GOP’s Trop-NET system documentation

Version 1.2 (December, 2014)

Jan Douša, GOP - Geodetic Observatory Pecný (jan.dousa@pecny.cz)

Summary

This document provides instructions for installing and maintaining of the processing system primary developed at the Geodetic Observatory Pecný (GOP) for ground-based GNSS near real-time (NRT) troposphere monitoring in support of numerical weather prediction (NWM).

Background information about the processing system including a general philosophy of its architecture (package modules) is described first. Then, a minimum hardware and software requirements are specified together with basic instructions for installing and configuring entire environment and Trop-NET package. The second part of the document describes the strategy for downloading data and mandatory products, data processing strategy, product filtering, converting and uploading. The data processing depends on purchased and installed Bernese GNSS software (BSW) and exploits the Bernese Processing Engine (BPE).

Content

Introduction

1.1 Ground-based GNSS near real-time processing system

Requirements needed in support of ground-based GNSS near real-time solution using a batch processing approach, such as Bernese are:

- data flow from permanent GNSS stations on hourly basis (RINEX format),
- precise products and models from the International GNSS Service (IGS),
- highly effective, robust and fully automated procedure, modularized and self-repairing with a minimum requirement on manual intervention,
- continuous monitoring and evaluation tools for feedback.

During past 15 years, GOP developed a core system for automated GNSS processing using BSW+BPE including data/product flow in support of various scientific applications. These were used in different update modes and for different purpose, namely:

- near real-time GPS regional troposphere monitoring (EGVAP)
- near real-time GPS+GLONASS regional troposphere monitoring (EGVAP)
- near real-time GPS global troposphere monitoring (EGVAP)
- ultra-rapid GPS and GLONASS orbit determination (IGS)
- rapid (daily), final (weekly) GPS solution for reference frame (EUREF)
- homogeneous re-processing of full EUREF permanent network (EUREF)
- daily/hourly data flow at the local data centre (EUREF).

All the above systems benefited at GOP from developing a common core library that is continuously maintained in support of additional flexibility and robustness. A near real-time GNSS processing in support of meteorology was one of early applications of the library development in 1999. At that time, limits in hardware and software, instabilities in data flow, a low quality of 24-hour predicted orbit products and models, required the solution to be designed as highly effective, fully self-supporting and maximally robust. During the following decade, further enhancements was done when extending the library for other challenging applications such as ultra-rapid orbit determination (2003), early GLONASS introduction (2008) and hourly global troposphere solution (2010). Obviously, all steadily leaded to improved robustness and flexibility libraries modules developed either for data processing, data and product flow, monitoring or evaluation.

The above mentioned was behind the idea of sharing the part of the library for other utilizations in GNSS meteorology. The Trop-NET package consists of the source code on top of the Bernese Processing Engine and GNSS software, several independent modules and the processing strategy. Both are aimed for a transfer of knowledge within the COST ES1206, project GNSS4SWEC, in order to a) facilitate the establishment of new analysis centres, b) improve the product coverage and its homogeneity in Europe and c) give a possibility to share optimally future developments and closely coordinate any update.

Although the Trop-NET pack solution was designed as a robust one, various problems usually occurs in relations to a) **IT environment** - correct functionality of involved hardware, operational system, Internet connectivity etc., b) unavailability or incorrectness of **input data, precise products** and **models** during download, mirror or conversion actions and c) **reconfiguring** or **upgrading** the processing system, Bernese GNSS software or other software components. An overall control over these effects is specific for each site and should remain under a local control. Developments of the monitoring of HW, SW and other IT components together with generating early warning message (except the warning/error messages from the processing itself) is highly recommended, but not provided as a complex solution via the Trop-NET pack.

1.2 System architecture

The overall system consists of different modules and tools supporting individual utilization and settings for different scenarios but still keeping common interfaces, Figure 1. However, the configuration in support of near real-time GNSS solution only is part of this document. Modules supporting distributed processing nodes are

- module for downloading data and products, mirroring mandatory precise models
- module for GNSS data processing using BSW and BPE
- module for uploading products

Additional central components of the system are currently maintained at GOP:

- maintaining the repository for Trop-NET pack distribution and update including a support for decentralized development and system versioning
- monitoring and comparing available near real-time products
- monitoring and long-term evaluating products
- documentation and mail exchanging information.

Additional modules are considered for future development, such as converting ZTD to IWV, IWV animation plots, local production monitoring etc.

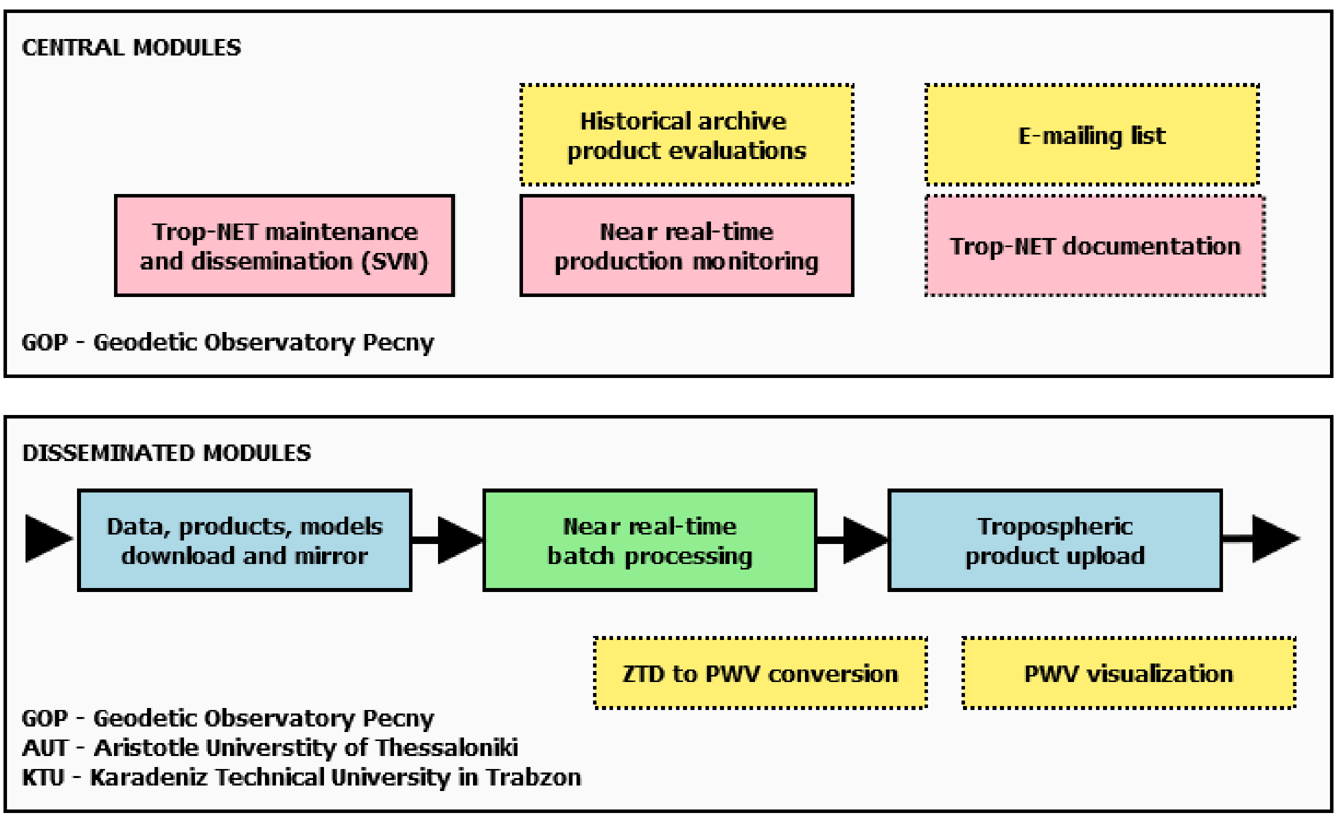

Figure 1: Trop-Net central and disseminated modules

System requirements

This section gives instructions on the hardware, operation system and software installation requirements, which are essential for the efficient and successful installation of GOP processing. This typically represents a need for a continuous maintenance and thus should be maximally in hands of the local staff.

