμβοι που ανήκουν στο μονοπάτι από το A στο Z στέλνουν πληροφορίες για την κατάσταση την οποία διατηρούν στον X
- Ο X θα ειδοποιήσει όποιους κόμβους πρέπει να ενημερωθούν για την άφιξη του στο σύστημα

Pastry

- **Ευριστική δρομολόγηση:** Κάθε κόμβος διατηρεί εκτός των άλλων και ένα σύνολο γειτόνων που τα μέλη του βρίσκονται κοντινότερα από κάποια άλλα χρησιμοποιώντας για παράδειγμα σαν μετρική απόστασης τον αριθμό των IP βημάτων

CAN



Σύνδεση κόμβου στο CAN

- διαλέγει ένα τυχαίο σημείο P στο χώρο
- ρωτά έναν κόμβο που ήδη βρίσκεται στο σύστημα να του επιστρέψει τον κόμβο n που η ζώνη του περιέχει το P
- χωρίζει την ζώνη του στη μέση και
- αναθέτει ένα από τα δύο μισά στο νέο κόμβο

Αποσύνδεση κόμβου στο CAN

- Αναθέτει τη ζώνη του σε έναν από τους γείτονες του
 - αν η ένωση των δύο ζωνών είναι έγκυρη, τότε μια νέα μεγαλύτερη ζώνη δημιουργείται.
 - Διαφορετικά, ο γείτονας αναλαμβάνει προσωρινά τη διαχείριση και των δύο ζωνών.
- Αν ένας κόμβος πέσει, τότε το CAN ορίζει ότι ο γείτονας με τη μικρότερη ζώνη θα αναλάβει και τη ζώνη του άλλου.

Τεχνικές Μείωσης της Καθυστέρησης Αναζήτησης στο CAN

- Μέτρηση του RTT
- Πολλαπλές Πραγματικότητες
 - Χρήση πολλαπλών χώρων συντεταγμένων

Σύγκριση Αλγορίθμων

	CAN	Chord	Pastry	Tapestry
Node state	d	$\log N$	$\log N$	$\log N$
Lookup	$dN^{1/d}$	$\log N$	$\log N$	$\log N$
Join	$dN^{1/d} + d \log(N)$	$\log^2 N$	$\log^2 N$	$\log^2 N$

- Κόστος λειτουργίας
 - Chord, Pastry, Tapestry έχουν τα ίδια κόστη
 - CAN σταθερό μέγεθος πινάκων δρομολόγησης, ωστόσο μεγαλύτερο κόστος αναζήτησης.
- Ανθεκτικότητα και ταυτόχρονες αλλαγές
 - Οι περισσότεροι αλγόριθμοι όταν αντιμετωπίζουν γεγονότα εισόδου ή αποτυχίας κόμβου υποθέτουν ότι είναι τα μοναδικά που συμβαίνουν

Σύγκριση Αλγορίθμων

- Βελτίωση δρομολόγησης
 - Pastry, CAN, και Tapestry χρησιμοποιούν ευριστικές μεθόδους
- Κακόβουλοι κόμβοι
 - Το Pastry χρησιμοποιεί πιστοποιητικά για την απόδειξη την ταυτότητας ενός κόμβου
- Αναζήτηση
 - Βασίζεται στο μοντέλο του κατανεμημένου πίνακα κατακερματισμού και σε ένα μοναδικό αναγνωριστικό

Το μέλλον των P2P

- Οι αλγόριθμοι P2P πιθανότατα θα έχουν το μεγαλύτερο αντίκτυπο.
- Οι εφαρμογές P2P έχουν τη μεγαλύτερη πιθανότητα επιτυχίας στο μέλλον.
- Οι πλατφόρμες P2P αποτελούν το τρίτο πιθανό σενάριο για τα συστήματα P2P