Network-Attached Storeage using Linux nano-computers

by

Greg Bishop

Linux Supporters Group Adelaide

Wed 5 Feb 2014

Describe BACKUP

Discuss RSYNC

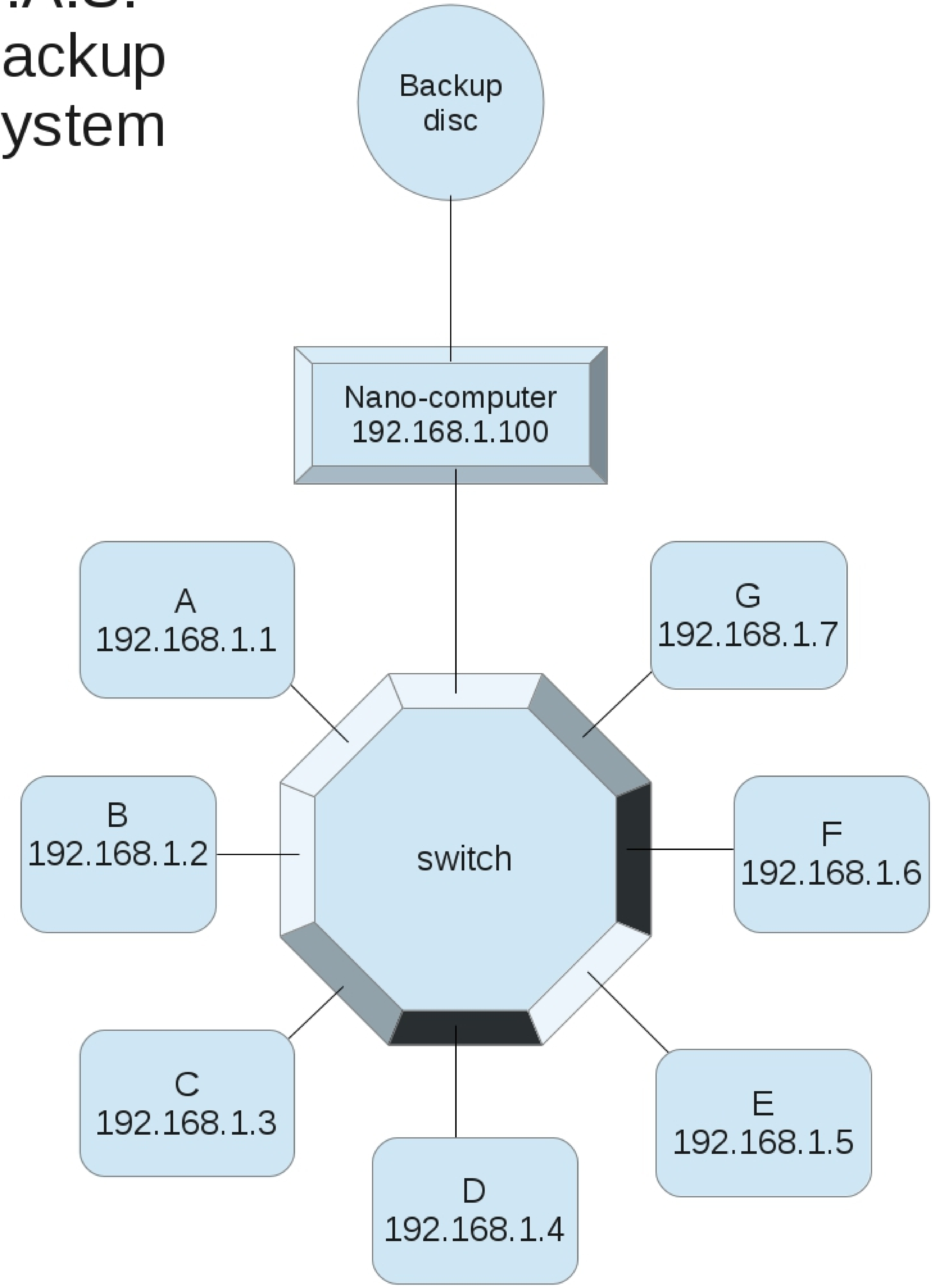
TRY:

