batman-adv scalability

Layer 2 Mesh Networks - Myths and Risks

Linus Lüssing

Freifunk Hamburg Geekend02, Sep. 2013

Outline

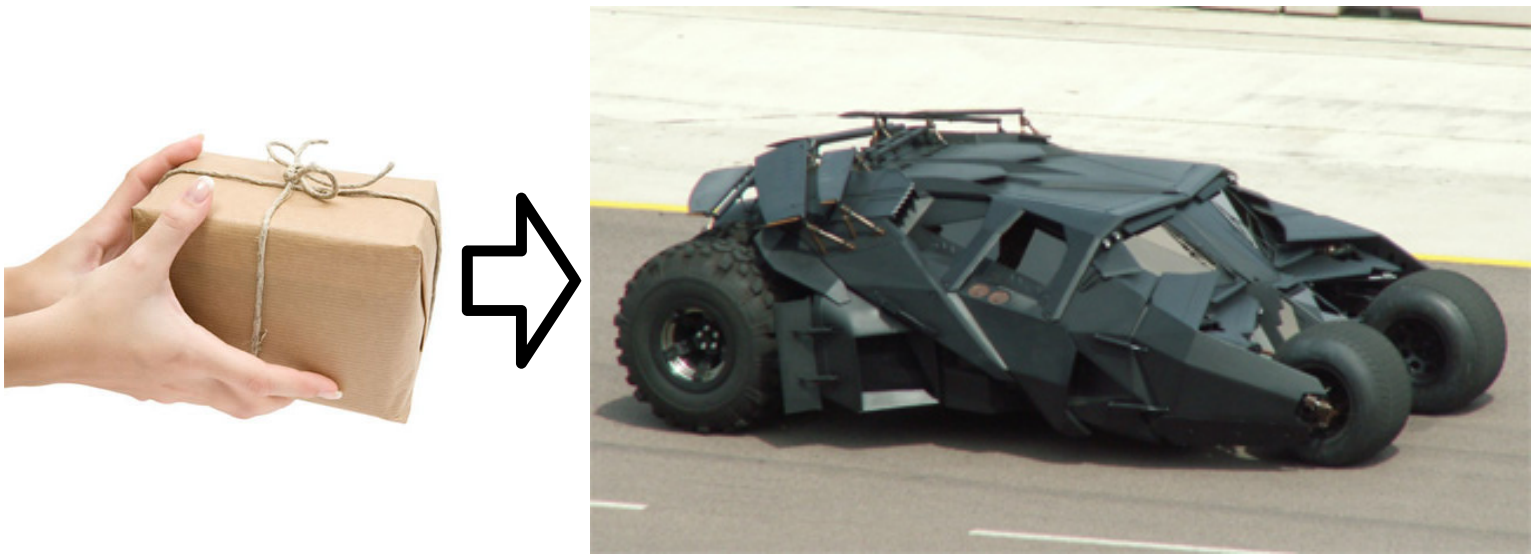
Outline

batman-adv: Big, Virtual Switch




batman-adv: Big, Virtual Switch

batman-adv: Encapsulation



- Encapsulates ethernet frames
- In own batman-adv header

Advantages

- Supports IPv4, IPv6, probably IPv42, ...
- Your non-IP / link-layer protocol?
- More flexible than Linux IP routing table:
  - Interface bonding
  - Network coding
  - ...
- Simple configuration
  - MAC addresses are unique
  - No IP subnet coordination
- Fast Roaming
- ...

Disadvantage: Overhead

Layer 3 mesh routing protocol:

- Mesh protocol overhead

Layer 2 mesh routing protocol:

- Mesh protocol overhead
- + Layer 2 specific overhead

Disadvantage: Overhead

IEEE 802.11s:

- Designed for $\sim$32 nodes

Disadvantage: Overhead

”Layer 2 Mesh Networks? Don’t Scale!”

Outline

- Experiences from Freifunk Lübeck

1 Node: Kernel panics

- 2009, pre 0.1 firmware
- Did not boot: batman-adv crashing

1 Node: Kernel panics

Black Box:

- 2009, pre 0.1 firmware
- Did not boot: batman-adv crashing

10 Nodes: Too Large Neighbourhood on VPN

Trying out batman-adv over VPN, using tinc:

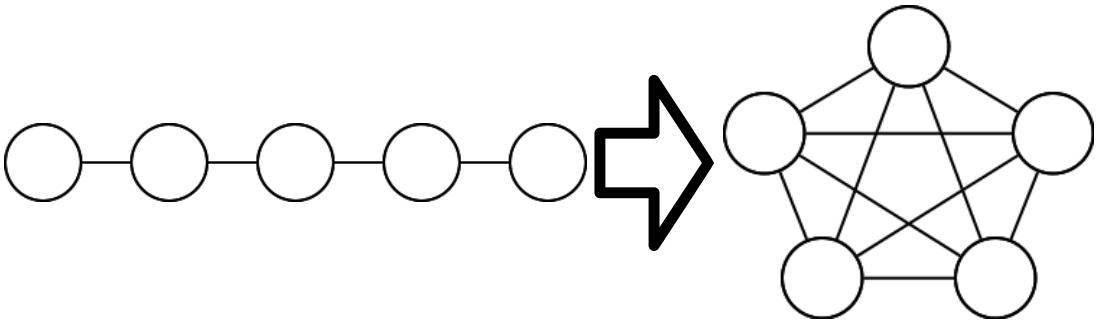
- tinc does meshing, too:
- Large neighbourhood on VPN: many rebroadcasts
- Overhead on DSL got close to 1MBit/s

