Κατανεμημένα Συστήματα

Συστήματα Peer-to-Peer (P2P)

Σκοπός

- Τα P2P είναι κατανεμημένες αρχιτεκτονικές που σχεδιάζονται με σκοπό τη διαμοίραση πόρων (περιεχομένου, αποθηκευτικού χώρου, κύκλων CPU) με απ' ευθείας ανταλλαγή χωρίς τη μεσολάβηση κάποιου ενδιάμεσου.
- Χαρακτηρίζονται από την ικανότητά τους να «απορροφούν» τα σφάλματα που συμβαίνουν και να αντέχουν τις προσωρινές αυξήσεις πληθυσμού χρηστών διατηρώντας ικανοποιητικά επίπεδα συνδεσιμότητας και απόδοσης.

Ορισμός

- Αυστηρός ορισμός: πρόκειται για πλήρως κατανεμημένα συστήματα, όπου όλοι οι κόμβοι είναι ομότιμοι υπό την έννοια της λειτουργικότητας και των έργων που εκτελούν. Δεν υπάρχουν «υπερκόμβοι» (π.χ. Kazaa) ή αρχιτεκτονικές κεντρικοποιημένου εξυπηρετητή που αναλαμβάνει έργα που έχουν μη κεντρικό ρόλο.
- Εναλλακτικός ορισμός: πρόκειται για μία κλάση εφαρμογών που εκμεταλλεύονται και χρησιμοποιούν πόρους που βρίσκονται οπουδήποτε στο Internet. Εδώ περιλαμβάνονται και συστήματα με κεντρικό εξυπηρετητή (π.χ. seti@home, Napster) ή εφαρμογές που ανήκουν στην περιοχή του grid computing.

Λειτουργικότητα Κόμβου

- Αναζήτηση άλλων κόμβων
- Τοποθέτηση και προσωρινή αντιγραφή περιεχομένου
- Δρομολόγηση
- Σύνδεση/αποσύνδεση από άλλους γειτονικούς κόμβους
- Εισαγωγή περιεχομένου
- Ανάκτηση περιεχομένου
- Αποκρυπτογράφηση και επιβεβαίωση περιεχομένου

Grids

- Τα Grids είναι κατανεμημένα συστήματα που επιτρέπουν τη μεγάλης κλίμακας συντονισμένη χρήση και διαμοίραση γεωγραφικά διασπαρμένων πόρων που βασίζονται σε μόνιμες, βασισμένες σε πρότυπες υποδομές, με σκοπό την υψηλή απόδοση.
- Θα μπορούσε κάποιος να πει ότι τα grids ασχολούνται κυρίως με τα θέματα που αφορούν την υποδομή και όχι την ανοχή σε σφάλματα, ενώ τα συστήματα P2P το αντίστροφο.
- Όσο περνούν τα χρόνια τα δύο είδη συστημάτων συγκλίνουν.

Κατηγοριοποίηση P2P Συστημάτων

— Τα συστήματα P2P έχουν υιοθετηθεί από ποικίλες εφαρμογές, διαφόρων κατηγοριών που περιλαμβάνουν τις παρακάτω:

- Επικοινωνία και συνεργατικότητα: εδώ ανήκουν συστήματα που παρέχουν υποδομή που διευκολύνει την απ' ευθείας, πραγματικού χρόνου επικοινωνία και συνεργασία μεταξύ διαφορετικών κόμβων (π.χ. Chat, IRC, κλπ.).
- Κατανεμημένος υπολογισμός: εδώ ανήκουν συστήματα που έχουν σαν σκοπό την αύξηση της υπολογιστικής ισχύος. Οι εργασίες χωρίζονται σε μικρά υποέργα που ανατίθενται σε διαφορετικούς κόμβους. Στα συστήματα αυτά απαιτείται συνήθως κάποια κεντρική διαχείριση (π.χ. Seti@home).

Κατηγοριοποίηση P2P Συστημάτων (2)

- Υποστήριξη υπηρεσιών Internet: p2p multicast systems, εφαρμογές ασφαλείας, εφαρμογές προστασίας απέναντι σε επιθέσεις τύπου άρνησης εξυπηρέτησης ή υιών, κλπ.
- Κατανεμημένες Βάσεις Δεδομένων: εφαρμογές όπου το σύνολο όλων των δεδομένων θεωρείται ότι αποτελείται από μη συμβατές τοπικές σχεσιακές βάσεις δεδομένων που διασυνδέονται μέσω κανόνων μετάφρασης και σημασιολογικών εξαρτήσεων μεταξύ αυτών (π.χ. Piazza).
- Διαμοίραση περιεχομένου: διαμοίραση ψηφιακού περιεχομένου και άλλου υλικού μεταξύ των κόμβων (π.χ. Napster, Kazaa, Chord)

P2P Συστήματα Διανομής Περιεχομένου

- P2P εφαρμογές
  - P2P συστήματα διαμοίρασης αρχείων
  - P2P συστήματα δημοσίευσης περιεχομένου και αποθήκευσης
- P2P υποδομές: παρέχουν τις παρακάτω υπηρεσίες
  - Αποτελεσματική τοποθέτηση δεδομένων και δρομολόγηση
  - Ανωνυμία
  - Διαχείριση συμπεριφοράς χρηστών

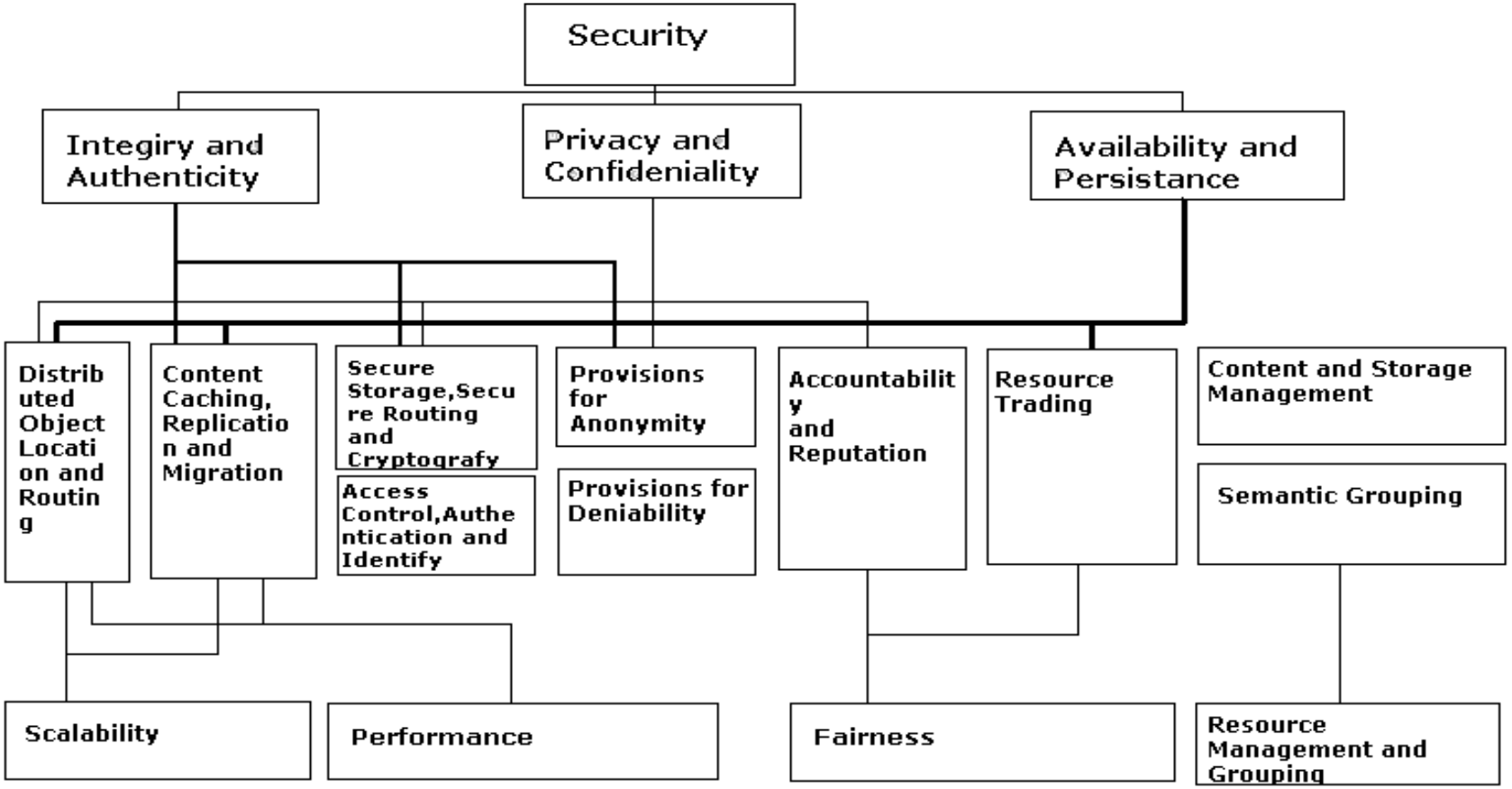
Χαρακτηριστικά Συστημάτων Διαμοίρασης Περιεχομένου

- Ασφάλεια
  - Πιστοποίηση και ακεραιότητα
  - Εμπιστευτικότητα
  - Διαθεσιμότητα και Μονιμότητα
- Δυνατότητα Κλιμάκωσης: απότομη αύξηση του αριθμού των κόμβων και των δεδομένων δεν θα έχει επιπτώσεις στην απόδοση και τη διαθεσιμότητα.
- Απόδοση
- Δικαιοσύνη: οι χρήστες προσφέρουν και εκμεταλλεύονται το παρεχόμενο υλικό με ισορροπημένο τρόπο.

