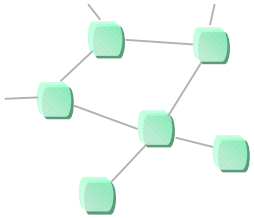
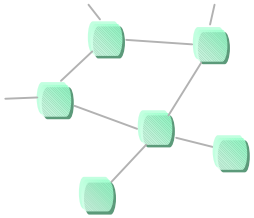


COLLABORATIVE WORK with GNUTELLA

by Valentin Mesaros (UCL)*

* Partial results of the cooperative work between
Valentin Mesaros, Bruno Carton and Brieuc Florent



GNUTELLA: CHARACTERISTICS

- Gnutella is a distributed system for file sharing
  - provide means for network discovery (viral diffusion)
  - provide means for file searching and sharing (network crosscut file sharing)
- Defines a network at the application level
  - hosts running `gnutella` protocol
  - it runs over `TCP/IP`
- Employs the concept of peer-to-peer
  - all hosts are equal (symmetry)
  - there is no central point
- Provides pseudo-anonymity
  - anonymous search, but reveal the `IP` addresses when downloading

GNUTELLA: PROTOCOL (I)

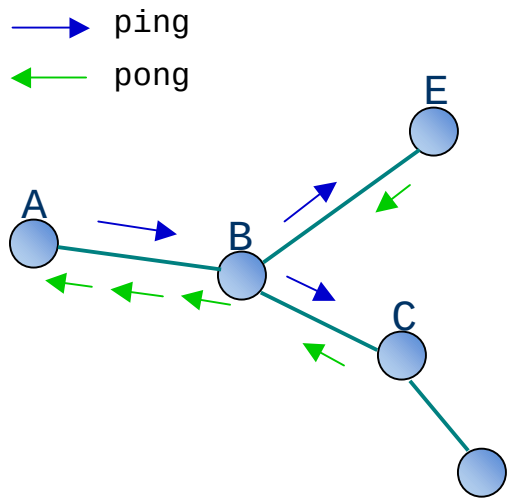


Fig 1. Network discovery

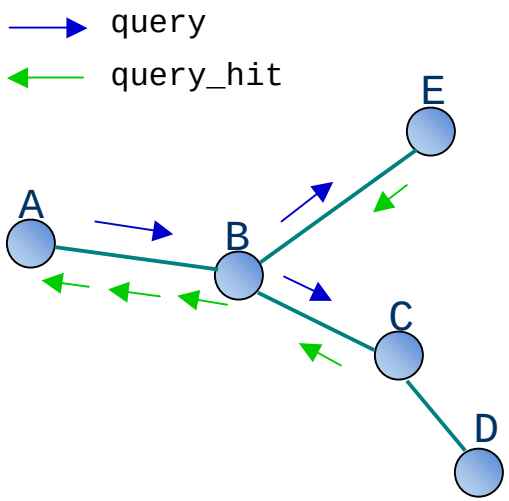


Fig 2. Network querying

- A discovers its *horizon* (e.g., TTL = 2)
  - send ping to its neighbors (broadcast)
  - ping msg is forwarded if TTL>0
- Receiving ping, B,C and E, respond pong
  - pong contains network info about its sender
  - B forwards pong msgs from E and C, to A

- A searches the network (e.g., TTL = 2)
  - send query to its neighbors (broadcast)
  - the query is forwarded if TTL > 0
- B,C and E, respond with query_hit
  - query_hit contains network info about where to download the file from
  - B forwards query_hit msgs from E and C, to A

GNUTELLA: PROTOCOL (II)