Changes in 0.2

October 2011

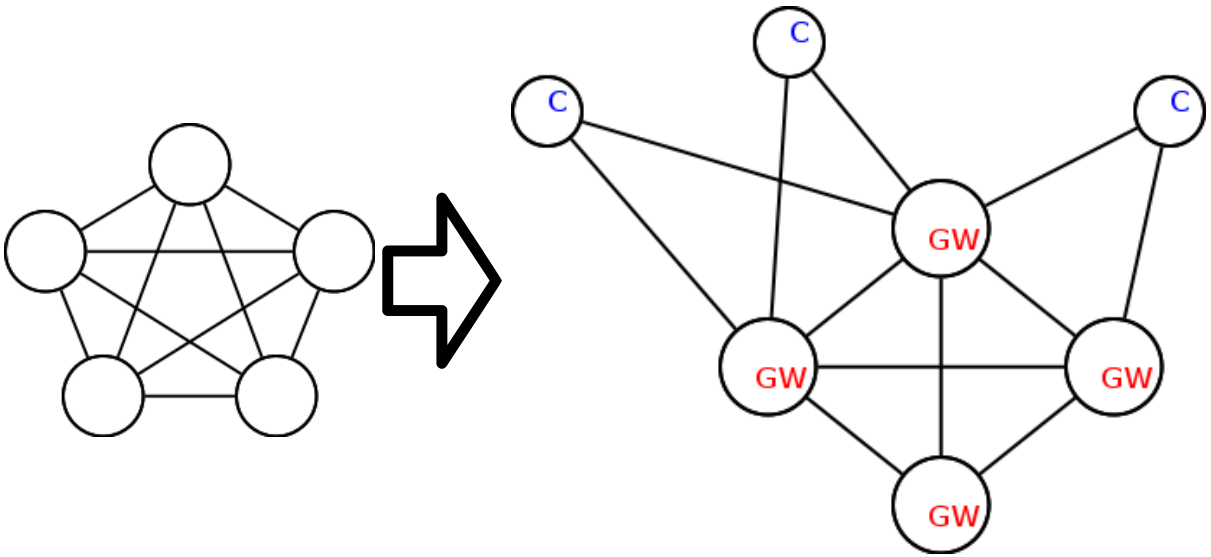
- Added tinc
- OGM (= route update + link quality) interval:
  - 1s -> 3s
- Should scale to 30 nodes then, right?

15-20 Nodes: Again, Too Large Neighborhood on VPN



- Wrong assumption about OGM scalability:
  - Linear to number of nodes in line topology
  - But squared to number of local neighbour nodes

