B.A.T.M.A.N. V: what's coming next?

Marek Lindner & Antonio Quartulli

B.A.T.M.A.N.-Advanced
www.open-mesh.org

May 14th, 2014
WirelessBattleMeshv7 - Leipzig

A few words about batman-adv...

The B.A.T.M.A.N. protocol was initiated in Berlin, 2006. The first edition was developed as a daemon, and moved to kernel space in 2007 to improve performance.
Characteristics:

- L2 routing (MAC address layer)
- runs on any Ethernet capable device (e.g. 802.3, 802.11, and 802.15.1)
- encapsulates incoming ether frames and handles all forwarding/delivery
- agnostic to IP or any L3 protocol
- supports non-mesh clients with gateway selection, roaming & more
- part of the Linux kernel, thus shipped by default in most Linux distributions (modprobe batman-adv)

What we are NOT going to talk about today

- backward compatibility (TVLV, compat number, etc)
- VLAN-ization of non-mesh client handling
- fragmentation v2.0 (we fragment everything!)
- extended AP isolation
- multicast improvements
- inter-connecting batman clouds
- layer2 anycast support
- DHT generalization (IPv6 address caching, ..)
- ...

Today's topics

- B.A.T.M.A.N. V introduction
- network-wide multi-interface optimizations
- protocol overview (ELP/OGMv2)
- throughput based metric
- current status / practical tips
- next steps

Network-wide multi-interface optimization

brief recap:

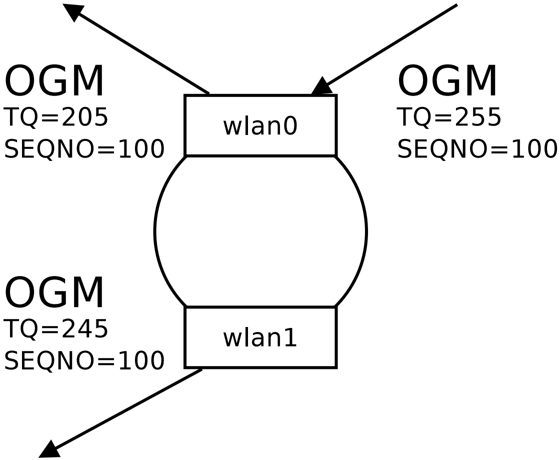
- batman-adv supports link-local multi-iface optimizations since early 2010
- results were good but we can do better ..

B.A.T.M.A.N. V roadmap (part I):

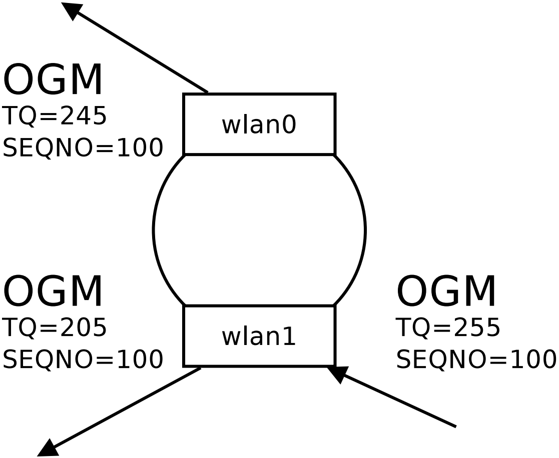
- differentiation between half duplex / full duplex
- take advantage of the many interfaces devices are powered with today
- maximizing traffic throughput by applying rules to the traffic flow

Network-wide multi-interface optimization (2)

in a nutshell:



Network-wide multi-interface optimization (3)



Network-wide multi-interface optimization (4)

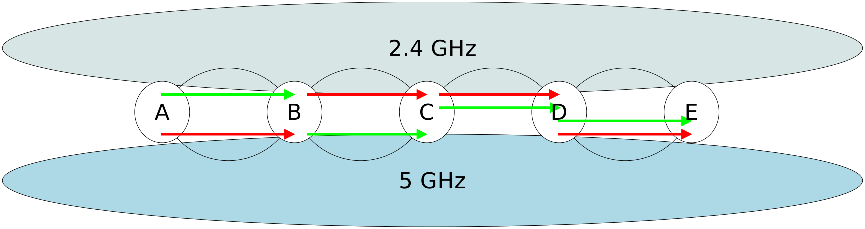
The tables:

- each interface has its own routing table
- the default table is used for traffic originating from the host itself