Χαρακτηριστικά Συστημάτων Διαμοίρασης Περιεχομένου (2)

- Δυνατότητα Διαχείρισης Περιεχομένου: δημοσίευση, αναζήτηση, ανάκτηση, διόρθωση, διαγραφή, διαχείριση αποθηκευτικού χώρου, υπηρεσίες μεταδεδομένων.
- Σημασιολογική ομαδοποίηση πληροφορίας:
  - ομαδοποίηση με βάση το ίδιο το περιεχόμενο
  - ομαδοποίηση με βάση τη θέση
  - ομαδοποίηση με βάση το δίκτυο
  - κλπ

Χαρακτηριστικά Συστημάτων Διαμοίρασης Περιεχομένου (3)



Κατανεμημένη Τοποθέτηση Αντικειμένων και Δρομολόγηση

- Το P2P σύστημα βασίζεται σε ένα "overlay" δίκτυο που διαμορφώνεται σαν ανεξάρτητο δίκτυο πάνω στο φυσικό δίκτυο.
- Η τοπολογία, η δομή, και ο βαθμός κεντρικοποίησης του "overlay" δικτύου, καθώς και οι μηχανισμοί δρομολόγησης και τοποθέτησης που υιοθετεί για τα μηνύματα και το περιεχόμενο είναι κρίσιμα για τη λειτουργία του συστήματος, καθώς επηρεάζουν την ανοχή σε σφάλματα, την αυτοσυντήρηση, την ασφάλεια, την ικανότητα κλιμάκωσης και την απόδοση του συστήματος.

Διάκριση Υπερκείμενων Δικτύων

- Με βάση την κεντρικοποίηση
  - Αμιγώς μη κεντρικοποιημένες αρχιτεκτονικές: οι κόμβοι λειτουργούν σαν servers και σαν clients (servents)
  - Μερικώς κεντρικοποιημένες αρχιτεκτονικές: κάποιοι κόμβοι δρουν σαν υπερκόμβοι, έχοντας ένα πιο σημαντικό ρόλο από άλλους. Σε περίπτωση σφάλματος οι υπερκόμβοι αντικαθίστανται από άλλους δυναμικά.
  - Υβριδικές μη κεντρικοποιημένες αρχιτεκτονικές: ένας κεντρικός εξυπηρετητής διευκολύνει την αλληλεπίδραση μεταξύ κόμβων διατηρώντας καταλόγους μεταδεδομένων που περιγράφουν τα δεδομένα που τηρούνται από τους κόμβους.

Διάκριση Υπερκείμενων Δικτύων (2)

- Με βάση τη δομή του δικτύου
  - Μη δομημένα: η τοποθέτηση αρχείων είναι εντελώς ανεξάρτητη από την τοπολογία του υπερκείμενου δικτύου (π.χ. Napster, Gnutella, Kazaa). Χρειάζονται αποτελεσματικοί μηχανισμοί αναζήτησης πληροφορίας.
  - Δομημένα: τα δεδομένα τοποθετούνται σε συγκεκριμένες θέσεις με βάση την τοπολογία (π.χ. Chord, CAN, Tapestry). Υπάρχει αντιστοίχηση ανάμεσα σε θέσεις και περιεχόμενο.
  - Χαλαρά δομημένα: π.χ. Freenet

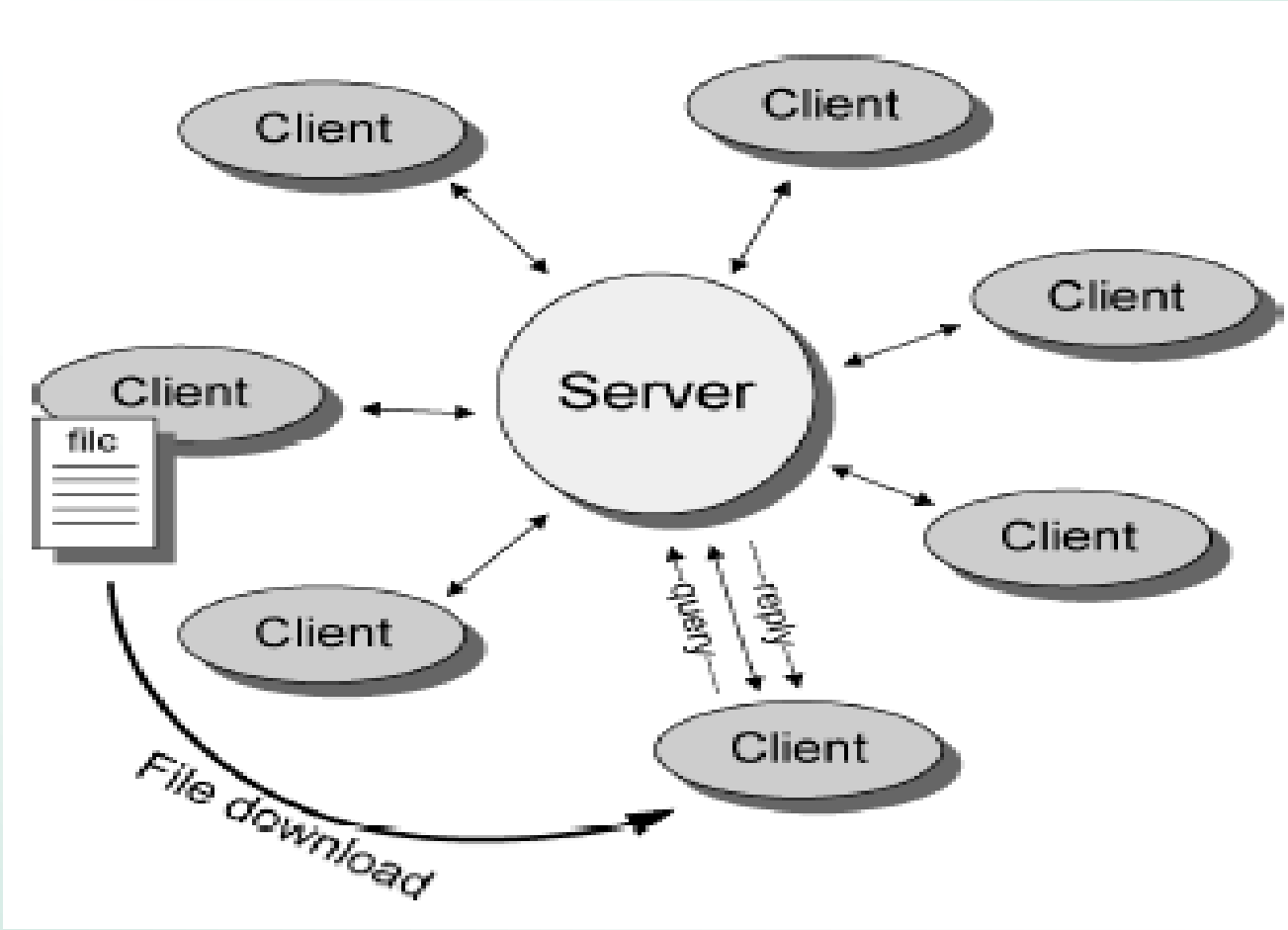
Διάκριση Υπερκείμενων Δικτύων (3)

	Centralization		
	Hybrid	**Partial**	**None**
Unstructured	Napster	Kazaa, Gnutella, Edutella	Gnutella
Structured Infrastructures			Chord, CAN
Structured Systems			PAST

Μη Δομημένες Αρχιτεκτονικές

- Υβριδικές Αποκεντρικοποιημένες
  - Όλοι οι clients συνδέονται με έναν κεντρικό υπολογιστή καταλόγου (central directory server) που διατηρεί:
    - έναν πίνακα με πληροφορίες σύνδεσης των καταχωρημένων χρηστών (π.χ. διεύθυνση IP, το εύρος ζώνης της σύνδεσης)
    - έναν πίνακα ο οποίος περιέχει μια λίστα με τα αρχεία που διατηρεί και διαμοιράζεται ο κάθε χρήστης, καθώς και τις περιγραφές των αρχείων σε μορφή μεταδεδομένων (π.χ. όνομα αρχείου, χρόνος δημιουργίας)

Μη Δομημένες Αρχιτεκτονικές (2)



Μη Δομημένες Αρχιτεκτονικές (3)

- Πλεονέκτημα: είναι απλά στην υλοποίηση και εντοπίζουν τα αρχεία γρήγορα και αποδοτικά
- Μειονέκτηματα:
  - είναι ευπαθή στον αυστηρό έλεγχο (censorship), τη νόμιμη δράση, την εποπτεία, την κακόβουλη επίθεση και την τεχνική αποτυχία
  - δεν έχουν ικανότητα κλιμάκωσης (unscalable)
- Παράδειγμα: Napster

Μη Δομημένες Αρχιτεκτονικές (4)

- Συστήματα που δεν ανήκουν στην υβριδική αποκεντρικοποιημένη κατηγορία μπορούν να χρησιμοποιήσουν κάποιον κεντρικό υπολογιστή σε περιορισμένη έκταση
  - π.χ. για το αρχικό bootstrapping συστήματος (π.χ. MojoNation [MojoNation, 2003]), ή
  - για την άδεια σύνδεσης νέων χρηστών στο δίκτυο με την παροχή πρόσβασης σε έναν κατάλογο υπαρχόντων χρηστών (π.χ. gnutellahosts.com για το δίκτυο gnutella).