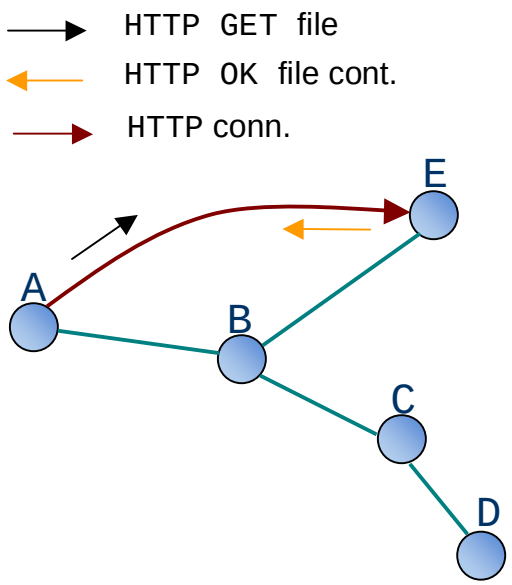


Fig 3. Download a file

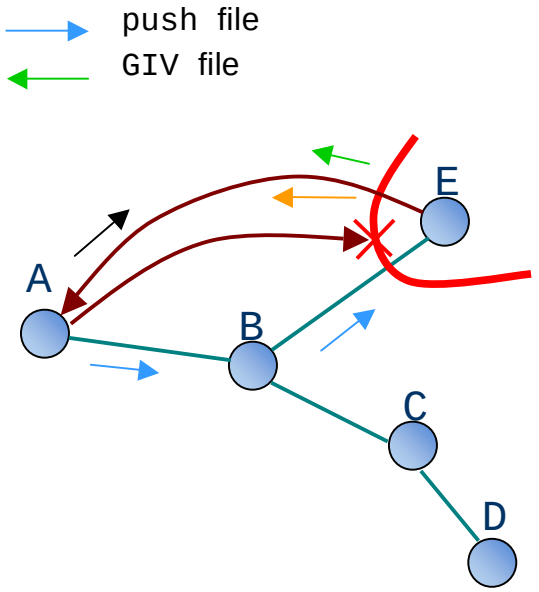
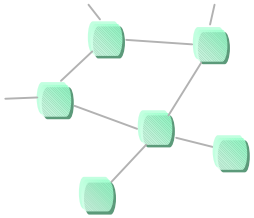


Fig 4. Download a firewalled file

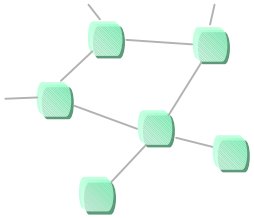
- A issues an HTTP query for a file found at E
  - initiate a TCP connection to E, for instance
  - send an HTTP GET file query to E
- E responds by sending the requested file
  - E acts as a web server sending the file content

- A fails initiating a TCP connection to E
- A instructs E to push the file
- E initiates a TCP connection to A
- E instructs A to issue the download (GIV msg)
- A issues an HTTP query for the file found at E
- E responds by sending the requested file



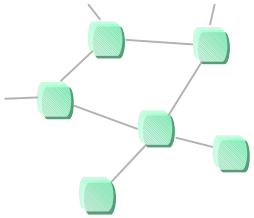
GNUTELLA: LIMITATIONS

1. Weak support for collaborative work
   - the only way to exchange information is point-to-point
2. Weak support for maintaining the network connectivity
   - through peer's *horizon*, or well-known *host caches*
3. Inefficient bandwidth usage for network discovery and querying
   - broadcast-based approach
4. Impossible to download files between two firewalled peers
5. No support for security
   - the shared information is not protected
   - risk of denial of service attacks