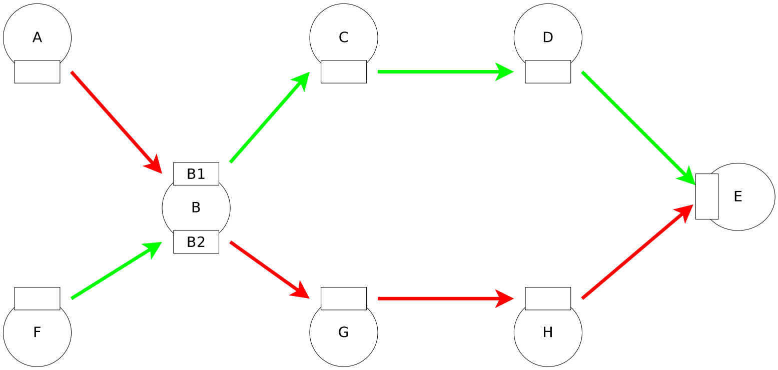
```
# batctl o
[B.A.T.M.A.N. adv master-b82b9b2, MainIF/MAC: wlan0/node2_wlan0 (bat0 BATMAN_IV)]
 Originator last-seen (#/255)     Nexthop [outIF]:   Potential nexthops ...
 node3_wlan0   0.670s   (255) node3_wlan0 [wlan0]: node3_wlan1 (255) node3_wlan0 (255)
 node1_wlan0   0.920s   (255) node1_wlan1 [wlan1]: node1_wlan1 (255) node1_wlan0 (254)

# batctl o -i wlan0
[B.A.T.M.A.N. adv master-b82b9b2, IF/MAC: wlan0/node2_wlan0 (bat0 BATMAN_IV)]
 Originator last-seen (#/255)     Nexthop [outIF]:   Potential nexthops ...
 node3_wlan0   0.560s   (252) node3_wlan1 [wlan1]: node3_wlan1 (252) node3_wlan0 (240)
 node1_wlan0   0.850s   (255) node1_wlan1 [wlan1]: node1_wlan1 (255) node1_wlan0 (238)

# batctl o -i wlan1
[B.A.T.M.A.N. adv master-b82b9b2, IF/MAC: wlan1/node2_wlan1 (bat0 BATMAN_IV)]
 Originator last-seen (#/255)     Nexthop [outIF]:   Potential nexthops ...
 node3_wlan0   0.260s   (253) node3_wlan0 [wlan0]: node3_wlan1 (240) node3_wlan0 (253)
 node1_wlan0   0.510s   (255) node1_wlan0 [wlan0]: node1_wlan1 (240) node1_wlan0 (255)
```

Network-wide multi-interface optimization (5)

The benefits:



Network-wide multi-interface optimization (6)

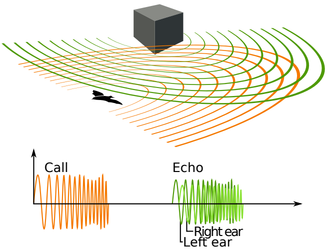


protocol overview

B.A.T.M.A.N. V roadmap (part II):

- evolution instead of revolution
- design goals
  - simplification
  - gain flexibility to better support diverse scenarios
  - reduce overhead

protocol overview - ELP



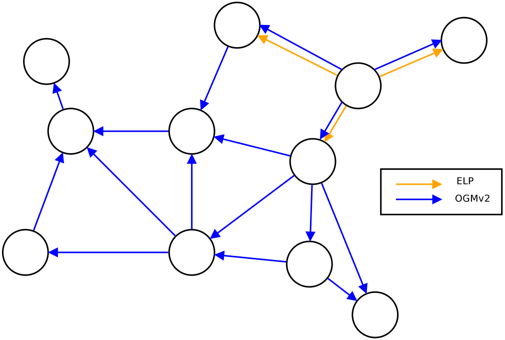
ELP - Echo Location Protocol

- link sensing & neighbour discovery
- no rebroadcast or forward of any kind
- short broadcast intervals

protocol overview - OGMv2

OGMv2 - Originator Message Protocol v2

- propagating routes in the mesh
- rebroadcast with stricter rules, yet simplified
- long broadcast intervals



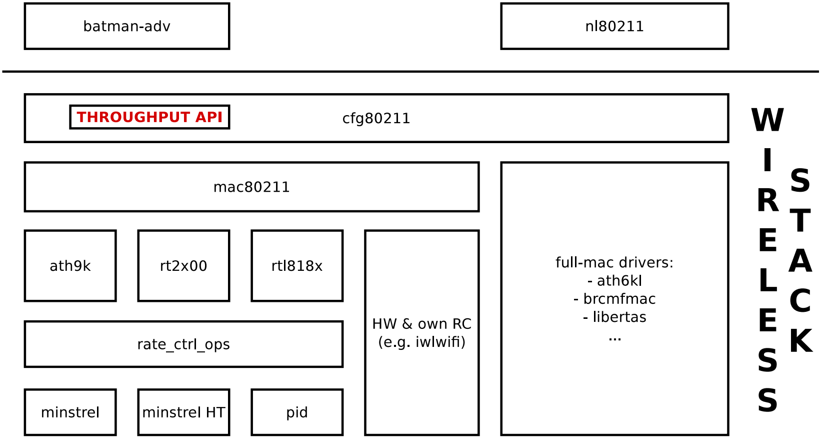
A new metric

The dream of **throughput based routing**. . .

