ANTA 2003

The Second International Workshop on Active Network Technologies and Applications

Active Web : active networking support for web transport

Laurent Lefèvre, Jean-Patrick Gelas

INRIA RESO Team – LIP Laboratory – Lyon, France

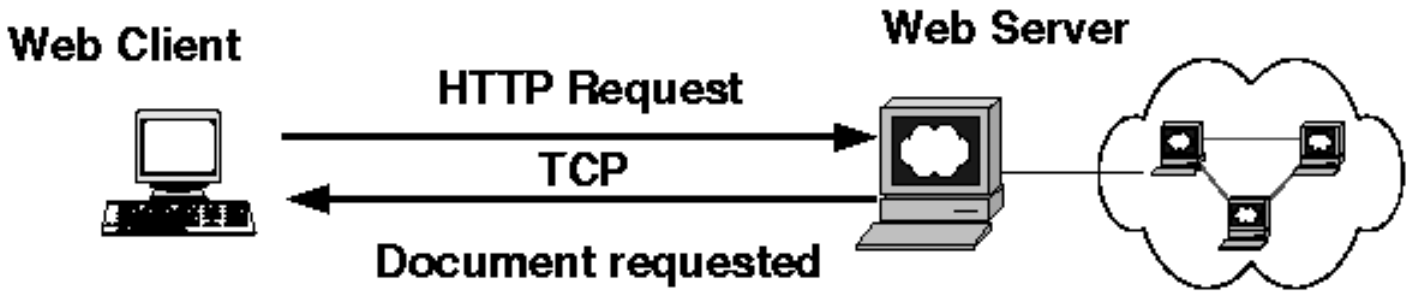
laurent.lefevre@inria.fr jpgelas@ens-lyon.fr

Background

- No killer application for AN !!
- Internet killer application :
  - Web applications
- What can active networks do for web applications and web transport ?

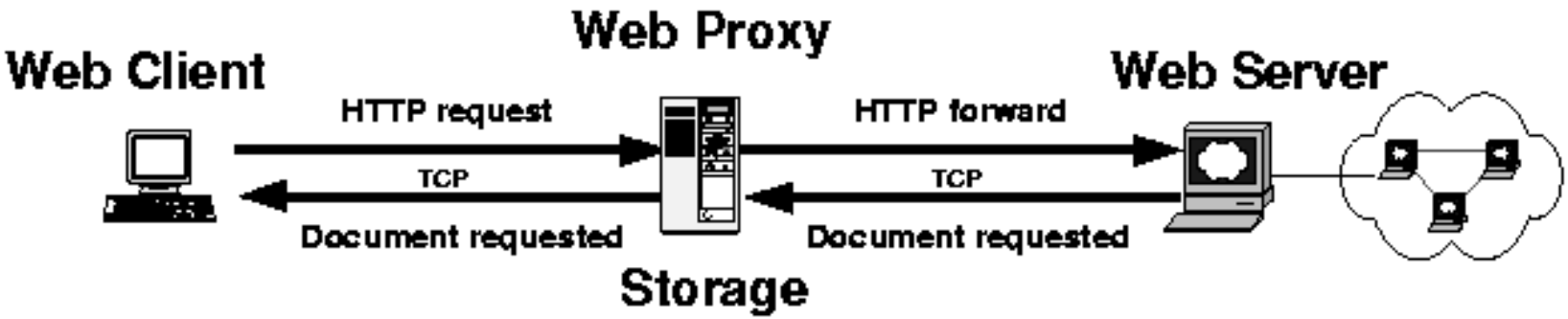
Primitive surfing

- Each requested document is provided by distant web server



Proxy web surfing

- Accessing distant files through Web proxy
- HTTP requests redirections
- Data caching “on the fly”



Improvements

- HTTP on UDP
  - Require to modify browsers,servers…
- How to « sometimes » benefit of UDP transport without modifying applications?

Active Web model requirements

- Supporting web applications and web transport
- Must support various transport protocols
  - Support TCP and UDP streams and new protocols
- Complete transparent solutions without modifying web equipments and applications (web browser, proxy, web server)
  - Avoid capsules, ANEP packets…
  - Use classical IP packets coming on specified ports