- Raspberry Pi
- BeagleBone Black
- CuBox

Assessment

Recommendations

N.A.S. Backup System



rsync backup

You'll need the rsync package
(rsync & rsyncd) on both computers

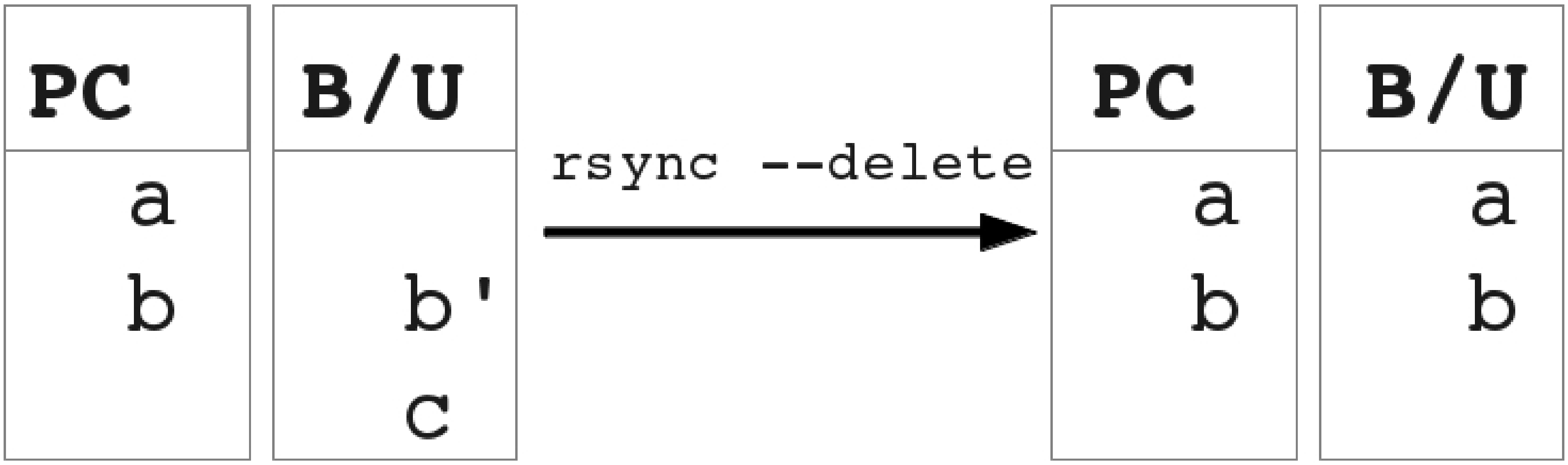
e.g. (referring to the previous diagram)

To:
backup /home directory onto backup disc
in the directory set aside for the
personal computer labelled PC 'A':

Use:

```
# /usr/bin/rsync -a --delete home/ \
      192.168.1.100:/mnt/backup/pc-a/home
```

Note *this 'gotcha'* using the option --delete:



PC: a, b

B/U: b', c

rsync --delete

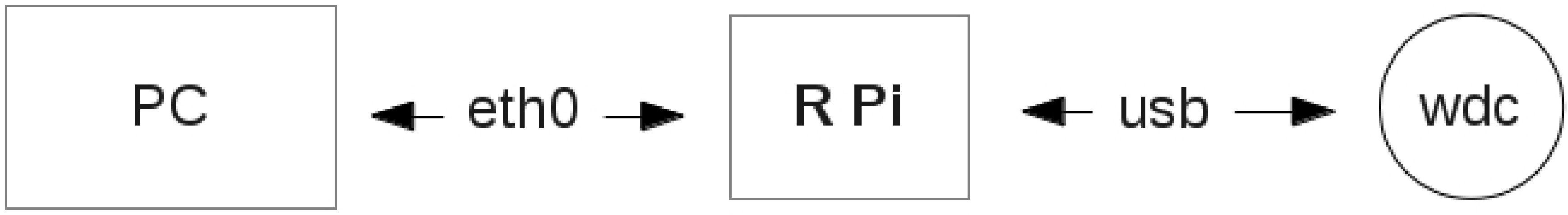
PC: a, b

B/U: a, b

```
#!/bin/bash

# /usr/local/bin/backup (this script)
# is a sample of a network-attached storeage
# (NAS) backup script
# this script is to be run as the root user
# usage: > su -c "/usr/local/bin/backup"

# to use rsync without needing to enter a root
# password for every directory, follow the
# recipe at the site
# http://www.linuxproblem.org/art_9.html
# it is also very useful to consult the
# article at the site
# http://bencane.com/2014/01/07/  \
# using-rsync-to-synchronize-a-local-and-
#                                     remote-directory/

# where is the remote computer with the backup
# disc attached to it?
nano_nas=192.168.1.100
remote=$nano_nas

# what is the path to the backup disc on
# the remote computer?
this_computer=$(hostname)
backup=/mnt/backup/$this_computer
```