- the idea is around for at least a decade
- existing approaches
- our motivation

Reading throughput in kernel space (WiFi)

Wireless links: query the Rate Control algorithm



Reading throughput in kernel space (WiFi) (2)

Example of RC statistics (MinstrelHT):

```
# cat /sys/kernel/debug/ieee80211/phy0/netdev:mesh0/stations/xx:xx:xx:xx:xx:xx/rc_stats

type           rate    thr    ewma    prob    retry     ....
HT20/LGI       MCS0    5.9    95.8   100.0     3        ....
HT20/LGI       MCS1    9.4    80.4    66.6     3        ....
HT20/LGI       MCS2   12.2    74.2   100.0     2        ....
HT20/LGI  T    MCS3   13.2    64.1   100.0     2        ....
HT20/LGI       MCS4   12.3    44.3   100.0     4        ....
 .....
```

Throughput in kernel space (3)

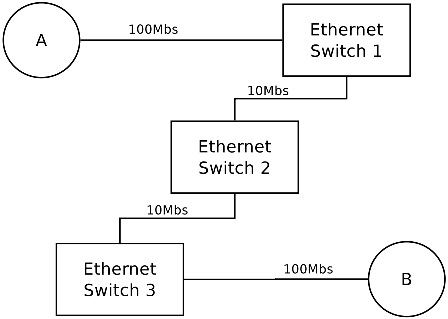
Wired links (common scenario):
read negotiated speed from the driver

Throughput in kernel space (3)

Wired links (common scenario):
read negotiated speed from the driver

it is easy, but it is not the best approach

Example:



Throughput in kernel space (4)

But what about:

- VPN links
- tunnels
- non-mac80211 WiFi drivers
- better Ethernet throughput estimation
- . . .

manual configuration? what else?

Throughput meter in batman-adv

- started as Google Summer of Code project in 2012
- re-implementation of TCP on batman-adv
- no need for IPs (uses batman-adv identifiers)
- measures the ”payload“ throughput (no packet overhead)
- can also be used from userspace (using batctl)

Example:

```
root@NodeA:~# batctl tm -t 3000 NodeB
Throughput meter called towards NodeB
Test duration 3010ms.
Sent 20247000 Bytes.
Throughput: 6.41 MB/s (53.81 Mbps)
```

Current limitations

- different RC algorithms (batman-adv is not aware!)
  - different API implementation
  - need for a different probing schema
- throughput != throughput
- real world testing

Current status

- **cfg/mac80211 patches** under review by linux-wireless people and close to integration
- a working **B.A.T.M.A.N. V prototype** is available in our git repository (ordex/batman_v branch)
- a working **throughput meter** protorype is available in our git reporitory (ordex/bw_meter branch)

How to use B.A.T.M.A.N. V on my node

The batman-adv kernel module is already able to host more than one routing algorithm

```
# cat /sys/kernel/debug/batman_adv/routing_algos
BATMAN_IV
BATMAN_V (only if compiled into the module)
```

How to use B.A.T.M.A.N. V on my node

The batman-adv kernel module is already able to host more than one routing algorithm

```
# cat /sys/kernel/debug/batman_adv/routing_algos
BATMAN_IV
BATMAN_V (only if compiled into the module)
```

Benefits:

- the routing algorithm can be changed at runtime

```
# echo BATMAN_V >/sys/module/batman_adv/parameters/routing_algo
# batctl if add -m bat0 wlan0
```

How to use B.A.T.M.A.N. V on my node

The batman-adv kernel module is already able to host more than one routing algorithm

```
# cat /sys/kernel/debug/batman_adv/routing_algos
BATMAN_IV
BATMAN_V (only if compiled into the module)
```

Benefits:

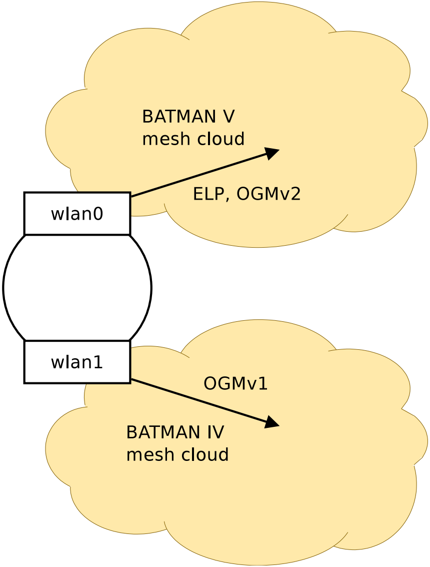
- the routing algorithm can be changed at runtime

```
# echo BATMAN_V >/sys/module/batman_adv/parameters/routing_algo
# batctl if add -m bat0 wlan0
```

- both algorithms can be used at the same time (on two different interfaces)

```
# echo BATMAN_IV >/sys/module/batman_adv/parameters/routing_algo
# batctl if add -m bat1 wlan1
```

How to use B.A.T.M.A.N. V on my node (2)



Thank you for your attention

Questions?