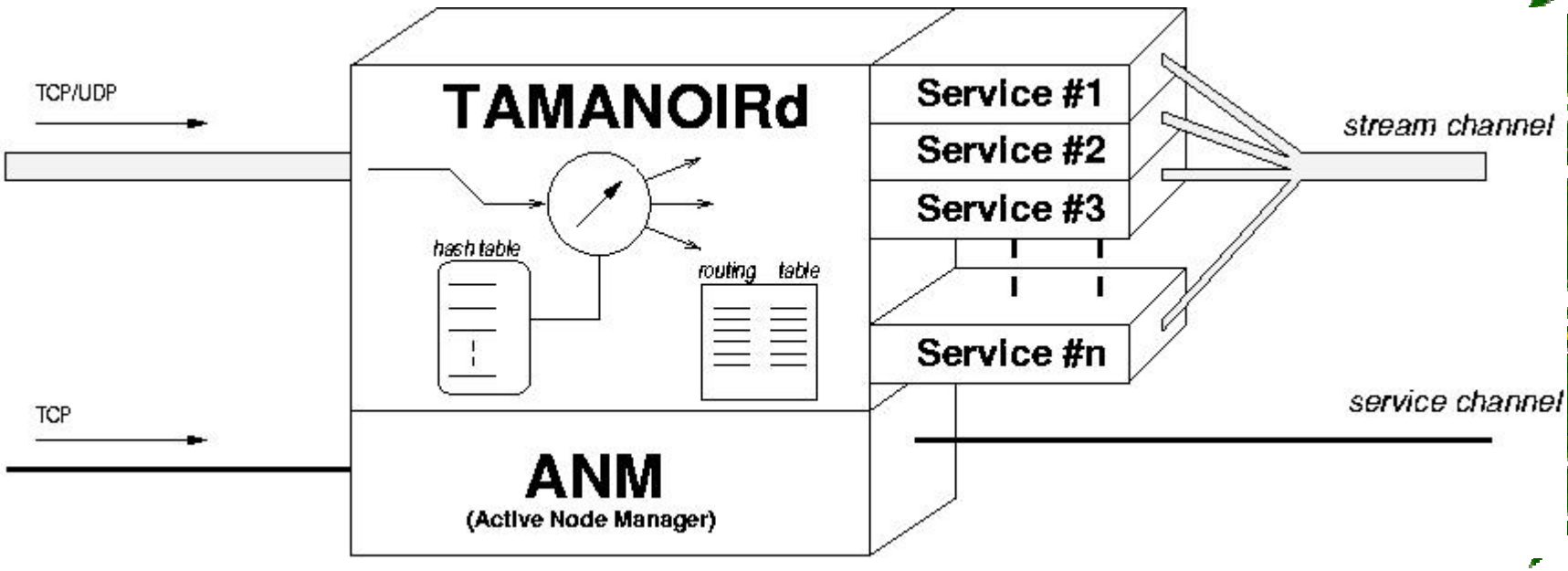
Active Web model requirements

- Easily tuneable by users to dynamically deploy personalized services
  - Active nodes must react to http request
  - AN keep state for each users
- Dynamically improvement
  - New services dynamically deployed
  - Services modified and tuneable on the fly
- Supported by existing Active nodes
  - Based on Tamanoir Execution Environment

Tamanoir Project

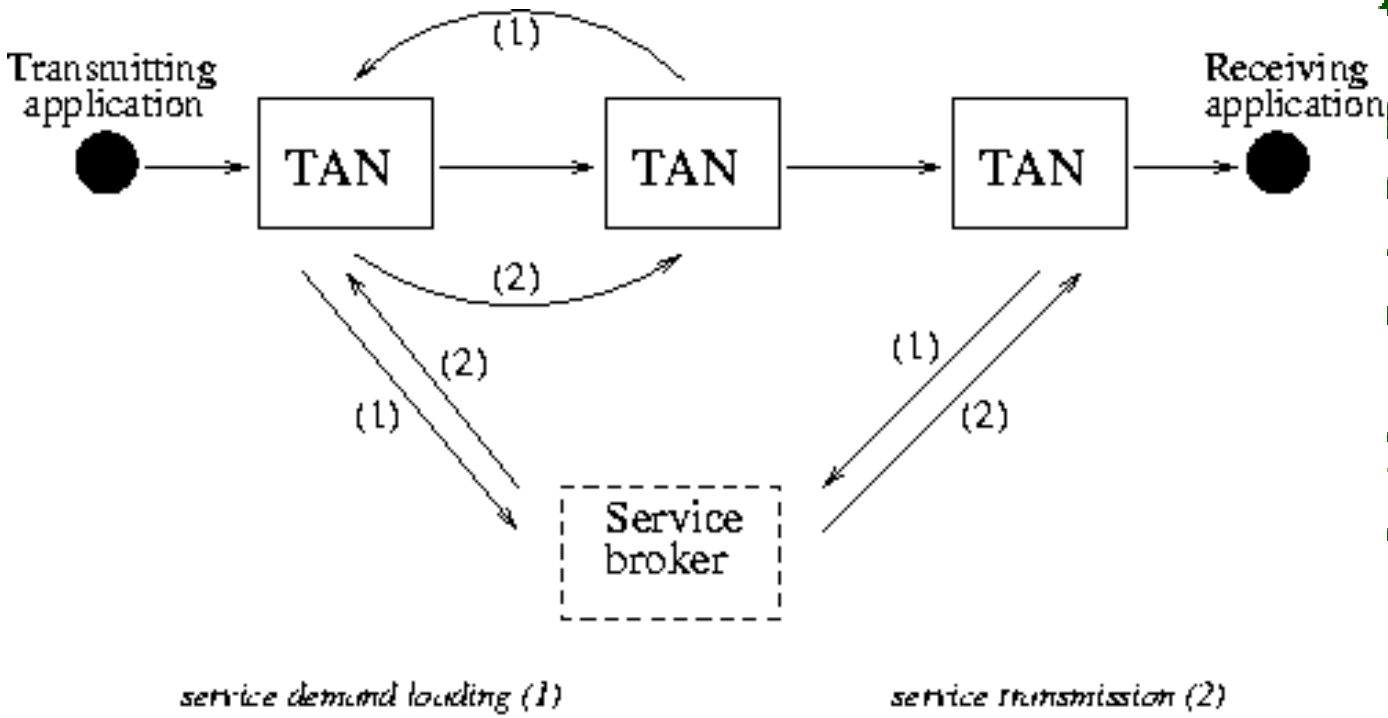
- Long term running project since 1999
- Execution Environment based on Java
- Distributed multi-threaded architecture
- Dynamic deployment of services
- Streams : UDP and TCP
- Supports services in kernel and on distributed resources *(see next talk !)*

