

```

12:24 <@s-srv1> 2010-05-22 10:24:01 DL1XXX__ DB0XYZ_B (from: d-srv1)
12:31 <@s-srv1> 2010-05-22 10:31:29 DL2XXX__ DB0XYZ_B (from: d-srv1)
12:36 <@s-srv1> 2010-05-22 10:36:31 DK0XYZ__ DB0XYZ_B (from: d-srv1)
12:45 -!- te0st-4 [te0st@80.116.64.236] has joined #dstar
12:46 -!- te0st-3 [te0st@80.116.124.47] has quit [Ping timeout: 121 seconds]
12:49 <@s-srv1> 2010-05-22 10:48:58 DL3XXX__ DB0ZYX_B (from: d-srv1)
[13:00] [@u-dl1bff(+i)] [5:#dstar(+mnt)]
[#dstar]
    
```

new table: **sync_mng_external**

target_cs	last_mod_time	arearp_cs	zonerp_cs
DL1XXX	2010-05-22 10:24:01	DB0XYZ B	DB0XYZ
DL2XXX	2010-05-22 10:31:29	DB0XYZ B	DB0XYZ
DK0XYZ	2010-05-22 10:36:31	DB0XYZ B	DB0XYZ
DL3XXX	2010-05-22 10:48:58	DB0ZYX B	NOCALL99

Only the „target_cs“ and „arearp_cs“ are reported over the IRC channel. The software looks for the corresponding „zonerp_cs“ in table „sync_mng“. If it doesn't find it there, the repeater cannot contact the remote module. In this case, the special value „NOCALL99“ is put in the „zonerp_cs“ field.

incoming „insert“ statement from USTRUST via „dsipsvd“ with out-of-date routing information:

```

insert into sync_mng values('DL2XXX ', '2010-05-22 08:30:15',
    now(), now(), 'dl2xxx', 'DB0XYZ C', 'DB0XYZ ',
    'DL2XXX ', 'DB0XYZ ', '10.40.181.88', 'f')
    
```

```

CREATE OR REPLACE FUNCTION process_sync_mng_external() RETURNS TRIGGER AS $$
DECLARE
    ext_data sync_mng_external%ROWTYPE;
BEGIN
    SELECT * INTO ext_data FROM sync_mng_external
        WHERE target_cs = NEW.target_cs;

    IF NOT FOUND THEN
        RETURN NEW;
    END IF;

    IF ( NEW.last_mod_time < ext_data.last_mod_time )
        AND NOT ( ext_data.zonerp_cs = 'NOCALL99' ) THEN
        NEW.arearp_cs := ext_data.arearp_cs;
        NEW.zonerp_cs := ext_data.zonerp_cs;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_sync_mng_ext
BEFORE INSERT OR UPDATE ON sync_mng
FOR EACH ROW EXECUTE PROCEDURE process_sync_mng_external();
    
```

PostgreSQL trigger
trigger_sync_mng_ext

Every „insert“ or „update“ to the table „sync_mng“ will be checked by the trigger function. If there is data available in table „sync_mng_external“ with a more recent timestamp then the newer (and better) data is used.

existing table: **sync_mng**

target_cs	last_mod_time	arearp_cs	zonerp_cs
DL1XXX	2010-05-22 07:21:01	DB0XYZ B	DB0XYZ
DL2XXX	2010-05-22 08:30:15	DB0XYZ B	DB0XYZ
DK0XYZ	2010-05-21 19:36:31	DB0XYZ B	DB0XYZ



