MAAY :
A self-adaptive peer network for efficient document search

Joaquín KELLER, Frédéric DANG-NGOC, Daniel STERN
France Telecom Research & Development

Google, …

Centralized architecture: A server farm of ~10^5 machines

Good:

- Exhaustiveness (~40%-60%)
- The web structure is used in page relevance estimation (pageRank algorithm)

Bad:

- Centralized: bottleneck, point of failure
- Privacy issues: all requests are logged
- Update cycle is looong: 2-3 months
- Transient web servers are invisible
- Impersonal responses, relevance is generic

Gnutella, KaZaA, …

Peer networks: ~10^5 nodes for gnutella, 10^6 nodes for KaZaA (web: 10^8 "pages")

Good:

- Transient nodes contribute to the system
- Search is performed on up-to-date data

Bad:

- Only the most replicated/popular data is accessible
- High latency
- Referenced documents are unstructured

MAAY

Basic principles:

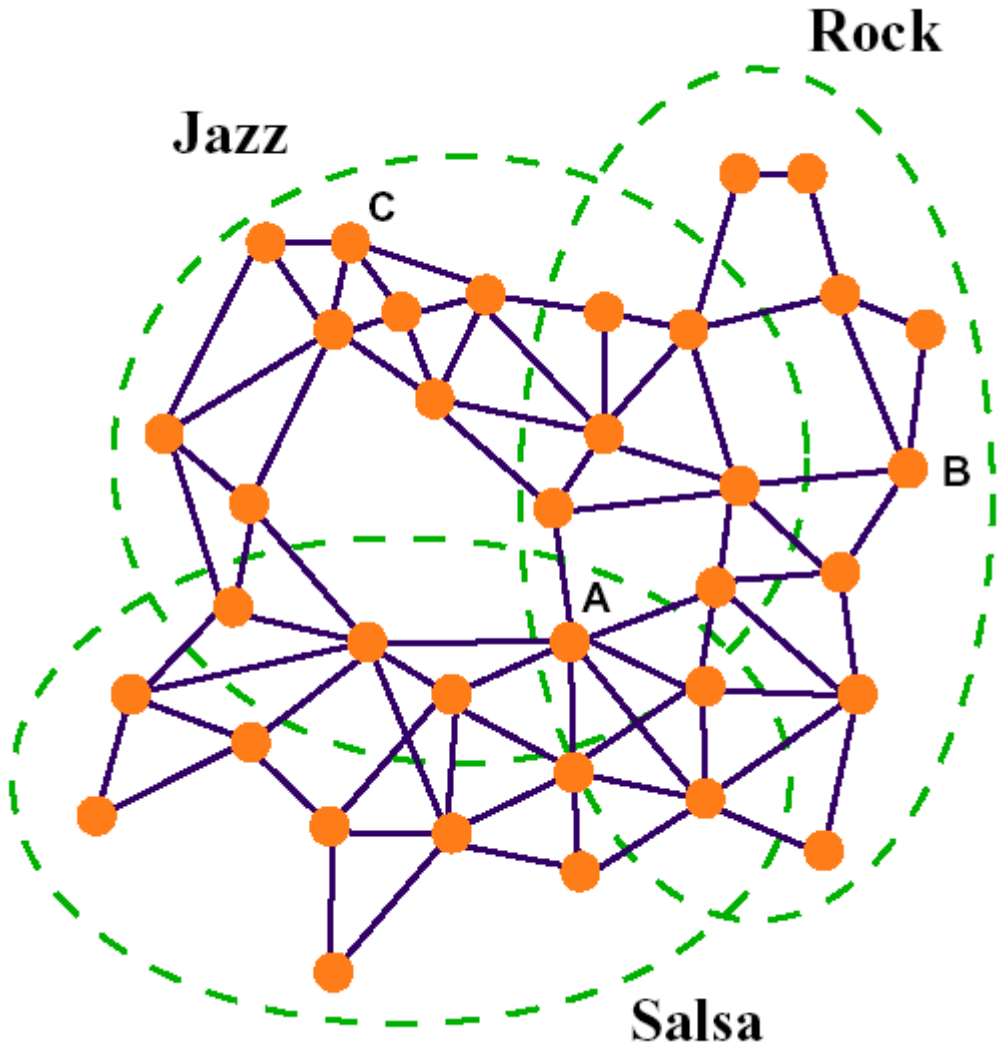
- Maay is a peer network
- Searches are keyword based (like in google or gnutella)
- **Nodes tends to cluster by *semantic proximity***
  - ⇨ So, for a given node, relevant/interesting content is nearby.