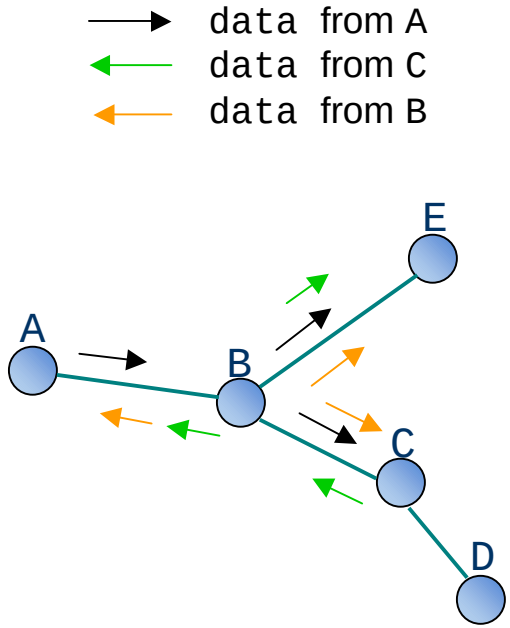
GNUTELLA: POSSIBLE EXTENSIONS

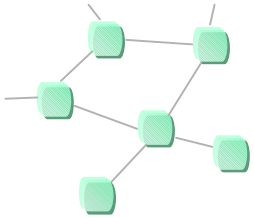
- 1. Extension for collaborative work
  - add a message for data transport (e.g., raw, `XML`, `Oz` strct.)
- 2. Maintain the network connectivity
  - when any of a peer's neighbors dies, try to connect to of the neighbors of the latter
- 3. Extension for network monitoring
  - employ a lease-based event model
  - tunable from the user level
- 4. Extension for file sharing between two firewalled peers
  - delegate the task to a third party



EXTENSION FOR COLLABORATIVE WORK

- Introduce `data` message
  - the msg payload is set at the upper level
  - e.g. of payload: raw data, `XML`, `Oz` structures
- Data diffusion
  - `data` can be sent via one or more peer interfaces
  - `data` is routed as `ping` msg is; based on `TTL`
- Implement ones own protocol at higher level
  - `data` msg can be used to specialize `gnutella`
- Resemblance with `IP` Multicast
  - data can be shared by every peer
  - consequence : somehow, have the peers grouped