Tamanoir Active Node (TAN)
Execution environment

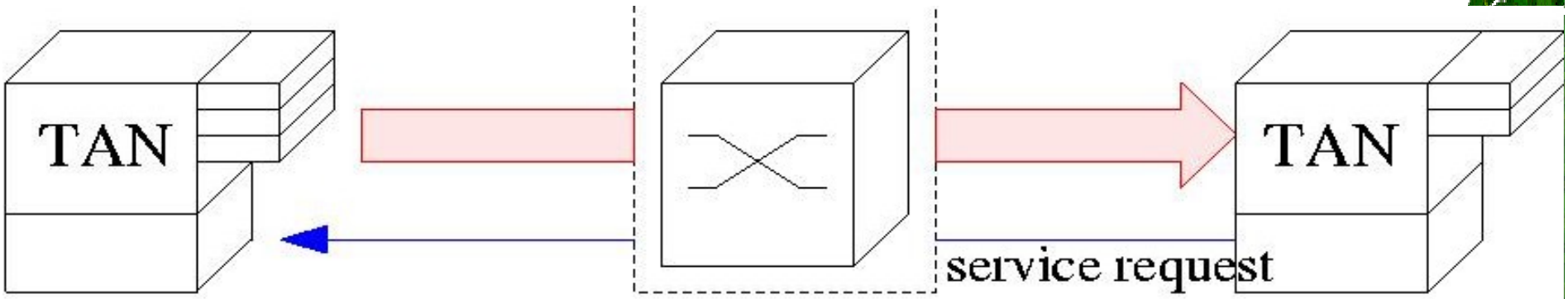


Active service deployment

- From application / middleware
- From TAN
- From a Service Broker

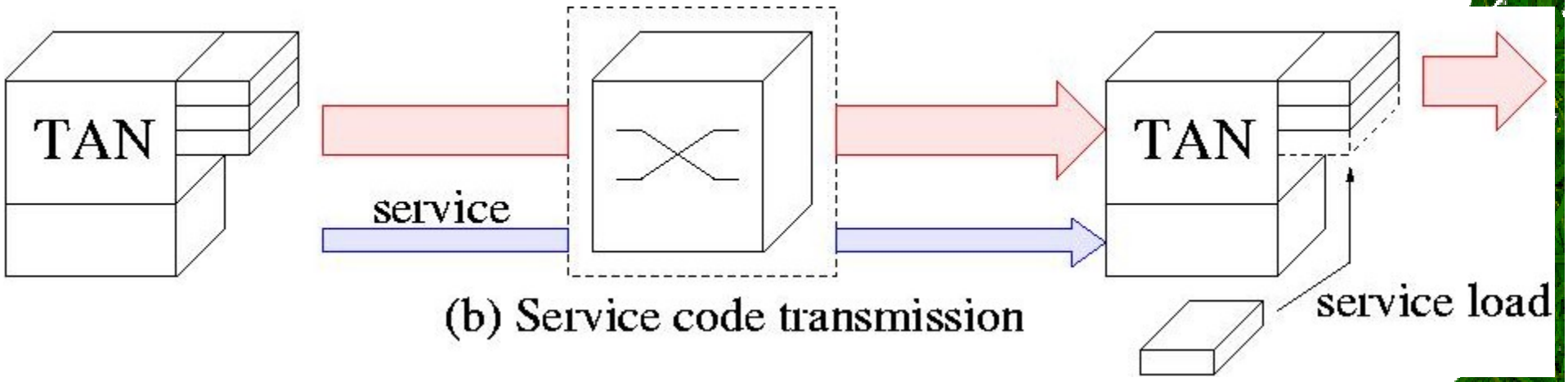


Service deployment problem



(a) Request of new service

Service deployment problem



(b) Service code transmission

Problem : What about data packets while the service is deployed ?

Active service deployment

- Service deployment in active nodes
  - UDP : loss of packets
  - TCP : slow down

⇒ On the fly storage of packets during service install (IBP – LOCI / UTK)

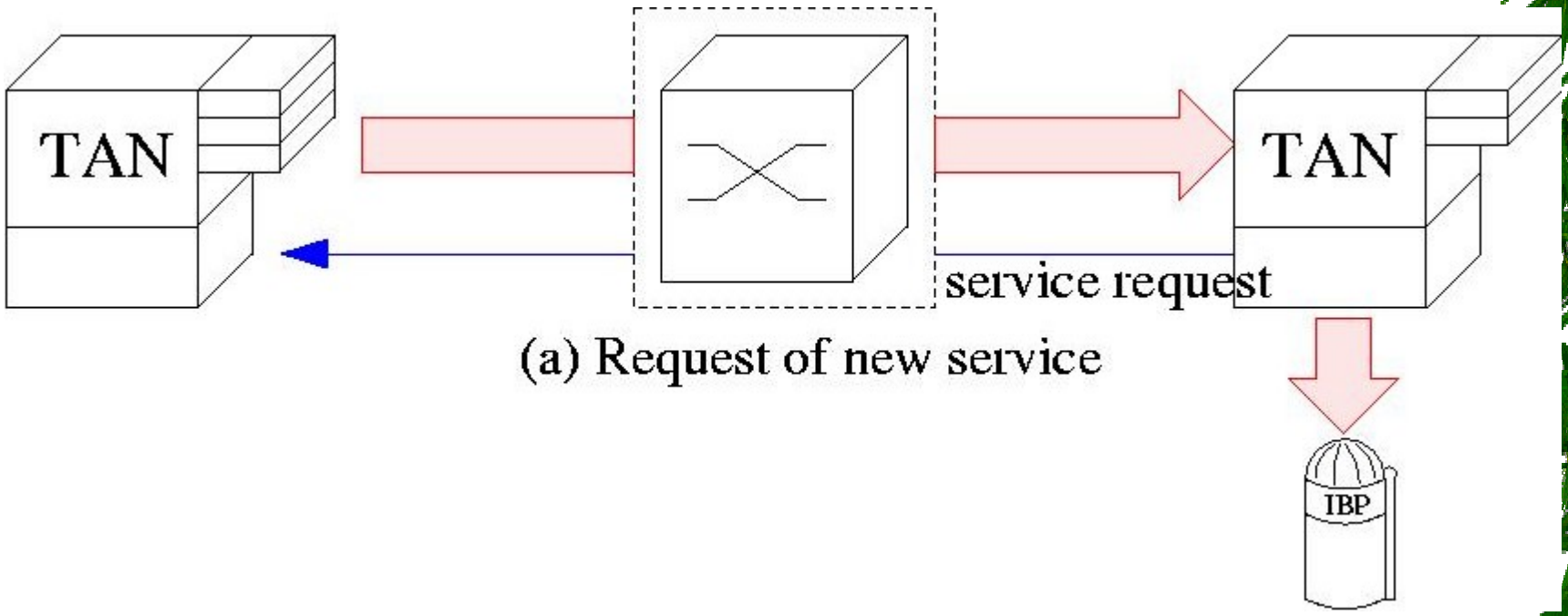
  ⇒http://www.loci.cs.utk.edu

Internet Backplane Protocol

- Proposed by LOCI Lab (UTK)
- Middleware for managing and using remote storage
- best-effort storage service, (the allocation semantic is weaker than typical storage services, such as files)
- IBP Allocations are time-limited
- Storage resources available for direct public access at network intermediate nodes.
- Allocation and scheduling of storage resources are exposed to the network.

