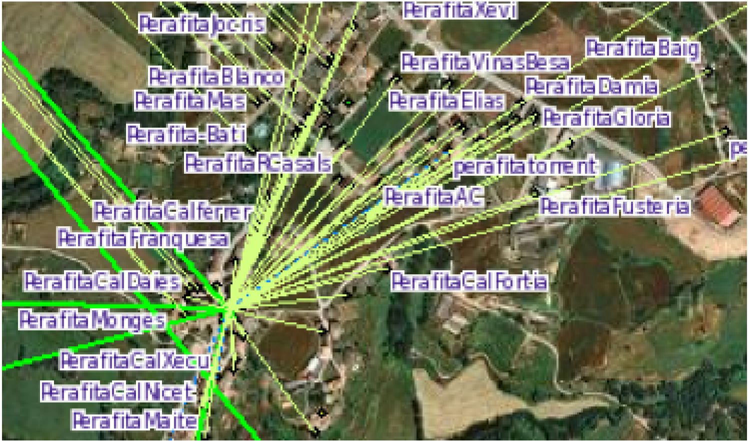
Diversity-aware routing

Juliusz Chroboczek
Univerité de Paris 7
`jch@pps.jussieu.fr`

17 March 2011

Interference in mesh networks

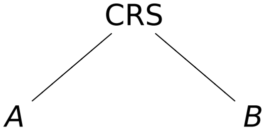
Mesh networks :
multiple radio links in a single geographical area.



These links interfere.

Types of interference

Inter-flow interference (thanks to BP):



Intra-flow interference (thanks to Z):

BabelZ deals with intra-flow interference only.
(For now?)

Intra-flow interference

Due to intra-flow interference, throughput decreases with the number of hops :

A —— B	100 %
A —— B —— C	50 %
A —— B —— C —— D	25 %
$\dots$	
7 hops	$< 10\%$

Throughput stabilises after 7–8 hops, at less than 10 % of the link capacity.

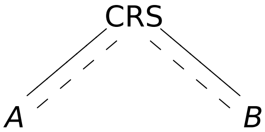
Solving interference

Solving/minimising interference :

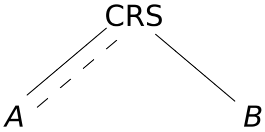
- hybrid (wired/wireless) networks ;
- artificial obstacles ;
- directional antennas ;
- multiple radio frequencies.

Multiple frequencies : inter-flow

If nodes have multiple radios, it is possible to avoid interference:



Easier case:



Multiple frequencies : intra-flow

Easy case:

Difficult case:

Babel and BabelZ

Babel was carefully design to allow

- flexible metrics and
- flexible route selection.

(As a matter of fact, RFC 6126 does not define any particular metric! Example metrics are provided in an appendix.)

BabelZ experimental branch:

- sends channel information in route updates;
- avoids intra interference;
- inter interference is further work.

Links and routes

A link is a single hop :

$$A \longrightarrow B$$

A route is a sequence of links :

$$A \xrightarrow{l_1} B \xrightarrow{l_2} C \xrightarrow{l_3} D$$

We write "$\cdot$" for concatenation :

$$r = l_1 \cdot l_2 \cdot l_3.$$

Metrics

A link l has a cost $\mathrm{cost}(l) \in C$.
A route r has a metric $\mathrm{metric}(r) \in M$.
Operations :

$$\begin{aligned} \oplus &: C \times M \to C \\ \preceq &: M \times M \end{aligned}$$

Given a route r and a link l,

$$\mathrm{metric}(l \cdot r) = \mathrm{cost}(r) \oplus \mathrm{metric}(m)$$

A route r is “better” than a route r' when

$$\mathrm{metric}(r) \preceq \mathrm{metric}(r').$$

The goal of the routing protocol is
to compute the set of routes of smallest metric.

Examples of metrics

Shortest-hop routing (RIP, OLSR-RFC):

- $\text{cost}(l) = 1$;
- $c \oplus m = c + m$;
- $m \preceq m'$ when $m \leq m'$.

ETX (OLSR-ETX):

- $\text{cost}(l)$ depends on (pre-ARQ) packet loss;
- $c \oplus m = c + m$;
- $m \preceq m'$ when $m \leq m'$.

