Peer-to-peer услуги

Цел

Дава възможност за споделяне на данни и ресурси в голям мащаб без използване на отделно управлявани сървъри и съответстващата с тях инфраструктура.

Предоставяне на разпределени услуги и приложения използвайки данните и ресурсите на персонални компютри и работни станции, свързани в различни мрежи.

Характеристики на P2P системите

- Дизайнът им осигурява всеки потребител да допринася с ресурси към системата
- Въпреки разликата в предоставените ресурси, всички възли имат една и съща функционалност и отговорност
- Операциите не зависят от наличието на централна услуга за администриране
- Могат така да са проектирани, че да предлагат ограничена анонимност на предлагащите и използващите ресурсите
- Ключов момент за ефикасността на операциите е изборът на алгоритъм за разполагане на данните между много хостове и начинът на достъп до тях

Особености на P2P системите

- Различни собственици на компютри и мрежови връзки – няма гаранция за наличен ресурс, дали е свързан и ли без отказ.
- Наличността на процесите и компютрите е непредсказуема.
- Услугите не могат да се базират на гарантиран достъп.

Развитие на P2P системите

- 1-ва генерация – Napster music exchange service (2001)
- 2-ра генерация – Приложения за споделяне на файлове, предоставящи голяма разширяемост, анонимност и отказоустойчивост (Gnutella, Kazaa, BitTorent)
- 3-та генерация – поява на Middleware слоеве за приложно-независимо управление на разпределени ресурси в глобален мащаб. P2P Middleware платформи (Pastry, Chord, Kademlia).

P2P системи 3gen

- Платформите позволяват разполагане на ресурси на множество компютри в Интернет.
- Маршрутизиране на съобщенията към тях по заявка на клиентите
- Клиентите нямат грижа как и къде са разположени ресурсите.
- Гарантират доставката на заявките за ограничен брой от хопове.
- Дават възможност за разполагане на реплики на ресурси на налични хостове.

Идентифициране на ресурсите

- Използват се глобални уникални идентификатори (GUID) – обикновено се генерират като хеш от някое или от всички състояния на ресурс.

Routing overlays

- **Routing overlay (RO)** – разпределен алгоритъм за локализиране на възлите и обекти
- Middleware отговаря за маршрутизирането на заявките от всеки клиент до хост, който съдържа заявения обект.
- “Overlay” – маршрутизиращият механизъм е на приложно ниво и е различен от този на мрежовото.

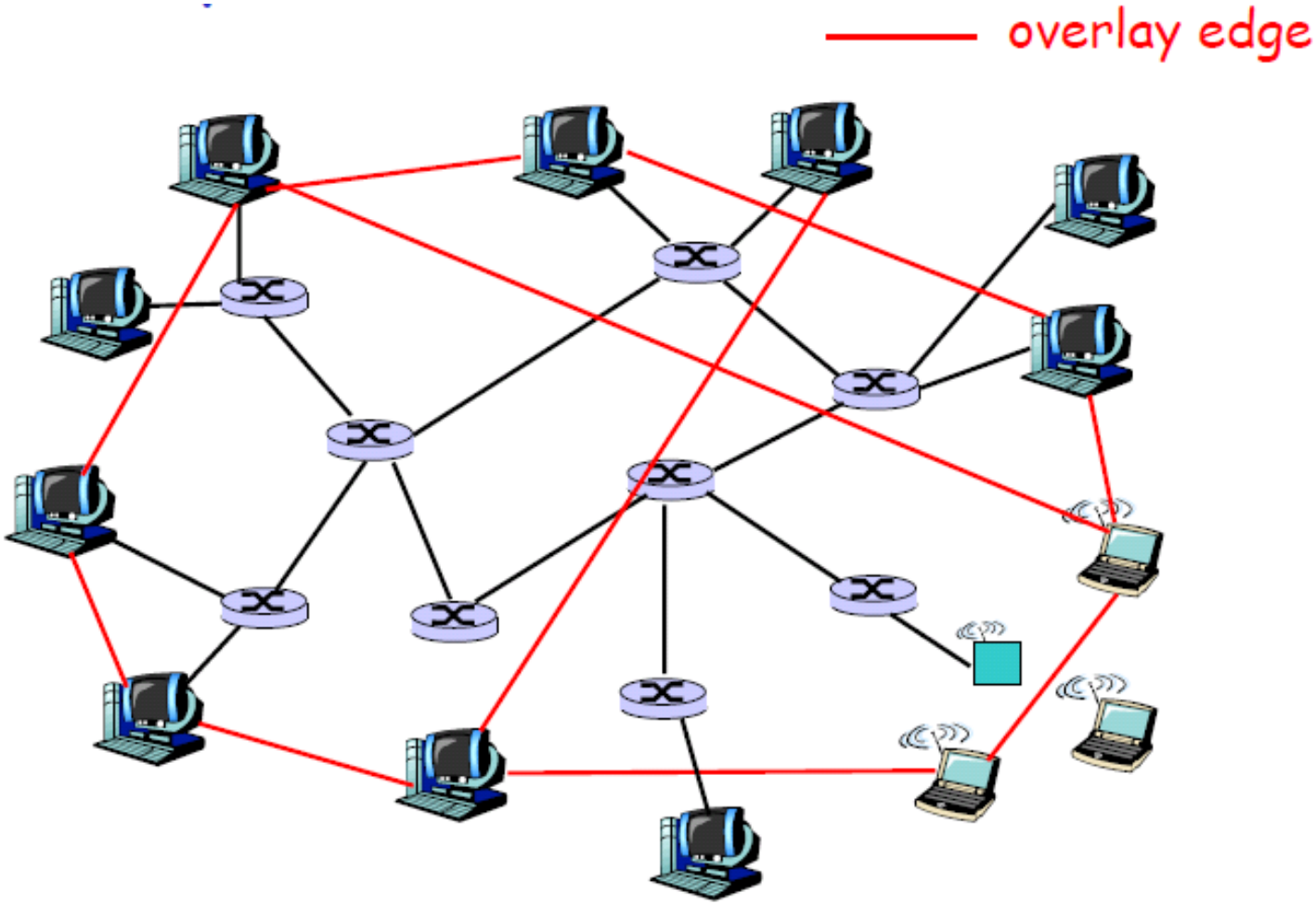
Routing overlays

- Механизмът осигурява възел да може да достъпва обект чрез маршрутизиране на всяка заявка през последователност от възли, използвайки познанията на всеки от тях за локализиране на обекта.
- RO поддържа знания за местоположението на всички реплики на обектите и доставя заявките до най-близкия активен възел, имащ копие на изисквания обект.