2.1 Hardware and OS preparation

General requirements/recommendations about the hardware and operational system are following:

- stable and reliable Internet connectivity (>5Mbps recommended)
- optimally (recommended) independent Linux/Unix servers for a) core data processing with Bernese GNSS Software and b) data/products local data centre and product archive
- a conservative Linux operational system such as Debian is recommended for keeping a high stability of the server and a long-term support without frequent updates, but keeping important security patches)
- if possible, servers should be designed for an uninterrupted processing (e.g. reliable cooling, UPC, RAID, optionally: redundant power-source, rack design, durable CPU/RAM)
- it is recommended to install the root user as well as operating system at an individual partition at each server keeping it separate from an intensively used working areas. This makes the Linux system more robust from unpredictable errors and, generally, facilitates fixing any file system problem at more extensively used working areas

Minimum characteristics of the core data centre (and archive) if it differs from the processing server:

- no special requirements, but should be designed for a reliable data archive (disk space, redundancy) and a stable Internet connectivity
- *autofs* package to support automated NFS mounts from the processing server to the data server (recommended)
- minimum 150GB free disk storage (recommended for the local data archive) at a separated disk partition from the operating system in order to avoid any Linux system corruption due to extensive IO operations at working area

Minimum characteristics of the core processing server:

- OS Linux (required), optimally Debian GNU/Linux or its clone using deb-package system (recommended)
- minimum 300GB free disk storage (recommended for the processing itself only) at a separated disk partition from the operating system in order to avoid any Linux system corruption due to extensive IO operations at working area
- minimum CPU 1.5 GHz, optimally 4-16 cores (always better multi-core processors than a single ultra-fast processor)
- minimum 4GB RAM (strongly recommended, while it is always better to have more memory in support of more stations processed, more parameters and higher time resolution etc.)

2.2 Supporting software

Prior the Bernese GNSS Software (BSW) installation, specific libraries and software tools need to be installed. These are usually standard packages in Debian GNU/Linux (or its clone like Ubuntu):

- *bash (usually default), csh and perl* – scripting languages (required)
- *gcc, g++ and gfortran* – compilers (required)
- *make* – compilation tool (required)
- *qt3/qt4* – development libraries for BSW V5.0/V5.2 menu compilation)
- zlib1g-dbg – development compression library(required, used by menu)
- *wget and lftp* – download and mirroring tools (required)
- *recode* – tool for format conversions (required)
- Subversion (SVN) – version control system (required)
- *tar/gzip/ncompress* – archiving and compressing tools (required)
- *jed* – fast and powerful text editor (recommended)
- gv – EPS/PDF viewer (recommended)
- *gnuplot (optionally GMT)* – plotting tools (recommended)
- *ImageMagick* – graphics library (recommended)

Additional software for GNSS can be installed later with the GOP system:

- *RNXCMP* – compression tools *(http://terras.gsi.go.jp/ja/crx2rnx.html)*
- *teqc – UNAVCO's transform, quality check and* editing tool for GNSS data *(http://www.unavco.org/software/data-processing/teqc/teqc.html)*

2.3 Bernese GNSS Software

The Bernese GNSS Software V5.2 (alternatively V5.0) should be properly installed, according to BSW documentation (http://www.bernese.unibe.ch/), at the processing server and setting BSW environment (by running $X/EXE/configure.pm):

1. setting BSW environment (LOADGPS.setvar)
2. creation of campaign area ($P)
3. creation of user working area ($U)
4. creation of temporary area ($T)
5. install latest online BSW updates
6. successful compilation of all executables in $XG
7. successful compilation of the menu system in $XQ
8. running BSW example campaign for the proper installation testing

A good knowledge of the BSW is important for the maintenance and possible further enhancements. An interactively step-by-step GNSS processing according to BSW documentation and provided examples is recommended prior developing automated BSW applications.

Local data centre/repository

This section describes requirements for the preparation of input data pool (local data center) and its interface for supporting the GOP routine processing system. It should be prepared by the local staff, since it typically represents a need for a continuous maintenance. Two main motivations for separating the local repository from the GNSS processing module are: a) avoiding any internet-dependent problems in data processing, b) avoiding any duplication in data/product download when more variants of the data processing are maintained in parallel.

For a reliable routine NRT ZTD production, which is requested by E-GVAP (http://egvap.dmi.dk), timely and stable input data (and meta-data) are necessary together with mandatory precise products and models. Only a self-repairing, effective and robust procedure for data downloading or mirroring can guarantee a robust NRT ZTD estimation with minimalistic manual interventions. It is thus recommended to invest a reasonable time for a development in this area. The download module of the Trop-NET pack was designed for this goal and may be flexibly adapted for any specific tasks.

3.1 Data and meta-data

For near real-time GNSS analysis, hourly RINEX observation files from the network of local/national GNSS stations need to be at the local data centre (data pool) with a minimum latency (optimally a few, but up to 10-15 minutes for the majority of the stations). For a reliable processing, additional core (fiducial and remote) stations are needed in order to realize reference frame and to estimate absolute values of tropospheric parameters. About 10-15 such core stations, distributed in a distance of 500-1500km from the centre of the network, are recommended. Most of the core stations should be part of the IGS/EUREF network in order to guarantee a stable geodetic datum for both coordinate and tropospheric solutions.

The RINEX observation data could be stored locally in RINEX or CRINEX (Hatanaka compression) formats and, optimally, further compressed by *gzip/compress* tools. The data pool structure is recommended to follow a standard way of data archiving in time-specific directories (e.g. *nrtdata/<YEAR>/<DOY>/<HR>/*), rather than organized station by station. This facilitates its usage as well as maintenance (e.g. removal of historical files etc.). If data pool is used only to feed the data processing (i.e. not used as a long-term repository), it is recommended to have a background process that regularly cleans up the data after a specified period and thus keeps a constant use of the disk space over time.

Station meta-data are very important for the GNSS data processing as an independent source of information. For all stations, so called site-log files (at least minimalistic ones) are requested for a keeping a full history of changes at each station. The GOP processing system uses independent meta-data for cross-checking the RINEX headers. In the case when a significant discrepancy occurs, the station is automatically excluded from the processing and can be included only after a manual intervention (safe mode). For preparing the standard EUREF and IGS site-log files, it is recommended to use the EPN CB web-interface as provided at the URL http://epncb.oma.be/_networkdata/sitelogsubmission/index.php (using option without the registration and without log-file submission, see the right hand side of the web). Up-to-date site-log files should be stored also at the data pool in a single directory (e.g. *site_log*).

3.2 Precise models and products

Besides hourly RINEX observations, additional precise products and models are needed in regular updates:

- precise ultra-rapid orbit and ERP products, SP3c & ERP format (optimally IGS products from any IGS data centre: CDDIS, BKG, IGN, see http://www.igs.org) or, alternatively, CODE products (ftp://ftp.unibe.ch/aiub/CODE/)
- merged GNSS navigation messages, e.g. mirror from the IGS global data center ftp://cddis.gsfc.nasa.gov/gnss/data/hourly/<YEAR>/<DOY>/hour<DOY>0/<YR>n.Z or from other source (BKG, IGN, GOP data centres)
- antenna phase centre offset and variation models, ATX format (e.g. mirror of EPNCB ftp://epncb.oma.be/pub/station/general/)
- BSW supporting files – satellite crux-file (SAT_YEAR.CRX), receiver descriptions (RECEIVER.), differential code biases (DCB), GPS-UTC (GPSUTC.), but optimally as a mirror of the BSW ftp ftp://ftp.unibe.ch/aiub/BSWUSER52/GEN/)

All these should be mirrored into the local data centre (data pool) with well defined directory structure (e.g. orbits/<gpsweek>, brdc/, BSWUSER52/, EPNCB/).

Installation of Trop-NET basic modules

An easy system installation and update is maintained at GOP using the Subversion (SVN) – a version control system. The Subversion manages files and directories, and the changes made to them, over time. This allows a group of users to install, recover, update and synchronize from any software version in central repository (hosted by GOP).

4.1 Installation from GOP repository

To enable an authenticated access to the GOP repository, you need to 1) generate a unique secure-shell (SSH) key, 2) sent a public part of the key to GOP and wait for the activation, 3) check out the Trop-NET pack from the repository and 4) complete the installation according to instructions.

If you haven't got yet a secure key (*$HOME/.ssh/id_dsa.pub*), you can generate it using the following command:

 # ssh-key-gen -t dsa -N "your sentence"

The command creates two files in *$HOME/.ssh* directory: the private part of the key (*id_dsa*), which you should redistribute only for your purpose (e.g. more accounts), and the public part (id_dsa.pub) that you should provide to GOP for activating the access to the Trop-NET repository. You may safely send the public part via e-mail. After message from GOP, that your access was activated, you can download (checkout) from the repository the installation pack using these commands:

export SVN_SSH="ssh -i $HOME/.ssh/id_dsa"
svn checkout "svn+ssh://gnssmet@studna.pecny.cz/trunk" GNSSMET

If *GPSUSER* directory already exists in your, move it away (e.g. *cp -r GPSUSER GPSUSER.BSWxx*) and create a symbolic link to *GNSSMET* in your home directory:

cd $HOME
ln -s GNSSMET GPSUSER

At this step, you have got a basic installation of three Trop-NET modules - download, upload and GNSS processing using BSW (V50/V52) + BPE. Next step will consist of the setup of the environment and its integration with the standard Bernese GNSS software environment.