Every IRC client sends a „PING“ command to the IRC server every 30 seconds. The IRC server has to respond to this command with a „PONG“ answer. If this „PONG“ is not received within 30 seconds, the client closes the connection and tries to connect to the server again. If the client changes its IP address this „PING-PONG“ communication fails and results in a new connection to the IRC server within 1 minute.

```
12:24 <@s-srv1> 2010-05-22 10:24:01 DL1XXX__ DB0XYZ_B (from: d-srv1)
12:31 <@s-srv1> 2010-05-22 10:31:29 DL2XXX__ DB0XYZ_B (from: d-srv1)
12:36 <@s-srv1> 2010-05-22 10:36:31 DK0XYZ__ DB0XYZ_B (from: d-srv1)
12:45 -!- te0st-4 [te0st@80.116.64.236] has joined #dstar
12:46 -!- te0st-3 [te0st@80.116.124.47] has quit [Ping timeout: 121 seconds]
12:49 <@s-srv1> 2010-05-22 10:48:58 DL3XXX__ DB0ZYX_B (from: d-srv1)
[13:00] [@u-d11bff(+i)] [5:#dstar(+mnt)]
[#dstar]
```

The IRC server also sends „PING“ commands. After 2 minutes, the server closes the old connection.

```
12:45 -!- te0st-4 [te0st@80.116.64.236] has joined #dstar
```

IRC channel nickname is converted to capital letters

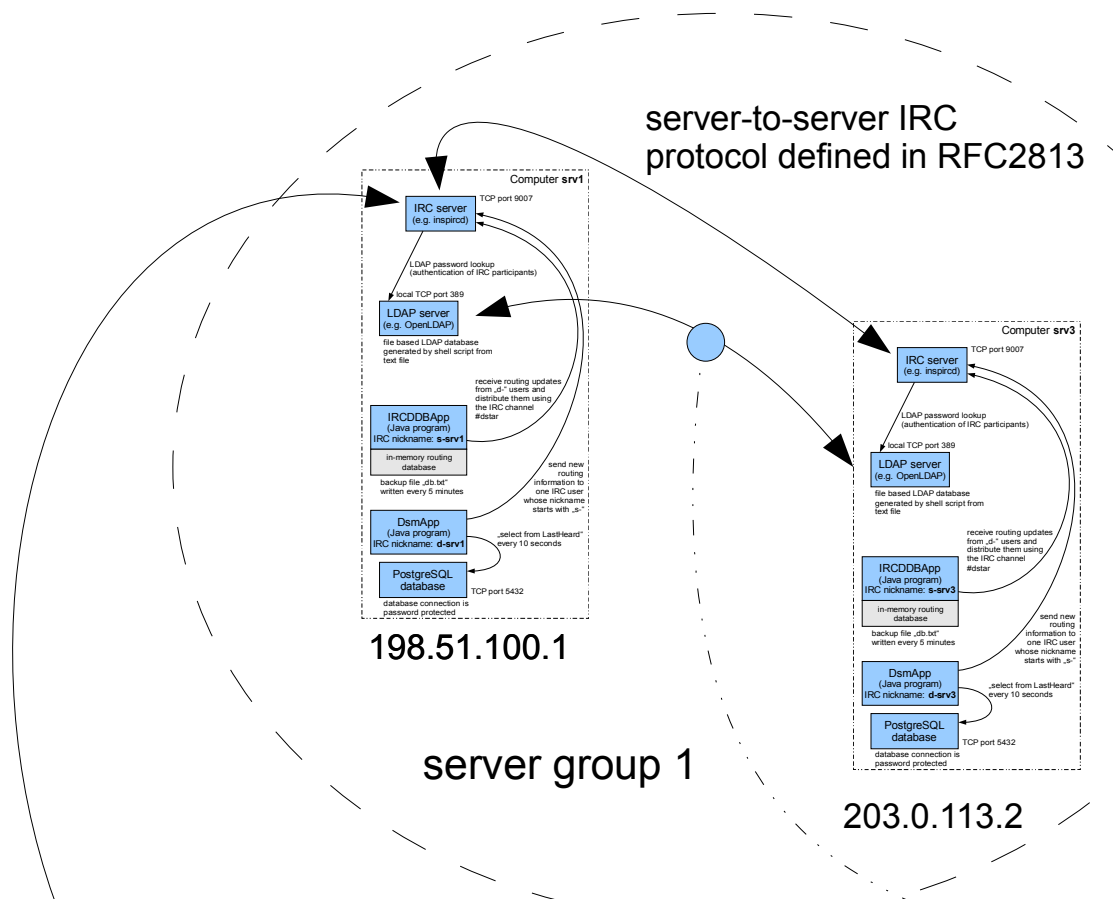
the „host“ part of the IRC server login message is the IP address of the repeater

```
update sync_gip set zonerp_ipaddr='80.116.64.236' where zonerp_cs='TE0ST'
```