Overlay граф

- Виртуален ръб (Virtual edge) – TCP конекция или указател към IP адрес
- Поддържане на overlay:
  - Периодично пингване на съседен възел
  - Ако съседен възел отпадне, създава нов ръб
  - Стартиране на нов възел

Overlay мрежа



Разпределени изчисления

- P2P системите позволяват използване на изчислителната мощ на свързаните компютри

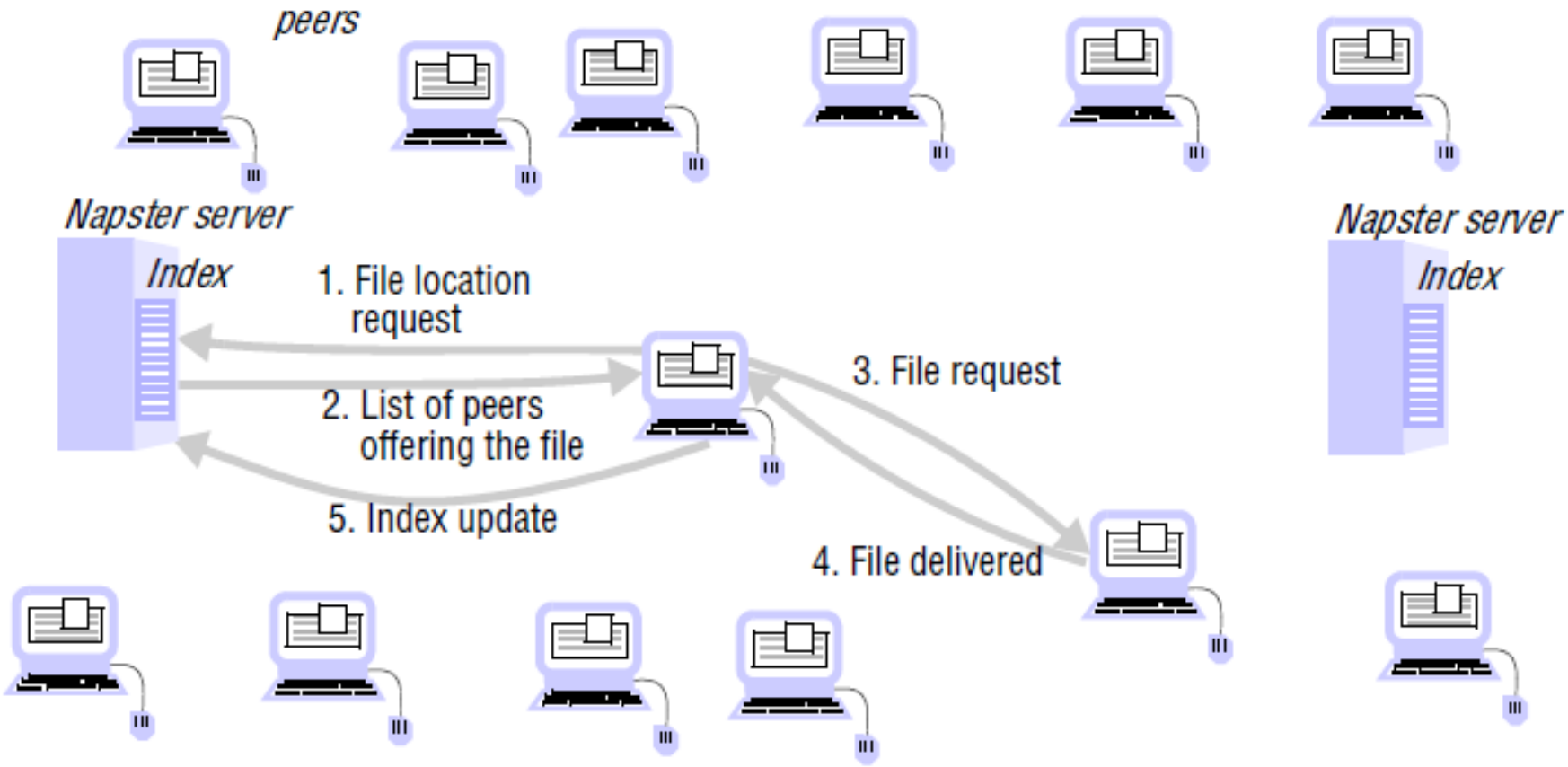
<u>Пример:</u>

Проект SETI@home – част от проекта *Search for Extra-Terrestrial Intelligence*.

Разпределя се потока данни от радиотелескоп в 107 sec единици по 350Kb към машините на доброволци. Разпределянето и координацията на резултатите се поддържа от единичен сървър.

През 2002 са използвани 3.91 милиона персонални компютри за обработка на 221 мил. работни единици за 12 месеца, средно 24.36 teraflops изчислителна мощ.