4.2 Setup environment

The variable *$C* points to the BSW root directory if you have properly installed the Bernese GNSS software. If the original file *LOADGPS.setvar* still remains in *$X/EXE*, create a symbolic link (or a copy) at your home directory with the new name *LOADGPS_Vxx* (*xx* is the Bernese version, e.g. 50) using the command:

ln -s $X/EXE/LOADGPS.setvar $HOME/LOADGPS_Vxx

Then edit *LOADGPS_Vxx* file by adding following lines (compare with example *$HOME/GPSUSER/RC/LOADGPS*):

 # add local paths
 # =========
addtopath "$HOME/bin"

addtopath "$U/bin"

 # Default CAMPAIGN
 # ===========
export DEFAULT_CAMPAIGN="MET_HOUR"
. $U/bin/camp_path "$DEFAULT_CAMPAIGN"

The default processing campaign is predefined as *MET_HOUR* and this campaign should be setup in *$X/PAN/MENU_CMP.INP* (manually or via Bernese MENU)

! List of Campaigns
! ----------------
CAMPAIGN 2
"${P}/EXAMPLE"
"${P}/MET_HOUR"
widget = uniline

For the campaign *MET_HOUR* the top directory and working subdirectory need to be created either using BSW MENU or manually, e.g.:

mkdir -p $P/MET_HOUR
cd $P/MET_HOUR
mkdir ATM BPE CRD DAT ERR INF OBS ORB ORX OUT PLT RAW RUN SOL STA TAB

Finally, modify your user-account environment with additional paths and aliases. It is recommended to copy file from the installation area to its standard name and location at *$HOME* so it is used to initialize environment immediately after the login:

cp $HOME/GPSUSER/RC/BASH_ALIASES $HOME/.bash_aliases

Check that at the user account file *.profile* exists in home directory and it includes *.bashrc* file. If it is disabled, uncomment it. Then check that also the *.bashrc* contains optional inclusion of the user account localization using the file *.bash_aliases* as follows

If [-f ~/.bash_aliases]
then . ~/.bash_aliases
fi

You may anytime accommodate the *.bash_aliases* file with your preferable setting, however, check that at least these library path are correctly setup

export PATH=$PATH:$HOME/GPSUSER/bin # GOP tools
export PATH=$PATH:$HOME/GPSUSER/DC # download
export PATH=$PATH:$HOME/GPSUSER/TEQC # TEQC
export PATH=$PATH:$HOME/GPSUSER/RNXCMP # RNXCMP

The above setting guarantees that all important programs are implicitly found in the environment. The location of *teqc* tool should either be physically in one of these directories (*$U/TEQC* recommended) or, alternatively, you add your path to a specific location or you create a symbolic link to the location:

cd $HOME/GPSUSER/TEQC/
ln -s <absolute-path-to-teqc-executable> teqc

The Hatanaka RINEX compression tools are available from the repository (original source (http://terras.gsi.go.jp/ja/crx2rnx.html) and can be compiled as follows:

cd $HOME/GPSUSER/RNXCMP
./make # compile CRX2RNX and RNX2CRX

At this stage, close all terminals opened from the server (or ask for changing user at your X-windows) and do a first login in order to check that your environment was setup correctly. If there is no warning message after the opening of new terminal (or shell) and the following environment variables are reasonably defined, e.g.

echo $U → displays your GNSSMET directory (=GPSUSER)
echo $C → displays Bernese root directory
echo $T → displays Bernese temporary directory
echo $P → displays Bernese campaign directory etc..

Then you have most likely setup environment successfully. If there is any warning/error reported, try to fix it following the message. In the case of success, continue with creating several remaining mandatory links in the Trop-NET environment:

cd $U
ln –s OPT.Vxx OPT
cd OPT/
./link_GEN MET

The last command updates all links to general MENU files (*MET_GEN/MENU_**) in all other subdirectory thus keeping them fully consistent with the original one (*MET_GEN*).

4.3 Complete the installation

In order to complete the installation several additional actions are recommended.

1) Setup a mail server machine to enable sending e-mails to Internet by linux *mail* command. In order to support redistribution of warning messages from the processing via e-mail, list one or more e-mail addresses in the file *$HOME/.forward*. Check if it works via the following command (direct mail and redistributed mail):

echo "direct mail test" | mail –s "subject: testing email" <e-mail address>
echo "redistributed test" | mail –s "subject: testing email" $(whoami)@localhost

2) It is recommended to create several root directories at user *$HOME*, where various data will be located from the download module

REPO → local repository home (can also be link to a remote location via NFS)
DC.cfg → settings files (see download configuration)
DC.log → logging files (log, get, wget, conv, warn, err)
DC.garb → collection of garbage files (not successfully converted)
DC.cache → caching directory (to help in effective automated source identification)
DC.db → database of successful downloads (line per download)

4.4 Automated scheduling

Individual modules and their steps can be started manually or automatically. Interactive starts of individual modules are recommended for initial system testing system, for understanding all individual steps or to identify any potential problem after first installation.

Scheduling start of modules is provided via the Linux *cron* daemon. First, *cron* table need to be prepared according to the *crontab* format. Second, the *cron* table (file) need to be activated by this command

crontab <crontable-file>

Anytime, active *cron* commands can be de-activated or simply listed from the command line as follows

crontab –r # command list de-activation

crontab –l # show active command list

Finally, following table shows the standard *crontab* schedule for the three Trop-NET modules.

```
SHELL=/bin/bash
USER=gnss_ac
PATH=.:/home/gnss_ac/bin:/home/gnss_ac/GPSUSER/bin:/home/gnss_ac/GPSUSER/DC/:/home/gnss_ac/GPSUSER/TEQC:/home/gnss_ac/GPSUSER/
RNXCMP:/bin:/usr/bin:/usr/bin/X11:/:

# MIN HOUR DAY MON DOW COMMAND
# -----------------------------
# data, product and model downloads
# -----------------------------
*/5 * * * * DC.bin/get_RINEX-6h.sh      > $HOME/CRON/cron-get-rinex.log     2>> $HOME/CRON/cron-get-rinex.err
*   * * * * DC.bin/conv_RINEX_H.sh      > $HOME/CRON/cron-conv-rinex.log   2>> $HOME/CRON/cron-conv-rinex.err
50 * * * * DC.bin/get_MIRROR.sh         > $HOME/CRON/cron-mirror.log        2>> $HOME/CRON/cron-mirror.err
59 * * * * DC.bin/get_SP3_daily.sh      > $HOME/CRON/cron-get-orbits.log    2>> $HOME/CRON/cron-get-orbits.err
# -----------------------------
# list exctractions
# -----------------------------
58 * * * * GPSUSER/bin/extr_SITE --XtrF REPO/sit_log/list_IGS    "REPO/sit_log/IGS/*log"   > $HOME/CRON/list-igs.log 2>&1 &
58 * * * * GPSUSER/bin/extr_SITE --XtrF REPO/sit_log/list_EUR   "REPO/sit_log/EUR/*log" > $HOME/CRON/list-eur.log 2>&1 &

# -----------------------------
# BPE processing
# -----------------------------
20   * * * *   . $HOME/.profile ; GPSUSER/WORK/met_hour.pl -tod    > $HOME/CRON/cron-bpe.log   2> $HOME/CRON/cron-bpe.err
# -----------------------------
# product upload
# -----------------------------
* * * * * lim_exec.pl -lim 45 "Upload_Table.pl --tab $HOME/GPSUSER/UPLOAD/remote_export.KTU --dir $HOME/GPSUSER/UPLOAD/" >>
$HOME/CRON/cron-upload.log 2>&1
# -----------------------------
# data cleaning
# -----------------------------
# 55 * * * *   clean_DIR "/home/gnss/REPO/orbits/*" 2 show
# 55 * * * *   clean_DIR "/home/gnss/REPO/nrtdata/*/*/*" 30 show
```

At the table, the first five columns represent setting mask for MINUTES, HOURS, DAY OF WEEK, MONTH and DAY OF WEEK followed by the scheduled command with arguments. The mask can be number, list of numbers separated by commas, '*' for any value, */5 for every 5 minutes etc (see man page for *crontab*). At the beginning of the table, some user environment can be setup for the shell environment started by *cron* daemon which, by default, uses very restricted setup e.g. for program path). At the second part, individual download processes are scheduled a) hourly RINEX data download at every fifth minute, b) parallel conversion of the data download, c) mirror of precise models and d) download of precise orbits and Earth Rotation parameters. Additional commands updates every hour a summary extraction (used for station configuration, see BPE processing) from *site-logs* for all EUREF and IGS stations.

Third part represents the schedule of the Bernese data processing (every hour at HH:20). Resulting troposphere products are searched at *$U/UPLOAD* directory every minute, i.e. if any file exists there, the upload procedure is started and after successful finish it is deleted. The last commands in the *crontab* table represent cleaning procedures for various locations.

Module for data and product downloads

Due to the different character of various download procedures (e.g. update rate, mirroring), specific downloading tasks are usually operated individually. The perl library supporting all such operations is located in *$U/DC* directory and the main program is called *data_center.pl*. The background software utilized is the *wget* program available as a standard Linux packages in most systems (can be also installed separately).