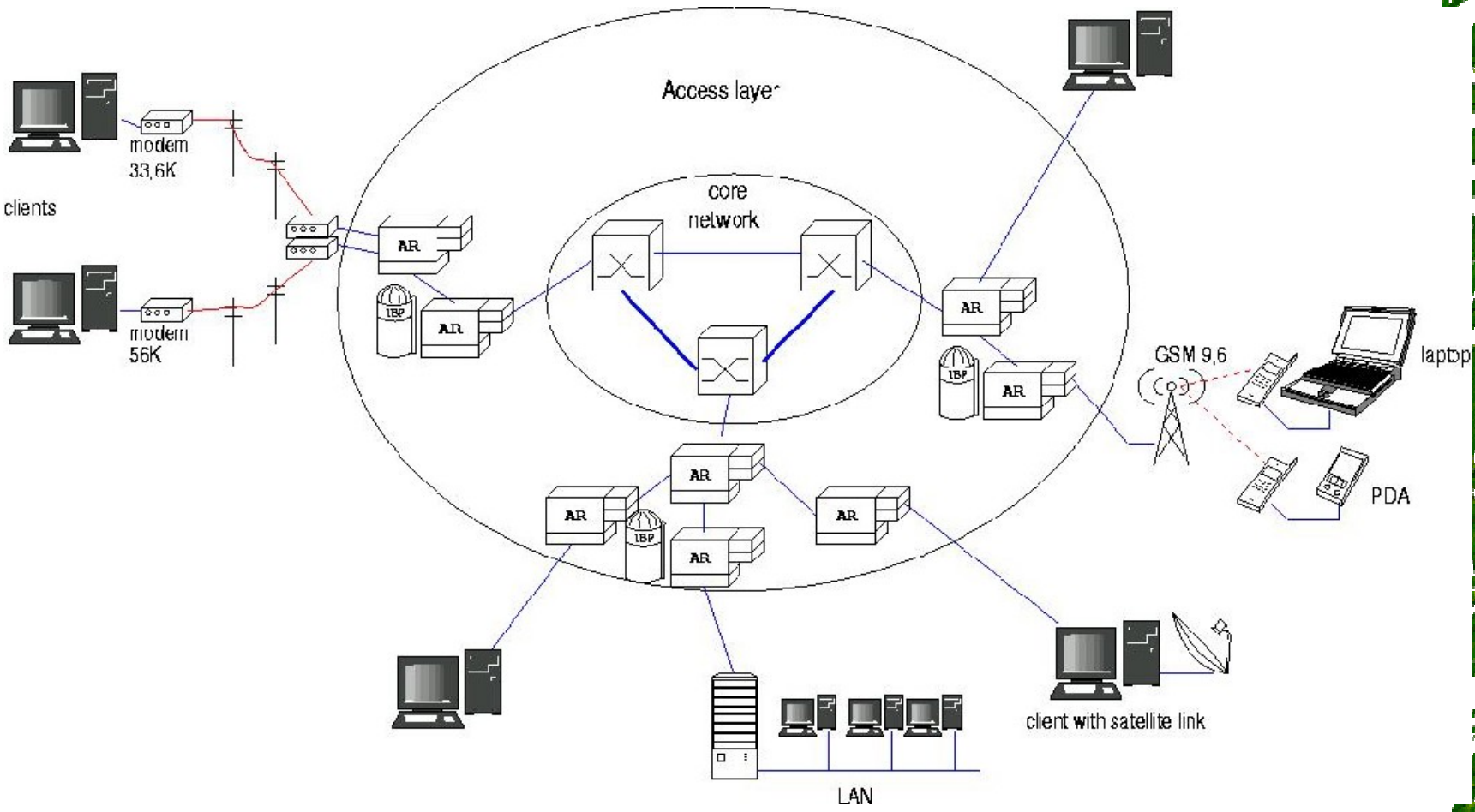
IBP logistical support for TAN -Service Deployment -

- Data caching
  - IBPService provides caching for data reaching a TAN while the appropriate service has to be installed



(a) Request of new service

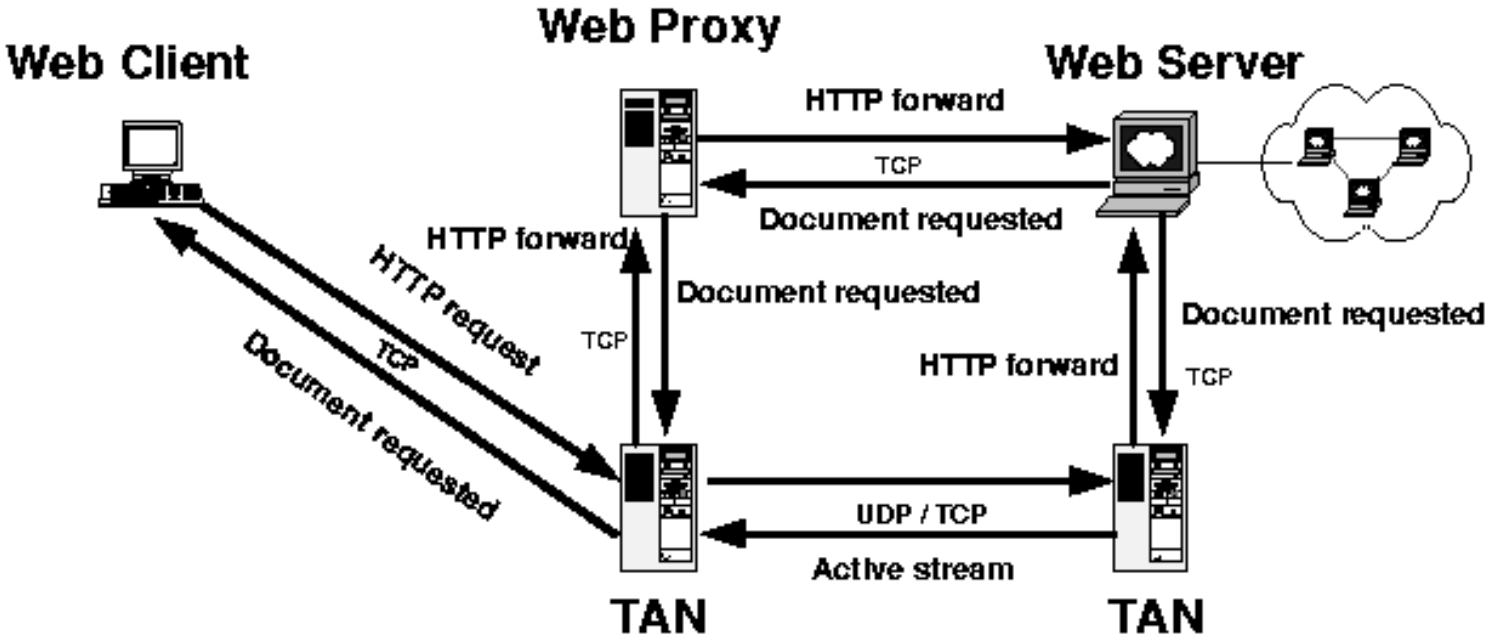
Active nodes locations



- Around backbones in access layer and edge networks
- Routers / Gateways, proxies