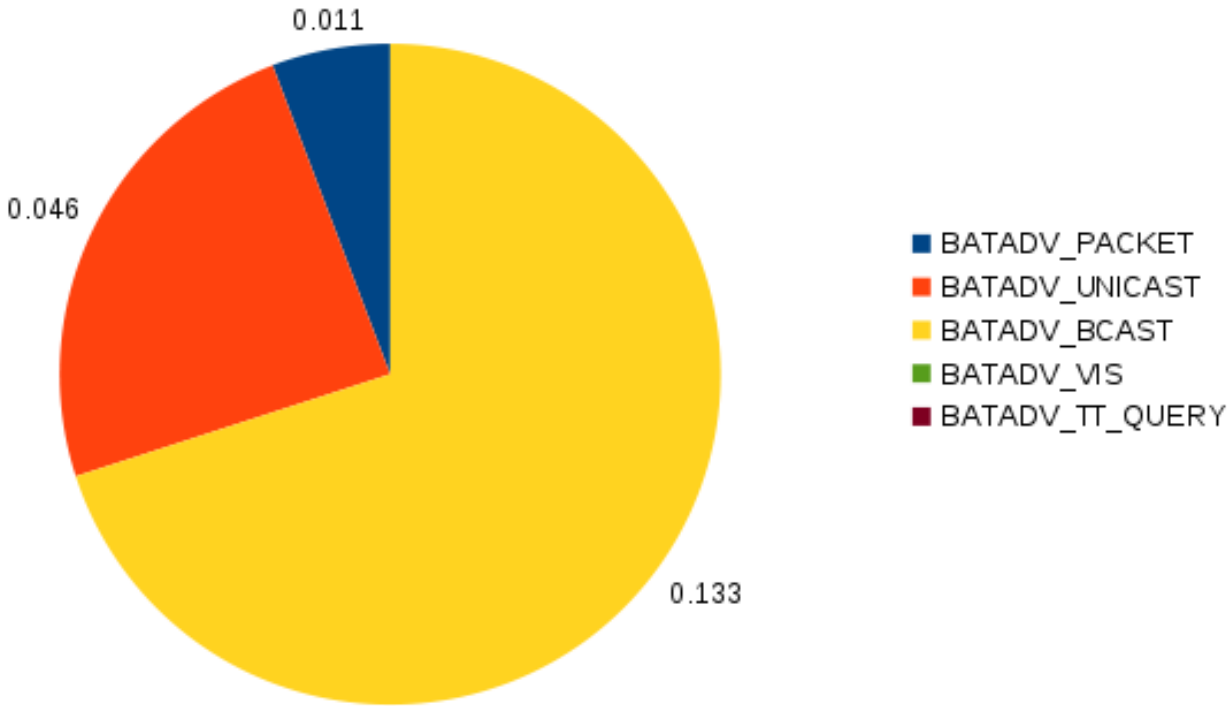
Changes in 0.3



August 2012: fastd

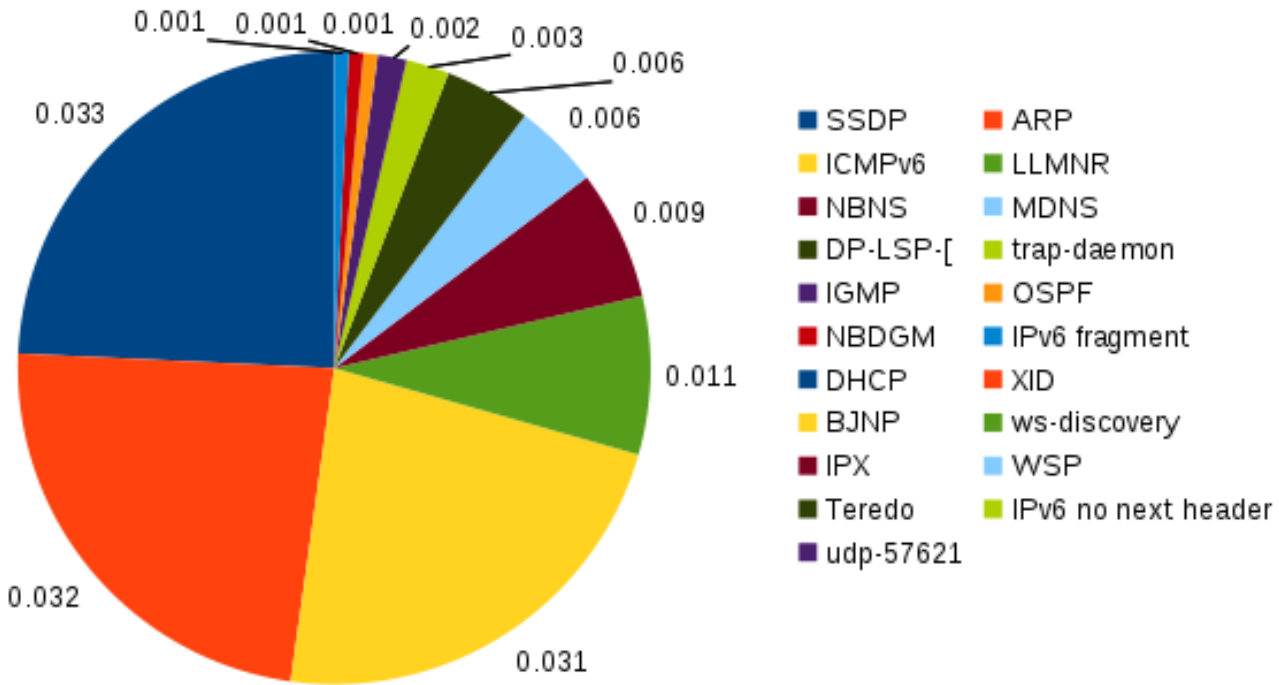
- Decreased neighbourhood size on VPN
- Peer-Groups: Connection to two of n gateway nodes

80 Nodes: Multicast Overhead



- Two wifi neighbours
- Measured on adhoc wifi interface
- Result: Losing about 25% airspace