Napster


peers
Napster server
Index
1. File location request
2. List of peers offering the file
3. File request
4. File delivered
5. Index update
Napster server
Index

Особености на Napster

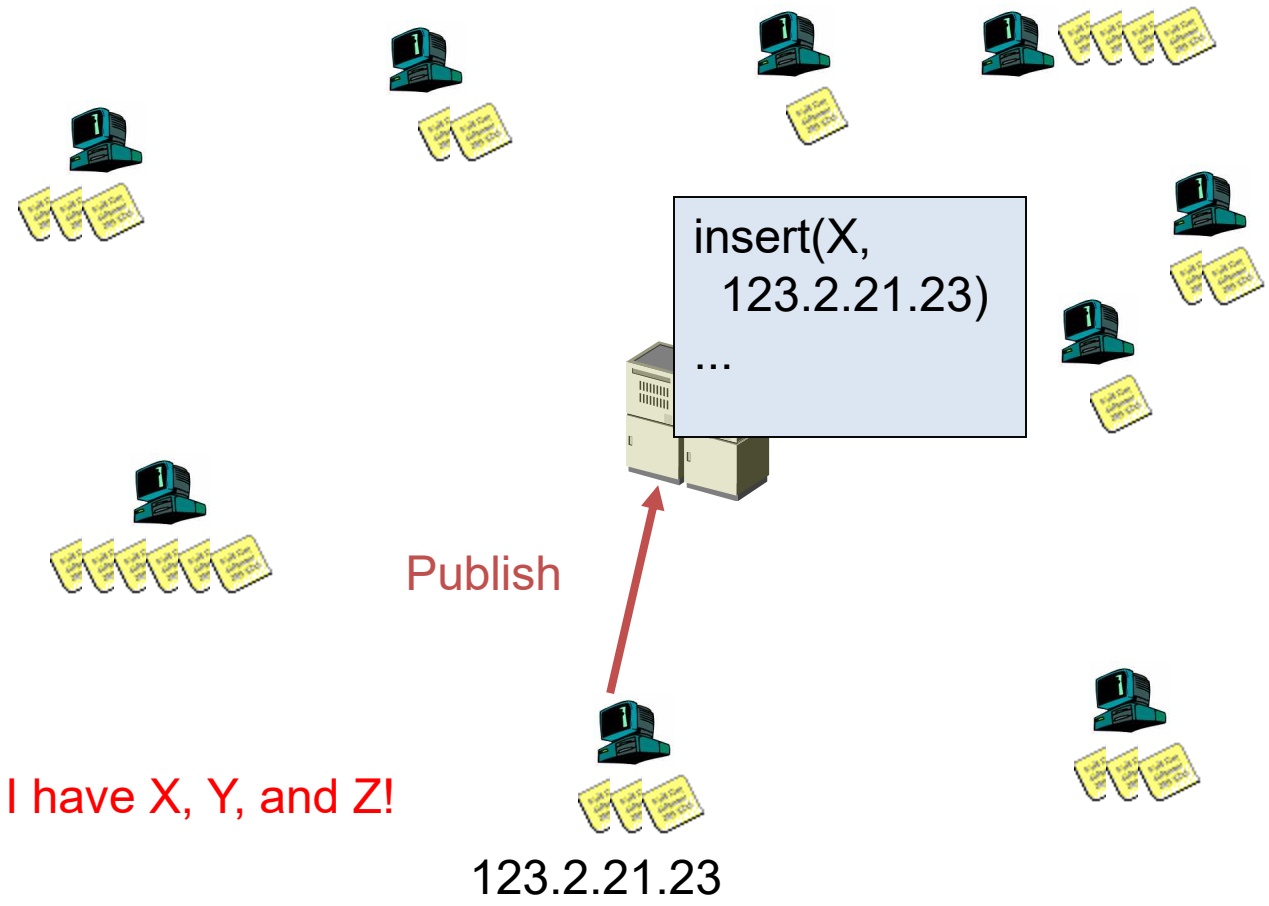
- Широкообхватна услуга, зависеща почти изцяло от данните и компютрите, свързани през Интернет.
- Поддържа унифициран индекс на налични файлове.
- Няма нужда всички реплики да са еднородни след обновявания – музикалните файлове никога не се обновяват
- Няма гаранция за наличността на отделни файлове – ако музикалния файл е временно недостъпен, той може да изтегли по-късно.