Μη Δομημένες Αρχιτεκτονικές (5)

- Πλήρως αποκεντρικοποιημένα συστήματα
  - Παραδείγματα: Gnutella, FreeHaven
  - Δεν υπάρχει κανένας κεντρικός συντονισμός των ενεργειών στο δίκτυο και οι χρήστες (***servents***) συνδέονται ο ένας με τον άλλον άμεσα
  - Το Gnutella χρησιμοποιεί το IP ως τη βασική υπηρεσία δικτύου
  - Η επικοινωνία μεταξύ servents ορίζεται σε μια μορφή πρωτοκόλλου επιπέδου εφαρμογής υποστηρίζοντας τέσσερις τύπους μηνυμάτων

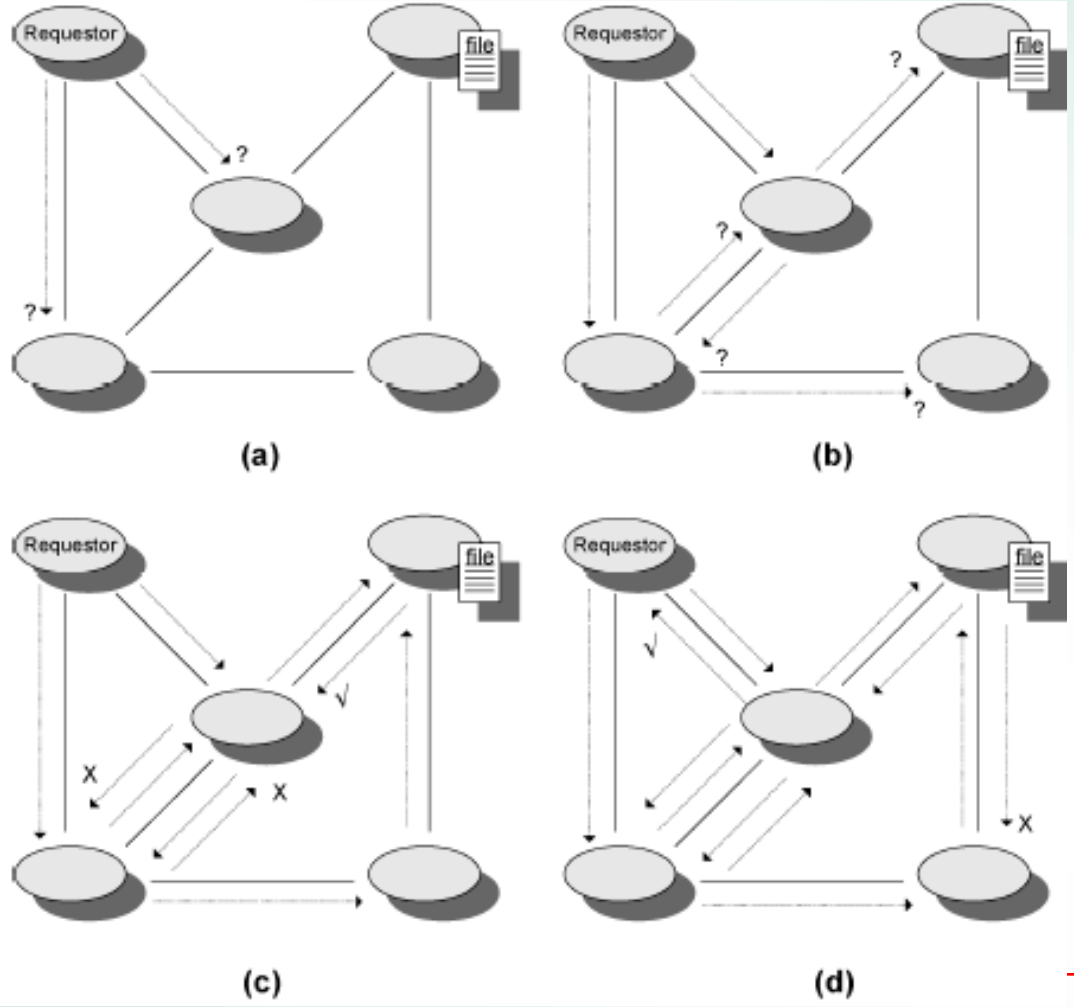
Μη Δομημένες Αρχιτεκτονικές (6)

- **Μηνύματα στο Gnutella:**
  - **Ping:** Ένα αίτημα για έναν host να δηλώσει την ύπαρξή του.
  - **Pong:** Απάντηση σε ένα μήνυμα Ping, το οποίο περιέχει την IP και τη θύρα του κόμβου που αποκρίνεται καθώς και τον αριθμό και το μέγεθος των διαμοιραζόμενων αρχείων.
  - **Query:** Ένα αίτημα αναζήτησης. Περιέχει το κείμενο αναζήτησης και τις ελάχιστες απαιτήσεις σε ταχύτητα του αποκρινόμενου host.
  - **Query Hits:** Απάντηση σε ένα Query μήνυμα. Περιέχει την IP, τη θύρα και την ταχύτητα απόκρισης του host καθώς και τον αριθμό των αρχείων που βρέθηκαν και το σύνολο των δεικτών τους.

Μη Δομημένες Αρχιτεκτονικές (7)

- Μετά την σύνδεση ενός node στο δίκτυο Gnutella, ο κόμβος στέλνει ένα *Ping* μήνυμα σε οποιοδήποτε κόμβο είναι συνδεμένος.
- Έπειτα, οι κόμβοι αυτοί στέλνουν ένα μήνυμα *Pong* που προσδιορίζουν μ' αυτό τον τρόπο τον εαυτό τους και επίσης διαδίδουν το μήνυμα *Ping* στους γειτονικούς nodes.
- Οι αρχικές αρχιτεκτονικές Gnutella χρησιμοποιούν μηχανισμούς καθολικής εκπομπής για να διανείμουν *Ping* και *Query* μηνύματα
- Για να περιοριστεί η διάδοση μηνυμάτων μέσω του δικτύου, σε κάθε επικεφαλίδα μηνύματος περιέχεται ένα πεδίο που ονομάζεται χρόνος ζωής (TTL). Σε κάθε hop, η τιμή αυτού του πεδίου μειώνεται και όταν φθάσει στη τιμή μηδέν, το μήνυμα απορρίπτεται.

Μη Δομημένες Αρχιτεκτονικές (8)



Μη Δομημένες Αρχιτεκτονικές (9)

- Μερικώς Κεντρικοποιημένα Συστήματα
  - Supernodes: κόμβοι που έχουν στόχο την δεικτοδότηση (indexing) και την ενδιάμεση αποθήκευση (caching) των αρχείων που υπάρχουν στο δίκτυο.
  - Οι peers επιλέγονται αυτόματα για να γίνουν supernodes εάν έχουν ικανοποιητικό εύρος ζώνης και επεξεργαστική ισχύ
  - Οι supernodes τοποθετούν σ' έναν κατάλογο τα διαμοιραζόμενα αρχεία που συνδέονται με τους peers και η proxy αναζήτηση γίνεται μέσω αυτών των peers. Όλες οι ερωτήσεις, επομένως, κατευθύνονται αρχικά στους supernodes.

Μη Δομημένες Αρχιτεκτονικές (10)

- Ο χρόνος ανακάλυψης μειώνεται
- Δεν υπάρχει κανένα σημείο αποτυχίας. Εάν ένας ή περισσότεροι supernodes καταστραφούν τότε οι κόμβοι που συνδέονται με αυτούς μπορούν να δημιουργήσουν νέες συνδέσεις με άλλους supernodes και το δίκτυο θα συνεχίσει να λειτουργεί κανονικά.
- Οι supernodes θα αναλάβουν μια μεγάλη μερίδα ολόκληρου του φορτίου του δικτύου, ενώ το μεγαλύτερο μέρος των κόμβων, οι οποίοι ονομάζονται "normal", θα είναι πολύ ελαφριά φορτωμένοι
- Παραδείγματα: Kazaa, Edutella

Μη Δομημένες Αρχιτεκτονικές (11)

- Επόμενη έκδοση του Gnutella:
  - Ένας μηχανισμός για την επιλογή supernodes οργανώνει το δίκτυο Gnutella σε μια διασύνδεση από *superpeers* και κόμβους πελάτες.
  - Όταν ένας κόμβος με αρκετή υπολογιστική ισχύ εισέρχεται στο δίκτυο, γίνεται αμέσως ένας *superpeer* και εγκαθιστά τις συνδέσεις με άλλα superpeers, δημιουργώντας ένα μη δομημένο επίπεδο δίκτυο από superpeers.
  - Εάν εγκαταστήσει έναν ελάχιστο αριθμό από συνδέσεις με κόμβους πελάτες μέσα σε ένα καθορισμένο χρόνο, παραμένει ένας *superpeer*. Διαφορετικά, μετατρέπεται σε κανονικό κόμβο πελάτη.

