Instructions

Use this script once you have been assigned a SixXS tunnel to connect that tunnel. Once your tunnel has been up for a week, you earn enough credits to request a subnet (/48 IPv6 allocation) which you can then use for hosts on your network. See the instructions in the SETUP VALUES section for more info.

See http://www.sixxs.net/main/ for info on how to request tunnels and subnets.

Any feedback on the script welcome in this forum topic.

```
#!/bin/sh
#
# SIXXS IPv6 startup script for WRT54G
# From http://www.dd-wrt.comhttp://www.dd-wrt.com/wiki/index.php/IPv6_startup_script

###################################
###        SETUP VALUES         ###
### Change these before running ##
###################################

# The IPv6 prefix (/64) of the tunnel initially assigned to you by SixXS (including the ::).
# Only the ::1 and ::2 are used from this range, for the PoP and your router, respectively.
# You need to keep the tunnel up for a week in order earn enough credits to request a /48 subnet
# (see below) which can then be used by hosts on your network.
TUNNELPREFIX="2001:db8:1111:1111::"

# The IPv4 address of the SixXS PoP you're using
SIXXS4="1.1.1.1"

# MTU - must match your SixXS tunnel settings and radvd's "AdvLinkMTU" value; 1280 is the default
# This script will give you a recommendation when you run it, but it should be 20 less than your
MTU=1280

# A /64 from your /48 subnet prefix assigned by SixXS, INCLUDING the /64
## This should match what you announce to the network with radvd (set in Administration -> Manage
# If you have not yet been assigned this (you need to keep the tunnel up for a week to earn enoug
##SUBNET="2001:db8:2222::/64"

# Space-separated list of extra IPs you want for your router, such as:
# * Local (RFC 4193, currently draft) address e.g. fdxx:xxxx:xxxx:yyyy::1/64; or
# * site-local (as per RFC 1918, but obseleted by RFC 3879) address e.g. fec0::1/64
# * Any other local or global fixed address you want (note: we already assign the above SUBNET ad
# This is optional, leave blank/commented out if not wanted.
###ADDITIONALIP6="fec0::1/64"

###################################
###     END OF SETUP VALUES     ##
###################################

logger -st sixxs -- "Starting IPv6 setup"

# Get the tunnel endpoint addresses
MYTUNNELIP="${TUNNELPREFIX}2"
SIXXSTUNNELIP="${TUNNELPREFIX}1"

# Get our external IPv4 (in case the IP isn't available, because we don't have a connection yet,
RETRIES=3
while [ -z ${EXTIP} ]; do
    # PPPoE devices are on ppp0, all others are on vlan1 or eth1; we try all
```

```
    for DEV in "ppp0" "vlan1" "eth1"; do
        EXTIP=`/usr/sbin/ip -4 addr show dev ${DEV} | grep inet | awk '{print $2}' | cut -d/ -f1`
        if [ -n "$EXTIP" ]; then
            logger -st sixxs -- "- External IP ${EXTIP} found on device ${DEV}"
            break 2
        fi
    done
    let RETRIES-=1
    if [ "$RETRIES" -gt 0 ]; then
      logger -st sixxs -- "- No external IP found; trying again in 10 secs..."
      sleep 10
    else
      logger -st sixxs -- "- No external IP found after 30 seconds; quitting"
      exit 1
    fi
done

# Create tunnel
logger -st sixxs -- "- Creating SixXS tunnel ${EXTIP} <--> ${SIXXS4}"
/usr/sbin/ip tunnel add sixxs mode sit local ${EXTIP} remote ${SIXXS4}

# Bring tunnel interface up explicitly
logger -st sixxs -- "- Bringing sixxs device up"
/usr/sbin/ip link set sixxs up

# Fix MTU and TTL
logger -st sixxs -- "- Fixing MTU and TTL of sixxs device"
/usr/sbin/ip link set mtu ${MTU} dev sixxs
/usr/sbin/ip tunnel change sixxs ttl 64

# Check for optimal MTU
# Max MTU is device's MTU minus 20 (IPv6 headers are 20 bytes bigger than IPv4)
RECMTU=`ip addr show dev $DEV | grep mtu | awk '{print $5}'`
RECMTU=`expr ${RECMTU} - 20`
if [ ${MTU} -lt ${RECMTU} ]; then
    logger -st sixxs -- "- Note: You could increase your MTU setting from ${MTU} to ${RECMTU} for
elif [ ${MTU} -gt ${RECMTU} ]; then
    ### TODO: What happens if MTU is too high?
    logger -st sixxs -- "- WARNING: Your MTU ${MTU} is too high - it should be ${RECMTU}. Change
fi

# Add extra IPs if any
if [ ! -z ${ADDITIONALIP6} ]; then
    for IP in ${ADDITIONALIP6}; do
   logger -st sixxs -- "- Adding fixed IP ${IP}"
   /usr/sbin/ip -6 addr add ${IP} dev br0
    done
fi

# Configure IPv6 endpoint on the tunnel
logger -st sixxs -- "- Configuring local IPv6 tunnel endpoint ${MYTUNNELIP}"
/usr/sbin/ip -6 addr add ${MYTUNNELIP}/64 dev sixxs

# Add default route
logger -st sixxs -- "- Adding default route through SixXS ${SIXXSTUNNELIP}"
/usr/sbin/ip -6 ro add default via ${SIXXSTUNNELIP} dev sixxs

if [ ! -z  ${SUBNET} ]; then
    # Add a network from the /48 subnet to br0
    logger -st sixxs -- "- Setting up subnet ${SUBNET}"
    /usr/sbin/ip -6 addr add ${SUBNET} dev br0
```

```
    ## # Start router advertisement daemon
    ## # No need to do this here; just use the setup in the web interface
    ## logger -st sixxs -- "- Starting radvd..."
    ## /usr/sbin/radvd -C /tmp/radvd.conf
else
    logger -st sixxs -- "- Note: No subnet configured yet - apply for one when you've kept the tu
fi

logger -st sixxs -- "SixXS IPv6 setup done"
```

radvd configuration

An example radvd.conf, to be used once you have been assigned a /48 subnet (see Administration -> Management in your router's web interface):

```
 interface br0 {
     AdvSendAdvert on;
     AdvHomeAgentFlag on;
     AdvLinkMTU 1280;
     MinRtrAdvInterval 3;
     MaxRtrAdvInterval 10;
     prefix [REPLACE WITH SUBNET]/64 {
         AdvOnLink on;
         AdvAutonomous on;
         AdvRouterAddr on;
     };
 };
```