Napster - операции

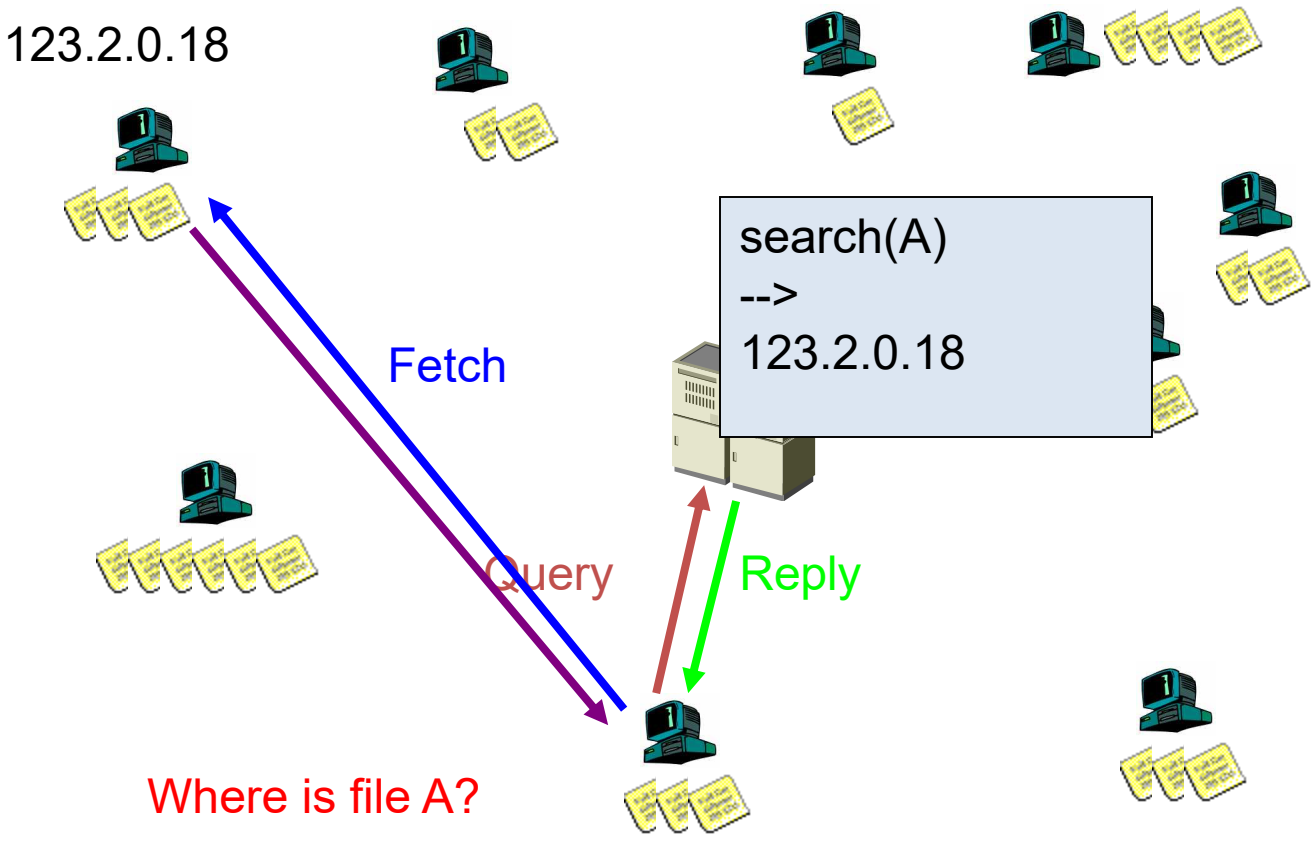
Централизирана база данни:

- **Join** – при стартиране клиентът се свързва с централен сървър
- **Publish** – съобщава списък от файлове на централния сървър
- **Search** – запитва сървъра, връща къде се съхранява заявения файл
- **Fetch** – изтегля файла директно от peer

Napster - Publish



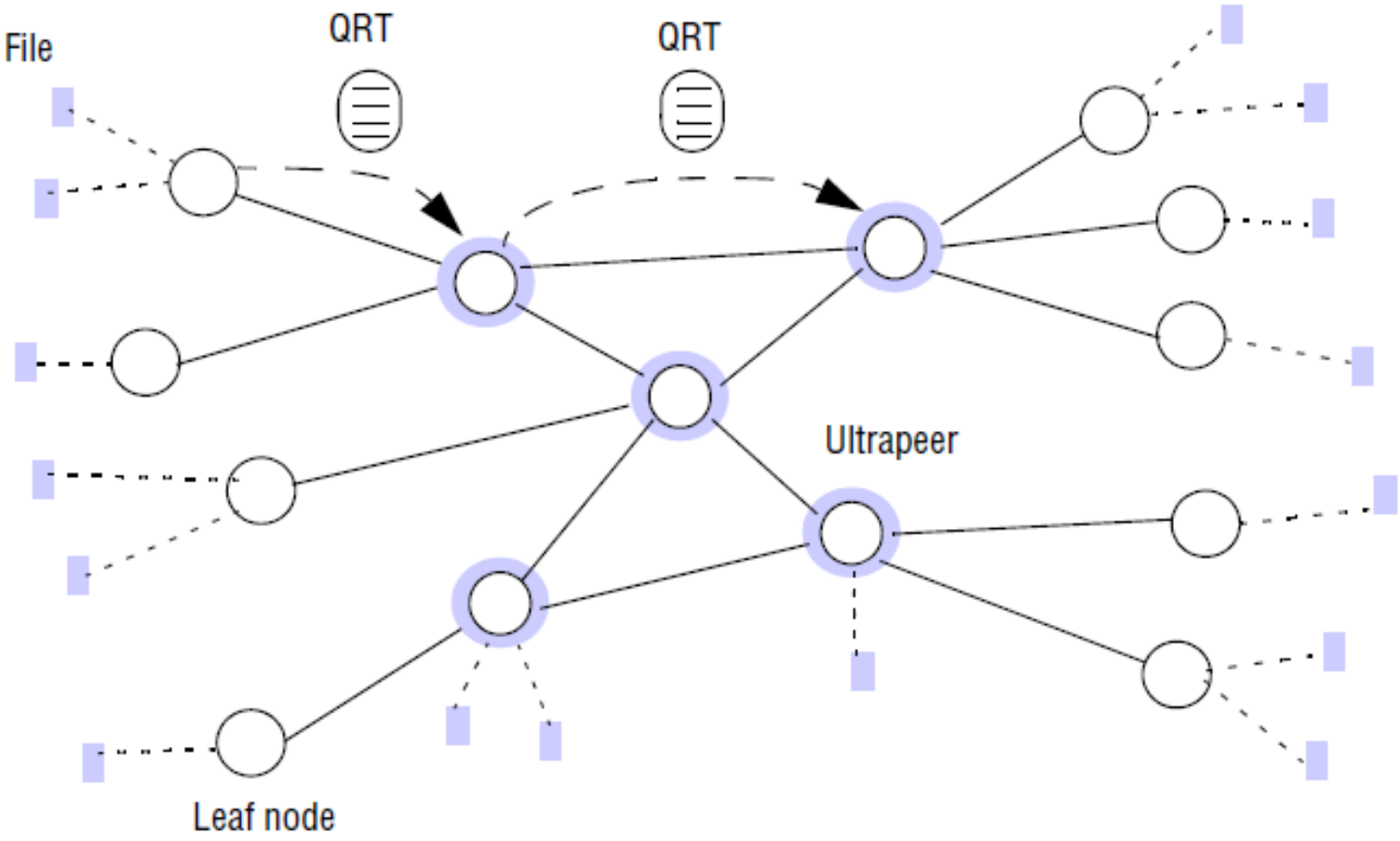
Napster - Search



Gnutella - особености

- Напълно децентрализирана система
- Наличие на възли с допълнителни ресурси - *ultrapeer*
- Формират ядрото на мрежата
- Останалите възли – *leaf nodes*
- *Leaf* възлите са свързани към малък брой *ultrapeers*.
- Редуциране на максималния брой хопове за пълно търсене
- Архитектурата – „хибридна“ и е адаптирана в Skype