Δομημένες Αρχιτεκτονικές

- Freenet
- Chord
- CAN
- Tapestry

Freenet

- Χαλαρά δομημένο σύστημα: οι κόμβοι μπορούν να κάνουν εκτιμήσεις (όχι με βεβαιότητα) για τον κόμβο που μπορεί να αποθηκεύσει το περιεχόμενο.
- Αποφυγή καθολικής εκπομπής μηνυμάτων αίτησης σε όλους τους γείτονές τους ή σ' ένα τυχαίο υποσύνολό τους.
- Χρησιμοποιεί την προσέγγιση chain mode propagation, όπου κάθε κόμβος λαμβάνει τοπικά μια απόφαση για το ποιος κόμβος στέλνει το μήνυμα αίτησης στον επόμενο.

Freenet (2)

- Τα αρχεία προσδιορίζονται από μοναδικά δυαδικά κλειδιά.
- Εφαρμογή μιας hash συνάρτησης σε ένα περιγραφικό κείμενο που συνοδεύει κάθε αρχείο καθώς αυτό αποθηκεύεται στο δίκτυο από τον αρχικό ιδιοκτήτη του.
- Κάθε κόμβος διατηρεί τα δεδομένα του,
- Διατηρεί έναν δυναμικό πίνακα δρομολόγησης που περιέχει τις διευθύνσεις και τα αρχεία άλλων κόμβων.
- Για την αναζήτηση ενός αρχείου, ο χρήστης στέλνει ένα μήνυμα αίτησης που καθορίζει την τιμή του *κλειδιού (key)* και το χρόνο *timeout* (hops to live).

Freenet (3)

- Χρησιμοποιεί 4 τύπους μηνυμάτων, που όλοι περιλαμβάνουν τον identifier του κόμβου (για την ανίχνευση επαναλήψεων), την τιμή hopes to live και τα identifiers των κόμβων της πηγής και του προορισμού
  - Data insert
  - Data request
  - Data reply
  - Data failed

Freenet (4)

- Οποιοσδήποτε κόμβος λάβει το data insert μήνυμα, ελέγχει πρώτα εάν το κλειδί υπάρχει ήδη.
- Εάν το κλειδί δεν βρεθεί, ο κόμβος ανατρέχει στο πιο “κοντινό” κλειδί (από άποψη λεξικογραφικής απόστασης) στον πίνακα δρομολόγησής του και προωθεί το data insert μήνυμα στον αντίστοιχο κόμβο.
- Σύμφωνα μ’ αυτόν τον μηχανισμό, τα νέα αρχεία τοποθετούνται στους κόμβους που έχουν αρχεία με παρόμοια κλειδιά.
- Αυτό συνεχίζεται εφ' όσον δεν παραβιάζεται το όριο hops to live.

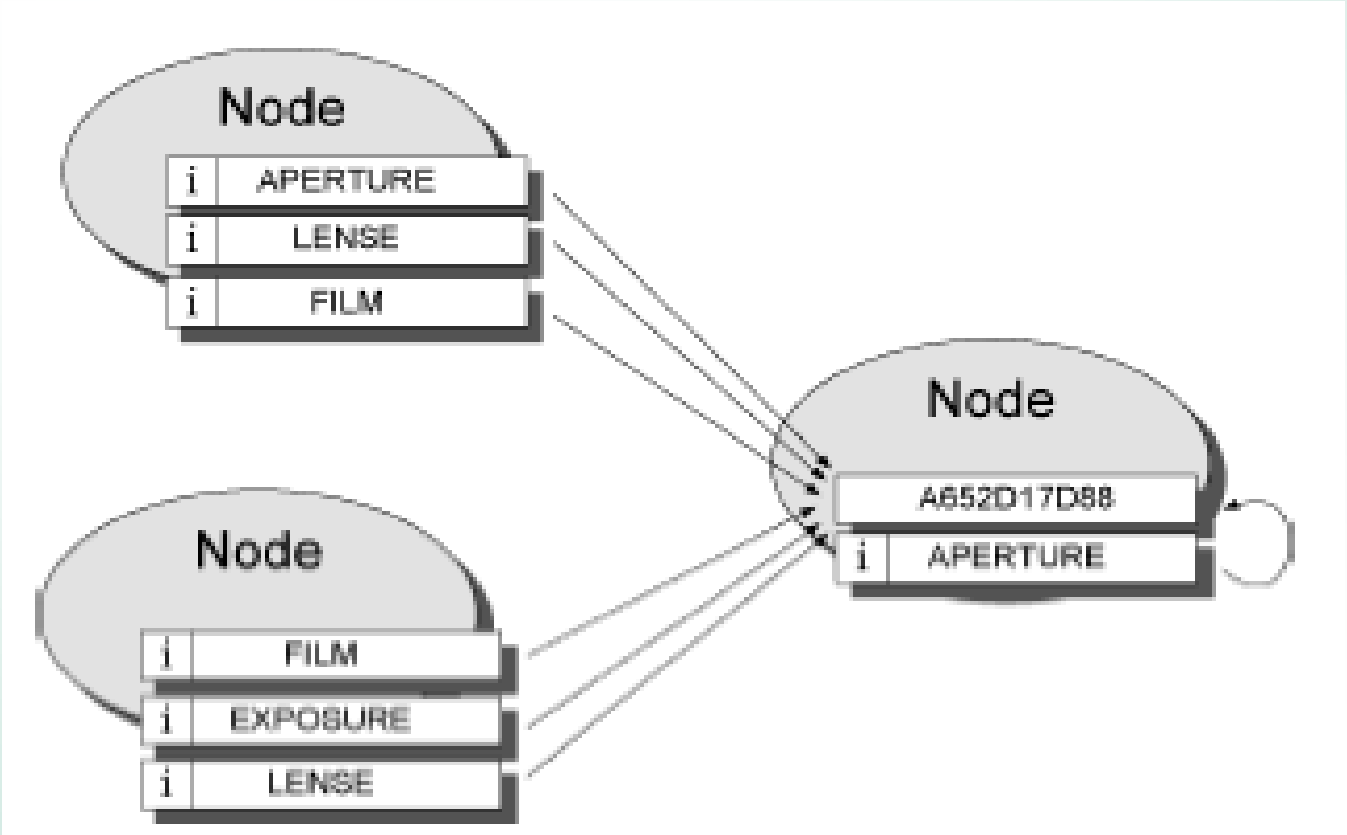
Freenet (5)

- Κατ' αυτό τον τρόπο, περισσότεροι από ένας κόμβοι θα αποθηκεύσουν το νέο αρχείο.
- Συγχρόνως, όλοι οι συμμετέχοντες κόμβοι θα ενημερώσουν τους πίνακες δρομολόγησής τους με τις νέες πληροφορίες
- Εάν το hopes to live όριο φτάνεται χωρίς σύγκρουση, ένα "all clear" αποτέλεσμα θα διαδοθεί πίσω στον αρχικό κόμβο (inserter), ενημερώνοντας τον ότι το insert μήνυμα ήταν επιτυχές.
- Εάν το κλειδί βρεθεί, ο κόμβος επιστρέφει το προϋπάρχων αρχείο σαν να έχει γίνει αίτηση για αυτό.

Freenet (6)

- Εάν ο κόμβος δεν διαθέτει το αρχείο που ο requestor αναζητά, προωθεί την αίτηση στο γείτονά του που είναι πιθανό να έχει το αρχείο, αναζητώντας το κλειδί του αρχείου στον τοπικό πίνακα δρομολόγησής του
- Οι κόμβοι αποθηκεύουν, επίσης, την ταυτότητα (ID) και άλλες πληροφορίες των αιτήσεων, προκειμένου να καθοδηγούν τα "Data reply" και τα "Data failed" μηνύματα.
- Εάν το ζητούμενο αρχείο βρεθεί, τότε ο αρχικός κόμβος λαμβάνει μια απάντηση που περιέχει το αρχείο, το οποίο αποθηκεύεται σε όλους τους ενδιάμεσους κόμβους για μελλοντικές αιτήσεις.

