

Enabling TI-RPC for IPv6

Linda Wu

linda.wu@sun.com

Outline

- TI-RPC Overview
- Changes to Solaris TI-RPC for IPv6
- Design goals
- Open Issues
- NFS over IPv6
- Conclusion

TI-RPC Overview

TI-RPC vs TS-RPC

- TS-RPC(Transport Specific) has a strong tie to IP. API likes: `svcdp_create()` or `clntudp_create()`.
- TI-RPC (Transport Independent RPC) has interface like `svc_create()`, `clnt_create()`
- TI-RPC Uses TLI (Transport Layer Interface), like `t_open()`, `t_alloc()`, `t_bind()`

TI-RPC Overview

Client creation API

Client = `clnt_create(hostname, prog, vers, nettype)`

`clnt_create()` does the following:

- Network protocol selection.
- Open communication endpoint
- Find remote host address and service port

TI-RPC Overview

server creation API

of Server handle =

```
svc_create(dispatch, prognum, versnum, nettype)
```

svc_create does the following:

- Network protocol selection
- Open communication endpoint
- Register with portmapper and rpcbind

TI-RPC Overview

/etc/netconfig

Example of /etc/netconfig:

<u>netid</u>	<u>semantic</u>	<u>flag</u>	<u>protofamily</u>	<u>proto</u>	<u>device</u>
udp	tpi_clts	v	inet	udp	/dev/udp

- TLI + /etc/netconfig makes TI-RPC
- Depends on the input selection to `clnt_create()` and `svc_create()` the corresponding netconfig structure will be retrieved by `getnetconfig()` API.

TI-RPC Overview

How `/etc/netconfig` is used?

- Device field is used to open communication end point
- netid field has been used in many places. most important place is in rpcbind database {prog, ver, netid, universal addr}
- Other fields are used mostly for filtering purpose. Currently nettype accepts the following: *null, visible, circuit_v, datagram_v, circuit_n, datagram_n, udp, tcp*

Changes to Solaris TI-RPC

New entries for IPv6 in /etc/netconfig

Treat IPv6 as a new protocol:

<u>netid</u>	<u>semantic</u>	<u>flag</u>	<u>protofamily</u>	<u>proto</u>	<u>device</u>
udp6	tpi_clts	v	inet6	udp	/dev/udp6
tcp6	tpi_cots	v	inet6	tcp	/dev/tcp6

Changes to Solaris TI-RPC

Other changes

- The universal address for IPv6 is: IPv6 address.p1.p2
- Well own port services listen on the same port for both IPv4 and IPv6. None-well known port servers listen on different port for IPv4 and IPv6.
- rpcbind has tcp6 and udp6 in the netid field to indicate is service is available over IPv6.
- portmapper is only support on IPv4 for interoperability

Changes to Solaris TI-RPC

Dual IP Stack

Server:

- Register with both protocol

Client:

- Try one first, if failed the other one
- The order can be customized by NETPATH or the order in the /etc/netconfig

Design Goal

- Provide backward compatibility
- Provide interoperability between Dual Stack IPv4/6 hosts and IPv4 only hosts
- Provide an easy migration for the previously written IPv4 RPC application to the Dual IPv6 and IPv4 environment.

Open Issues

How to handle `rpc_broadcast()`?

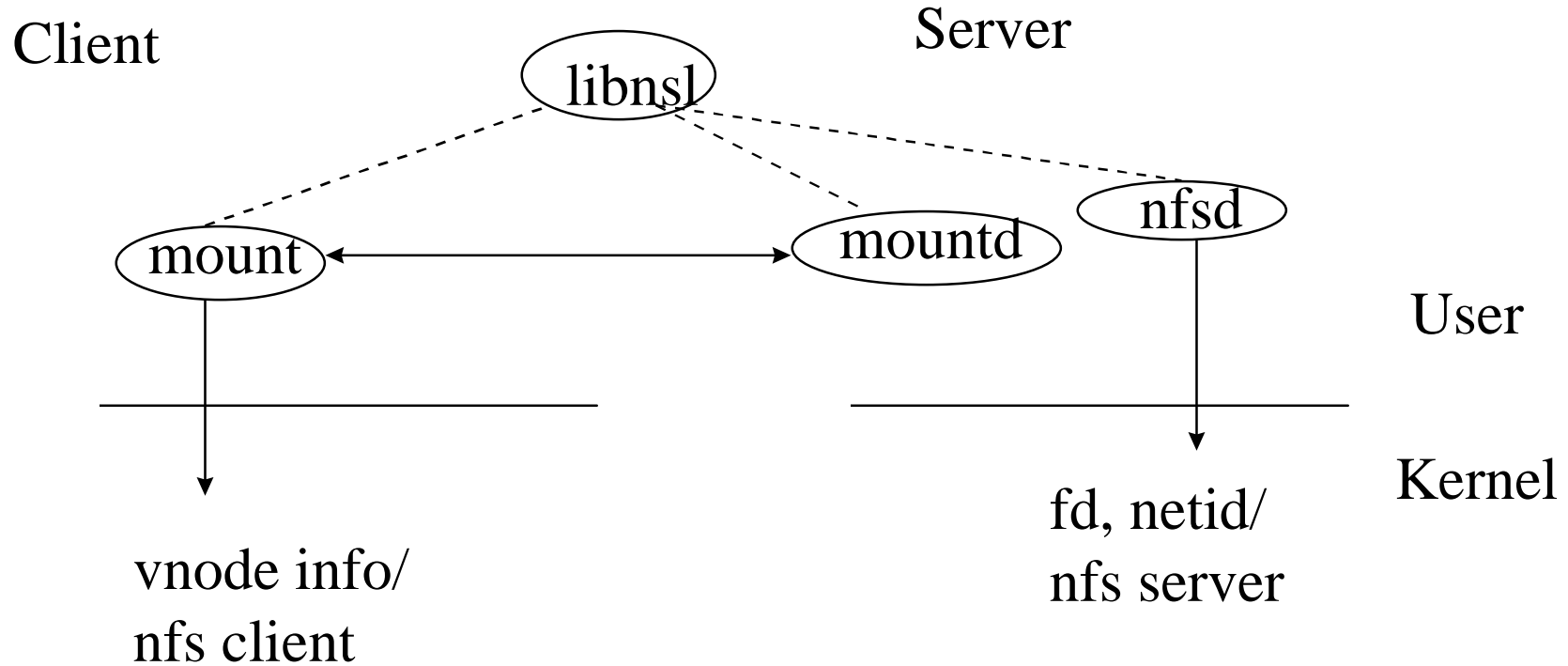
Server side:

- `rcpbind` register with a multicast address

Client side;

- If the hosts has dual stack, what is the mechanism?
send mutlicast request first over IPv6 , then broadcast
message over IPv4?

NFS over IPv6



libnsl: user library contains TI-RPC APIs

Conclusion

- New entry for IPv6 in /etc/netconfig
- Universal Address of IPv6 is: IPv6address.p1.p2
- netid in rpcbind database has udp6 and tcp6 for IPv6 services
- Dual IP stack host registers service with both stack. Client tries one protocol than the other.
- Portmapper is not going to be support in IPv6.
- The changes in user level RPC also effect running NFS over IPv6