Individual configuration can support job related to data and product download or precise model mirroring using individual configurations.

5.1 Configuration

The script is configurable through the file-based configuration and via option-based configurations (command line arguments of the *data_center.pl* program).

5.1.1 File-based configurations

The file-based configurations is split into two types of files

- data file definitions (.dat)
- source definitions (.src)

Both have similar column-wise structure, but different content. The fields (=columns) in a single record (=line) are separated by default with ':' (colon), which may be modified by setting the *Separator* variable at any single line within the file if the colon needs to be used in a valid value. All lines starting with '#' mark. First field (DATA ID or SOURCE ID) should be unique. Each field in any column, but the first one, may contain a multi-word when values are separated by a semicolon. When decoding, each line is checked for predefined time-specific variables `<year>`, `<yr>`, `<mn>`, *<doy>, <i>, <hr>, <gw>, <dw>* that are substituted with the working time (set as option-argument). Regular expressions are supported for file searching (*, ?, [], {}).

The data file defines DATA ID (1st column), data type (2nd column), file name (3rd column), SOURCE ID (4th column, later defined in Source definition), group name representing the first level directory of the program root (5th column), local relative path for group (6th column) and comment (last column). The example of IGS ultra-rapid orbit download is the following:

```
# NAME          ... arbitrary ID
# TYP[;subTYP]  ... data type (RNX, SP3+ERP [ci SP3, ERP], CRX, .., MIRROR, UNIQUE)
# MASK[;MASK]   ... filemask (using '*,[,],?'   +de-mask <doy>,<i>,<hr>,<gw>,<dw>, etc.)
# SRC[;SRC]     ... ordered source list
# GRP[;GRP]     ... group access: repository root located under "--out-directory=" (e.g. $HOME)
# LOCDIR        ... destination: sub-directory of repository root (GRP)
# ---------------------------------------------------------------------------
# NAME : TYP[;subTYP] : MASK[;MASK] : SRC[;SRC] : GRP[;GRP] : LOCDIR : comment
# ---------------------------------------------------------------------------
```

```
BRDN:  GPS,NAV      : brdc<doy>0.<yr>n.Z    : CDDIS-BRDN        : REPO : orbits/brdc/<year>      :
BRDG:  GLO,NAV      : brdc<doy>0.<yr>g.Z    : CDDIS-BRDG        : REPO : orbits/brdc/<year>      :

# IGS ultra-rapid orbits & ERPs (masked separately to 8 lines due to a local search!)
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_00.sp3*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_00.erp*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_06.sp3*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_06.erp*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_12.sp3*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_12.erp*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_18.sp3*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
IGU:  SP3+ERP;ULTRA : igu<gw><dw>_18.erp*   : IGSCB;CDDIS;BKG   : REPO : orbits/<gw>u            :
```

Note, that IGS ultra-rapid orbits (SP3) and Earth rotation parameters (ERP) are defined in eight records in order to guarantee all files will be downloaded for every day. The reason is that the program first check local file (mask) availability and starting the download request only in case that the file (mask) succeeds in local search, it may happen that other files will not be downloaded. Each line thus represents a single file guaranteed to be downloaded.

Setting more SOURCE IDs enables source automated selection for the download in sub-sequent runs. Although the order gives an initial priority for the source, the selection uses results cached from several previous downloads for the source and data mask. Nevertheless, it is not recommended to use more than 2-3 sources for the automated selection. The second example shows the setting of hourly RINEX file download from two sources in BKG for IGS and EUREF stations and applying file-mask with lower/upper cases:

```
# NAME          ... arbitrary (starting CDDIS?)
# TYP[;subTYP]  ... data type (RNX, SP3+ERP [ci SP3, ERP], CRX, .., MIRROR, UNIQUE)
# MASK[;MASK]   ... filemask (using '*,[,],?'   +de-mask <doy>,<i>,<hr>,<gw>,<dw>, etc.)
# SRC[;SRC]     ... ordered source list
# GRP[;GRP]     ... group access: repository root located under "--out-directory=" (e.g. $HOME)
# LOCDIR        ... destination: sub-directory of repository root (GRP)
# ---------------------------------------------------------------------------
# NAME : TYP[;subTYP] : MASK[;MASK] : SRC[;SRC] : GRP[;GRP] : LOCDIR : comment
# ---------------------------------------------------------------------------
 AUT1     : RNX;HOURLY : aut1*<doy><i>.<yr>[od]*;AUT1*<doy><I>.<yr>[OD]* : BKG_EUR : REPO : nrtdata/<yr><doy>/<I> :
 KTVL     : RNX;HOURLY : ktvl*<doy><i>.<yr>[od]*;KTVL*<doy><I>.<yr>[OD]* : BKG_EUR : REPO : nrtdata/<yr><doy>/<I> :

 ANKR     : RNX;HOURLY : ankr*<doy><i>.<yr>[od]*;ANKR*<doy><I>.<yr>[OD]* : BKG_IGS : REPO : nrtdata/<yr><doy>/<I> :
 BRST     : RNX;HOURLY : brst*<doy><i>.<yr>[od]*;BRST*<doy><I>.<yr>[OD]* : BKG_IGS : REPO : nrtdata/<yr><doy>/<I> :
```

Specific setting in the data file configuration may request a direct source mirroring. It is enabled using the keyword *MIRROR<filemask>* in the 3rd column instead of the file mask. Specific source hostnames may be also directly defined in the 4th column, i.e. replacing the SOURCE ID. Example file is to mirror supporting metadata, products or models (not the change of the separator):

```
# ---------------------------------------------------------------------------
# NAME : TYP[;subTYP] : MASK[;MASK] : SRC[;SRC] : GRP[;GRP] : LOCDIR : comment
# ---------------------------------------------------------------------------
Separator = '!'

EPNCB   ! LOG ! MIRROR<*>        ! ftp://ftp.epncb.oma.be/epncb/station/general/  ! REPO ! epncb/general !
SITLOG1! LOG ! MIRROR<*>        ! ftp://ftp.epncb.oma.be/epncb/station/log/      ! REPO ! sit_log/EUR !
SITLOG2! LOG ! MIRROR<*>        ! ftp://igscb.jpl.nasa.gov/igscb/station/log/    ! REPO ! sit_log/IGS !

DCB0! BSW ! MIRROR<[CP]?[PC]?????.DCB.Z> ! ftp://ftp.unibe.ch/aiub/CODE/<year>    ! REPO ! bswuser/DCB/<year> !
DCB1! BSW ! MIRROR<CODE.DCB.Z>           ! ftp://ftp.unibe.ch/aiub/CODE/<year>    ! REPO ! bswuser/DCB/<year> !
DCB2! BSW ! MIRROR<CODEFULL.DCB.Z>       ! ftp://ftp.unibe.ch/aiub/CODE/<year>    ! REPO ! bswuser/DCB/<year> !

BSWUSER1! BSW50 ! MIRROR<*>              ! ftp://ftp.unibe.ch/aiub/BSWUSER50/GEN ! REPO ! bswuser50/GEN !
BSWUSER2! BSW50 ! MIRROR<*>              ! ftp://ftp.unibe.ch/aiub/BSWUSER50/STA ! REPO ! bswuser50/STA !
BSWUSER3! BSW50 ! MIRROR<*DCB>           ! ftp://ftp.unibe.ch/aiub/BSWUSER50/ORB ! REPO ! bswuser50/DCB !

BSWUSER1! BSW52 ! MIRROR<*>              ! ftp://ftp.unibe.ch/aiub/BSWUSER52/GEN ! REPO ! bswuser52/GEN !
BSWUSER2! BSW52 ! MIRROR<*>              ! ftp://ftp.unibe.ch/aiub/BSWUSER52/STA ! REPO ! bswuser52/STA !
BSWUSER3! BSW52 ! MIRROR<*DCB>           ! ftp://ftp.unibe.ch/aiub/BSWUSER52/ORB ! REPO ! bswuser52/DCB !
```

Finally, the source file defines SOURCE ID (1st column), data type (2nd column), source hostname (3rd column), remote directory (4th column), and comment (last column). The example of IGS ultra-rapid orbit download is the following:

```
# NAME          ... arbitrary ID
# TYP[;subTYP] ... data type (RNX, SP3+ERP [ci SP3, ERP], CRX, .., MIRROR, UNIQUE)
# IP[;HOST]     ... ip address, voluntary HOST address
# DIR           ... remote directory (+ de-mask <sess>,<id>,..etc.)
# -----------------------------------------------------------------------
# NAME : TYP[;subTYP] : IP[;HOST] : DIR : comment
# -----------------------------------------------------------------------

CDDIS: SP3+ERP        : cddis.gsfc.nasa.gov    : /gps/products/<gw> :
CDDIS-BRDN: BRDN      : cddis.gsfc.nasa.gov    : /gps/data/daily/<year>/<doy>/<yr>n :
CDDIS-BRDG: BRDG      : cddis.gsfc.nasa.gov    : /gps/data/daily/<year>/<doy>/<yr>g :

IGSCB: SP3+ERP        : igscb.jpl.nasa.gov     : /pub/product/<gw> :
BKG:   SP3+ERP;ULTRA  : igs.bkg.bund.de        : IGS/products/orbits/<gw> :
```

5.1.2 Option-based configurations

The command-line options are useful for setting specific download requests that may be scheduled for various periodicities using different working period, data directory e.g. for different data types or actions.