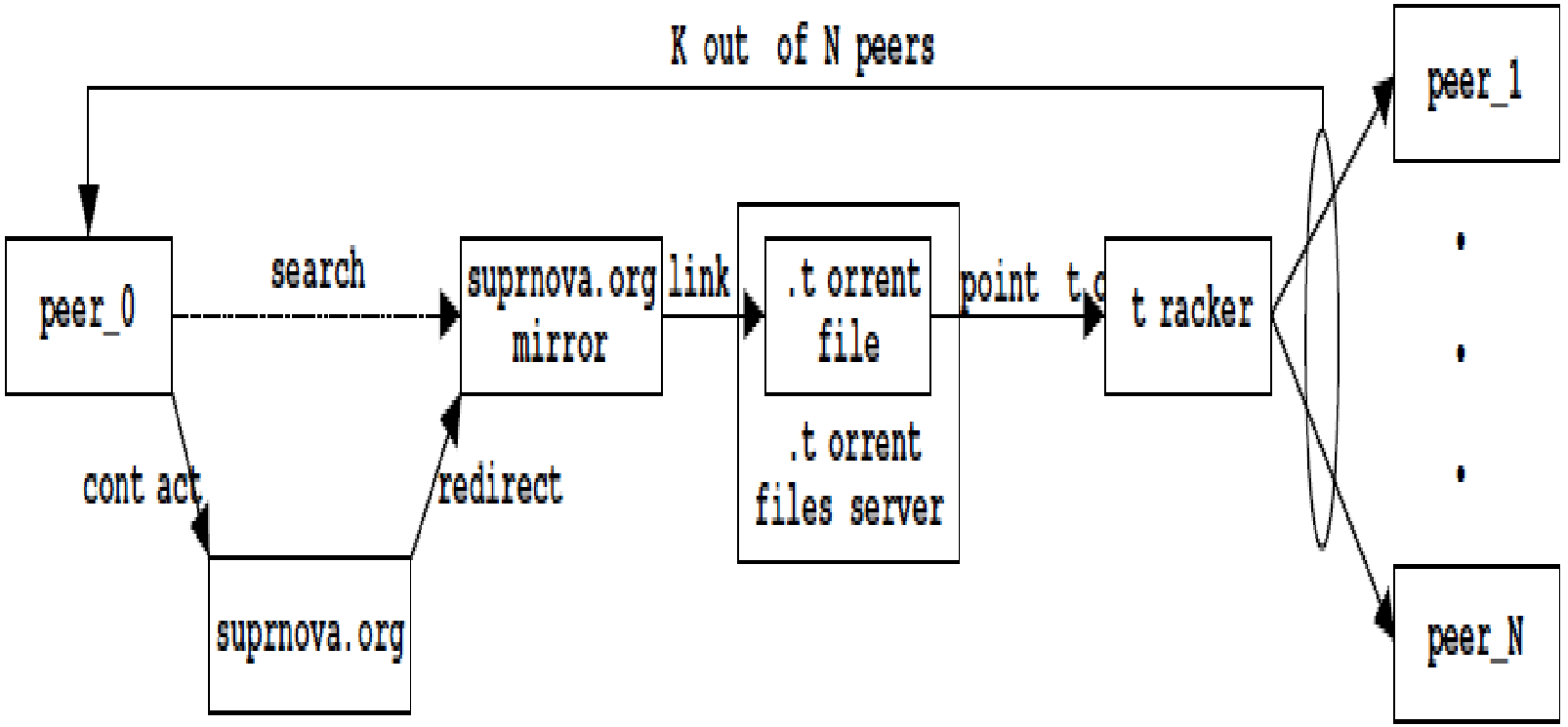
Freenet (7)



BitTorrent

- Έχει τρία πολύ βασικά χαρακτηριστικά:
  - Δεν χρησιμοποιεί κάποιο μηχανισμό αναζήτησης, αλλά ένα κεντρικό κατάλογο όπου γίνεται η αναζήτηση (suprnova)
  - Υιοθετεί μια πολιτική διαμοίρασης επιπέδου αρχείου και όχι τη συνήθη επιπέδου καταλόγου
  - Παρέχει ένα μηχανισμό ανταλλαγής μεταξύ των πελατών που κατεβάζουν το ίδιο αρχείο, ώστε να υπάρχει μεγαλύτερη δικαιοσύνη.

BitTorrent (2)



BitTorrent (3)

Added	Name	Filesize	Seeds	DLs	Quality	Submitter	Info
29-02	Castlevania (NES) completed in 12:23...	37 Mb	5	5	-	anonymous	link
17-03	Everquest2 Trailer	205 Mb	0	1	Windows	anonymous	-
07-03	FFX2:FMVS (Str8 FROM DVD)	919 Mb	1	14	PS2	T!dus_Beta	link
06-02	Partition Magic 8 0 español by Nitro...	50 Mb	4	2	N64	anonymous	-
30-11	Quake 1 beaten in 12 minutes, 23 sec...	166 Mb	8	5	Windows	haze	link
29-02	Rockman (NES) completed in 21:53 by ...	50 Mb	4	3	-	anonymous	link
15-11	S.T.A.L.K.E.R. Trailer 5 hi-res	47 Mb	0	1	Windows	Ezth Nahua	-

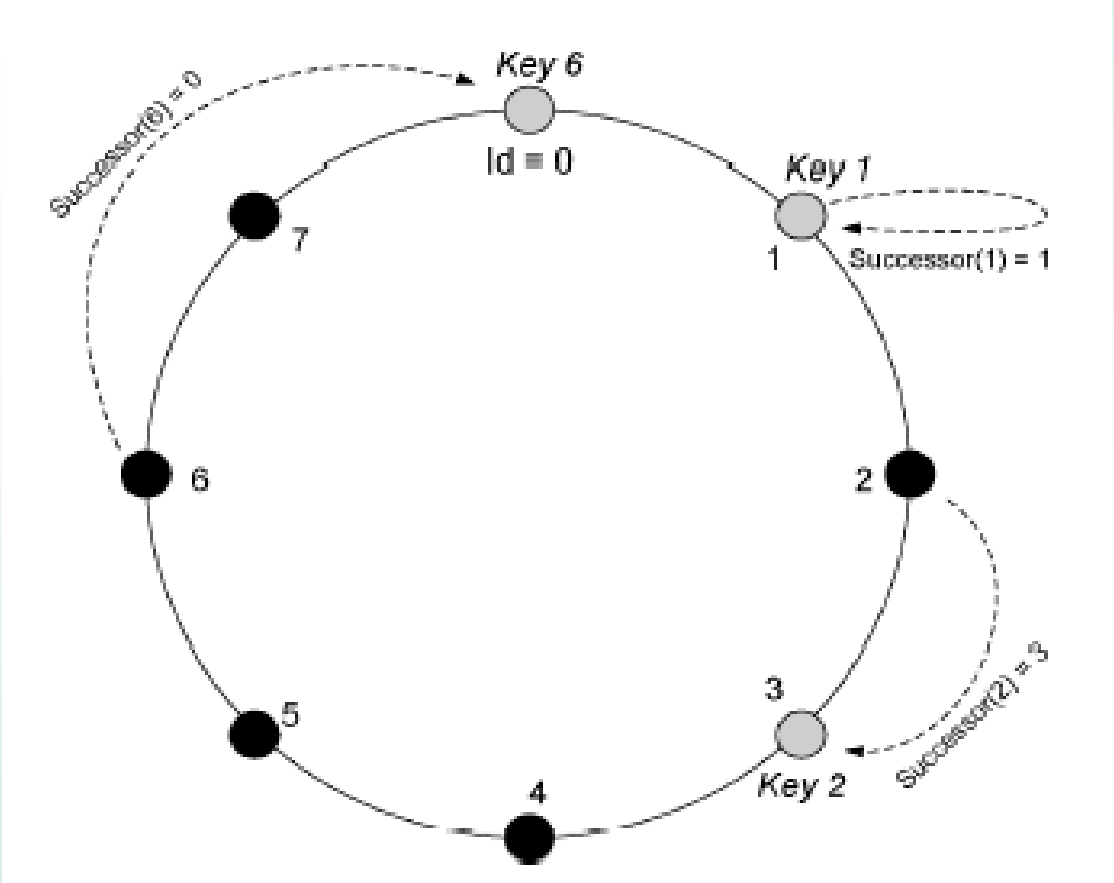
BitTorrent (4)

- .torrent : αρχείο μεταδεδομένων. Δείχνει σε ένα tracker
- tracker: διατηρεί πληροφορίες για τους peers που κατέβασαν πρόσφατα ή κατεβάζουν τώρα το αντίστοιχο περιεχόμενο και απαντά στο χρήστη με μια λίστα αυτών των peers.
- Ο peer μπορεί πλέον να εγκαταστήσει απ' ευθείας συνδέσεις με τους άλλους peers.
- Ένας tracker μπορεί να επιβλέπει το ταυτόχρονο κατέβασμα από πολλαπλά αρχεία
- Διαφορετικά .torrent αρχεία που αντιστοιχούν στο ίδιο αρχείο δεδομένων μπορεί να δείχνουν σε διαφορετικούς trackers.

