Plan B's /net: An adaptative file system for networks

Gorka Guardiola Múzquiz

paurea@lsub.org
http://plan9.escet.urjc.es/who/paurea

Roadmap

- Computing environments
- Plan B
- Name spaces
- `/net`
- Problems
- Related work
- Lessons

Today's computing environments

Heterogeneous trasient resources

- Different implementations
- Different interfaces
- Different properties
- More than one at a time
- Come and go

Computing environments (cntd)

Networks

- Used to access (talk to) services. I care about reading my e-mail, best latency least cost.
- But...
  - Now I have to specify Bluetooth, Wifi or Ethernet.
  - with different bandwidths, latencies, ...
  - many at a time
  - change explicitly
- ⇒ Applications (or system) get complex.

Plan B

Built for changing distributed environments

- No connections
- No binding (`open` banned).
- Single names for multiple instances
- Dynamic mounting for network resources
- Per application mount table (customizable).

Plan B (cntd.)

Example

```
: import any /usr/nemo!Dgsyc /usr/nemo
: import -b any /usr/nemo /usr/nemo
: ns
...
  /usr/nemo: nautilus!fs!/usr/nemo
  /usr/nemo: aquamar!fs!/usr/nemo
: cp /usr/nemo/doc/slides.mf /b/printer
```

Name spaces

Tool to detect resources and adapt.

- Like a private mount table.
- One per application.
- Ordered prefix table.
- Order fixed at programming time.
- Dynamic (automatic) mounts.

Name spaces (cntd.)

Contraints

- Describe properties of resources.
- Used to narrow resolutions.
- Resolved once per system call (no binding!).
- Just syntax: users give meaning.
- Example:

```
: cp  slides.ps!Tpostscript /b/printer
```

Plan B's `/net`

Interface to the network (like, for example, sockets)

- One file system per network
- A high level single interface (abstract).
- Users select by programming the name space

Plan B's /net (cntd.)