Active Web model

ActiveWeb architecture

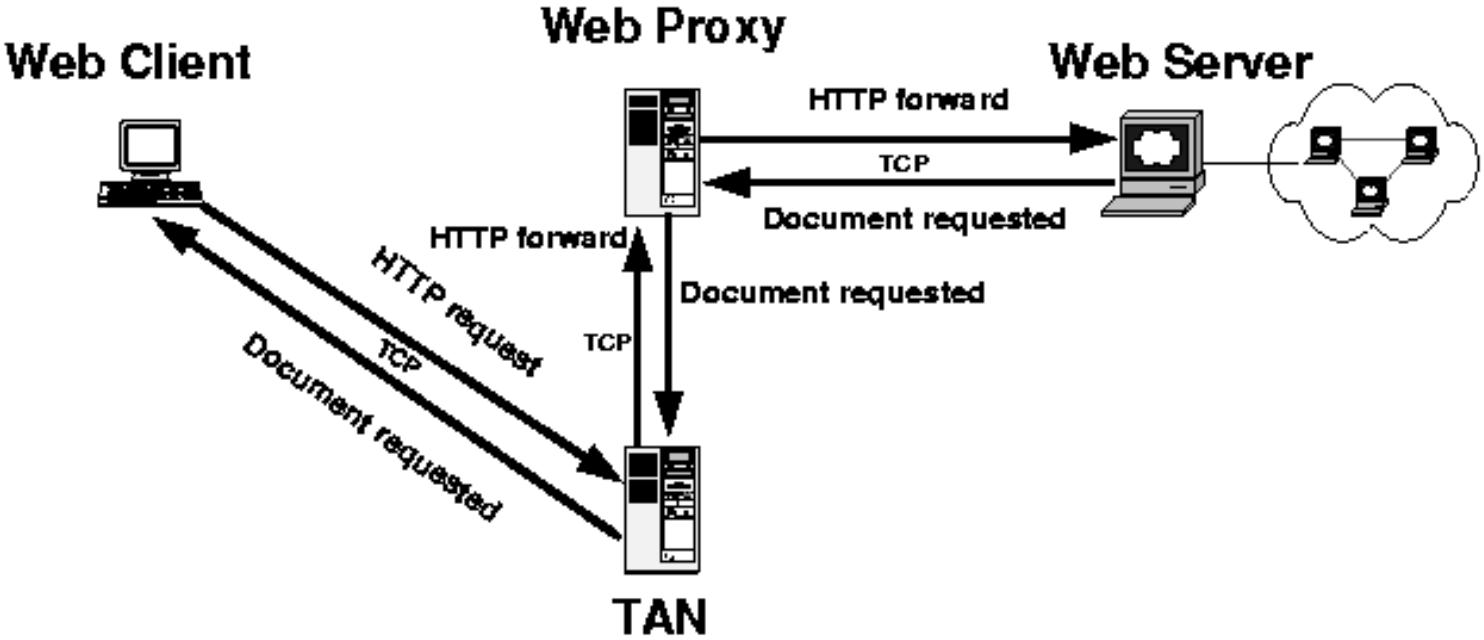


- Tamanoir active nodes (TAN) supports Web client, Web proxy and Web Server
- Dynamic adaptation services are deployed inside TAN and tuned by users through web browsers

Active Web service tuning

- Active Web services based on services blocks composition :
  - Forwarding blocks : TCP2TCP, UDP2DUP
  - Transport adaptation blocks : UDP2TCP, TCP2UDP
  - On the fly analysis of html documents to detect needed files
  - Content adaptation blocks : compression on the fly, stream adaptation, alert messages…
- Supports 6 Classes of Web Surfing (CoWS)