80 Nodes: Multicast Overhead - Types



- Service Announcements: SSDP, LLMNR, MDNS
- Address Resolution: ARP+ICMPv6

Changes in 0.3.2.1

April, 2013

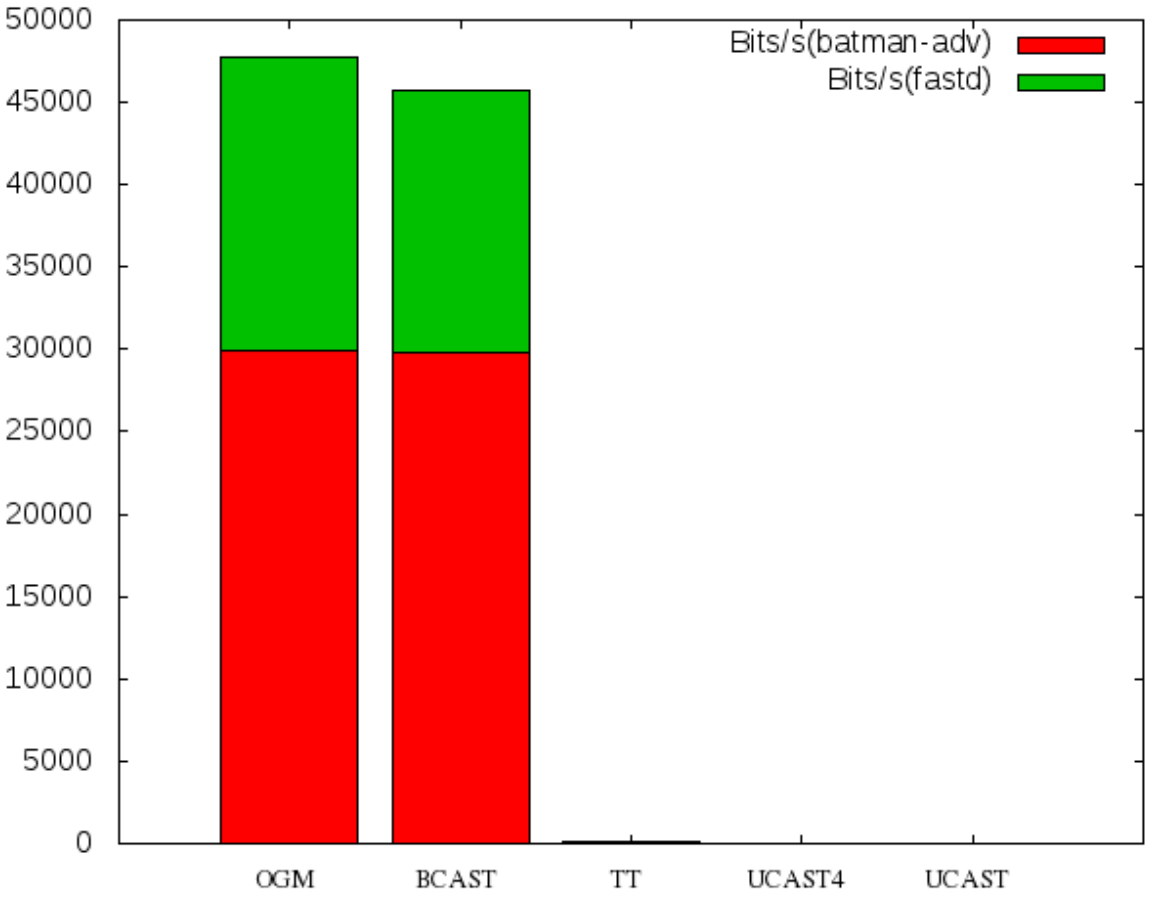
- batman-adv 2013.0.0: Distributed ARP Table
- Multicast Rate: 1MBit/s $\Rightarrow$ 12MBit/s
- Rebroadcasts on VPN: 3 $\Rightarrow$ 1
- Filter for non-essential multicast packets

Outline

Test Setup

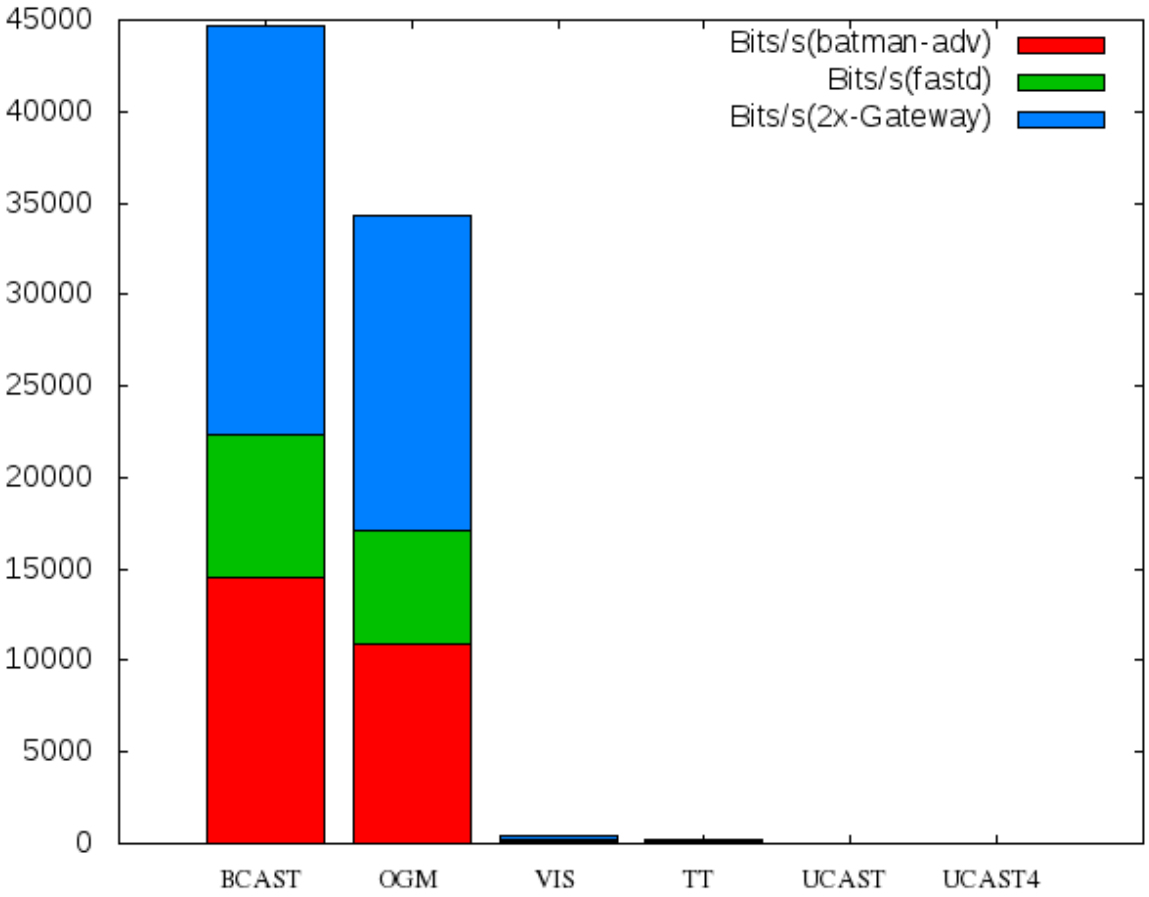
- tcpdump on fastd VPN tunnel interface
- Thu Sep 19 00:00:00 2013 - Thu Sep 19 23:59:59 2013