Gnutella



Gnutella – Query Routing Protocol

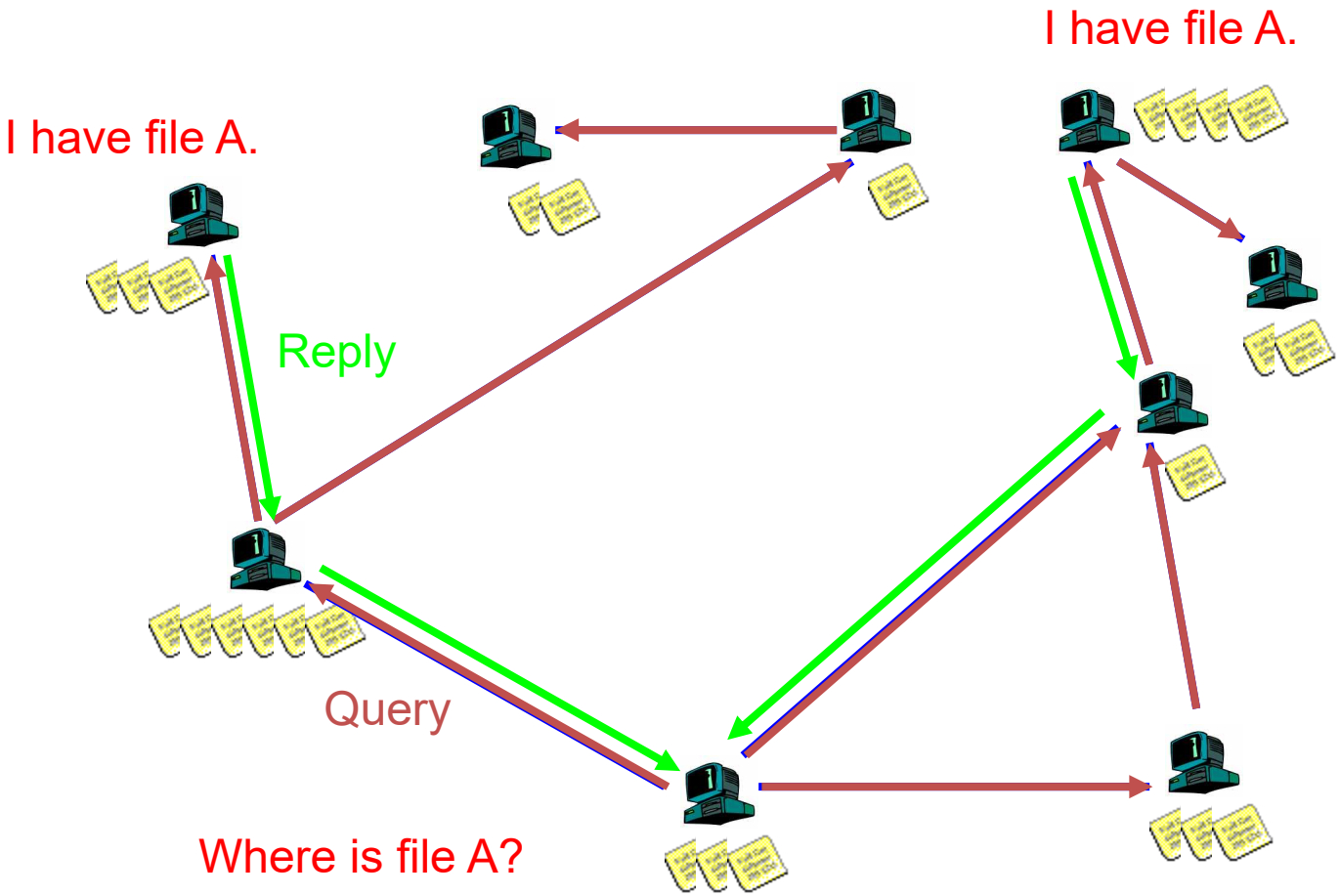
- Редуцира броя на запитвания от възлите
- Базира се на обмен на информация за файловете, съдържащи се във възлите.
- Вместо споделяне на информация за файловете директно, протоколът използва множество от числа като хеш на отделните думи на името на файла.
- Всеки възел поддържа Query Routing Table съдържаща тези хешове

Gnutella - операции

Множество запитвания:

- **Join** – при стартиране клиентът се свързва със свои съседни възли
- **Publish** – няма необходимост
- **Search** – запитва съседните възли, които питат своите и т.н. При намиране се връща отговор към запитващия възел
- **Fetch** – изтегля файла директно от peer