existing table: **sync_gip**

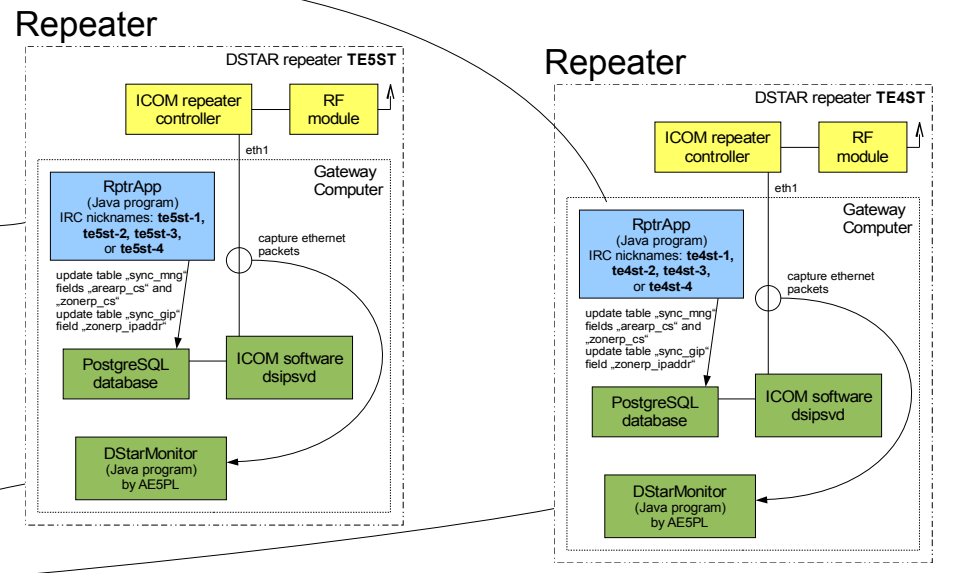
zonerp_cs	last_mod_time	zonerp_ipaddr	..
SZ1XX	2009-12-08 15:06:09	91.238.227.120	..
TE0ST	2010-05-01 10:24:01	80.116.64.236	..
VA2RX	2009-11-11 17:40:27	74.215.0.34	..