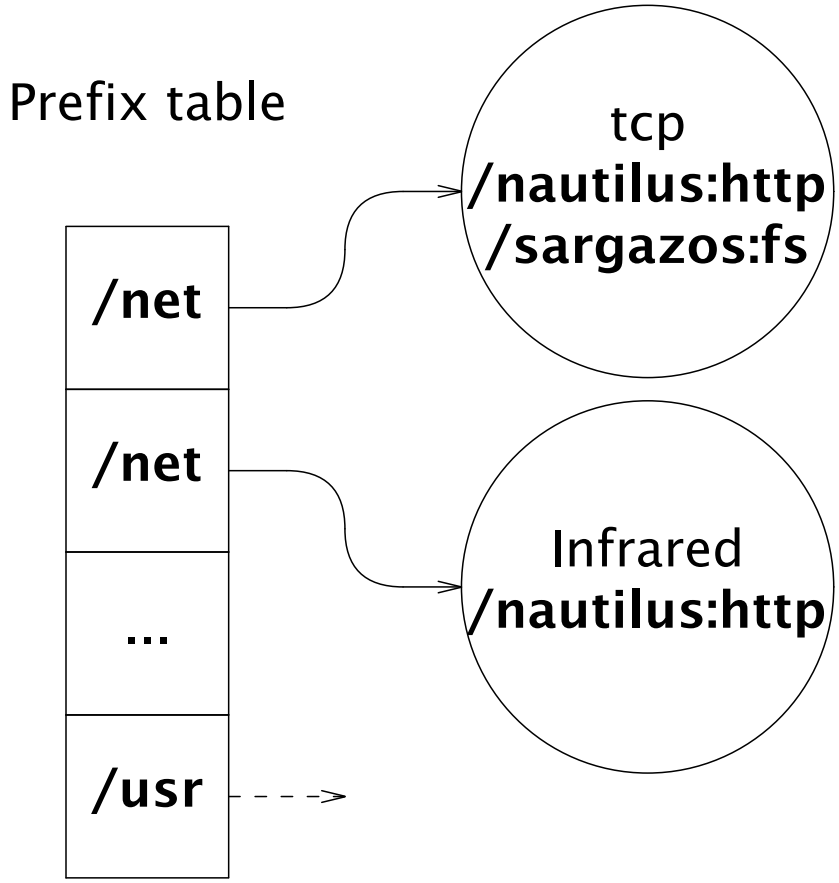
Naming as a tool

/net/machine!svc

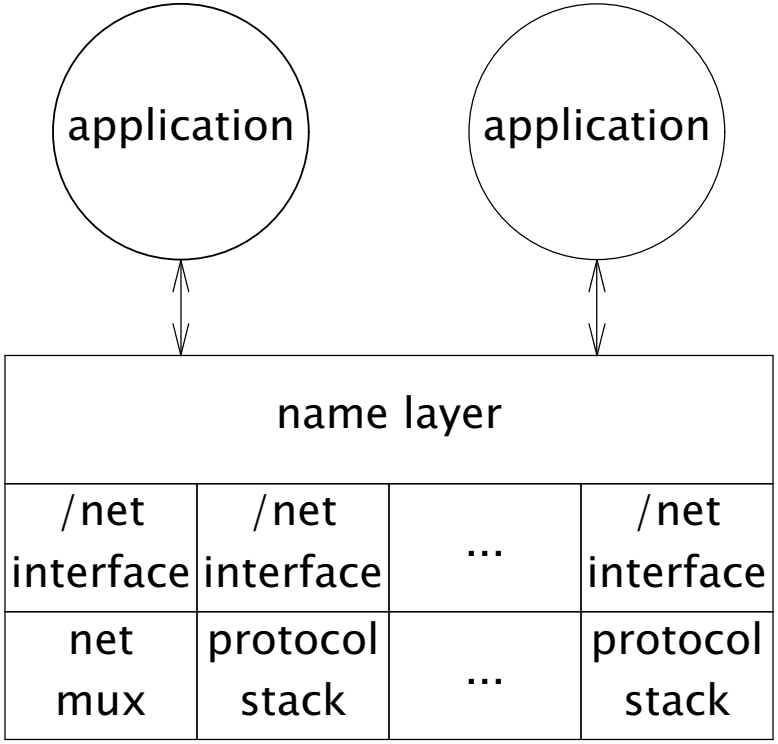
/net/any!svc

/net/local.N!svc

Use of namespaces


Prefix table
/net
/net
...
/usr
tcp
/nautilus:http
/sargazos:fs
Infrared
/nautilus:http

/net layers


application
application
name layer
/net interface
/net interface
...
/net interface
net mux
protocol stack
...
protocol stack

Plan B's `/net` (cntd.)

Semantics

- No connections (user's view)
- No addresses, names of machines and services
- Multiple networks at `/net`
- Dynamically mounted
- Constraints can narrow the search

Client

Create the endpoint

```
make("/net/nautilus:http:0");
```

Send the request

```
copy("/net/nautilus:http:0",0, "/mem",
&req, sizeof(req));
```

Client (cntd.)

Get the reply (n bytes)

```
n= copy("/mem", &repl, "/net/nautilus:http:0",
0,sizeof(repl));
```

Hang up

```
delete("/net/nautilus:http:0");
```

Server

In a loop do:

Make local endpoint

```
make("/net/local:http:0");
```

Get the request

```
n= copy("/mem", &req, "/net/local:http:0",
0,sizeof(req));
```

Server (ctnd.)

Log the peer's address

```
copy("/mem",&addr,
"/net/remote:http:0",sizeof(addr));

print("peer:%s \n", addr);
```

Server (ctnd.)

Process the request

```
do_req(req,repl);
```

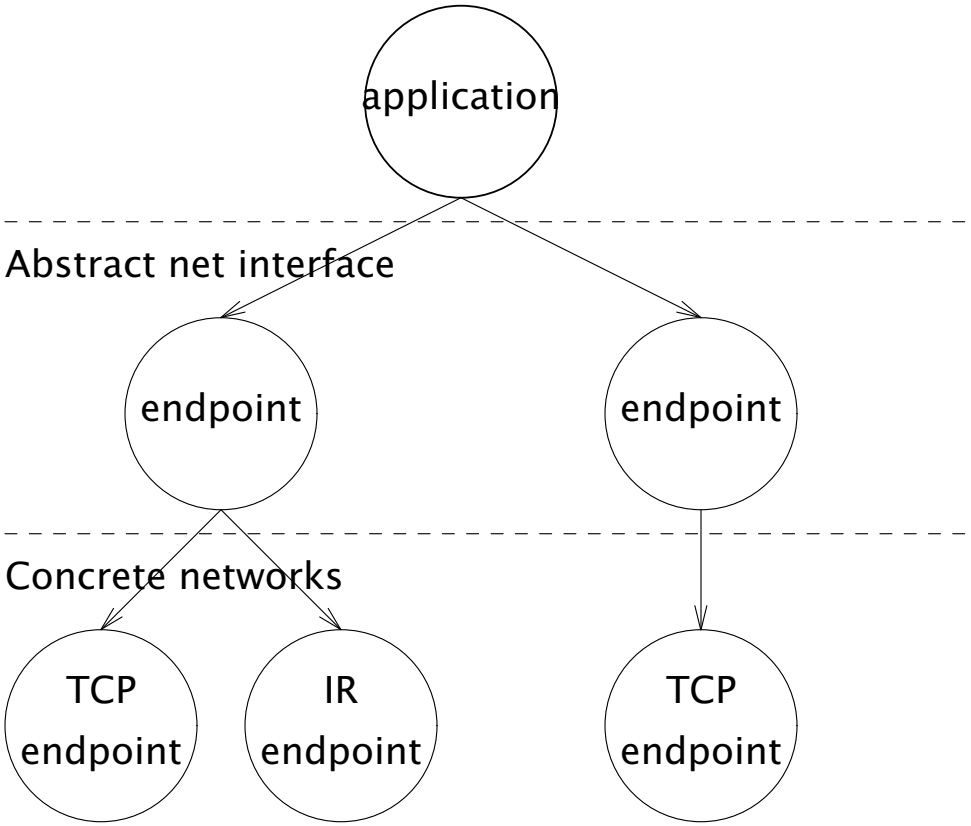
Send the reply

```
copy("/net/local:http:0",0,
"/mem",&repl, sizeof(repl));
```

Hang up

```
delete("/net/local:http:0");
```

Use of endpoints



Endpoints (ctnd.)