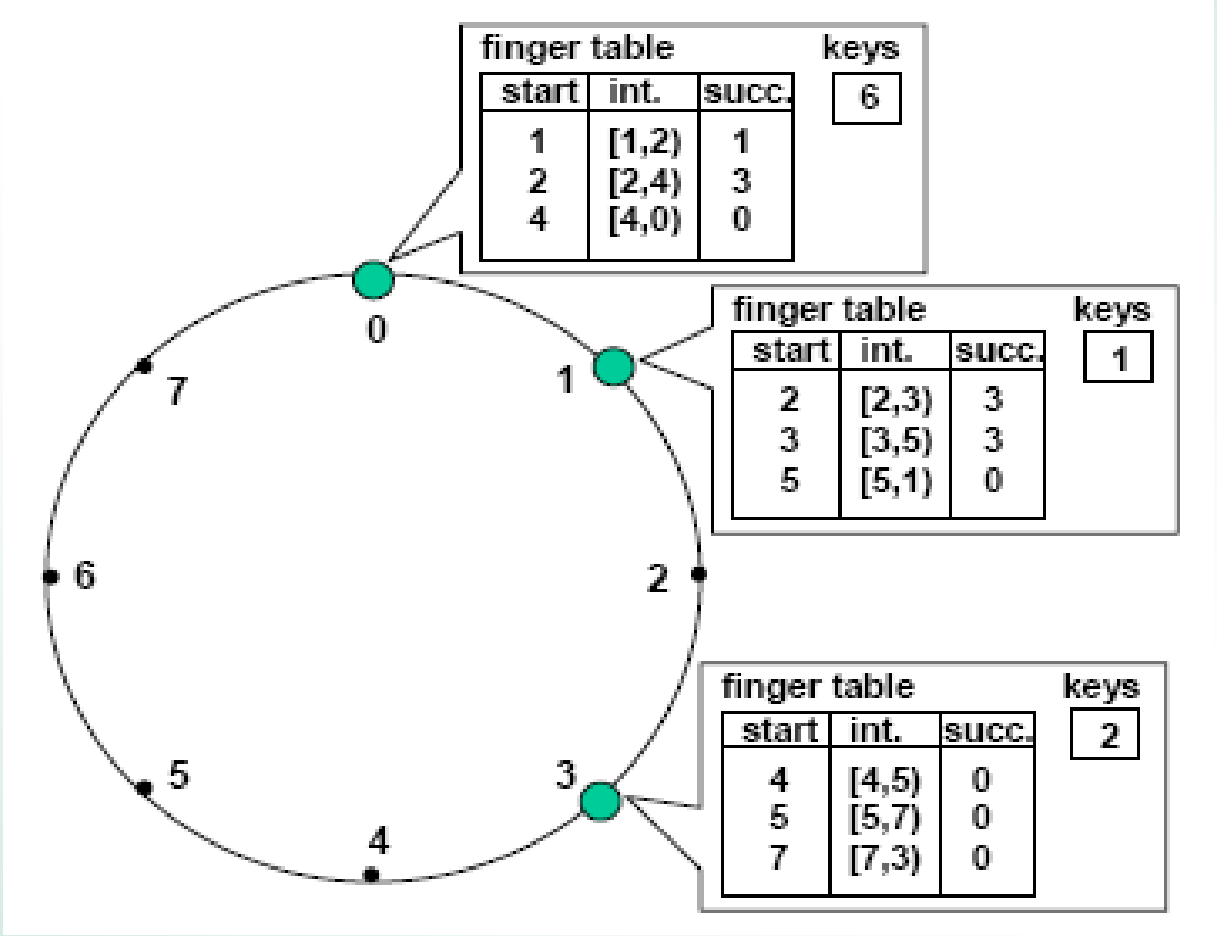
BitTorrent (5)

- Ένα αρχείο διαιρείται σε κομμάτια ίσου μεγέθους.
- Ένα κομμάτι διαιρείται περαιτέρω σε blocks.
- Το Block είναι η μονάδα μετάδοσης στο δίκτυο.
- Το πρωτόκολλο διατηρεί την πληροφορία για τα κομμάτια που έχουν ήδη κατέβει.
- Ο peer διατηρεί μια λίστα με τους peers που έχει συνδεθεί.
- Ο peer μπορεί να κάνει upload σε ένα υποσύνολο των παραπάνω peers.

Chord



Chord (2)



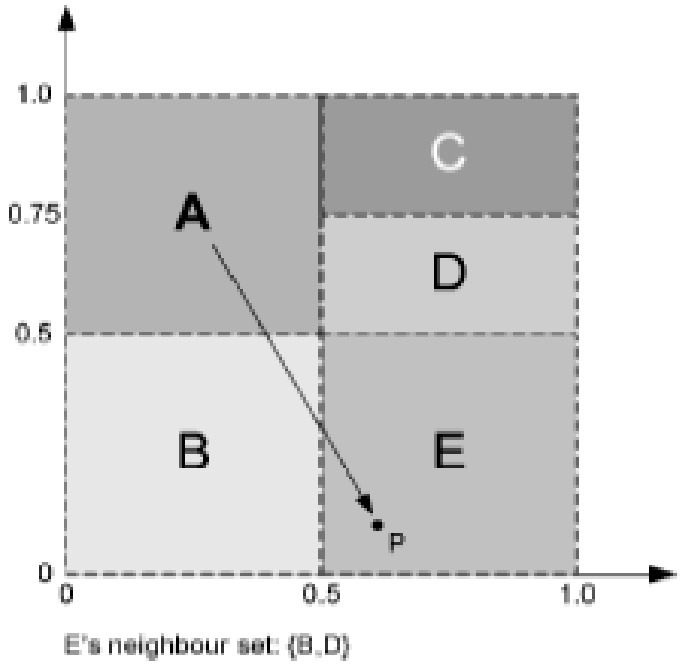
Chord (3)

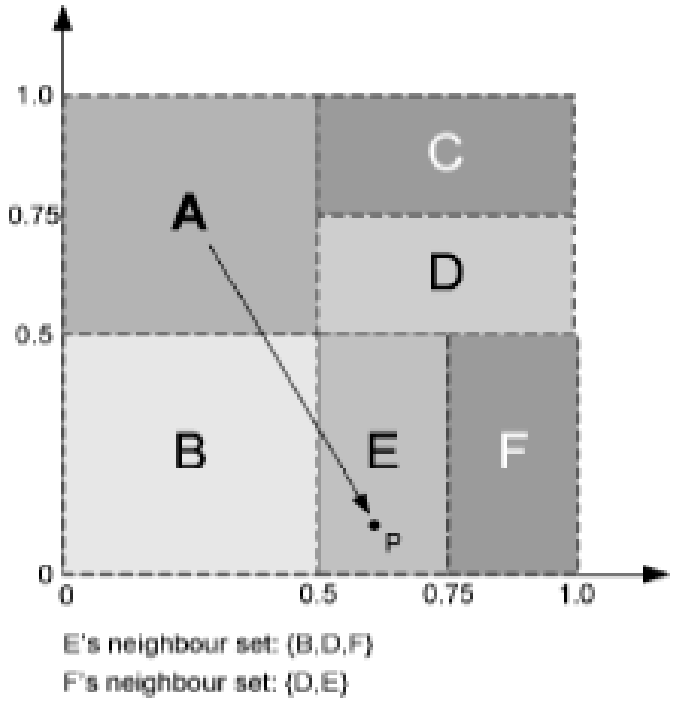
Notation	Definition
$finger[k].start$	$(n + 2^{k-1}) \bmod 2^m, 1 \leq k \leq m$
$.interval$	$[finger[k].start, finger[k+1].start)$
$.node$	first node $\geq n.finger[k].start$
successor	the next node on the identifier circle; $finger[1].node$
predecessor	the previous node on the identifier circle

CAN (Content Addressable Network)

- Κάθε κόμβος του δικτύου CAN αποθηκεύει ένα μέρος, το οποίο καλείται "ζώνη" (zone), του hash πίνακα καθώς επίσης και τις πληροφορίες για έναν μικρό αριθμό γειτονικών ζωνών στον πίνακα.
- Το CAN χρησιμοποιεί ένα εικονικό d-διαστάσεων Καρτεσιανό χώρο για να αποθηκεύει τα ζευγάρια *(κλειδί K, value V)*. Κάθε κόμβος αντιστοιχεί σε ένα τμήμα αυτού του διαστήματος που αποτελεί τη ζώνη του hash πίνακα.

CAN (2)





CAN (3)

- Οι CAN κόμβοι διατηρούν έναν πίνακα δρομολόγησης που περιέχει τις διευθύνσεις IP των nodes που έχουν τις γειτονικές ζώνες, για να επιτρέψει τη δρομολόγηση μεταξύ των αυθαίρετων σημείων στο χώρο.
- Η δρομολόγηση στο CAN επιτυγχάνεται ακολουθώντας την πορεία μέσω του Καρτεσιανού διαστήματος από την πηγή στον προορισμό.

Tapestry

- Το Tapestry είναι βασισμένο στους μηχανισμούς τοποθέτησης και δρομολόγησης που χρησιμοποιούνται στο *Plaxton mesh*.
- Το Plaxton mesh αποτελεί μια κατανεμημένη δομή δεδομένων που επιτρέπει στους κόμβους να τοποθετήσουν τα αντικείμενα και τα μηνύματα δρομολόγησής τους μέσω ενός αυθαίρετα-ταξινομημένου overlay δικτύου, χρησιμοποιώντας χάρτες δρομολόγησης μικρού και σταθερού μεγέθους.