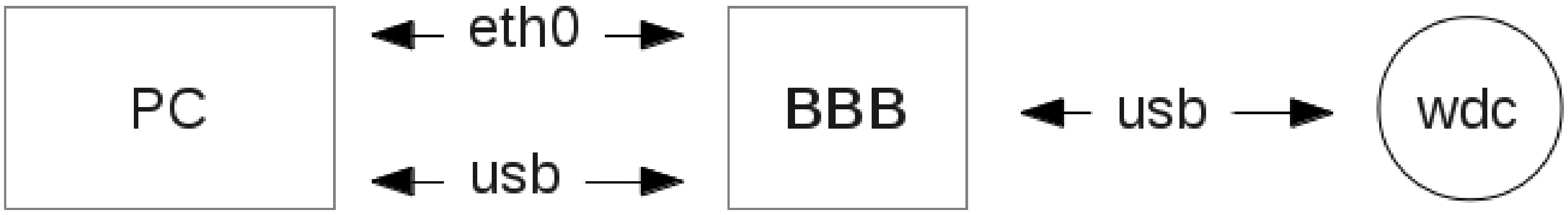
```
# which options of the rsync command
# do we want to use?
rsync_command= \
    "/usr/bin/rsync -av --delete --inplace"

# is the remote computer available?
echo -n "testing if $remote is up ... "
ping -c 1 -W 1 $remote 1>/dev/null 2>&1 \
|| { echo "sorry, $remote is down"; exit 1; }
echo "yep!"

# list the directories to back up,
# relative to the root directory (each one
# must already exist on backup disc)
dir_list="home boot etc usr/local var/log
           var/lib/dpkg var/lib/apt"

echo "$(/bin/date): starting backup to $remote"
pushd /
for dir in $dir_list
do
  $rsync_command /$dir/ $remote:$backup/$dir
done
popd
echo "$(/bin/date): finished backup to $remote"

exit 0
```

Results for **Raspberry Pi**



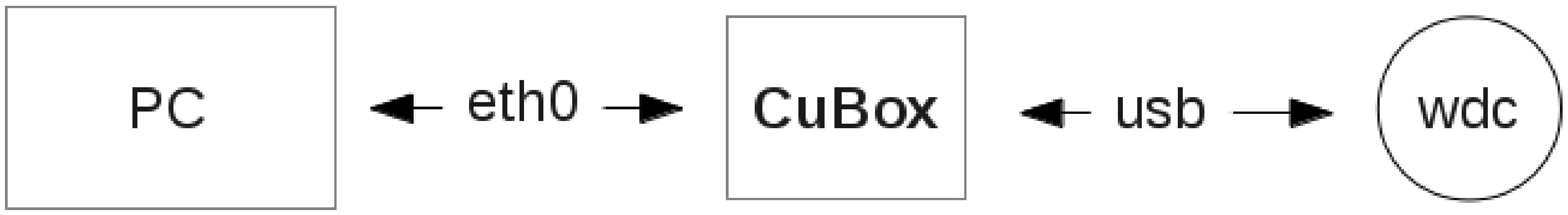
```
Start Personal Computer
Start Raspberry Pi (boots to eth0=192.168.1.100)
Plug in backup disk (Western Digital USB)
Run 'dmesg' -------> fails to recognise WDC
Add double-cable to two USB ports for more current
Run 'dmesg' --------> still fails to recognise WDC
Add powered USB hub between RPi and WDC
Reboot RPi
Run 'dmesg' ---> recognises WDC, creates /dev/sda1
Backup runs OK
```

Results for **BeagleBone Black**



```
Start Personal Computer
Start BBB (eth0=192.168.1.100; usb0=192.168.7.2)
Plug in backup disk (Western Digital USB)
Run 'dmesg' -------> fails to recognise WDC
Plug in WDC first
Reboot BBB
Run 'dmesg' --------> cannot spin up WDC disc
Add powered USB hub between BBB and WDC
Reboot BBB
Run 'dmesg' ---> recognises WDC, creates /dev/sda1
Backup runs OK
```

Results for **CuBox**



PC

eth0

CuBox

usb

wdc

Start Personal Computer

Start CuBox (boots to eth0=192.168.1.100)

Plug in backup disk (Western Digital USB)

Run 'dmesg' ---> recognises WDC, creates /dev/sda1

Backup runs OK

Assessment

Raspberry Pi:

USB current inadequate for the WDC
Requires an extra powered USB hub

BeagleBone Black:

No 'hotplug' of WDC, (old kernel?)
USB current inadequate for the WDC
Required extra powered USB hub

CuBox:

Worked straight out of the box

Recommendation

Even though the CuBox appears a little roughly finished and somewhat more expensive, it is the simplest reliable nano we tried. The BeagleBone Black and the Raspberry Pi might best be left to other more suitable projects.