The period definition is defined with the reference epoch *(--last-epoch*) and back interval from the reference epoch *(--back-time*). File-based configurations are then introduced with options *--dat-list* and *--src_list*. Various working directories may be defined for the program root (*--out-dir*), input area for files to be converted (*--inp-dir*), garbage area (*--grb-dir*), cache area (*--cache-dir*), file mask or area for log files (*--log-file*) and area for download database files (*--db-file*). Limit for the operation may be set with *--max-time* option. Various program actions may be activated via setting three options: *--check-data*, *--get-data* and *--convert-data*.

For the use of command line options it is practical to prepare starting shell scripts corresponding to specific download tasks such as shown in the first part of the basic *crontab* table (see automated scheduling). Examples of such basic operations in support of near real-time processing are provided in *$U/DC/bin* directory. Additionally, we recommend to use independent scripts for downloads if manually start the command, e.g. for testing, from those started via the *cron* deamon scheduling. The command line example of the program start may states as

```
./data_center.pl        --last-epoch="-t -h -1"                    \
                        --back-time="6h"                           \
                        --check-data                               \
                        --get-data                                 \
                        --noconvert-data                           \
                        --dat-list="$HOME/DC.set/RINEX_H.fil"      \
                        --src-list="$HOME/DC.set/RINEX_H.src "     \
                        --out-dir="$HOME/"                         \
                        --inp-dir="$HOME/DC.local/hourly/"         \
                        --grb-dir="$HOME/DC.garb/hourly/"          \
                        --cache-dir="HOME/DC.garb/hourly/"         \
                        --log-file="$HOME/DC.log/hourly/"          \
                        --db-file="$HOME/DC.db/"                   \
                        --max-time=30 --src-iter --verb 3
```

5.2 Repository maintenance

= To complete = Update, cleaning, monitoring

Module for GNSS data processing

6.1 Processing configuration

The processing module of the Trop-NET consists of several configuration and definition parts which will be described in following subsections

- Source definition of data, products and model
- Station in processing network
- Control variable in the Processing Control File (PCF) of the Bernese Processing Engine (BPE)
- Configuration of input panels for Bernese programs
- Update of PCV model

6.1.1 Data, products, model source configuration

Two files provide settings of data (other external information) and products sources - *$U/PCF/MET_dat* and *$U/PCF/MET_orb*, respectively. While the first column represents a unique keyword, all other columns represent setting values. The latter can consist of environment variables (e.g. *$HOME*) or the following specific time keywords that are substituted in runtime: <year>, <yr>, <mm>, <dd>, <mn>, <hr>, <mi>, <sc>, <sess>, <doy>, <gpsw>, <dow>. Comments are all lines starting with '#' character.

Standard example of the file *$U/PCF/MET_dat* is following

```
#LDC center source table    source/dir
#========================   ==========================================================
 REPO                       $HOME/REPO/nrtdata/<yr><doy>/<ID>
 GREECE                     $HOME/REPO/greece/<yr><doy>/<ID>
#
 DCBYRMN                    $HOME/REPO/bswuser/DCB/<year>/????<yr><mm>.DCB.Z
 DCBMASK                    $HOME/REPO/bswuser52/DCB/CODE.DCB
 SATCRUX                    $HOME/REPO/bswuser52/GEN/SAT_<year>.CRX
```

The first part describes two data locations and the second part three files with supporting external information – two Bernese DCB files (monthly and cumulated) and satellite information Bernese CRX file. The keywords are used in PCV scripts or network configuration as we will see later.

Example of *$U/PCF/MET_orb* settings for precise orbit and Earth Rotation Parameters (ERP) is following

```
#Id Remote orbit file name     Remote ERP file name       Remote directory
#== ========================== =========================  ==========================================
#.. IGS orbit products
#== ========================== =========================  ==========================================
#IF igs<gpsw><dow>.sp3.Z       igs<gpsw>7.erp.Z           $HOME/REPO/orbits/<gpsw>
#IR igr<gpsw><dow>.sp3.Z       igr<gpsw><dow>.erp.Z       $HOME/REPO/orbits/<gpsw>

 IU igu<gpsw><dow>_<hr>.sp3.Z  igu<gpsw><dow>_<hr>.erp.Z  $HOME/REPO/orbits/<gpsw>u
#IV igv<gpsw><dow>_<hr>.sp3.Z  igu<gpsw><dow>_<hr>.erp.Z  $HOME/REPO/orbits/<gpsw>u
#CO cou<yr><doy>_<hr>.sp3.Z    cou<yr><doy>_<hr>.sp3.Z    $HOME/REPO/orbits/<gpsw>u
#== ========================== =========================  ==========================================
#.. Broadcast orbits
#== ========================== =========================  ==========================================
 BG brdc<doy>0.<yr>n.Z         none                       $HOME/REPO/orbits/brdc/<year>
 BR brdc<doy>0.<yr>g.Z         none                       $HOME/REPO/orbits/brdc/<year>
```

The 2-char keyword (*ID*) at the beginning of each setting line represents a unique identification of a specific orbit and ERP product referenced in PCF file and scripts. Next three columns are mask that are substituted in runtime and represents remote orbit and ERP file name and directory. The Bernese files are then implicitly constructed in the campaign directory as follows: *$ORB/ID<yr><sess>.EXT*.

6.1.2 Network station configuration

The configuration of all processing stations in *<XXX>* analysis centre is primarily via *$U/STA/LIST_<XXX>.SIT*. The file contains these columns - 1) cluster number (0-9), 2) SITE name and DOMES, 3) start time (YEAR SESS), 4) end time (YEAR SESS), and 5) comma-separated list of repository sources (see Data configuration). If DOMES is not available, a non-blank string should be used instead, e.g. *XXXXXXXXX*. Any line starting with *'#'* is understood as a comment. The example configuration is below:

```
# SITE DOMES        from:      to:        data-sources                comments
#                   YEAR SESS  YEAR SESS
# ==== ==========   ==== ====  ==== ====   *************************** ****************
# core stations
# ==============
1 ANKR 20805M002    2001 0750  2019 0010   REPO
0 ARGI 10117M002    2001 0750  2019 0010   REPO
1 AUT1 12619M002    2001 0750  2019 0010   REPO,GREECE
0 BRUS 13101M004    2001 0750  2019 0010   REPO
1 BUCU 11401M001    2001 0750  2019 0010   REPO
1 GLSV 12356M001    2001 0750  2019 0010   REPO
```

If two or more data sources are defined, a searching loop overall all items (ordered) the list is performed until relevant data are found or end of the list. This network configuration file is used in various BPE scripts and to be applied in BPE processing it should be correctly defined in global variable *"V_STLIST"* in the PCF-file *$U/PCF/MET_HOUR.PCF*. The configuration of clusters can be checked by a visual inspection of network plot generated in *$U/STA* directory with one of the following commands (differs only in predefined scope areas in *$U/bin/ac_Map.gmt* plotting script)

```
#./00PLOT <SIT-file> <region> <site-log extraction-mask in REPO>
 ./00PLOT  GREECE.SIT MET_EU  "$HOME/REPO/sit_log/list*"
 ./00PLOT  GREECE.SIT MET_GR  "$HOME/REPO/sit_log/list*"
```

The first command produces a European figure while the second was configured with focus on Greece. The network configuration file (*$U/STA/LIST_<XXX>.SIT*) can be effectively used also for generating Bernese site information file (*STA-file)* using the following command in *$U/STA* (see shell script 00LIST)

```
#./list2sta.pl <SIT-file>  <STA-file> <log extraction-mask in REPO>
 ./list2sta.pl LIST_XXX.SIT NEW.STA    "$HOME/REPO/sit_log/list*"
```

For all active stations in the configuration file, the *list2STA.pl* perl script searches all important metadata from the summary list(s) in the local data repository as extracted from *sit-log* files. Although STA-file could be automatically updated, it is not recommended and to do always any change manually. It is strongly recommended in interactive mode and to avoid overwriting of existing STA-file and check all changes first by *diff* linux command as follows:

```
diff NEW.STA  LIST_XXX.STA
```

If any instrumentation change occurs at any station and is already reported in the RINEX header, the site is excluded from the processing and warning e-mails are sent regularly until STA-file is updated. Any update in *STA-file* should be also cross-checked with existence of new included receiver information in Bernese REC-file (e.g. *$X/GEN/RECEIVER.*) and antenna-dome pair in Bernese PCV-file (e.g. *$X/GEN/I08_1768.PHG*).