CoWS 1

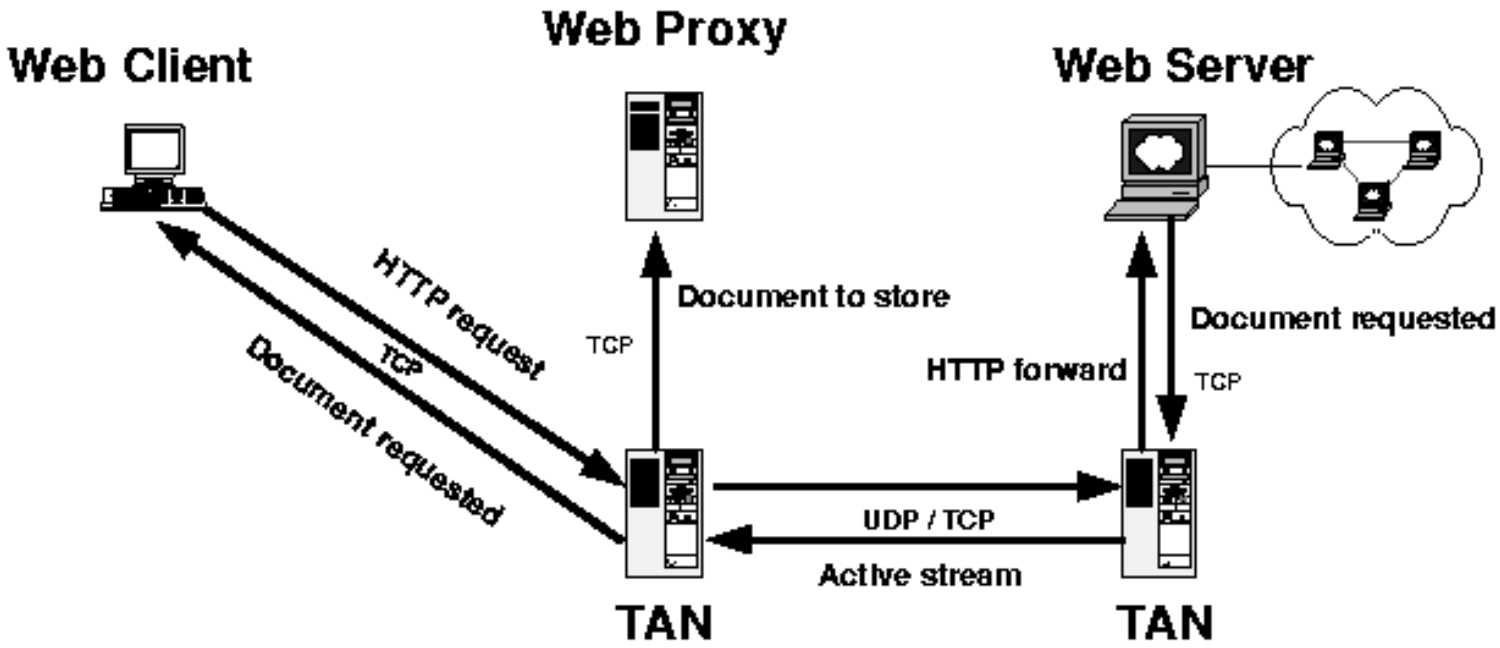


- All streams remain in TCP and cross active node (this allow transparent access to streams through active services and possibility of modifying stream contents on the fly for content adaptation)
- Deployment of TCP2TCP (forwarding) block services combined with adaptation and data content enhancement services
- Requires only one active node

Service blocks

- **TCP2TCP** : active forwarding service
  - receiving of an HTTP request like a Web Proxy;
  - forwarding HTTP request to local Web Proxy;
  - receiving TCP streams from local proxy;
  - optionally modifying data content for adaptation;
  - forwarding TCP streams to user browser.

CoWS 2,3

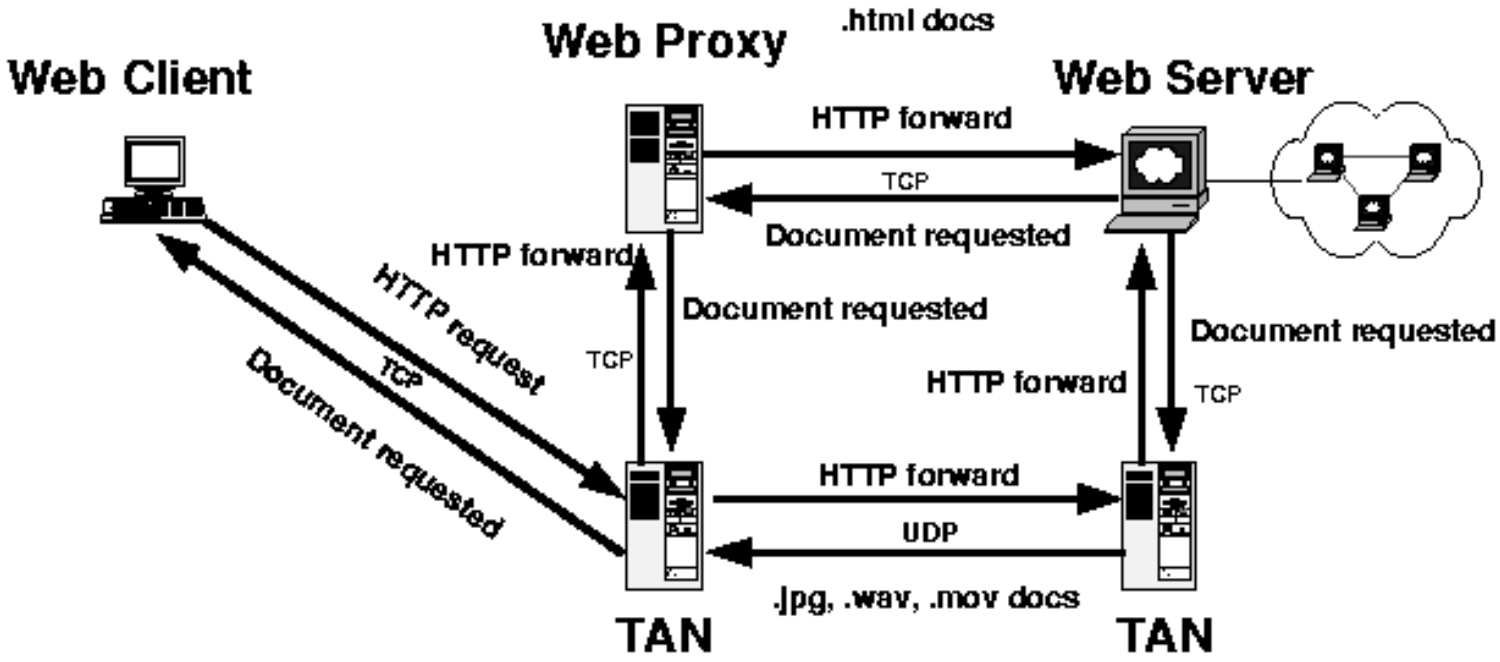


- CoWS 2 : all important streams (html, web programs like (shockwave, applet, javascript), style sheets) are transported with TCP while multimedia streams (pictures, video, sounds) arrive with UDP;
- CoWS 3 : all important streams (html, web applications, style sheets\ldots) arrive and files which size is under a given value arrive with TCP while multimedia streams use UDP. This CoWS is useful to guarantee good arriving of small important multimedia elements (icons, index bar..). The value is provided by users;
- Deployment of TCP2TCP (forwarding) services combined with UDP2TCP and TCP2UDP services + analysis of html document
- Require TANs located closely to the web server

Service blocks

- **TCP2UDP** : on entry active node (“near” web server)
  - Generating UDP packet header;
  - Data packing and adding of information inside packet payloads (packet number, file size…)
- **UDP2TCP** : on exit active node (“near” user)
  - TCP header generation and data packing;
  - Lost management : generation of empty packets to eventually deliver corrupted streams;
  - Management and generation of sequence numbers to detect UDP packets losses.
- Between TANs : « upgraded » UDP streams