Google exhaustiveness is an overkill:

- Most keywords Google reference (~ 1 million) are useless
  - ⇨ a given user will use, in its whole life, only few thousand words
- A lot of responses are irrelevant
  - ⇨ the meaning of a word depends on who is talking (eg "football")

In Maay the goal is perception of exhaustiveness:

- ⇨ Latency for unlikely/random keywords is longer


Rock
Jazz
C
B
A
Salsa

Semantic proximity

The semantic position of a node is determined by:

- Outgoing requests
- Content made available to other nodes
- Downloaded documents by the user

User behavior + local indexing of stored content

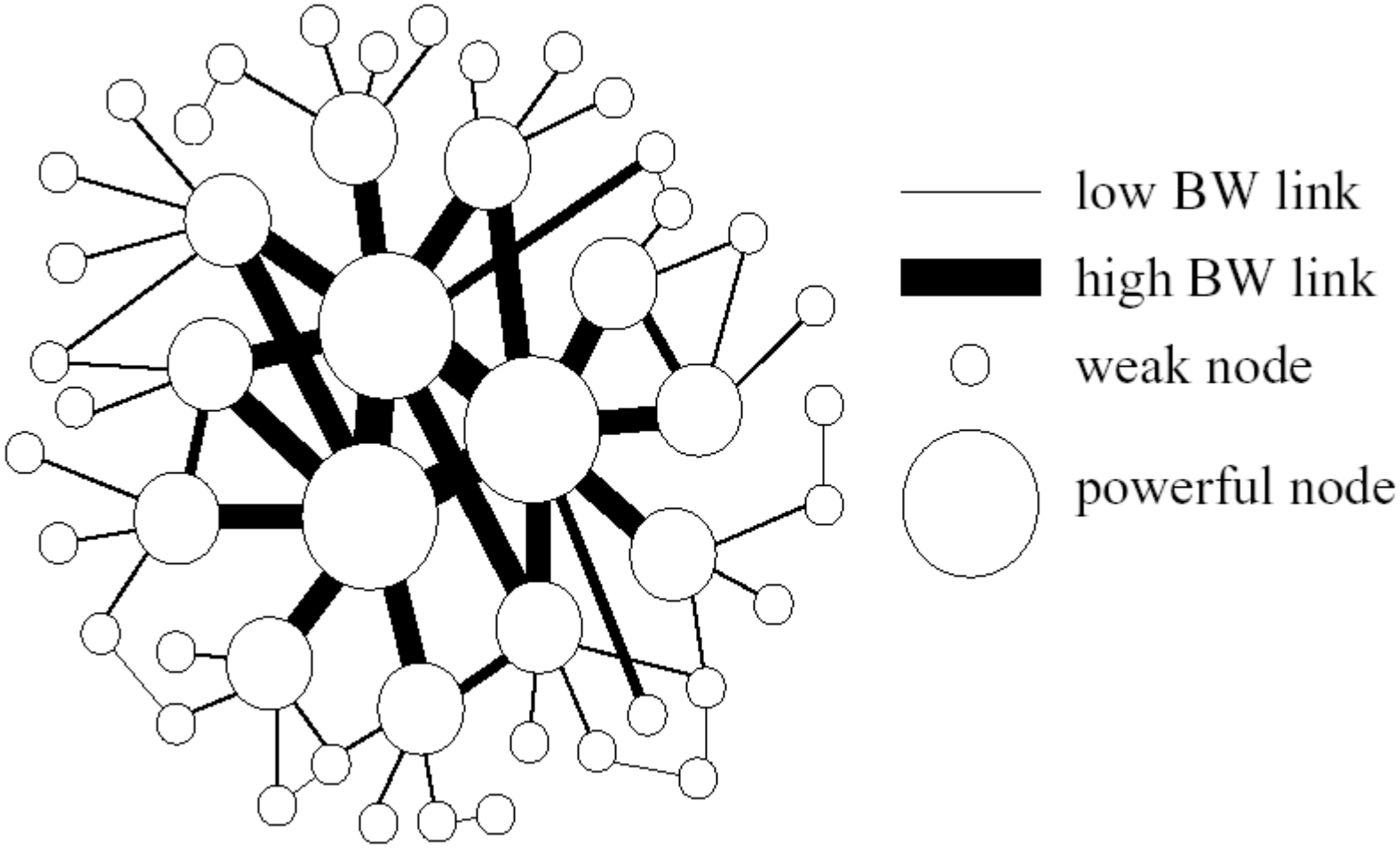
The semantic distance to other nodes is determined by:

- Requests and responses received (and possibly forwarded)
- Documents requested for download

Users' behavior + available content

Load balancing – topology optimisation

Preferential attachment to powerful (processing power) nodes:



Routing

Routing table:

A node keeps track of all nodes viewed once

- Keyword → List of (node, score for keyword)
- Node → Affinity, Power

Routing:

If the node does not match the keyword (no doc available)

The request is forwarded to the most suitable nodes

(some of these nodes should be alive)

If no node match the keyword

The request is forwarded using the nearest keyword

(Arbitrary paths are built)

Responses are routed back using the same route, so the node learns from other requests

MAAY deployment

Today:
Browsers keep track of all visited pages
Servers keep track of all visitors

Where is the content ? In browsers caches !!!

MAAY deployment:
Transform web servers + web browsers in MAAY nodes
Make use of the data already stored

Examples of such networks : Gnutella, KaZaA, Bittorrent, …

Conclusion and ongoing work

MAAY makes an intensive use of storage and bandwidth

but storage and bandwidth become cheaper faster than CPU

Storage : hard drives 1$/GB (in stores)

Bandwidth: ADSL2+ 25Mb/s, optic fiber 1 Tb/s (in labs)

More storage and bandwidth imply more online content:

Will Google + actual web be enough to access content ?

To do:

- Simulator
- A tiny implementation for intranets
- Model the MAAY network using small word and scale free networks

➔ Improve/refine algorithms