Represent services on machines

Use random numbers to disambiguate

Client

Local

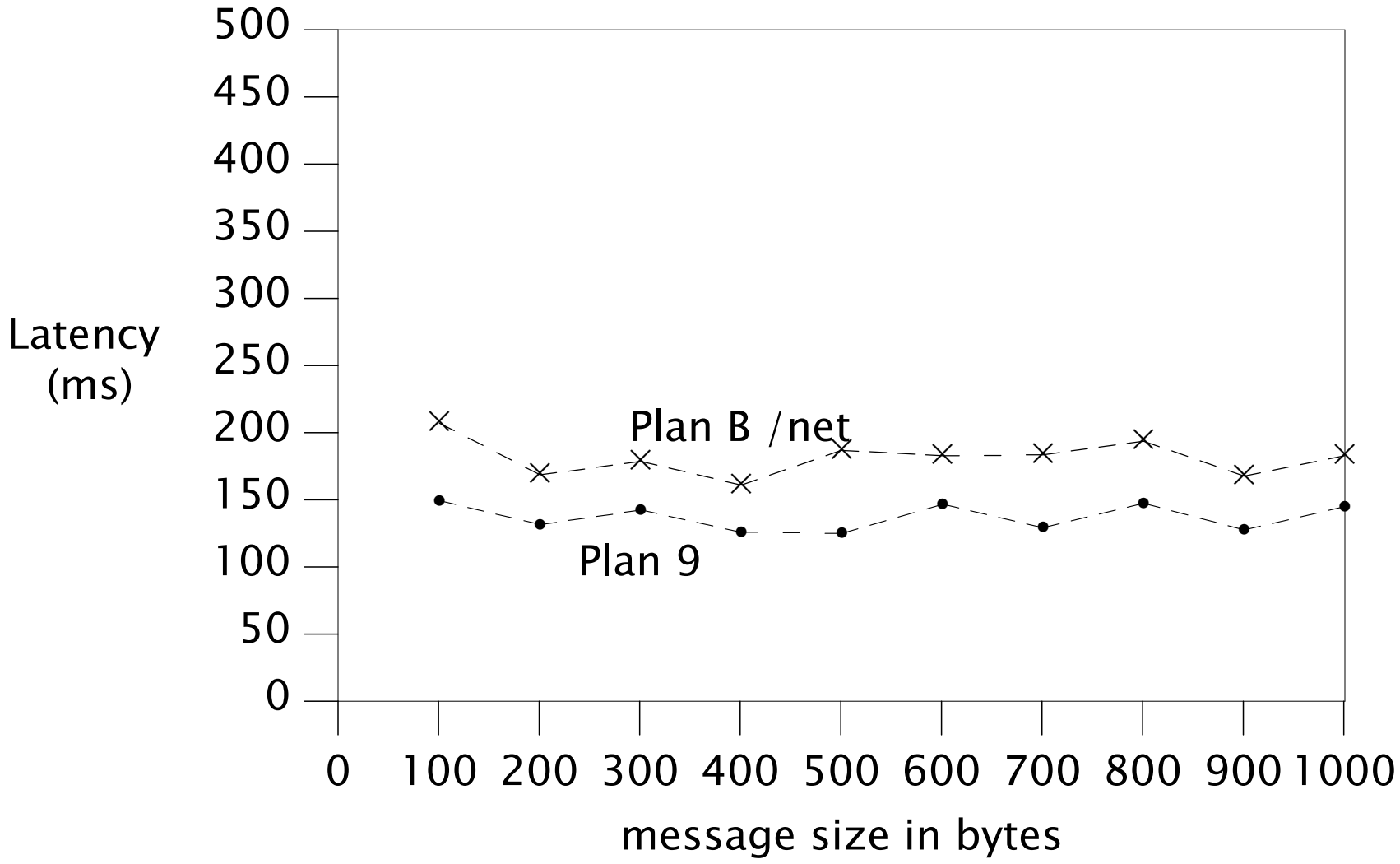
Remote

Raw

Measures (ctnd.)

Only has sense to compare with Plan 9 (same protocol stack).

Measures (ctnd.)


Latency
(ms)
500
450
400
350
300
250
200
150
100
50
0
Plan B /net
Plan 9
0
100
200
300
400
500
600
700
800
900
1000
message size in bytes

Problems

- Resolving each time (not much)
- Letting the user know
- Qos, numeric constraints

Related work

Mobile sockets, overlay networks etc.

None deal with heterogeneous networks.

Lessons

Be careful with timeouts

Connections are evil

Names have power

WiP

We had a native and hosted version on 2nd edition.

We had our web server running on different nets

We are now on 3rd edition, halfway between Plan B and Plan 9.

A rewrite for this edition will be done.

Questions?

http://lsub.org/ls/planb.html

paurea@lsub.org