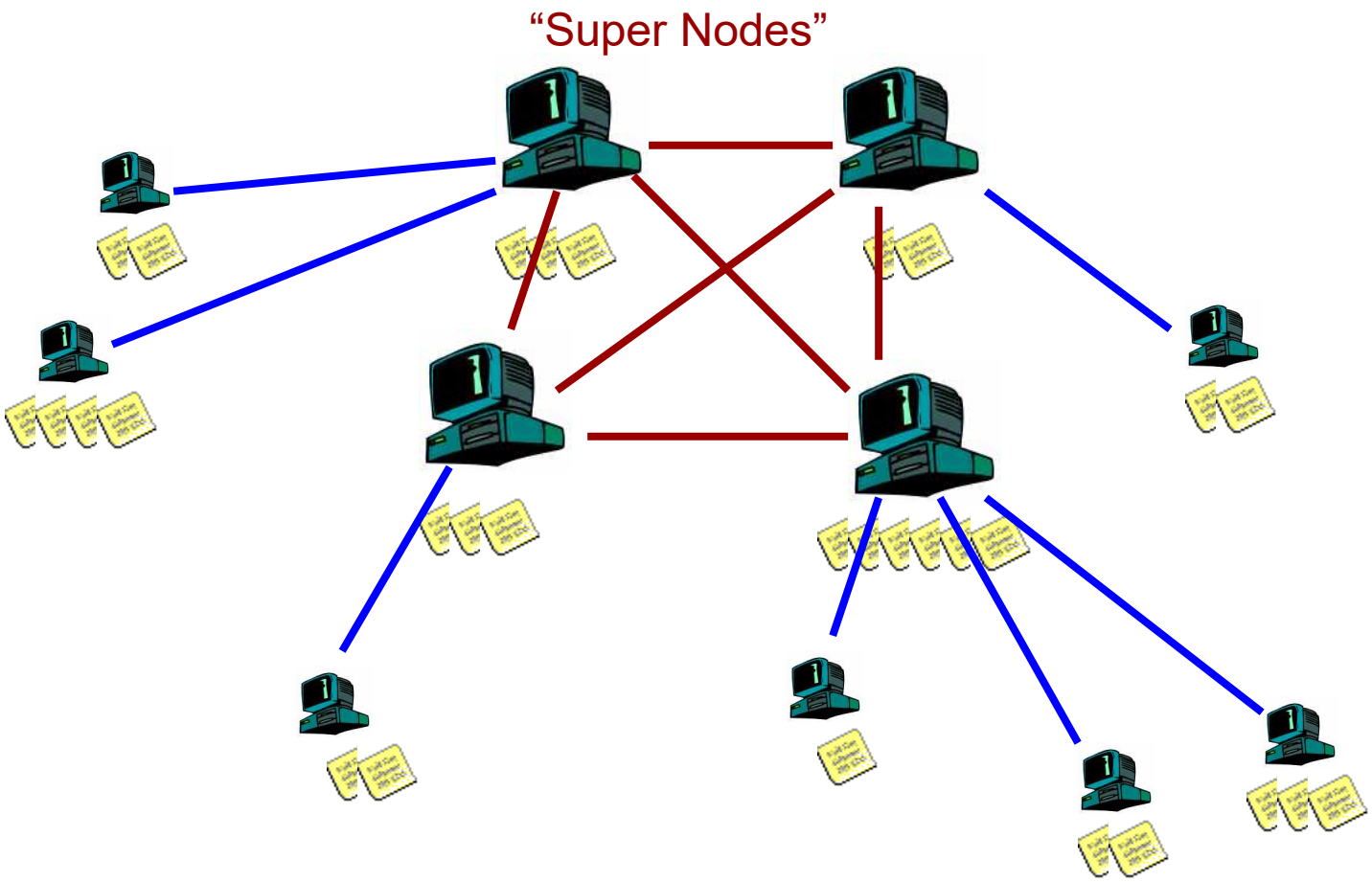
Gnutella - Search



KaZaA - особености

- Наличие на супервъзли
- Интелигентни запитвания
- Консолидиране на запитванията – много свързани възли могат да имат само няколко файла, а изпращането на заявки към подвъзли може да заеме повече bandwidth отколкото самият отговор