6.1.3 Processing control file configuration

Processing Control File (PCF) of the Bernese Processing Engine consists of the definition of all processing step including definition of their dependencies and directories with input panels (1st section in PCF), optional parameters and possible jumps or parallelization in processing steps (2nd section in PCF) and global variables (3rd section in PCF). In this section, global variables that control principal processing settings will be introduced only. Detailed description of all PCF processing steps will be provided in following sections.

```
VARIABLE DESCRIPTION                              DEFAULT
8******* 40************************************** 16**************
V_BSWVER Bernese GNSS Software version            52                      # 50 or 52
V_SATSYS Satellite systems                        GPS                     # GPS, ALL, GLONASS
V_CMP    3-char campaign identification           MET
V_ORB    ID for precise orbits                    IU
V_BRD    ID for broadcast orbits                  BG
V_HINT   Processing window [h]                    4                       # [h]
V_STLIST Station list file (+ext)                 LIST_AUT                # name.SIT
V_INFSTA Station information file name            LIST_AUT                # name.STA
V_INFBLQ Ocean loading correction file name       FESX_AUT                # name.BLQ
V_INFPCV PCV-INFO file (+ext)                     I08_1808.PHG            # full name
V_INFSAT Satellite information file               SATELLIT.I08            # full name
V_INFREC Receiver information file                RECEIVER.               # full name
#
V_APRFIX A priori FIX file                        METEO                   # name.FIX
V_REFFIX Reference FIX file                       RF$Y$S+0                # name.FIX
V_REFCRD A priori CRD file                        IGS_$M$Y                # name.CRD
#
V_RNXCRX Rinex/Site-Crux file                     METEO                   # name.CRX e.g. MET_$Y+0
V_SATCRX Sat-Crux file                            MET_$Y+0                # name.CRX e.g. METEO
#
V_EGMMOD Earth gravity model                      EGM2008_SMALL.          # full name e.g. JGM3.
V_SUBMOD Subdaily pole model                      IERS2010XY              # full name e.g. IERS2000
V_NUTMOD Nutation model                           IAU2000R06              # full name e.g. IAU2000
V_TIDMOD Ocean tide model                         OT_FES2004              # full name e.g. OT_CSRC.TID
V_TPOMOD Solid Earth tide model                   TIDE2000                # full name e.g. TIDE2000
V_JPLEPH JPL planetary ephemerides                DE405                   # full name e.g. DE200
V_GMFVMF Global or Vienna mapping function        GMF                     # GMF or VMF
```

Global BPE variables are defined in the 3rd section of the PCF file and it can be used in any PCF script using *$bpe->{NAME}* or in any MENU panel using *$(NAME)* where *NAME* is a variable without prefix "*V_*". The Bernese version can be explicitly defined in *B_BSWVER* variable (implicitly used Bernese V5.0). *V_SATSYS* pre-define global use of satellites system (GPS, ALL, GLONASS etc.). Processing interval [in hours] of the original data batch is defined in *V_HINT* variable and the campaign 3-char abbreviation in *V_CMP*. Precise and broadcast orbit *ID* selection is defined in *V_ORB* and *V_BRD*, respectively. Station list for the network definition is setup in *V_STLIST* variable. Several *V_INFxxx* variables and *V_xxxMOD* define file names of various models or supporting information files. A priori list of fiducial stations, actual list and reference coordinate file is defined in *V_APRFIX*, *V_REFFIX* and *V_REFCRD* variables, respectively. A priori satellite and station Bernese CRUX-files is defined in *V_RNXCRX* and *V_SATCRX*.

6.1.4 Bernese input panels

Configuration of Bernese menu panels are located in $U/OPT directory and controls all Bernese programs started from PCF file. These may be setup via Bernese menu or manually via a text editor (recommended only to advanced Bernese users). In the Trop-NET system, panels are used to pre-configure specific settings while all common settings are optimally setup via global variables such as e.g. GNSS system, elevation cut-off angle, precise orbit selection, batch processing, common precise models and supporting information files. Some of these common configurations are automatically set via Bernese *putkey* command in PCF scripts, while others (e.g. models) are simply defined in panels using BPE global variables (see PCF configuration).

Automated setting was designed in support of a simple control over the most important options so that a full consistency of corresponding settings is guaranteed across the BPE processing. Additionally, it supported a flexible design of more general PCF scripts based on a minimalistic default panels in $U/OPT directories. Different variants of individual steps can be simply provided with particular PCF parameters as provided in 2nd section of PCF file.

6.1.5 PCV model update

Model for antenna phase centre offset and variations (PCV) need to be updated mainly if new receiver antenna occurs in processing network or if a GNSS satellite constellation is modified. First, new PCV model has to be downloaded in the exchange format (ATX), second it needs to be converted to Bernese PHG-file and, third, updated file needs to be reconfigured in the processing. The first and second step can be done manually or using shell script in *$U/PCV* and command

```
#./00PCV <REPO-path>
 ./00PCV "$HOME/REPO/"LIST_XXX.STA
```

The first argument consists of the full path to the local repository (data centre) that includes subdirectory bswuser52 (or bswuser50) where the original BSW support ftp server from AIUB is mirrored. If it is not locally available, default path to IGS/EUREF distribution is downloaded. Second argument is the reference system identification with which the PCV model should be consistent (e.g. I05, I08, E05, E08). Third argument is the campaign name and the fourth is the active Bernese STA-file. Using the last argument, user can optionally add an individual ATX models (in a single file) that should be added to the official IGS or EUREF models.

After creating the new Bernese PHG-file, the user has to move it to the *$X/GEN/* directory and to update global variable *"V_INFPCV"* in *$U/PCF/MET_HOUR.PCF*.

6.2 NRT strategy for troposphere monitoring

Several aspects need to be particularly considered for near real-time GNSS troposphere estimates updated on hourly basis:

a) High efficiency of low latency GNSS processing.
b) Long-term receiver coordinate stability and their consistency with actual troposphere estimate.
c) System robustness with minimum manual interventions.

We will discuss how the Trop-NET system solves such needs.

6.2.1 Spatial and temporal solution stacking

Processing clusters together with normal equation stacking is very useful strategy to obtain an efficient solution in near real-time (Dousa, 2001). However, in order to support a global solution and more reliable ambiguity resolution for long baselines, the processing batch was extended to the 4-hour session (Dousa 2013). The lowest processing level, which includes raw data, consists of the redundancy of 3 hours data pre-processing as it is showed in Figure 3.

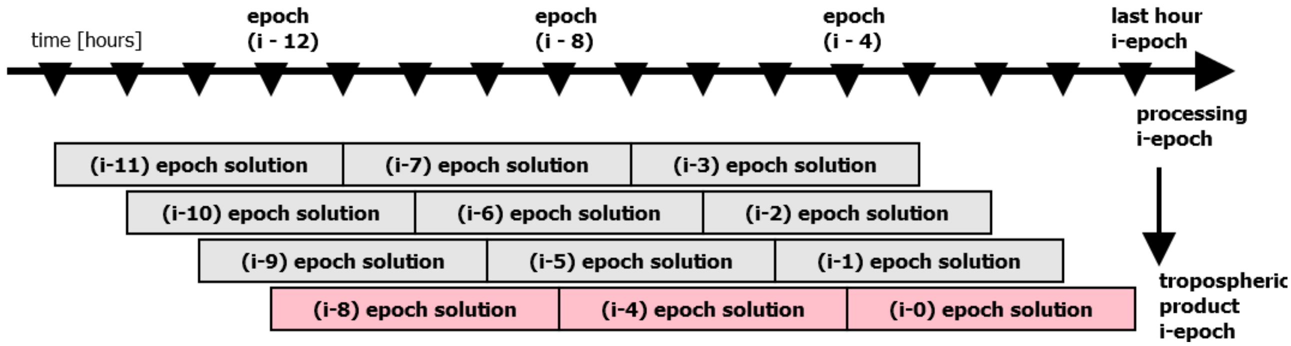

Figure 2: Trop-NET processing redundancy

Three basic levels of the Trop-NET processing stacking is shown Figure 3. These are described as follows

- Processing of network cluster using 4 hours of data batch
- Stacking in spatial domain to generate a single-session solution for a whole network
- Stacking in temporal domain to provide a multi-session solution based on previous normal equations.

Sub-network solutions for all defined clusters (see Section Processing clusters and parallelization) are intermediately created using the GPSEST Bernese program (specific *GPSxxx_P* PCF scripts). Such solutions of *X* clusters are temporarily saved in the form of normal equations and immediately stacked in spatial domain (generic *ADD_CLU* PCF script). Resulting single-session normal equations are intermediate products saved for this session as well as for all sessions in the following 30 days. Using specific number of current and recent single-session normal equations, the temporal stacking (generic *ADD_SEQ* PCF script) is applied for various procedures of parameter estimations.