DNS: server-group-1.example.com.
198.51.100.1
203.0.113.2

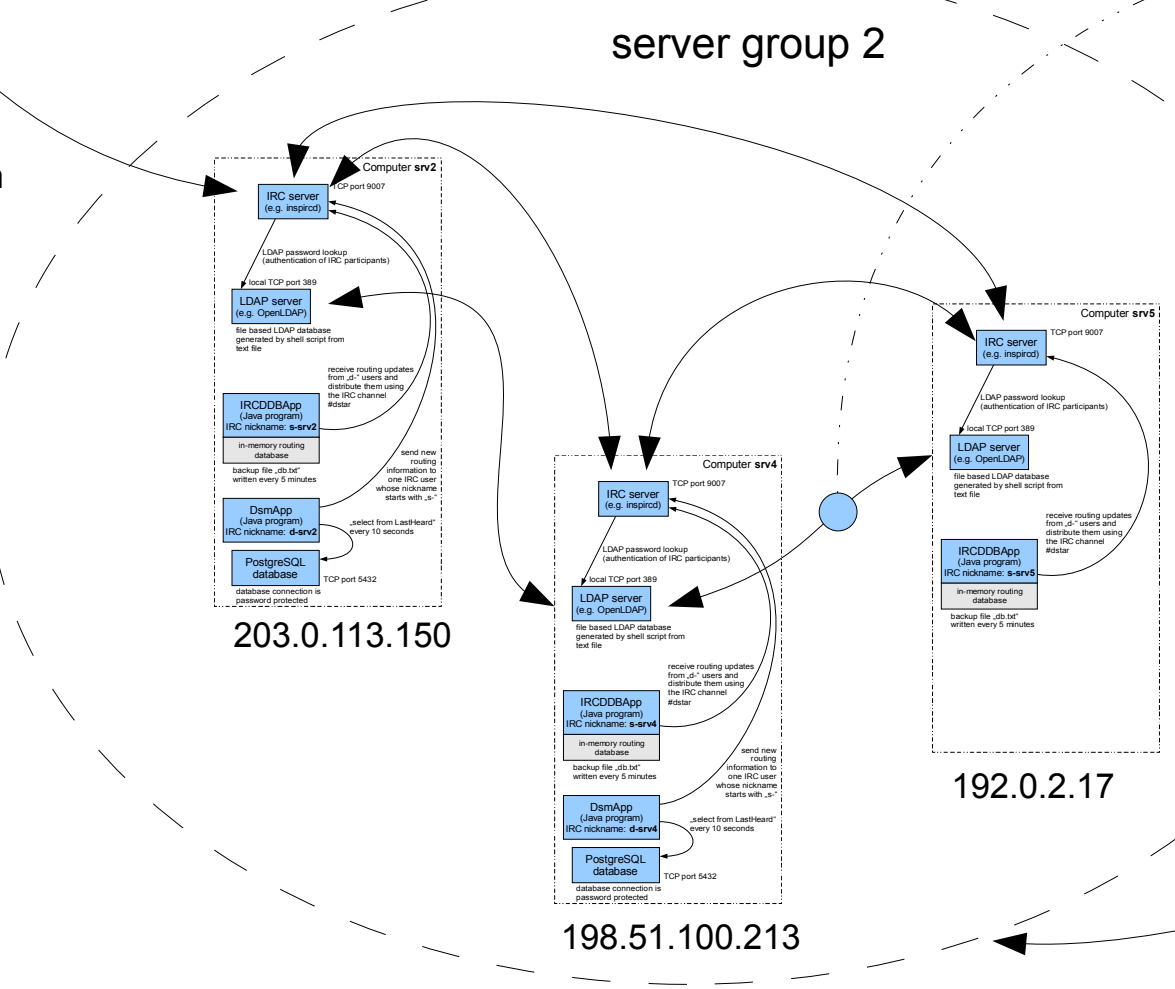
select IRC server IP address and PostgreSQL server IP address with DNS round robin



server-to-server IRC protocol defined in RFC2813

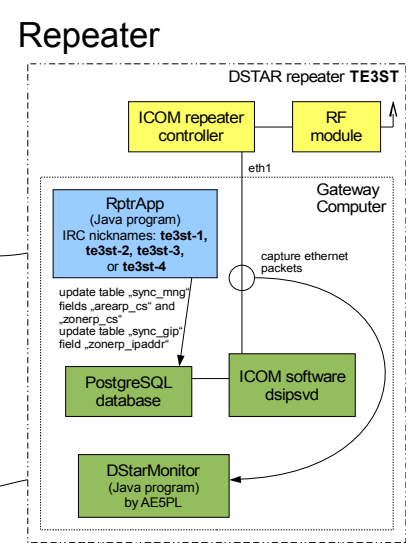
user account information has to be synchronized within a server group

(there should be more than one connection between groups of servers)



DNS: server-group-2-irc.example.com.
203.0.113.150
198.51.100.213
192.0.2.17

select IRC server IP address and PostgreSQL server IP address with DNS round robin separately



DNS: server-group-2-postgres.example.com.
203.0.113.150
198.51.100.213

