规约采集处理思路

零, +++++ 基本定义 +++++

typedef int proto\_point\_no; /\* 规约点号 \*/

typedef QString mms\_var\_ref; /\* MMS变量参引 \*/

typedef int fert\_point\_id; /\* 采集点id \*/

typedef int scada\_point\_id; /\* SCADA点id \*/

typedef int link\_id; /\* 链路id \*/

typedef int station\_id; /\* 变电站id \*/

typedef QVariantHash fert\_one\_data\_value; /\* 采集点数据完整值 \*/

typedef QVariantHash scada\_one\_data\_value; /\* SCADA点数据完整值 \*/

typedef struct

{

mms\_var\_ref;

fert\_one\_data\_value;

}GRADE\_ONE\_DATA\_TYPE; /\* 一类数据 \*/

typedef struct

{

mms\_var\_ref;

fert\_one\_data\_value;

}GRADE\_TWO\_DATA\_TYPE; /\* 二类数据 \*/

以下以mms\_var\_ref为例详细说明数据处理过程, 以proto\_point\_no为模型的类似.

一, +++++ 通信点和采集点id之间的映射关系 +++++

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 每一个链路拥有两个hash, key值和value值分别采用点号或参引和采集点id.

\* UpDataFertPointHash 为上行映射,点号或参引映射到采集点id.

\* DownDataFertPointHash 为下行映射,采集点id映射到点号或参引.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

QHash<mms\_var\_ref, fert\_point\_id> UpDataFertPointHash;

QHash<fert\_point\_id, mms\_var\_ref> DownDataFertPointHash;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 整个系统所有采集点的hash, key值采用采集点id, value值采用链路id.

\* FertPointLinkHash 为采集点id映射到链路id.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

QHash<fert\_point\_id, link\_id> FertPointLinkHash;

二, +++++ 数据处理过程 +++++

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 上行数据处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1, 在一个链路上送的通信报文中处理"点号或参引"的值,

2, 链路级数据缓冲处理, (可选)

3, 根据UpDataFertPointHash找到"采集点id",

4, 把采集点id和值发送给后端,

5，系统级数据缓冲处理. (可选)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 下行数据处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1, 从界面或应用接收到采集点id数据后,

2, 系统级数据缓冲处理, (可选)

3, FertPointLinkHash中找到隶属的链路id,

4, 然后在该链路的DownDataFertPointHash中找到"点号或参引",

5, 链路级数据缓冲处理. (可选)

三, +++++ 链路级数据缓冲处理 +++++

每一链路都拥有有以下数据:

QHash<mms\_var\_ref, fert\_one\_data\_value> UpStoredRecentData; /\* 上行数据缓存区 \*/

QQueue<GRADE\_ONE\_DATA\_TYPE> UpCachedGradeOneData; /\* 上行一类数据:用于处理低频变化数据(例如遥信) \*/

QQueue<GRADE\_TWO\_DATA\_TYPE> UpCachedGradeTwoData; /\* 上行二类数据:暂存性的数据(例如交互或命令数据) \*/

QQueue<mms\_var\_ref> UpCachedGradeThreeData; /\* 上行三类数据:周期性刷新数据(例如遥测) \*/

QHash<mms\_var\_ref, fert\_one\_data\_value> DownStoredRecentData; /\* 下行数据缓存区 \*/

QQueue<GRADE\_ONE\_DATA\_TYPE> DownCachedGradeOneData; /\* 下行一类数据:用于处理低频变化数据(例如遥信) \*/

QQueue<GRADE\_TWO\_DATA\_TYPE> DownCachedGradeTwoData; /\* 下行二类数据:暂存性的数据(例如交互或命令数据) \*/

QQueue<mms\_var\_ref> DownCachedGradeThreeData; /\* 下行三类数据:周期性刷新数据(例如遥测) \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 上行 链路级数据缓冲处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

接收到的所有一类数据和三类数据全部要保存到缓存区中UpStoredRecentData中(二次数据不需要),

UpStoredRecentData时刻保存着该链路的所有最新数据, 在保存到UpStoredRecentData之前, 必须先更新暂存区.

一类数据暂存区, 采用FIFO队列缓存, 数据变化频度低(遥信类), 重复数据不能折叠(在队列中依次排队,保持其原有时序和值),

CachedGradeOneData的入队判断标准是,其值是否发生变化(质量码或标志位等)或UpStoredRecentData中不存在(第一次采集).

二类数据暂存区, 采用FIFO队列缓存, 属于交互或命令数据, 该数据不需要保存到UpStoredRecentData中.

三类数据暂存区, 采用FIFO队列缓存, 数据变化频度高(遥测类), 重复数据可以折叠(在队列中不需要保存其完整值),

CachedGradeThreeData的入队判断标准是, 其值变化超过门槛值(质量码发生变化一般不考虑)或UpStoredRecentData中不存在(第一次采集).

上行数据暂存区发送给SCADA处理时:

按照UpCachedGradeOneData,UpCachedGradeTwoData,UpCachedGradeThreeData的优先顺序依次发送处理.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 下行 链路级数据缓冲处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

从界面或应用接收到的一类数据, 仍然按照入队原则放入QQueue<GRADE\_ONE\_DATA\_TYPE> DownCachedGradeOneData, 并更新DownStoredRecentData;

从界面或应用接收到的二类数据, 放入QQueue<GRADE\_TWO\_DATA\_TYPE> DownCachedGradeTwoData;

从界面或应用接收到的三类数据, 仍然按照入队原则放入QQueue<mms\_var\_ref> DownCachedGradeThreeData, 并更新DownStoredRecentData;

下行数据暂存区在组装报文时:

仍然按照UpCachedGradeOneData,UpCachedGradeTwoData,UpCachedGradeThreeData的优先顺序依次处理, 依次组装报文下发.

四, +++++ 系统级数据缓冲处理 +++++

QHash<link\_id, station\_id> LinkToStationHash; /\* 链路id到变电站id \*/

QHash<station\_id, link\_id> StationToLinkHash; /\* 变电站id到链路id \*/

QHash<scada\_point\_id, fert\_point\_id> FertToScadaPointHash; /\* SCADA点id到采集点id \*/

QHash<fert\_point\_id, scada\_point\_id> ScadaToFertPointHash; /\* 采集点id到SCADA点id \*/

QHash<scada\_point\_id, scada\_one\_data\_value> ScadaStoredRecentData; /\* 系统数据缓存区 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 上行 系统级数据缓冲处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SCADA在接收到上行数据后, 根据FertToScadaPointHash找到SCADA点id,

对一类数据和二类数据要保存到ScadaStoredRecentData中, 对三类数据做响应动作后续处理.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 下行 系统级数据缓冲处理

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1, SCADA在接收到界面或应用下发的命令后, 应该组装成二类数据, 并按照接口发送给指定的链路.

2, 需要下发的数据, 可以组装成一类数据和三类数据, 并按照接口发送给指定的链路.