KaZaA

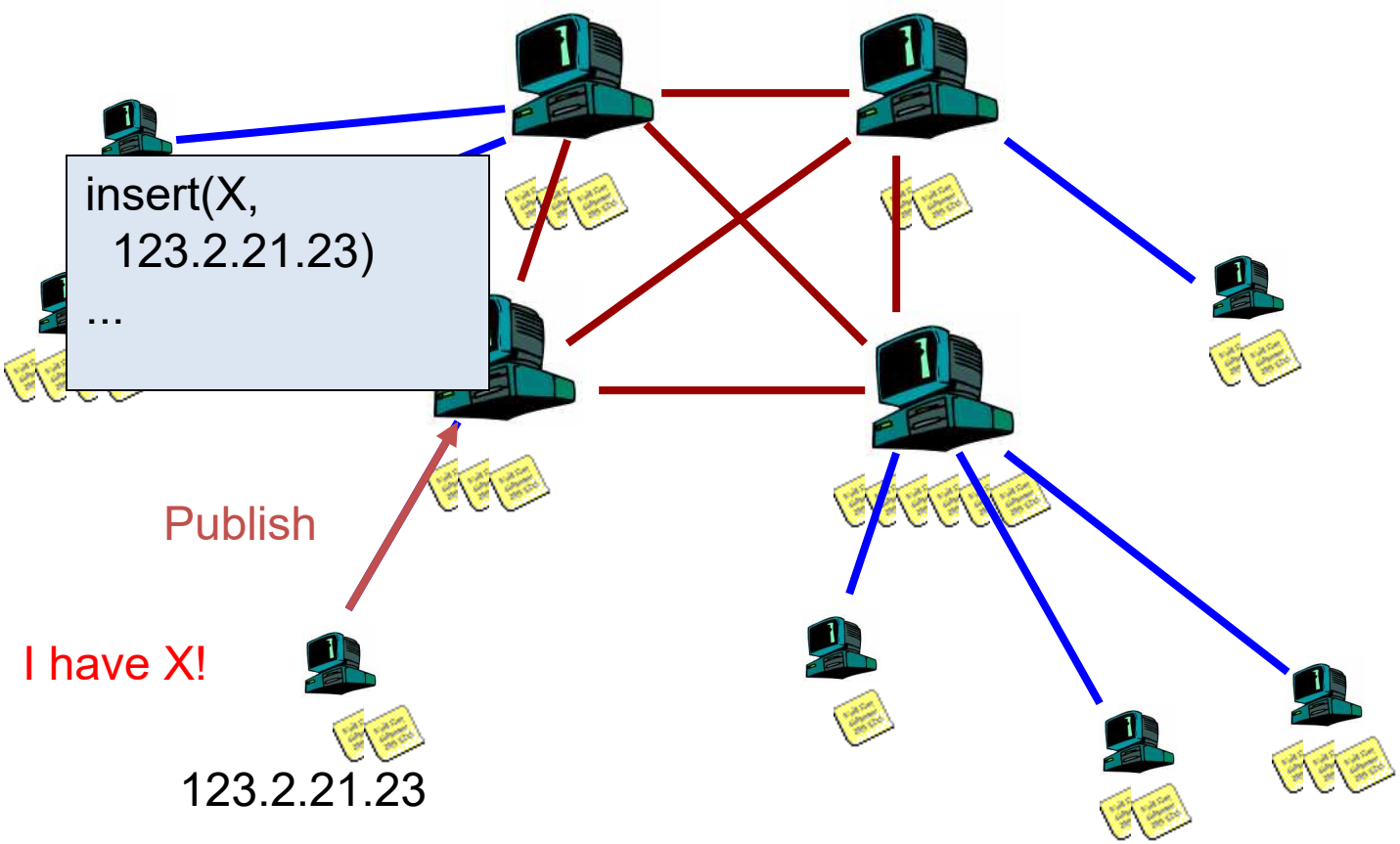


KaZaA - операции

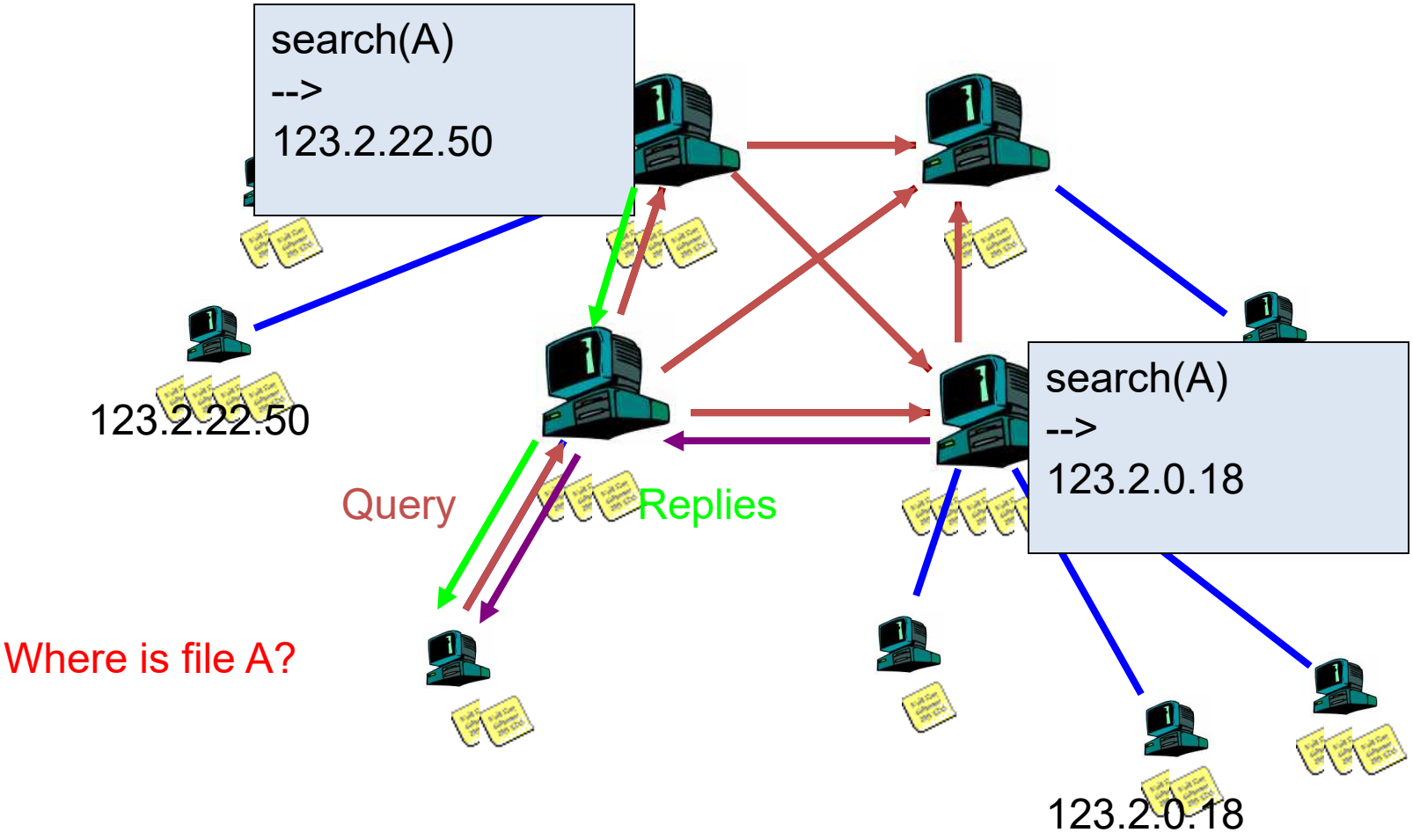
„Интелигентно“ изпращане на множество запитвания

- **Join** – при стартиране клиентът се свързва супервъзел
- **Publish** – изпраща списък с файлове към супервъзел
- **Search** – запитва директно супервъзел. Супервъзлите разпространяват заявката между тях си
- **Fetch** – изтегля файла директно от peer. Може едновременно да изтегля от множество peers