RX by batman-adv type, average Bits/s



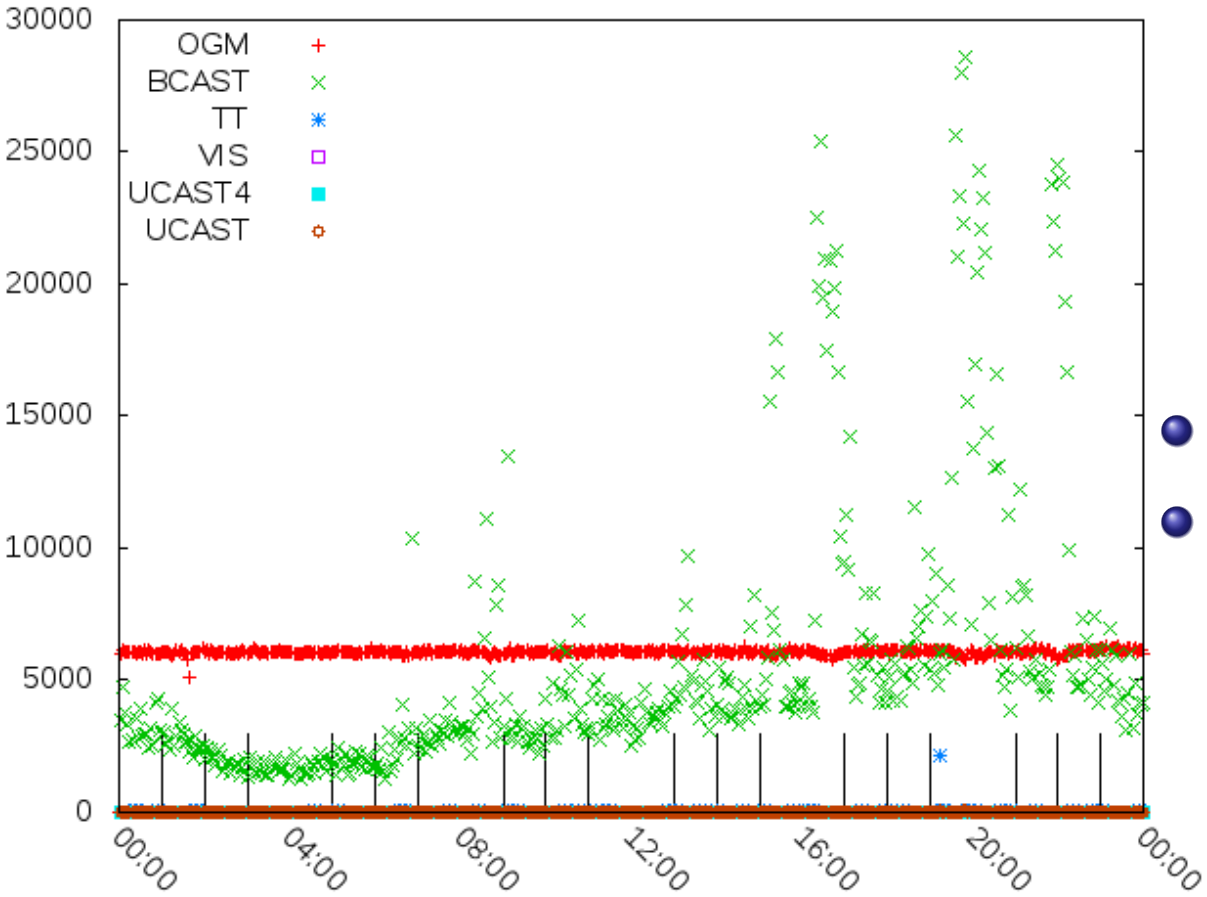
- fastd: +66Bytes headers
- $\sim$ 94KBit/s average
- 1.5% of ADSL, 6MBit/s