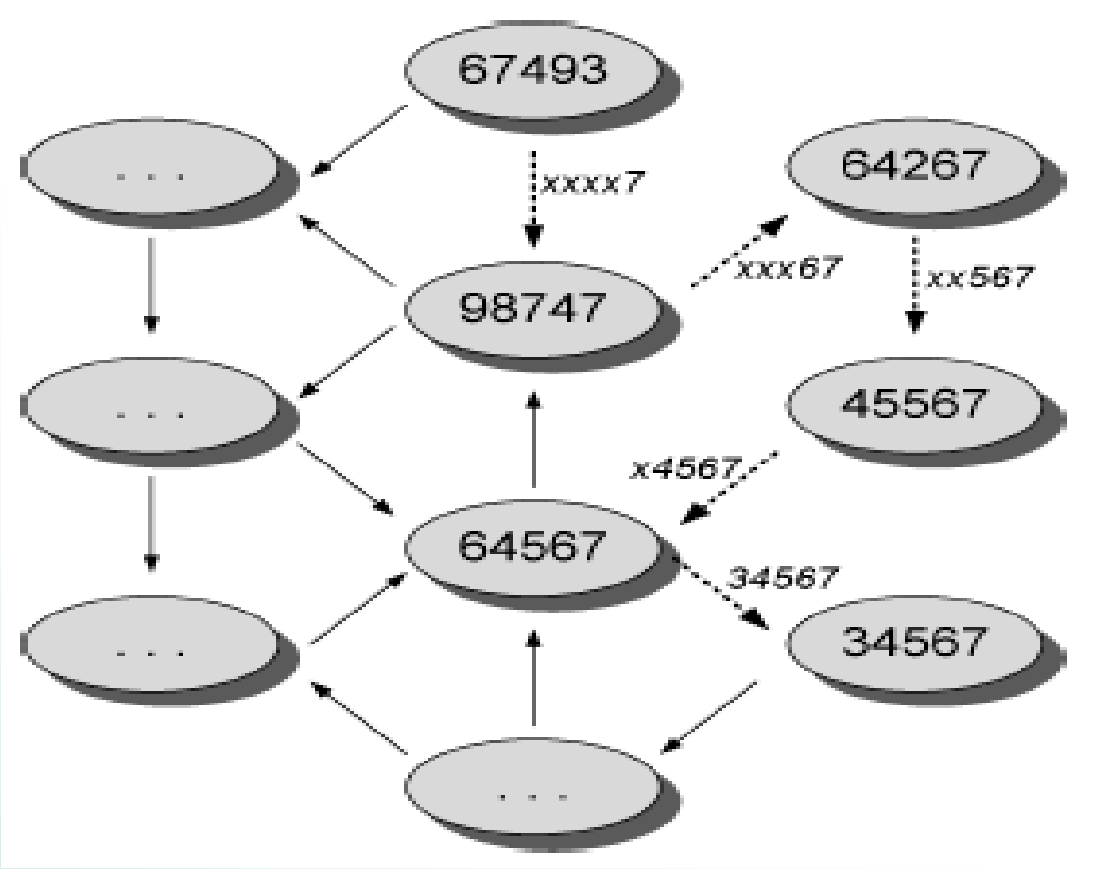
Tapestry (2)

- Κάθε κόμβος διατηρεί έναν *χάρτη γειτόνων (neighbor map)*, όπως φαίνεται στο παράδειγμα.
- Ο neighbor map έχει πολλαπλά επίπεδα.
- Κάθε επίπεδο l περιέχει τους δείκτες των κόμβων, των οποίων οι ταυτότητες πρέπει να αντιστοιχούν στα ψηφία l (τα x αντιπροσωπεύουν οποιαδήποτε ψηφία).
- Κάθε εγγραφή (entry) στο χάρτη γειτόνων αντιστοιχεί σε έναν δείκτη του πιο κοντινού κόμβου στο δίκτυο του οποίου η ταυτότητα ταιριάζει με τον αριθμό στο χάρτη γειτόνων, μέχρι μια θέση ψηφίων.

Tapestry (3)

	Level 5	Level 4	Level 3	Level 2	Level 1
Entry 0	07493	x0493	xx093	xxx03	xxxx0
Entry 1	17493	x1493	xx193	xxx13	xxxx1
Entry 2	27493	x2493	xx293	xxx23	xxxx2
Entry 3	37493	x3493	xx393	xxx33	xxxx3
Entry 4	47493	x4493	xx493	xxx43	xxxx4
Entry 5	57493	x5493	xx593	xxx53	xxxx5
Entry 6	**67493**	x6493	xx693	xxx63	xxxx6
Entry 7	77493	x7493	xx793	xxx73	xxxx7
Entry 8	87493	x8493	xx893	xxx83	xxxx8
Entry 9	97493	x9493	xx993	xxx93	xxxx9

Tapestry (4)



Tapestry (5)

- Χρησιμοποιεί έναν *root node* για κάθε αντικείμενο, ο οποίος χρησιμεύει στο να παρέχει έναν σίγουρο κόμβο από τον οποίο το αντικείμενο μπορεί να εντοπισθεί.
- Όταν ένα αντικείμενο *a* εισάγεται στο δίκτυο και αποθηκεύεται στον κόμβο n_s, ένας root node n_r ανατίθεται σε αυτό με τη χρήση ενός ντετερμινιστικού αλγορίθμου.
- Ένα μήνυμα δρομολογείται έπειτα από το n_s στο n_r, αποθηκεύοντας τα δεδομένα υπό μορφή χαρτογράφησης (object id a, storer id n_s) σε όλους τους κόμβους.

Tapestry (6)

- Κατά τη διάρκεια μιας ερώτησης θέσης, τα μηνύματα που προορίζονται για το a δρομολογούνται αρχικά προς το n_r, έως ότου συναντήσει έναν κόμβο που περιέχει την (a, n_s) αντιστοιχία.
- Μειονεκτήματα:
  - την ανάγκη για τη σφαιρική γνώση που απαιτείται για την ανάθεση και τον προσδιορισμό των κόμβων ρίζας και
  - την ευπάθεια των root nodes.

Data Hash Tables (DHTs)

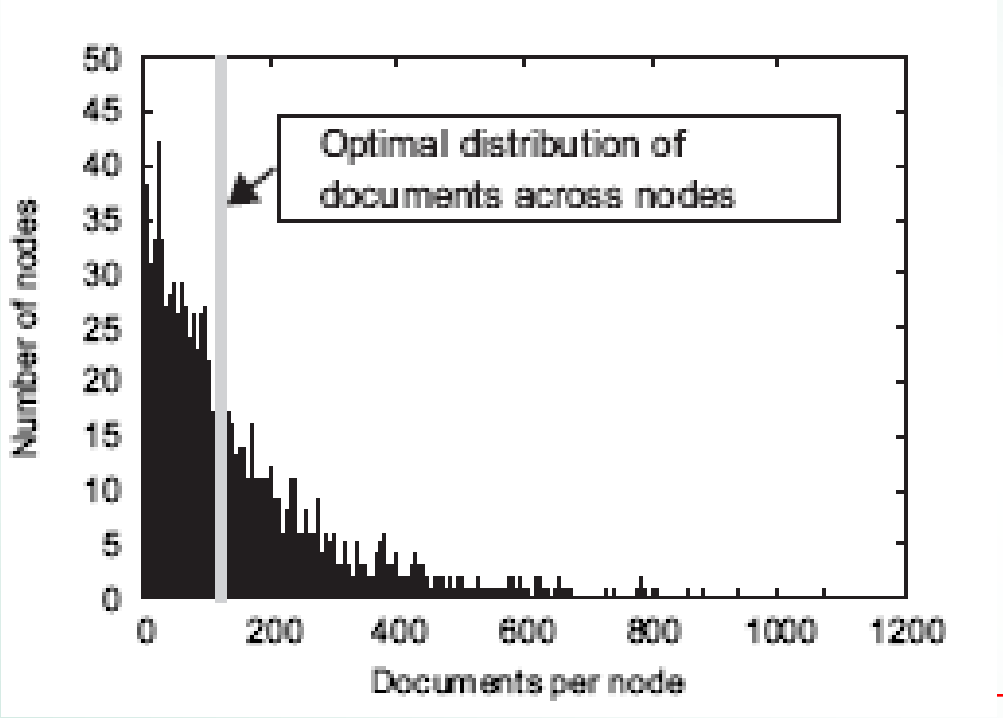
- Σχεδιάστηκαν για τα δομημένα peer-to-peer συστήματα
- Αντιστοιχεί τα δεδομένα σε ένα γραμμικό χώρο διευθύνσεων και διανέμει τα κλειδιά και τα αντίστοιχα δεδομένα τους με δομημένο τρόπο.
- Ο χώρος χωρίζεται σε διαστήματα, τα οποία αντιστοιχίζονται σε ατομικούς κόμβους.
- Κάθε κόμβος είναι υπεύθυνος για την διαχείριση των δεδομένων που υπάρχουν στο διάστημα που ελέγχει.
- Σχηματίζει ένα κατανεμημένο hash πίνακα.