CoWS 4



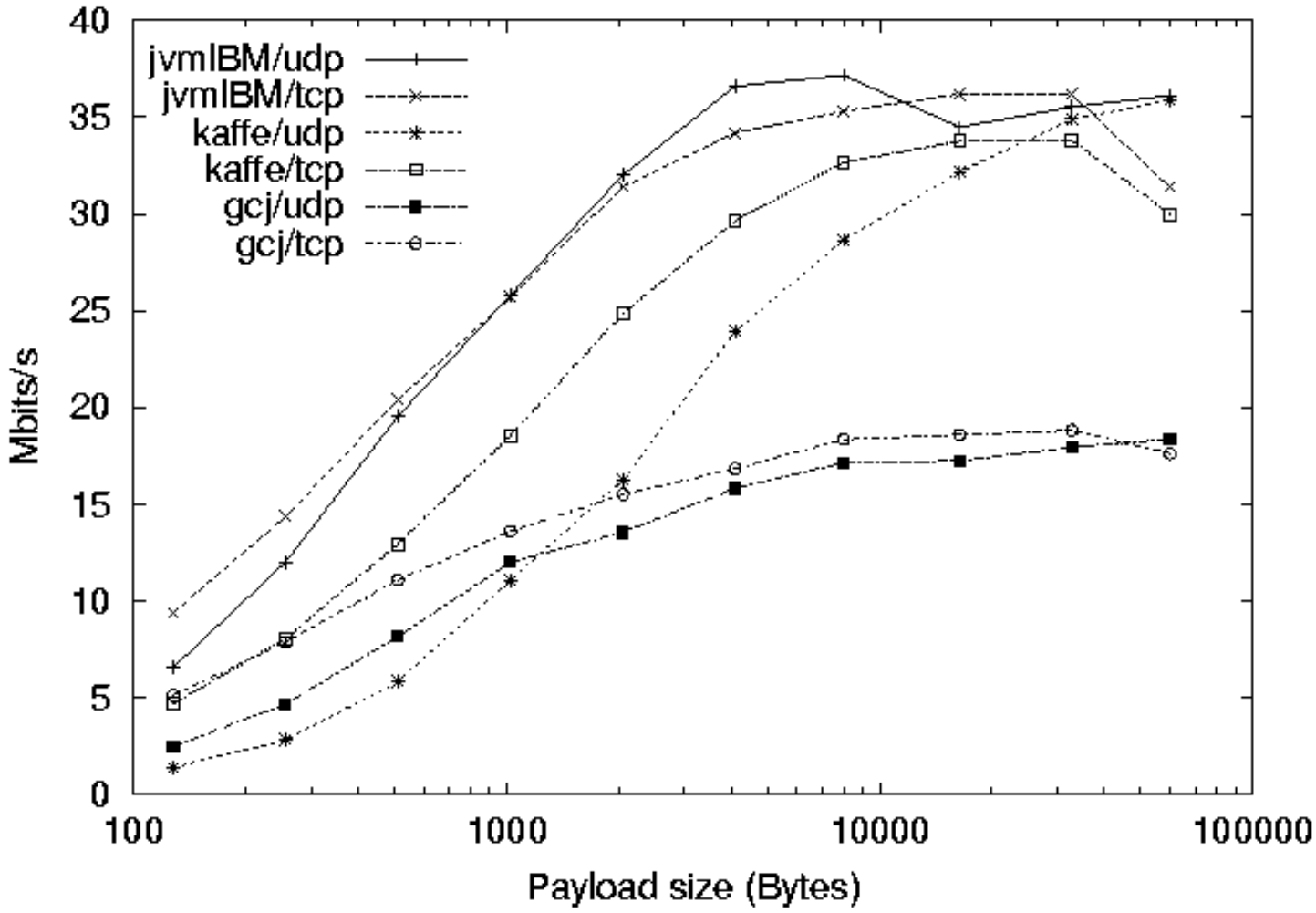
- CoWS 4 : all streams arrive with UDP and same data are requested in parallel by the proxy in TCP. Proxy loads requested files and can quickly provide them in case of important packets lost during UDP transfer;
- Deployment of TCP2TCP (forwarding) services combined with UDP2TCP and TCP2UDP services
- Require TANs located near the web server

CoWS 5,6

- CoWS 5 : all streams arrive with UDP transport protocol. In this CoWS, users trust network capabilities and want to quickly surf even if they lose few data. This is the most optimistic CoWS;

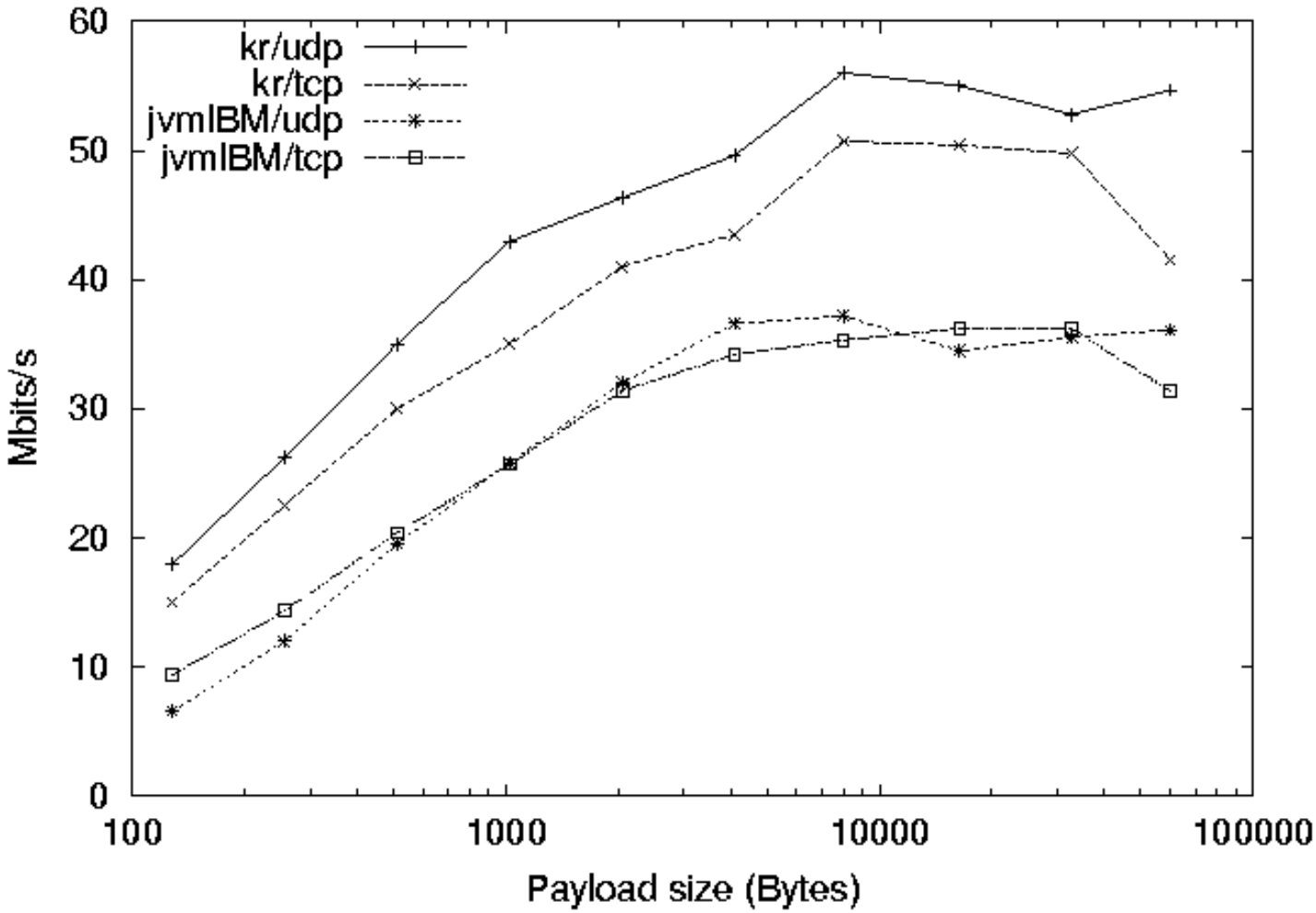
- CoWS 6 : automatic tuning of service. Local active node dynamically chooses the CoWS depending on network state and data frequency and size. (to be implemented)

