

Transfer a file incrementally over a flapping connection

Shipboard outages of network can happen when the ship turns. This can be unpredictable. I use this technique to "set and forget" transfer data over HiSeasNet, using a combination of *rsync*, *SSH* and the *Bourne-Again SHell (BASH)* on a UNIX-type system. You'll need to set up password-less SSH keys to do this.

1. Create an ssh-key for your shipboard host, if you haven't already done so. `man ssh-keygen` for more info on this. **This is a potential security risk**, so you should understand the caveats and benefits.

```
ssh-keygen
chmod 0600 ~/.ssh/id*
cat ~/.ssh/id_rsa.pub | ssh me@myShoreServer "cat - >> ~/.ssh/authorized_keys"
```

2. Login to a shore-based server, use the faster link to stage the data:

```
ssh me@myShoreServer
wget -N -nd http://someServer/some.file
```

3. On the local shipboard system, use *rsync* to grab the data in a loop that will execute forever until *rsync* succeeds. The *-P* flag will move resume a partially transferred file, so once data is moved once, it will stay on board and be updated; the *-c* flag will ensure the shore-side and shipboard final file(s) checksum to be the same.

```
while ! rsync -Prtz myShoreServer:/dir/to/data/some.file /local/target/dir; do echo "`date -u` Restarting rsync..."; done
```

If you want the data to show up, but not hog all the bandwidth while you leave this alone for hours (days?), use the *--bwlimit* flag. *--bwlimit* is calculated in kiloBytes per second (kBps), not kilobits per second (kbps), so a value of 2 kBps per second is 16 kbps.

```
while ! rsync --bwlimit=2 -Prtz myShoreServer:/dir/to/data/some.file /local/target/dir; do echo "`date -u` Restarting rsync..."; done
```