Data Hash Tables (DHTs) (2)

- Οι αιτήσεις για δεδομένα ενός διαστήματος πρέπει να απαντηθούν από τον κόμβο που είναι υπεύθυνος για αυτό το διάστημα.
- Εάν μια αίτηση για ένα άλλο διάστημα φτάσει σε έναν κόμβο, αυτός πρέπει να την προωθήσει άμεσα στο κατάλληλο διάστημα σύμφωνα με τις πληροφορίες δρομολόγησης του DHT
- Όλες οι εκτιμήσεις όσον αφορά την πολυπλοκότητα ενός DHT σε σχέση με τα κόστη αναζήτησης, διαχείρισης και αποθήκευσης, βασίζονται στην υπόθεση ότι τα δεδομένα είναι σχεδόν εξ ίσου μοιρασμένα.

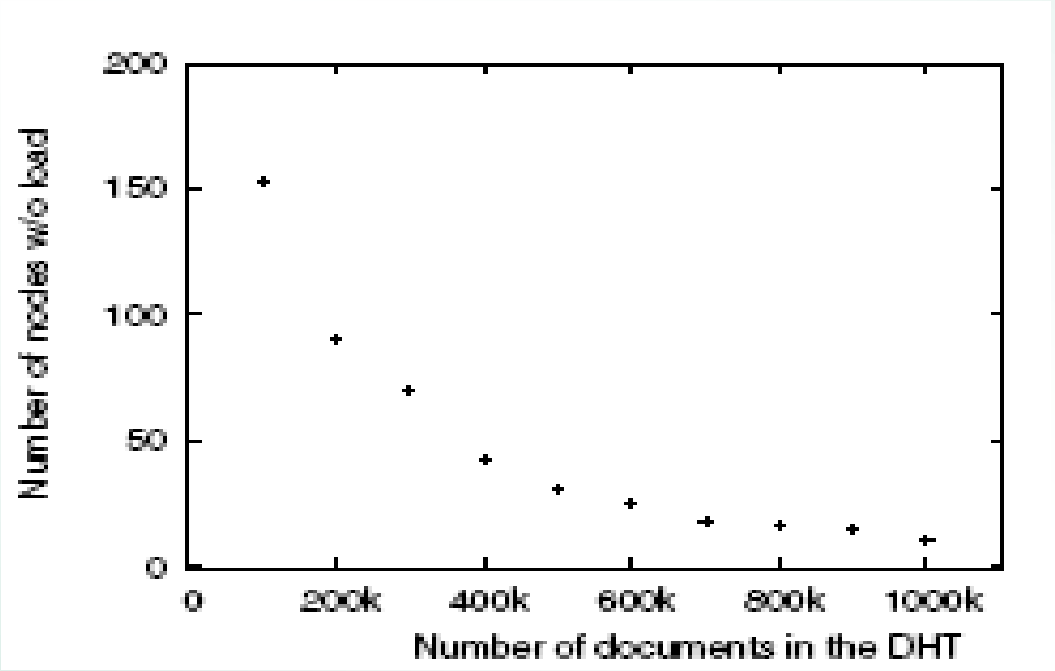
Data Hash Tables (DHTs) (3)

- Πείραμα προσομοίωσης σε Chord σύστημα: Η συχνότητα κατανομής του DHT των κόμβων αποθηκεύοντας έναν σταθερό αριθμό documents



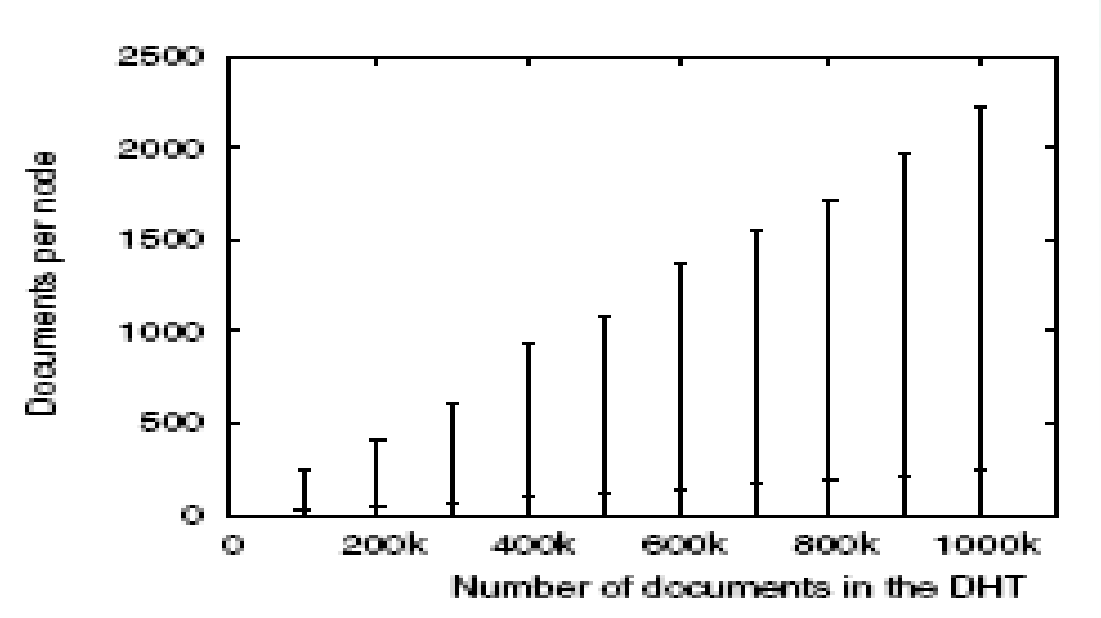
Data Hash Tables (DHTs) (4)

— Πείραμα προσομοίωσης σε Chord σύστημα : Ο αριθμός των κόμβων που δεν έχουν documents



Data Hash Tables (DHTs) (5)

— Πείραμα προσομοίωσης σε Chord σύστημα : Ο μικρότερος, ο μέσος όρος και ο μεγαλύτερος αριθμός των documents ανά κόμβο



Data Hash Tables (DHTs) (6)

- Είναι φανερό πως δεν μπορεί να επιτευχθεί ίση κατανομή των δεδομένων με τη χρήση μιας απλής hash συνάρτησης.
- Συνεπώς, θα πρέπει να αναπτυχθούν επιπλέον μηχανισμοί για την εξισορρόπηση του φορτίου δεδομένων μεταξύ των κόμβων.
- Το φορτίο ενός συστήματος με *N* κόμβους θεωρείται εξισορροπημένο, αν το φορτίο των δεδομένων κάθε κόμβου είναι το *1/N* του συνολικού φορτίου.

DHTs - Chord

- Το Chord χρησιμοποιεί μια παραλλαγή της consistent hashing για να ορίσει τα κλειδιά στους κόμβους.
- Η consistent hashing χρησιμοποιείται για την εξισορρόπηση του φορτίου, δεδομένου ότι κάθε κόμβος λαμβάνει κατά προσέγγιση τον ίδιο αριθμό κλειδιών, και περιλαμβάνει σχετικά λίγη μετακίνηση των κλειδιών όταν οι κόμβοι συνδέονται και αποσυνδέονται από το σύστημα.
- Η consistent hash συνάρτηση αναθέτει σε κάθε κόμβο και κλειδί έναν identifier από m bits χρησιμοποιώντας μια βασική hash συνάρτηση, όπως την SHA-1

DHTs – Chord (2)

- Ο identifier ενός κόμβου προσδιορίζεται χρησιμοποιώντας τη συνάρτηση hash πάνω στη διεύθυνση IP του κόμβου
- Ο identifier ενός κλειδιού παράγεται με την ίδια πράξη πάνω στο κλειδί.
- Το μήκος του identifier θα πρέπει να είναι αρκετά μεγάλο ώστε η πιθανότητα δύο κόμβοι ή κλειδιά να πάρουν την ίδια τιμή hash για τον identifier να είναι αμελητέα.
- Μηχανισμοί successor και finger table

DHTs – Chord (3)

- ΘΕΏΡΗΜΑ $1^{Ο}$: Για οποιοδήποτε σύνολο N κόμβων και K κλειδιών, με μεγάλη πιθανότητα ισχύει:
  - Κάθε κόμβος είναι υπεύθυνος για το πολύ (1+ϵ)K/N κλειδιά,
  - Όταν ένας $(N+1)^{st}$ κόμβος συνδέεται ή αποσυνδέεται από το δίκτυο, η ευθύνη για O(K/N) κλειδιά αλλάζει χέρια
- ΘΕΏΡΗΜΑ $2^{Ο}$: Με μεγάλη πιθανότητα, ο αριθμός των κόμβων, ο οποίος πρέπει να συνδεθεί για να βρεθεί ένας successor μέσα σε ένα δίκτυο με N κόμβους είναι $O(\log N)$.

DHTs – Chord (4)

- ΘΕΏΡΗΜΑ $3^{Ο}$: Με μεγάλη πιθανότητα, κάθε συνδεόμενος ή αποσυνδεόμενος κόμβος σε ένα Chord δίκτυο, με N κόμβους, θα χρησιμοποιεί μηνύματα της τάξης του $O(\log_2 N)$, για να επανεγκαταστήσει τις σταθερές δρομολόγησης και τα finger tables του Chord.
- Για να απλοποιηθούν οι μηχανισμοί σύνδεσης και αποσύνδεσης, κάθε κόμβος στο Chord διατηρεί έναν *predecessor pointer*.