TX by batman-adv type, average Bits/s



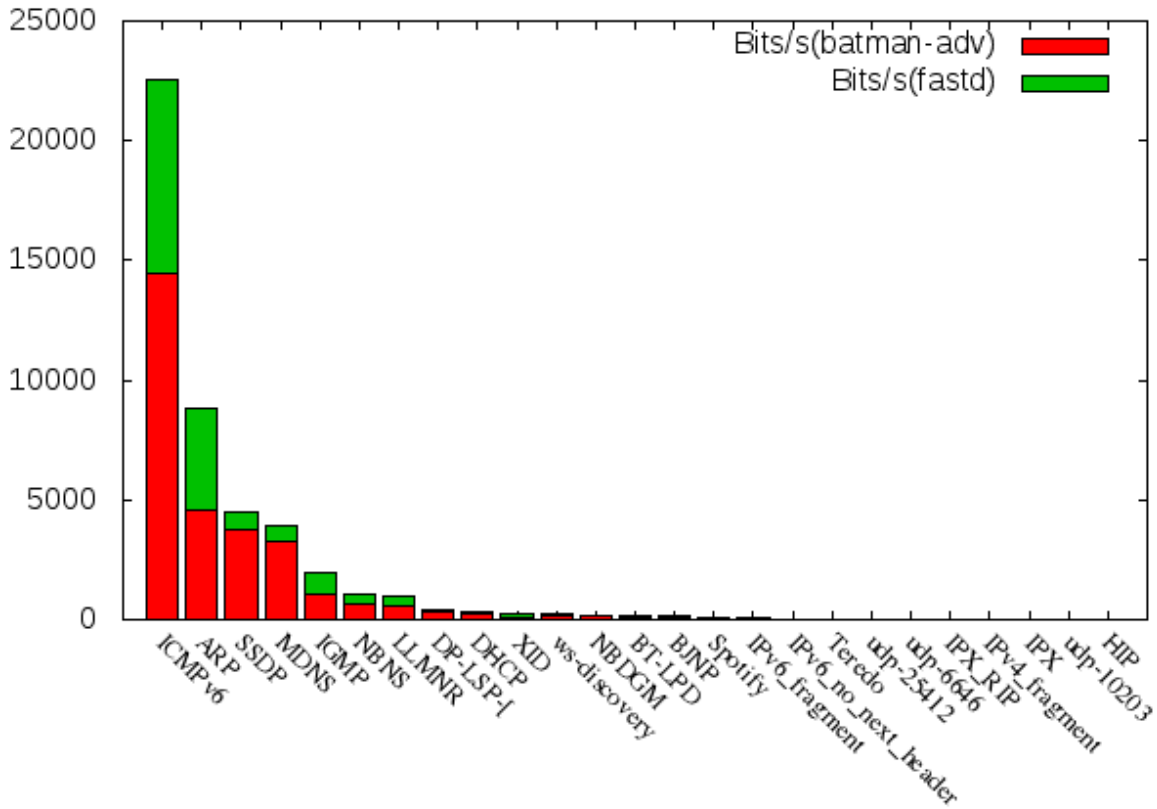
- ∼ 80KBit/s average
- 14% of ADSL, 576KBit/s

RX by batman-adv type, Packets/180s



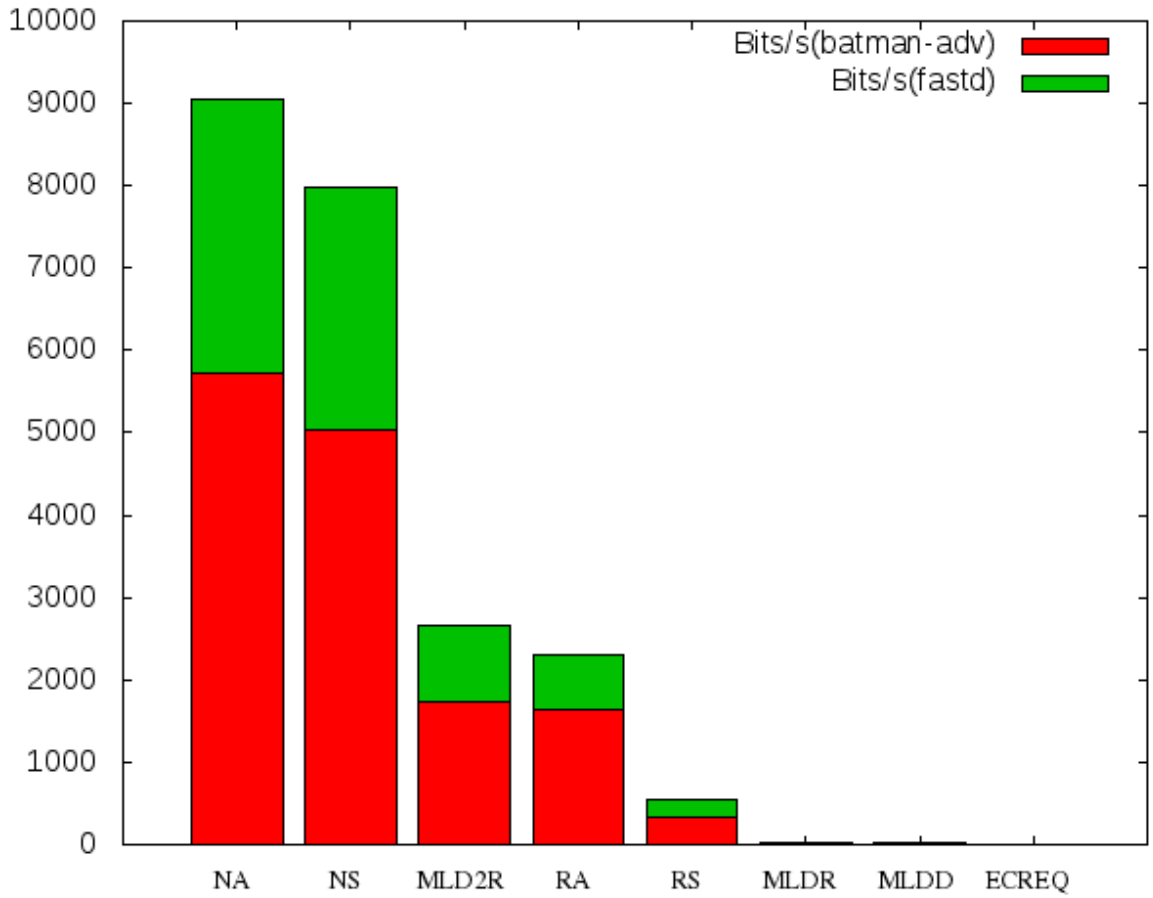
- OGM $\not\propto$ #clients
- BCAST $\propto$ #clients