Performance evaluation



For one http stream : TCP2TCP and UDP2UDP block

Performance evaluation



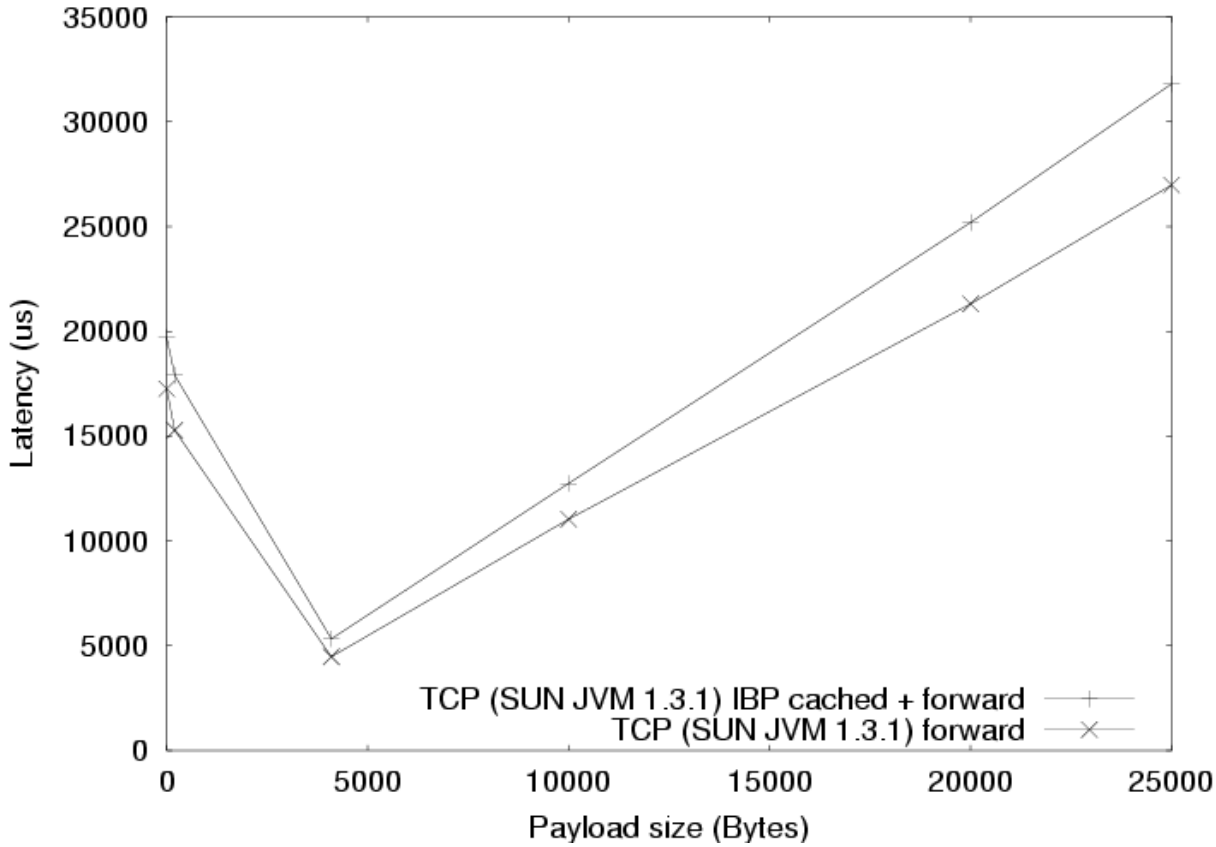
For basic forwarding service TCP2TCP : deployment inside kernel

Adding storage inside active network equipments

- Avoid using proxy cache
- Allow storage on the fly on corrupted and incomplete streams (udp)
- Use IBP depots already deployed with Tamanoir active nodes to store data

Performance evaluation of IBP services for Tamanoir

- Interaction between TAN and IBP
  - Independent processes
  - Socket-based communication
  - IBP services creates IBP Capabilities (pointers to IBP allocations)



Conclusions and future works

- Transform an active node into web proxy assistant. Active services can be located on proxy machine or active router
- Another way to use active node
- Interoperable solution with existing platforms : more AN, better services..
- Need to evaluate E2E benefits for end-user in specific scenarios
- Design and implement CoWS 6 with monitoring tools (NWS)

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.