Examples of metrics (2)

Hybrid routing (Babel):

- as shortest-hop on wired links;
- as ETX on wireless links.

Maximise throughput:

- cost(l) is the throughput;
- $c \oplus m = \min(c, m)$;
- $m \preceq m'$ when $m \geq m'$.

Routing properties

Since there are so many metrics to choose from, what are the properties that a metric must satisfy.
A metric MUST be strictly monotonic:

$$m \preceq c \cdot m$$

Intuitively, shorter routes are better than longer routes.
A metric SHOULD be isotonic:

$$\text{if } m \preceq m' \text{ then } c \cdot m \preceq c \cdot m'.$$

If a metric is not isotonic, Babel still converges, but might do so nondeterministically.

Maximising diversity

In order to avoid intra-flow interference, maximise diversity:

- design metrics to be explicitly aware of diversity.

Three approaches implemented in BabelZ:

- avoid hopping on the same interface (no memory) (`-z 1`);
- avoid hopping on interfering frequencies, no memory (`-z 2`);
- avoid hopping on interfering frequencies, with memory (`-z 3`).

Avoid same interface

`-z 1`

Very old idea:
prefer exiting through a different interface:

$$c \oplus m = c + m \quad \text{if the last hop of } r \text{ and } l \text{ are the same interface}$$

$$c \oplus m = \frac{1}{2}c + m \quad \text{otherwise}$$

Avoid interfering — no memory

-z 2

Slight improvement:
prefer a channel that doesn't interfere with the last hop:

$$c \oplus m = c + m \qquad \text{if the last hop of } r \text{ and } l \text{ interfere}$$

$$c \oplus m = \frac{1}{2}c + m \qquad \text{otherwise}$$

Advantages:

- deals with the case when a single node has multiple radios at the same frequency;
- nice side effect: automatically prefers wired interfaces
  (they don't interfere with anyone else).

Avoid interfering — with memory (1)

-z 3

Prefer a channel that interferes with none of the previous hops:

$$
\begin{aligned}
c \oplus m &= c + m && \text{if some hop in } r \text{ and } l \text{ interfere} \\
c \oplus m &= \frac{1}{2}c + m && \text{otherwise}
\end{aligned}
$$

Improvement: deals with the situation when interference is not local, notably with JBOLs.

Avoid interfering — with memory (2)

-z 3

Implementing non-interference with memory requires knowing about the full set of channels taken by a node.

This information is encoded as an extra sub-TLV within Babel's update TLVs. It will be silently ignored by RFC 6126 Babel.

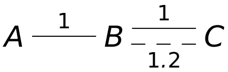
Isotonicity of non-interference

Isotonic:

$$\text{if } m \preceq m' \text{ then } c \cdot m \preceq c \cdot m'$$

Intuitively, it says that Liberal economics works.

All the diversity metrics are non-isotonic:



$$\text{metric}(-) \quad \preceq \quad \text{metric}(--)$$

but

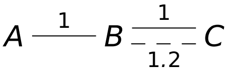
$$\text{metric}(- \cdot -) \quad \succeq \quad \text{metric}(- \cdot --)$$

Next-hop routing

```
while(true) {
      if(there is a better route)
            switch to the better route
}
```

Because of strict monotonicity, this process converges to a local optimum.
Isotonicity ensures that the local optimum is a global optimum.

Local and global optima



$$\text{metric}(-) \preceq \text{metric}(--)$$

but

$$\text{metric}(-\cdot--) \succeq \text{metric}(-\cdot-)$$

- *B* selects the - link; this is optimal for *B*;
- this is not optimal for *A*, which cannot improve its situation.

We have a local optimum that is not global.

Solutions to non-isotonicity

Non-isotonicity can be solved by using multiple routing tables in a single node:

- difficult to implement;
- not implemented yet (?).

In BabelZ, we ignore the problem.

BabelZ will converge to a local-only optimum in some cases.
Deal with it.

Conclusions and further work

BabelZ:

- almost ready to be merged into Babel trunk (but crashed yesterday);
- interoperates with RFC 6126 Babel;
- shown to work well in limited tests;
- first medium-scale test tomorrow?

Further work:

- fine-tuning;
- identify theoretical criteria for isotonicity;
- finalise the protocol and write it up.