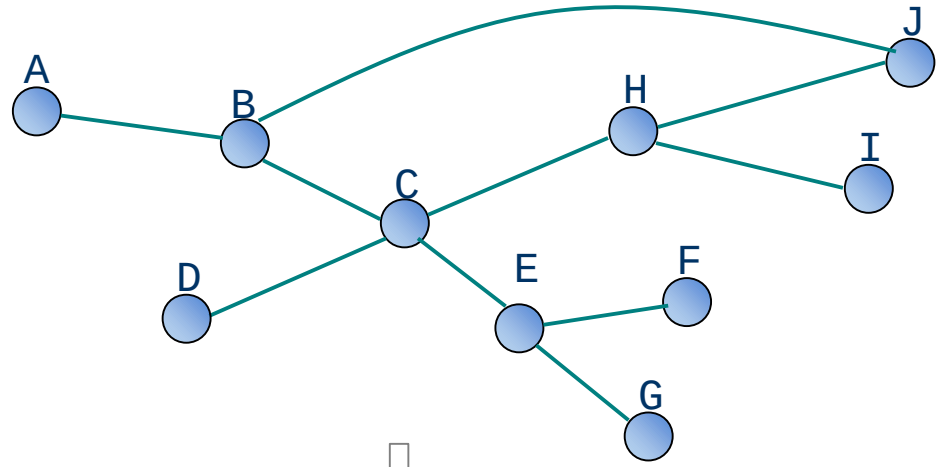


MAINTAIN GRAPH CONNECTIVITY

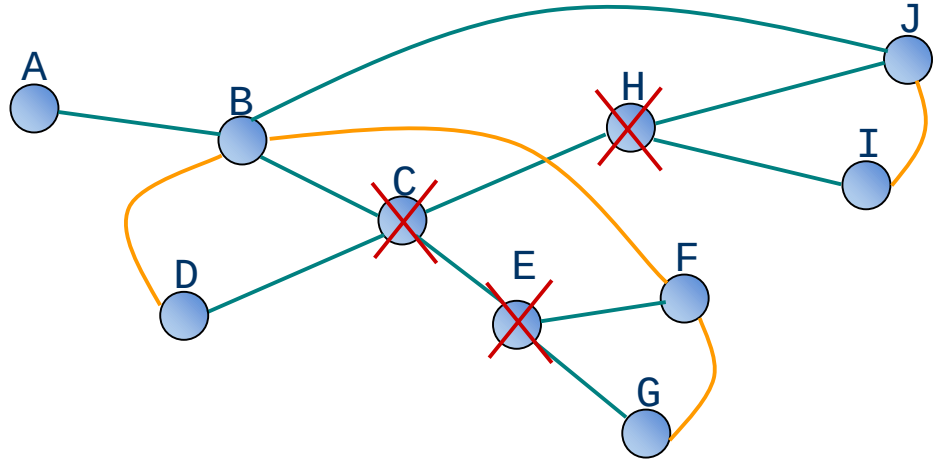
- Introduce `cping` and `cpong` messages
  - `cping` is the same as `ping`, but it triggers `cpong`
  - `cpong` contains info about the connectivity of its sender (i.e., who it is connected to)
  - a peer can know more about the topology of its horizon
- When its neighbor fails, it must do the followings:
  1. try to connect to all of the neighbors of the dead peer
  2. if all of the neighbors of the dead peer are dead, repeat step 1 for their neighbors
  3. when a connection succeeds, check whether the remaining neighbors need to be connected to

MAINTAIN GRAPH CONNECTIVITY

before
peer failures

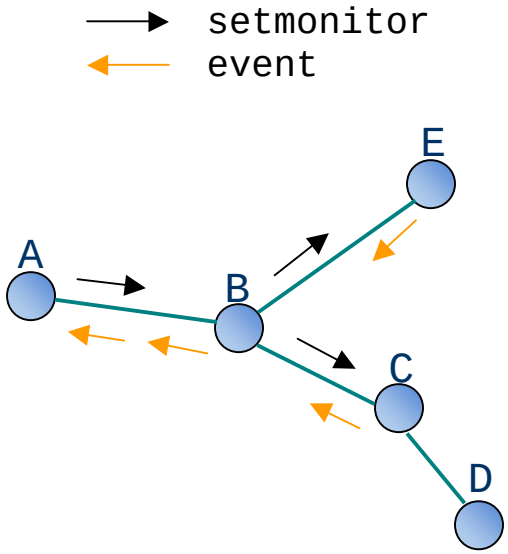


after
peer failures



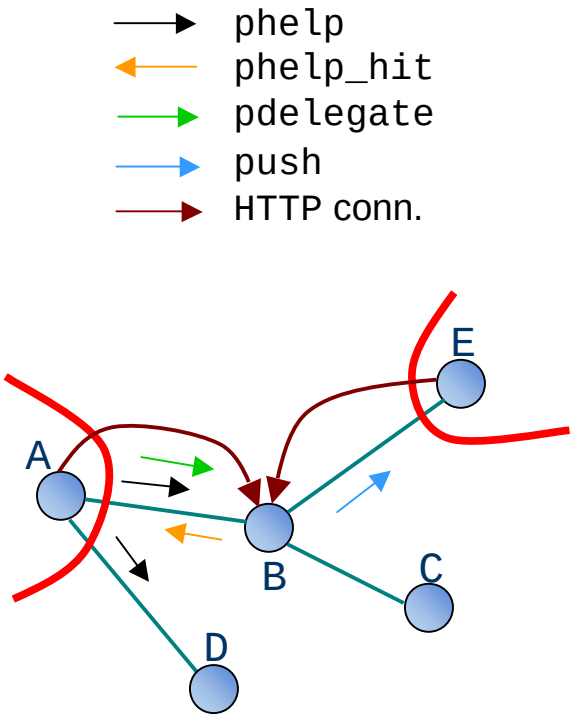
EXTENSION FOR NETWORK MONITORING

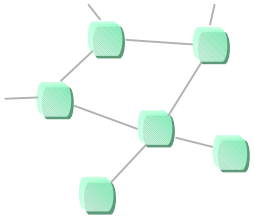
- Introduce `setmonitor` message
  - `setmonitor` registers for one or more events
  - indicate a `lease` for the requested events
- Introduce `unsetmonitor` message
  - `unsetmonitor` unregisters for one or more events
- The events are requested for a certain horizon
- Introduce `event` message
  - `event` issued whenever the requested event occurs
  - `event` msg is routed to the requester
  - event triggering is controllable from the user level