KaZaA – Publish



KaZaA – Search



BitTorent - особености

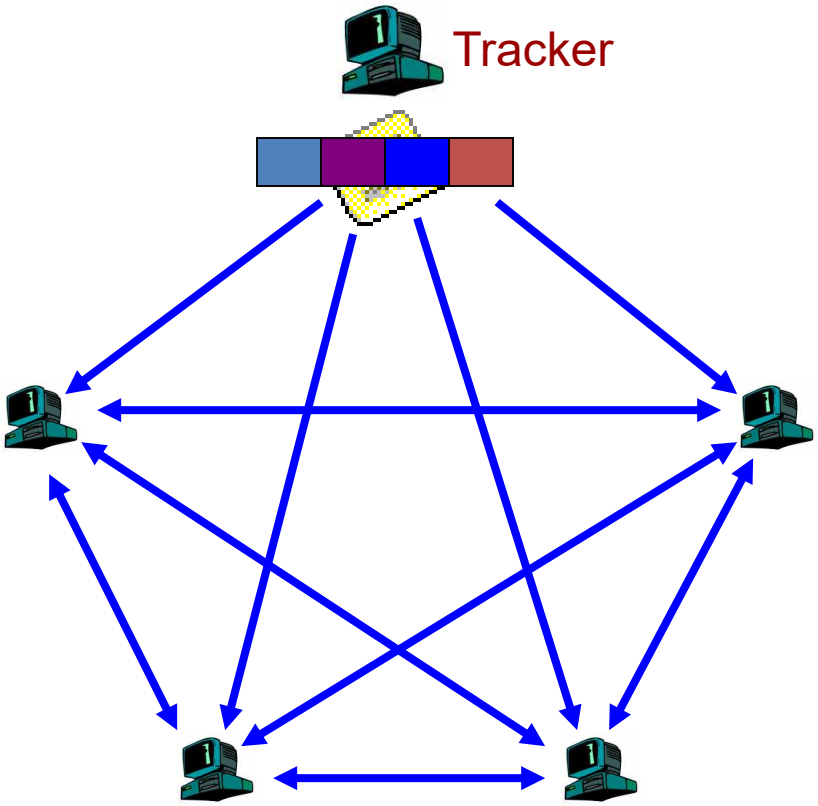
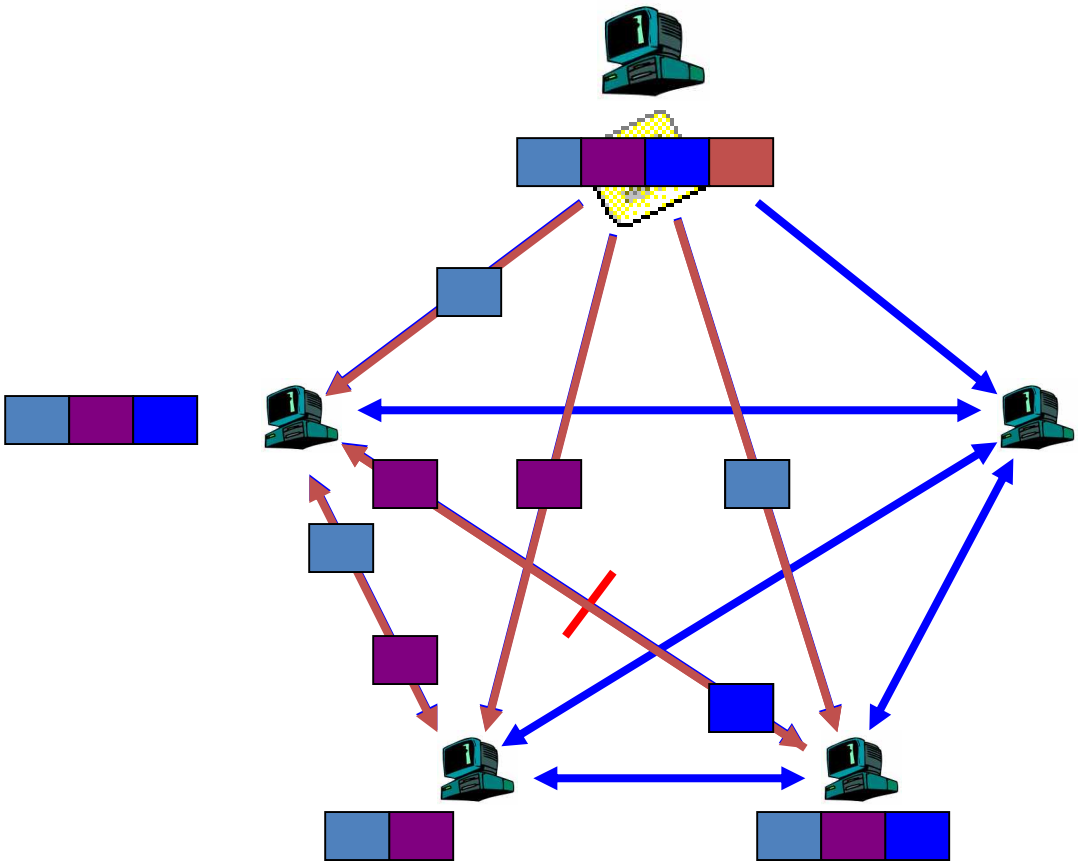
- Фокусиран на ефикасно изтегляне, не търсене
- Единичен публикуващ, множество изтеглящи
- Използва стратегия *tit-for-tat* – за да получиш файл, трябва и да дадеш (колкото повече споделяш, толкова по-бързо изтегляш)
- Различни части от файла могат да се теглят едновременно от множество възли

BitTorent - операции

Swarming:

- **Join** – при стартиране клиентът се свързва към tracker сървър и получава списък с peers (seed компютри)
- **Publish** – стартира tracker сървър
- **Search** – няма
- **Fetch** – изтегля части от файла директно от peers. Качва части от файл

BitTorent – Publish/Join



BitTorent – Fetch