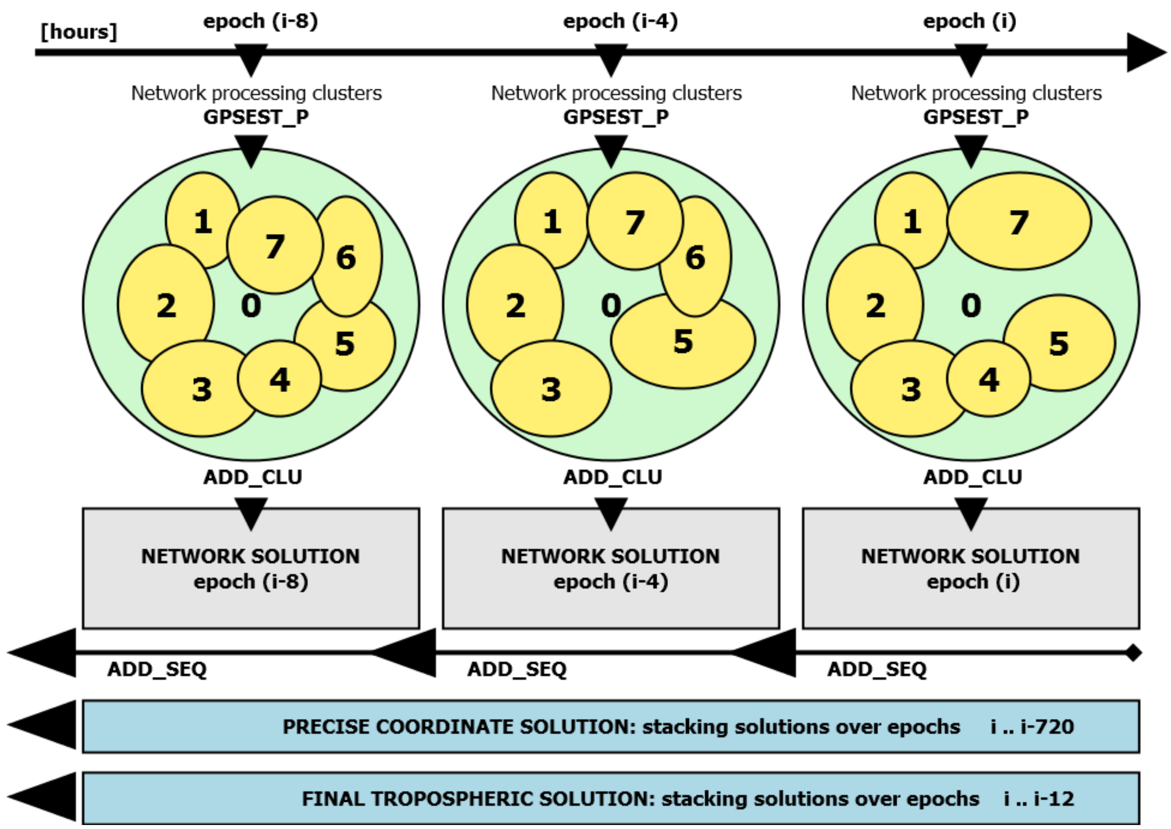

Figure 3: Trop-NET processing in network clusters, spatial and temporal stacking solutions

6.2.2 Estimated parameters

The following parameters still need to be estimated in the network double-difference solution:

- Tropospheric path delays
- Receiver coordinates
- Initial phase ambiguities

All parameters are estimated sequentially and iteratively within each 4-hour processing session. In such session, raw RINEX data are processed and precise models and satellite positions are introduced as known. First, initial coordinates and tropospheric path delays are estimated and reference datum setup using previous solution stacking. These parameters are introduced into the step of resolving integer phase ambiguities.

The coordinates are then estimated from a long time-span (up to 30 days) and with introducing all integer phase ambiguities. Remaining ambiguities are still estimated as float values. The coordinates can be also introduced from any external post-processing solution. In order to make the processing as much as simple and independent of such external process, it is preferred to estimate them during each hourly run. Other advantages are a) all models are implicitly guaranteed to be consistent between coordinates and troposphere, b) coordinates are automatically updated and do not depend on additional external information about the time changes and c) new station can be configured for a single process supporting an automatic product initialization for such site after reasonable time (in our case two days for initial coordinate estimates).

Finally, tropospheric parameters are estimated from a 12-hour using session of the normal equation combination in temporal domain. Currently, ambiguities are estimated simultaneously in this step since the stability of such tropospheric results is still a bit higher.

6.2.3 Processing clusters and parallelization

Different parallelization strategies are performed in various processing steps. Any step that requests a full correlation among all baselines utilizes the parallelization based on the definition of regional clusters. These are typically used for parameter modelling and for saving normal equations. All steps that allow an independent site or baselines processing use the parallelization based on automatically created groups of sites or files of requested size. These are for example in RINEX conversions, code smoothing, standard point positioning and ambiguity resolution (if inter-baseline correlations are neglected). For example clusters of up to 5-15 baselines are created for ambiguity resolution, while dimensions of other clusters are more defined by availability of CPUs (defined in $U/PAN/*CPU file and should not exceed number of physical CPUs).

The network processing clusters (i.e. regional clusters) are predefined in the station configuration file (.SIT), see the section network configuration. This configuration is used as a priori information only and when any cluster is too small, it is automatically merged with any other one. On the other hand, automatically generated size-specific clusters utilise several scenarios how the groups are created: a) sorting available zero-difference/single-difference files, b) stations from an actual list of available station or c) using cluster definitions from any previous step within the PCF file (to keep a consistency clusters).

6.2.4 Processing control and warnings

The processing itself is configured to provide warning and error reports via automated mailing to the localhost (*.forward* setting and mail server configuration is recommended, see section complete installing). The distribution may contain also an email to deliver SMS if the mobile provider supports it (and user allowed it). It is, however, recommended to use mailbox for the filtering only the error messages (i.e. those containing "*ERR"* keyword).

It is useful to see from the command line if the processing is actually running or not, and if the status of the last run. For this purpose there exists the command that can be started from anywhere

watch_log

A full archive of historical processing statuses (i.e. the success and run time) is available in *$AUTO/<YEAR>/<DOY>* directories. Using the linux *grep* command, one may easily search all successful or failed processing runs, respectively:

grep "Error: 0 " $AUTO/????/???/*
grep "Error: 1 " $AUTO/????/???/*

6.3 Bernese processing steps

= To complete =

6.3.1 Session preparation

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
  1 FILE_TAB              L_DAT:0
  2 FILE_TAB              L_STA:0
  3 FILE_TAB              L_ORB:0  L_ORB:1
#
  5 FILE_TAB              L_NEQ:0  L_NEQ:1  L_NEQ:2  L_NEQ:3  L_NEQ:4  L_NEQ:5   L_NEQ:6
  6 FILE_TAB              L_NEQ:7  L_NEQ:8  L_NEQ:9  L_NEQ:10 L_NEQ:11 L_NEQ:12  L_NEQ:13
  7 FILE_TAB              L_NEQ:14 L_NEQ:15 L_NEQ:16 L_NEQ:17 L_NEQ:18 L_NEQ:19  L_NEQ:20
  8 FILE_TAB              L_NEQ:21 L_NEQ:22 L_NEQ:23 L_NEQ:24 L_NEQ:25 L_NEQ:26  L_NEQ:27
```

6.3.2 A priori coordinates

= To complete =

```
12 APR_CRD                                R:EPN08  C:met
```

6.3.3 Precise orbits

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
#20 GEN_BRD                        O=BG+0   M=-17+5  OFFS=0   CONT&CLK UPD
#21 GEN_ORB               HOUR=XX  O=CO+0   M=-17+5  OFFS=0   CONT&CLK GET
 22 GEN_ORB               HOUR=XX  O=IU+0   M=-17+5  OFFS=0   CONT&CLK GET
```

6.3.4 Data decoding

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
 28 CRZRNXAP              _28_     HOURLY
 29 CRZRNX_P              PARALLEL _28_
 30 CHK_CRZ
 31 CCRINXAP              _31_     HOURLY
 32 CCRINX_P              PARALLEL _31_
 33 RXOBV3AP              _33_     SMOOTH   EPO:AUTO PERC:55
 34 RXOBV3_P              PARALLEL _33_
 35 RXOBV3AP              _35_     NON-SMT  EPO:AUTO PERC:55  CONTINUE
 36 RXOBV3_P              PARALLEL _35_
 38 MRG_CRD               RNX_???  $REFCRD
```

6.3.5 Data pre-processing

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
 70 CODSPPAP              _70_     STD:ORB  BRD:BRD  CLK:ORB  DCB:FILE
 71 CODSPP_P              PARALLEL _70_
 72 CHK_COD               NEXTJOB  070
#
# Baseline definition (OBS-MAX) in clusters
 75 SNGDIFAP              _75_     OBS-MAX
 76 SNGDIF_P              PARALLEL _75_
 77 ADD_BASL                       OBS-MAX
#
# Phase preprocessing in clusters
 80 MAUPRPAP              _80_     STD:ORB  ELV:7
 81 MAUPRP_P              PARALLEL _80_
 82 CHK_MAU               NEXTJOB  080
#
```

```
# Data editing in clusters + satellite orbit checking
 90 GPSEDTAP                _90_     STD:ORB  O:4R     S:0.001  ELV:7
 91 GPSEDT_P                PARALLEL _90_
 92 CHK_EDT                 NEXTJOB  090      075
```