RX by multicast type, average Bits/s



- Bandwidth hog: ICMPv6
- Many nodes with old firmware:
  - DAT not running ideal
  - Still much: SSDP+MDNS+ NBNS+LLMNR
- Size vs. Rate

RX by ICMPv6 type, average Bits/s



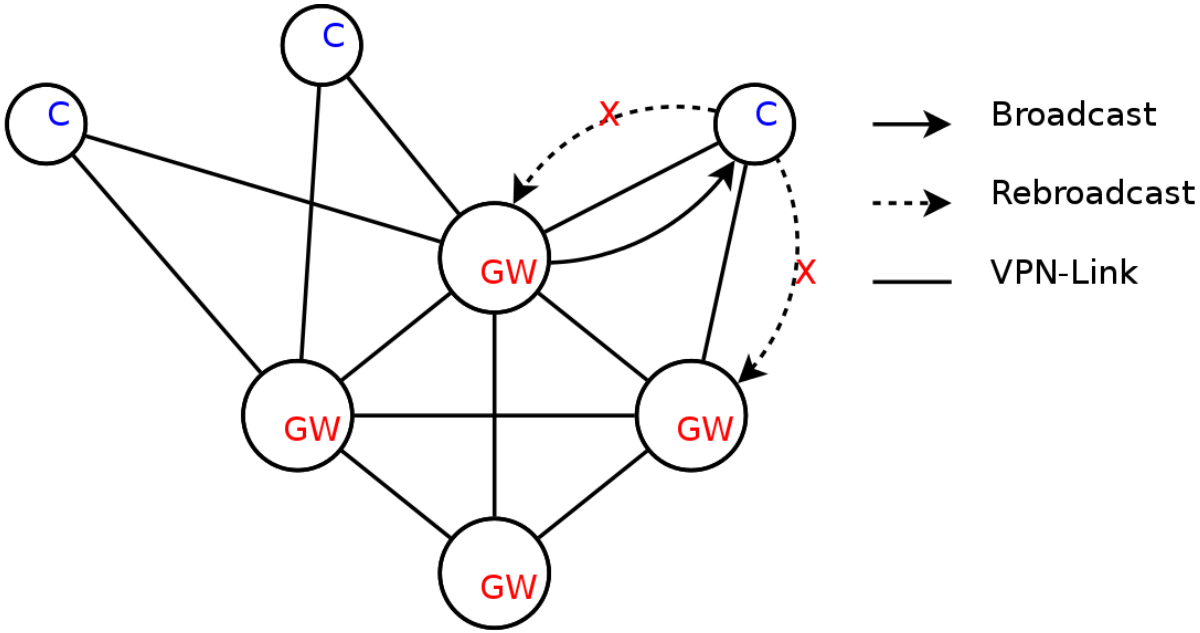
- Most: Neighbor Discovery

Statistics: Conclusions

- OGM and layer 2 specific multicast overhead about the same
- IPv6 ND is currently the largest layer 2 specific overhead
- Getting close to the ADSL upload limit

Outline

”Split horizon” for multicast payload frames



- No rebroadcasts for packets on/from VPN interface

”Split horizon” for multicast payload frames

```
Thu Sep 26 21:14:08 UTC 2013
 eth1  /  traffic statistics

                           rx         |       tx
--------------------------------------+------------------
  bytes                    19.83 MiB  |       10.39 MiB
--------------------------------------+------------------
          max              572 kbit/s  |      340 kbit/s
      average            90.25 kbit/s  |    47.31 kbit/s
          min               32 kbit/s  |       16 kbit/s
--------------------------------------+------------------
  packets                    223391    |          102620
--------------------------------------+------------------
          max                758 p/s   |         395 p/s
      average                124 p/s   |          57 p/s
          min                 38 p/s   |          16 p/s
--------------------------------------+------------------
  time                 30.00 minutes

Thu Sep 26 21:44:08 UTC 2013
```

