



wireless mesh networks

10 steps to speedup your
mesh-network by factor 5

Bastian Bittorf

<http://www.bittorf-wireless.com>

berlin, c-base, 4. june 2011



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



1 Agenda

2 CPU/Architecture

- efficient use of CPU
- rate-selection

3 Airtime

- avoid slow rates
- separate channels

4 Compression

- like modem: V.42bis
- iproute2/policy-routing
- compress data to inet-gateway
- slow DSL-lines?

5 Cache

- local HTTP-Proxy
- Gateway HTTP-Proxy



use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



efficient use of CPU

use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



use modern software

- Kernel 2.6.39
 - better scheduler
 - better memory management
- swapon /dev/ramswap0
- fast links needs fast hardware
- Q: what power is needed to route 30mbit/s from LAN to WIFI?



use a better rate-selection-algorithm

- **minstrel_ht**
- mac80211
- needs kernel 2.6



use a better rate-selection-algorithm

- minstrel_ht
- mac80211
- needs kernel 2.6



use a better rate-selection-algorithm

- minstrel_ht
- mac80211
- needs kernel 2.6



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



avoid slow rates

Safe airtime

- help your routing-protocol by throwing away slow links
- avoid air pollution
 - option beacon_int 10000
- avoid slow rates
 - option mcast_rate 6000
- also affects management frames (by accident?)
- maybe: list basic_rate 6000
- later: ETT-metric



separate channels

Split transport-network from access-network

- Channel A: adhoc-network
- Channel B: ap-network
- both devices are LAN/LAN wired
- olsr-tweaks:
 - option LinkQualityAlgorithm etx_ffeth
 - option mode ether



separate channels

Split transport-network from access-network

- Channel A: adhoc-network
- Channel B: ap-network
- both devices are LAN/LAN wired
- olsr-tweaks:
 - option LinkQualityAlgorithm etx_ffeth
 - option mode ether



separate channels

Split transport-network from access-network

- Channel A: adhoc-network
- Channel B: ap-network
- both devices are LAN/LAN wired
- olsr-tweaks:
 - option LinkQualityAlgorithm etx_ffeth
 - option mode ether



Split transport-network from access-network

- Channel A: adhoc-network
- Channel B: ap-network
- both devices are LAN/LAN wired
- olsr-tweaks:
 - option LinkQualityAlgorithm etx_ffeth
 - option mode ether



Split transport-network from access-network

- Channel A: adhoc-network
- Channel B: ap-network
- both devices are LAN/LAN wired
- olsr-tweaks:
 - option LinkQualityAlgorithm etx_ffeth
 - option mode ether



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)



like modem: V.42bis

Compressed tunnel to inet-gateway

- one tunnel for all clients on one router
- vtund / lzo
- openvpn / lzo
- ipip-tunnel / compression ???
- internal network traffic is not compressed
- needs dynamic tunnels (end-2-end)

- use policy-routing:

```
echo 50 XY »/etc/iproute2/rt_tables
ip rule add from $HNA/$MSK prio 30002 table XY
ip rule add to $HNA/$MSK prio 30001 lookup main
ip route add default \
    via $TUNNEL_NEIGH_IP \
    dev $TUNNEL_DEV \
    table XY
ip route flush cache
```



compress data to inet-gateway

Compressed tunnel to inet-gateway

- ```
ip tunnel \
 add tunnel0 \
 mode ipip \
 remote 10.10.1.1 \
 local 10.10.99.1 \
 compress lzo

ip link set dev ipip0 up
ip address add 172.16.1.2/24 dev tunnel0
```



slow DSL-lines?

