This code needs a rework to support versioning, because the outer message encoding in e.g. handle_discover_reply operates on a "while(!p.end())" basis. These structures need to be lists of bufferlists instead, where each bufferlist is something versioned.

Oh right, so as Zheng reminds me in his patch for #12105, we do have the CEPH_MDS_PROTOCOL value for these changes.

I don't know... it still feels like it would be better to not rely on a global protocol version for all these fields. Anyone else have feelings?

yes, it's better to not rely on CEPH_MDS_PROTOCOL. (CEPH_MDS_PROTOCOL will make the rework easier)

Mmmm. For over-the-wire encodings that don't go to disk, it's only about whether cross-version daemons can communicate or not.

I think it might be worth maintaining that communication as much as possible, but I'm not certain where the best place to put in the versioning is — for instance we might want to do a per-message or per-session encoding version instead that we keep track of instead?

Tracker changed from Bug to Feature

Category set to 90

Subject changed from Inodes, Dirs encoded unversioned in encode_replica to mds: use versioned wire protocol; obviate CEPH_MDS_PROTOCOL

Target version set to v13.0.0

Component(FS) MDS added
I'm working on this, please assign this to me

- Status changed from New to In Progress
- Assignee set to dongdong tao

have added the support for MDentryLink and MDentryUnlink, next step is CInode::encode_lock_state/CInode::decode_lock_state, there are two big switch, need to break down to small encode/decode functions for each state.

- Category deleted (90)
- Target version changed from v13.0.0 to v14.0.0
- Tags set to multimds,upgrade

- Target version changed from v14.0.0 to v15.0.0

- Status changed from In Progress to Fix Under Review
- Start date deleted (06/22/2015)
- Tags deleted (multimds,upgrade)
- Pull request ID set to 20160
- Labels (FS) multimds added

- Status changed from Fix Under Review to Resolved