”Split horizon” for multicast payload frames

```
Thu Sep 26 21:14:06 UTC 2013
 eth2  /  traffic statistics

                           rx          |        tx
---------------------------------------+------------------
  bytes                    19.59 MiB   |         3.24 MiB
---------------------------------------+------------------
          max              488 kbit/s   |        56 kbit/s
      average            89.16 kbit/s   |     14.73 kbit/s
          min               32 kbit/s   |         4 kbit/s
---------------------------------------+------------------
  packets                    222931     |            28916
---------------------------------------+------------------
          max                635 p/s    |           57 p/s
      average                123 p/s    |           16 p/s
          min                 45 p/s    |            7 p/s
---------------------------------------+------------------
  time                 30.00 minutes

Thu Sep 26 21:44:06 UTC 2013
```

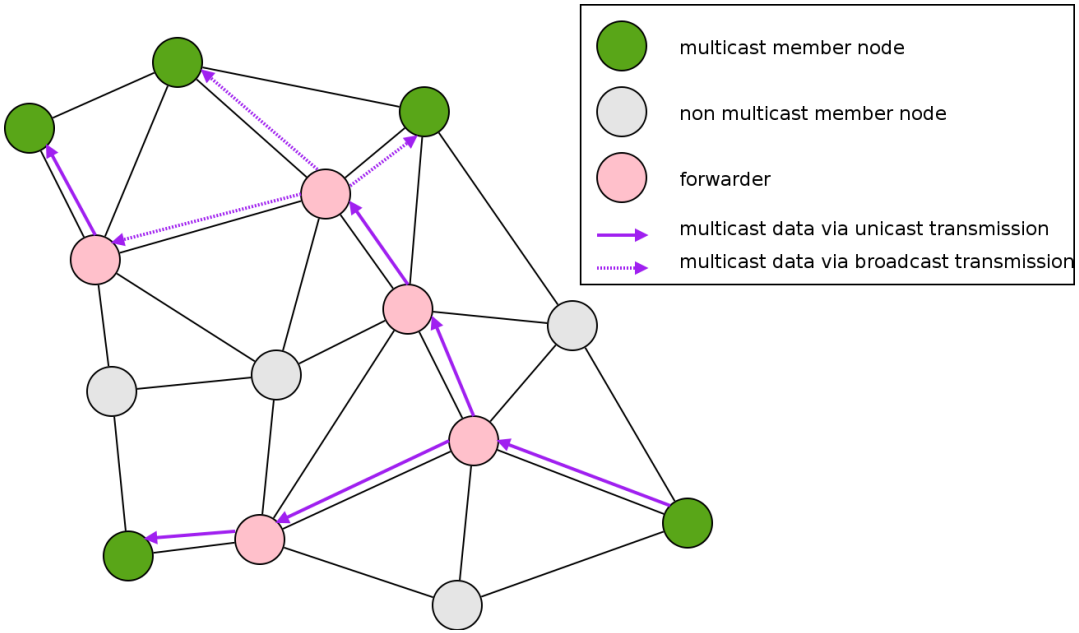
”Split horizon” for multicast payload frames

- Here: 47.31KBit/s vs. 14.73KBit/s
- Eliminates next bottleneck: ADSL upload

Distributed Address Table

- Distributed Hash Table for IPv6 Neighbor Discovery
- Like current DAT (Distributed ARP Table), but for IPv6, too
- Eliminates current largest ICMPv6 overhead:
  - IPv6 Neighbor Discovery
- Status: Patchset submitted, not upstream yet

Multicast Awareness



- Send multicast packets to interested nodes only
- Removes most ICMPv6 overhead:
  - Neighbor Solicitations, MLD Reports, ...
- Status: First basic patchset submitted

B.A.T.M.A.N. IV - Echo Location Protocol

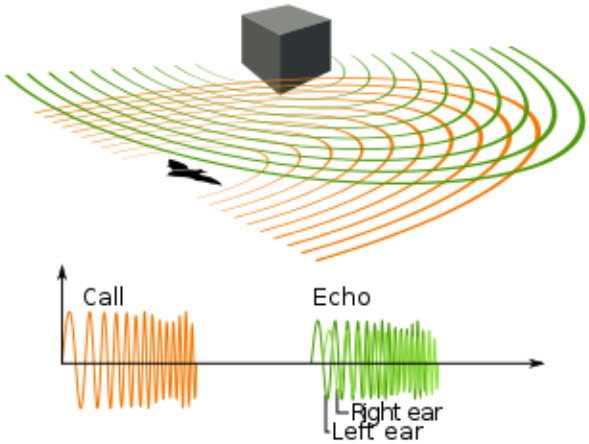


Image Source: Petteri Aimonen, Wikimedia Commons, CC-BY-SA

- Perform link quality measurements with own packet type: ELP
- Reduced overhead through different intervals for OGMs and ELP
- Easier to optimize OGM propagation

Script-Kiddie: Mallory

- Playing with physical or virtual link layer
- Local disaster (mostly?)

Super Villain: Joker

- Knows batman well
- Global disaster

Conclusion

- Current batman-adv algorithm is optimized for sparse networks
  - Keep node neighbourhood small
- Common LL-Service-Announcement protocols:
  - Don’t scale... :(
- With this flat and VPN architecture, batman-adv / layer 2 meshing works with:
  - 80 nodes: without multicast filters
  - 300+ nodes: with multicast filters
- Eliminating Layer 2 specific overhead:
  - Is on the horizon