
Subject: [PATCH v2 10/15] SUNRPC: service request network namespace helper introduced

Posted by [Stanislav Kinsbursky](#) on Wed, 25 Jul 2012 12:57:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

This is a cleanup patch - makes code looks simpler.

It replaces widely used `rqstp->rq_xprt->xpt_net` by introduced `SVC_NET(rqstp)`.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

```
fs/lockd/host.c      | 2 +-
fs/nfs/callback_xdr.c | 4 ++-
fs/nfsd/export.c     | 4 ++-
fs/nfsd/nfs4idmap.c  | 4 ++-
include/linux/sunrpc/svc.h | 2 ++
5 files changed, 9 insertions(+), 7 deletions(-)
```

diff --git a/fs/lockd/host.c b/fs/lockd/host.c

index 0084ab8..f9b22e5 100644

--- a/fs/lockd/host.c

+++ b/fs/lockd/host.c

```
@@ -331,7 +331,7 @@ struct nlm_host *nlmsvc_lookup_host(const struct svc_rqst *rqstp,
 struct nsm_handle *nsm = NULL;
 struct sockaddr *src_sap = svc_daddr(rqstp);
 size_t src_len = rqstp->rq_daddrlen;
- struct net *net = rqstp->rq_xprt->xpt_net;
+ struct net *net = SVC_NET(rqstp);
 struct nlm_lookup_host_info ni = {
 .server = 1,
 .sap = svc_addr(rqstp),
```

diff --git a/fs/nfs/callback_xdr.c b/fs/nfs/callback_xdr.c

index e64b01d..742ff4f 100644

--- a/fs/nfs/callback_xdr.c

+++ b/fs/nfs/callback_xdr.c

```
@@ -863,7 +863,7 @@ static __be32 nfs4_callback_compound(struct svc_rqst *rqstp, void
 *argp, void *r
 .drc_status = 0,
 .clp = NULL,
 .slotid = NFS4_NO_SLOT,
- .net = rqstp->rq_xprt->xpt_net,
+ .net = SVC_NET(rqstp),
 };
 unsigned int nops = 0;
```

```
@@ -879,7 +879,7 @@ static __be32 nfs4_callback_compound(struct svc_rqst *rqstp, void
 *argp, void *r
 return rpc_garbage_args;
```

```

    if (hdr_arg.minorversion == 0) {
-   cps.clp = nfs4_find_client_ident(rqstp->rq_xprt->xpt_net, hdr_arg.cb_ident);
+   cps.clp = nfs4_find_client_ident(SVC_NET(rqstp), hdr_arg.cb_ident);
    if (!cps.clp || !check_gss_callback_principal(cps.clp, rqstp))
        return rpc_drop_reply;
    }
diff --git a/fs/nfsd/export.c b/fs/nfsd/export.c
index ba23349..d77acd0 100644
--- a/fs/nfsd/export.c
+++ b/fs/nfsd/export.c
@@ -929,7 +929,7 @@ struct svc_export *
rqst_exp_get_by_name(struct svc_rqst *rqstp, struct path *path)
{
    struct svc_export *gssexp, *exp = ERR_PTR(-ENOENT);
-   struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+   struct nfsd_net *nn = net_generic(SVC_NET(rqstp), nfsd_net_id);
    struct cache_detail *cd = nn->svc_export_cache;

    if (rqstp->rq_client == NULL)
@@ -960,7 +960,7 @@ struct svc_export *
rqst_exp_find(struct svc_rqst *rqstp, int fsid_type, u32 *fsidv)
{
    struct svc_export *gssexp, *exp = ERR_PTR(-ENOENT);
-   struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+   struct nfsd_net *nn = net_generic(SVC_NET(rqstp), nfsd_net_id);
    struct cache_detail *cd = nn->svc_export_cache;

    if (rqstp->rq_client == NULL)
diff --git a/fs/nfsd/nfs4idmap.c b/fs/nfsd/nfs4idmap.c
index dae36f1..fdc91a6 100644
--- a/fs/nfsd/nfs4idmap.c
+++ b/fs/nfsd/nfs4idmap.c
@@ -546,7 +546,7 @@ idmap_name_to_id(struct svc_rqst *rqstp, int type, const char *name,
u32 namelen
    .type = type,
};
int ret;
-   struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+   struct nfsd_net *nn = net_generic(SVC_NET(rqstp), nfsd_net_id);

    if (namelen + 1 > sizeof(key.name))
        return nfserr_badowner;
@@ -571,7 +571,7 @@ idmap_id_to_name(struct svc_rqst *rqstp, int type, uid_t id, char *name)
    .type = type,
};
int ret;
-   struct nfsd_net *nn = net_generic(rqstp->rq_xprt->xpt_net, nfsd_net_id);
+   struct nfsd_net *nn = net_generic(SVC_NET(rqstp), nfsd_net_id);

```

```
    strcpy(key.authname, rqst_authname(rqstp), sizeof(key.authname));
    ret = idmap_lookup(rqstp, idtoname_lookup, &key, nn->idtoname_cache, &item);
diff --git a/include/linux/sunrpc/svc.h b/include/linux/sunrpc/svc.h
index 40e0a27..d83db80 100644
--- a/include/linux/sunrpc/svc.h
+++ b/include/linux/sunrpc/svc.h
@@ -278,6 +278,8 @@ struct svc_rqst {
    struct task_struct *rq_task; /* service thread */
};

+#define SVC_NET(svc_rqst) (svc_rqst->rq_xprt->xpt_net)
+
+/*
+ * Rigorous type checking on sockaddr type conversions
+ */
```