EXTENSION FOR FIREWALLED PEERS

- Both, A and E, are beyond firewalls
- A intends to download a file from E :
  - A asks for third party help (i.e., `phelp`)
  - B responds positively (i.e., `phelp_hit`)
  - A asks B to do the job (i.e., `pdelegate`)
  - B does the downloading from E as it were the requester (i.e., `push`)
- A gets the respective file from B
  - A connects to B and downloads the file using HTTP





Gnutella vs. Global Store

1. Centralization

   GS : internal coordinator for performance (no single point of failure)

   Gnutella : completely decentralized

2. Transactions

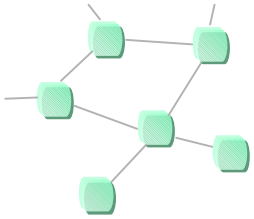
   GS : the shared data may be changed only within transactions

   Gnutella : permit simultaneous inconsistent views

3. Network scalability

   GS : each node is a replication point

   Gnutella : handle very large number of nodes



APPLICATION: PostIt

- PostIt is a collaborative application:
  - common forum for message exchange
  - fully replicated
- PostIt implemented over GS (Fig 1.)
  - rapid reaction to failures
  - consistency assured by GS
- PostIt over gnutella-extended (Fig 2.)
  - e.g., make use of the data message
  - the connectivity of the network is eventually maintained
  - consistency is implemented at an upper level (i.e., specialized protocol layer)

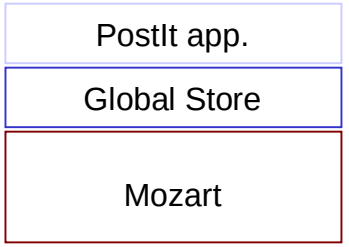


Fig 1. app. layering over GS

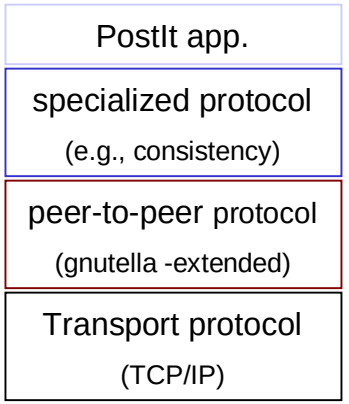
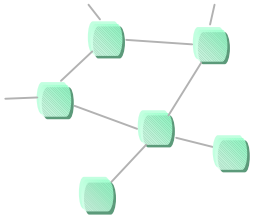
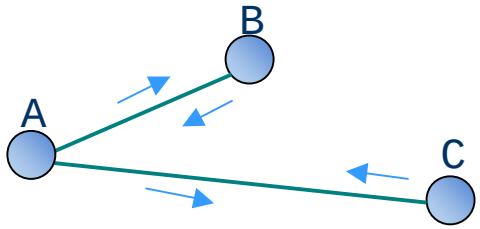


Fig 2. app. layering over gnutella

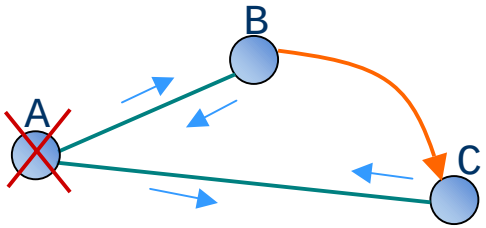


DEMO: PostIt over gnutella

1. have a number of peers running PostIt app.



2. after peer failures, the graph remains connected keeping the app. consistency



3. the remaining nodes continue communicating