6.3.6 Reference frame realization

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
 95 DATUMPRE              REF_FIX  STA_EXCL
 96 COMB_CLU                                I=4R     O=4R
 97 COMB_SEQ                                I=4R     O=R4     T=720:4
 98 CHK_REF               NEXTJOB  097
```

6.3.7 Ambiguity resolution

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
# Prepare preliminary TRP+CRD (to start and set up ambiguity resolution)
150 GPSESTAP              _150_    STD:ORB  C:R4     O:RR     ELV:7   X:GPS
151 GPSEST_P              PARALLEL _150_
152 COMB_CLU                                I=RR     O=RR
#
# L6+L3 ambiguity solution
171 GPSSIGAP              _171_    STD:ORB  LC:L6+L3 BAS:9999 I:RR     E:10     X:GPS
172 GPSSIG_P              PARALLEL _171_
173 CHK_SIG                                 LC:L6+L3
```

6.3.8 Coordinates estimates

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
# Coordinates amb-free/fix solution (24 days cummulated 360 x 4h_NEQs = 720h)
373 GPSESTAP              _173_    STD:ORB  A:FX     O:4W     W:-4+0   ELV:7
374 GPSEST_P              PARALLEL _173_
375 CHK_EST
376 COMB_CLU                                I=4W     O=4W
377 COMB_SEQ                                I=4W     O=W4     T=720:4
```

6.3.9 Troposphere estimates

= To complete =

```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
# Troposphere estimation for amb-free/fix
390 GPSTRPAP              _390_    STD:ORB  A:FX,FR  O:0X,0W  W:-4+0   C:W4
391 GPSTRP_P              PARALLEL _390_
392 CONTINUE
393 COMB_CLU                       C=W4     I=0W     O=0W
394 COMB_SEQ                       C=W4     I=0W     O=W0     T=12:4
395 COMB_CLU                       C=W4     I=0X     O=0X
396 COMB_SEQ                       C=W4     I=0X     O=X0     T=12:4
```

6.3.10 Product filtering and encoding

= To complete =

```
PID USER          PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
## Product generation, web monitoring update, session backup&cleaning
510 MET_EXPO              ACC:AUT  UP:W0    DIR:met  CO:W0,X0 XX:W0,X0 PL:W0,X0
511 MET_EXPO              ACC:AUT  UP:W0,X0 DIR:gop
512 MET_EXPO              ACC:AUT  UP:W0,X0 DIR:aut
513 MET_EXPO              ACC:AUT                    AR:W0,X0
```

6.4 Product archive & cleaning

= To complete =

```
PID USER          PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   PARAM8
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8*******
## Backup (and clean) current (and previous) sessions
920 FILE_TAB              BCKP:0   BCKP:2   BCKP:4   BCKP:6   BCKP:8   BCKP:10  BCKP:12
921 FILE_TAB                       D_TMP:2  D_TMP:4  D_TMP:6  D_TMP:8  D_TMP:10 D_TMP:12
922 FILE_TAB              D_NEQ:30 D_NEQ:40 D_NEQ:50          D_ORB:30 D_ORB:40 D_ORB:50
```

Module for product upload

The tropospheric product upload (or other files or products) is organized via a) specifying a local directory that is regularly searched by command from the *crontab* table (see example in automated scheduler installing) and b) configuration of local upload file mask to be searched in the directory, its name at the remote host (optional) and the remote path via URL settings which may contains also credentials for the secure upload. Both protocols FTP and SFTP are supported as given in the host URL. Similarly as in the download module, the upload also works with the time-specific substitution for searched file mask which may be limited for a specific time span.

The upload request is realized with *$U/bin/Upload_Table.pl* script regularly started from the *crontab* table (e.g. every minute). Internally, the upload uses the *lftp* sophisticated file transfer program. If one or more files relevant to the predefined mask in the configuration occur in the directory, a standard sequence of steps of the upload starts as follows:

- Connect to the host and change to the host directory
- Iterate over files to be uploaded
    - Upload the file to the remote directory with remote filename and prefix “_”
    - Move the uploaded file to the final remote name (remove the prefix)
    - Remove file from the local directory
- Leave the connection.

If any step fails, the program quit and the procedure may repeat next time until the local file disappears. The maximum time interval for upload is also set (and may be modified with command line option or in configuration file) to avoid a problem with frozen connections or with a multiple start (the *crontab* settings should reflect this limit too).

7.1 Upload configuration

The upload module is configured via file using three

```
# Allow identification of <yr> <year> <doy> <sess> <sesi> <i> <I> <hr> <mn> <dd> <wk> <dw>
# ===================================================================================
# file mask    remote-file    URL    e.g.
# cost_?_?_????????????_????????????_mult_gopx.dat.gz                  ---  ftp://<user>:<password>@<hostname>/<dir>
# cost_?_?_<year><mn><dd><hr><mn>_<year><mn><dd><hr>_mult_gopx.dat.gz  ---  ftp://<user>:<password>@<hostname>/<dir>

# Single file upload limit in [s] then the ftp process is killed
# ==============================================================
limit = 30

# GOP monitoring (anonymous FTP)
# ===============
 gop/cost*_????????*_mult_???1.dat.gz  ---  ftp://ftp.pecny.cz/incoming/COST_TRP
 gop/cost*_????????*_mult_???2.dat.gz  ---  ftp://ftp.pecny.cz/incoming/COST_TRP
```

Examples of maintenance

8.1 Add new station

Follow sections on download configuration, network settings and upload configuration. These steps should be performed (order is recommended):

- Configure station for the download in advance and check RINEX are regularly downloaded
- Add new station in SIT-file (network configuration) including a proper cluster identification
- Check if receiver, antenna and dome exists in *$X/GEN/RECEIVER* and *$X/GEN/I08_week.PHG*
- Add new stations to *$U/TRP/<XXX>_NRT* configuration file for the COST-716 format encoding (includes geoid height, country name)

8.2 New receiver or satellite

Follow the section PCV configuration. Check if files *$X/GEN/RECEIVER.* and *$X/GEN/SATELLITE.I08* are up-to-date, e.g. via comparing to *$REPO/bswuser52/GEN/RECEIVER.* file or with the one downloaded from anonymous AIUB ftp server.

8.3 Debugging

= To complete =

8.3.1 Download testing

= To complete =

8.3.2 Processing testing

= To complete =

8.3.3 Upload testing

= To complete =

References

- Dousa J, Vaclavovic P (2015), Evaluation of ground-based GNSS tropospheric products at Geodetic Observatory Pecny, In: IAG Symposia Series, Springer, Vol. 143
- Douša J, Bennitt GV (2013), Estimation and evaluation of hourly updated global GPS Zenith Total Delays over ten months, GPS Solut, Springer, 17:453–464
- Douša J. (2012), Developments of the GLONASS ultra-rapid orbit determination at Geodetic Observatory Pecný, In: Geodesy of Planet Earth, S. Kenyon, M.C. Pacino, U. Marti (eds.), IAG Symposia Series, Springer, 136:1029-1036.
- Douša J (2010), Precise near real-time GNSS analyses at Geodetic observatory Pecný - precise orbit determination and water vapour monitoring, Acta Geodyn Geomater, 7(157), 1-11.
- Douša J (2010), The impact of errors in predicted GPS orbits on zenith troposphere delay estimation, GPS solut, Springer, 13(3):229-239.
- Douša J (2004), Precise Orbits for Ground-Based GPS Meteorology: Processing Strategy and Quality Assessment of the Orbits Determined at Geodetic Observatory Pecný, J Met Soc Japan, 82:371-380.
- Douša J (2003), Evaluation of tropospheric parameters estimated in various routine analysis, Phys Chem Earth, 29(2-3):167-175.
- Douša J (2002), On the Specific Aspects of Precise Tropospheric Path Delay Estimation in GPS Analysis, In: Vistas for Geodesy in the New Millenium, IAG Symposia series, Adam, J. and K.-P. Schwarz (eds.), Springer, 125:285-290.
- Douša J (2001), Towards an Operational Near-real Time Precipitable Water Vapor Estimation, Phys Chem Earth, Part A, 26(3):189-194.
- Douša J (2001), The Impact of Ultra-Rapid Orbits on Precipitable Water Vapor Estimation using Ground GPS Network, Phys Chem Earth, Part A, 26/6-8, pp. 393-398.
- Douša J, Mervart, L. (2001), On Hourly Orbit Determination, Physics and Chemistry of the Earth, Part A, 26/6-8, pp.555-560.