## Compressed tunnel from inet-gateway to data-centre

- use lzo-compressed tunnel to server with better connection



# Caching HTTP-Proxy on each device

- Polipo
  - 8 Gigabyte USB-Sticks ( 10 Euro)
  - `mount -t btrfs -o compress,ssd /dev/sda1 /tmp/usb0`
  - needs 32MB RAM





# Caching HTTP-Proxy on each device

- Polipo
- 8 Gigabyte USB-Sticks ( 10 Euro)
- `mount -t btrfs -o compress,ssd /dev/sda1 /tmp/usb0`
- needs 32MB RAM



# Caching HTTP-Proxy on each device

- Polipo
- 8 Gigabyte USB-Sticks ( 10 Euro)
- `mount -t btrfs -o compress,ssd /dev/sda1 /tmp/usb0`
- needs 32MB RAM



## Caching HTTP-Proxy on each device

- Polipo
- 8 Gigabyte USB-Sticks ( 10 Euro)
- `mount -t btrfs -o compress,ssd /dev/sda1 /tmp/usb0`
- needs 32MB RAM



## Caching HTTP-Proxy on each device

- Polipo
- 8 Gigabyte USB-Sticks ( 10 Euro)
- `mount -t btrfs -o compress,ssd /dev/sda1 /tmp/usb0`
- needs 32MB RAM



# Caching HTTP-Proxy on gateway

- Polipo
- maybe squid on strong hardware



# Caching DNS-Resolver

- dnsmasq
- each local dnsmasq asks a central dnsmasq
- easy...



# Caching DNS-Resolver

- dnsmasq
- each local dnsmasq asks a central dnsmasq
- easy...



synchronise everything

## synchronise regulary

- unison
- always 1hop
- always in inet2node direction
- `while WifiIsIdle; do unison A B; done`





synchronise everything

## synchronise regulary

- unison
- always 1 hop
- always in inet2node direction
- `while WifiIsIdle; do unison A B; done`



synchronise everything

## synchronise regulary

- unison
- always 1 hop
- always in inet2node direction
- `while WifiIsIdle; do unison A B; done`



synchronise everything

## synchronise regulary

- unison
- always 1 hop
- always in inet2node direction
- `while WifiIsIdle; do unison A B; done`



compress to zero

# local Ad-Blocker

- Polipo
- easylist.txt + adblock2polipo.py
- <http://spiralofhope.com/polipo.html>



compress to zero

# local Ad-Blocker

- Polipo
- easylist.txt + adblock2polipo.py
- <http://spiralofhope.com/polipo.html>



compress to zero

# local Ad-Blocker

- Polipo
- easylist.txt + adblock2polipo.py
- <http://spiralofhope.com/polipo.html>



## layer7-QoS for better Layer8 experience

- shape everything
- layer7: <http://l7-filter.clearfoundation.com/>
- web + games fast
- downloads slow (every connection, where conntrack detects >1 Megabytes)
- class for internal-traffic (unison-cache-synching)



## layer7-QoS for better Layer8 experience

- shape everything
- layer7: <http://l7-filter.clearfoundation.com/>
- web + games fast
- downloads slow (every connection, where conntrack detects >1 Megabytes)
- class for internal-traffic (unison-cache-synching)





## layer7-QoS for better Layer8 experience

- shape everything
- layer7: <http://l7-filter.clearfoundation.com/>
- web + games fast
- downloads slow (every connection, where conntrack detects >1 Megabytes)
- class for internal-traffic (unison-cache-synching)



## layer7-QoS for better Layer8 experience

- shape everything
- layer7: <http://l7-filter.clearfoundation.com/>
- web + games fast
- downloads slow (every connection, where conntrack detects >1 Megabytes)
- class for internal-traffic (unison-cache-synching)



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues





## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



## what could be done better?

- in-kernel compressed ipip-tunnel (packet aggregation?)
- dynamic one-hop-tunnels with olsr
- hardware-supported compression
- automagic layer7-framework-builder
- do everything with IPv6
- do bandwidth aware routing
- having a second mesh-routing-table for "bulk"
- having package polipo-dev / unison
- having HT40 meshing
- having dynamic powercontrol aka minstrel\_blues



# thanks to...

- $\LaTeX$  (beamer class)
- openWRT
- linux
- mac80211 / minstrel\_ht
- ath9k / b43
- <https://github.com/bittorf/kalua>



# thanks to...

- $\LaTeX$  (beamer class)
- openWRT
- linux
- mac80211 / minstrel\_ht
- ath9k / b43
- <https://github.com/bittorf/kalua>