

Homenet Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2015

M. Wasserman
Painless Security
C. Hopps
Deutsche Telekom
J. Chroboczek
University of Paris-Diderot (Paris 7)
March 5, 2015

Homenet IS-IS and Babel Comparison
draft-mrw-homenet-rtg-comparison-02.txt

Abstract

This document is intended to provide information to members of the IETF Home Networks Working Group (Homenet WG), so that we can make an informed decision regarding which routing protocol to use in home networks. The routing protocols compared in this document are: The Babel Routing Protocol (Babel) and The Intermediate System to Intermediate System Intra-Domain Routing Protocol (IS-IS).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 6, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	4
2.	Protocols and Extensions Included in Comparison	5
2.1.	IS-IS Protocol and Extensions	5
2.2.	Babel Protocol and Extensions	6
3.	Routing Algorithms	6
3.1.	Link State Algorithm	6
3.2.	Loop-Avoiding Distance-Vector Algorithm (Babel)	6
3.3.	Discussion	7
4.	Convergence Times	7
4.1.	IS-IS	7
4.2.	Babel	7
4.3.	Discussion	8
5.	Autoconfiguration	8
5.1.	IS-IS	8
5.2.	Babel	8
5.3.	Discussion	8
6.	Support for Source-Specific Routing	9
6.1.	Source-Specific Routing in IS-IS	9
6.2.	Source-Specific Routing in Babel	9
6.3.	Discussion	9
7.	Support for Link Metrics	10
7.1.	Link Metrics in IS-IS	10
7.2.	Link Metrics in Babel	10
7.3.	Discussion	11
8.	Support for Stub Networks and Stub Routers	11
8.1.	IS-IS Support for Stub Networks and Stub Routers	12
8.2.	Babel Support for Stub Networks	12
8.3.	Discussion	13
9.	Support for non-IEEE 802.2 link layers	13
9.1.	IS-IS	13
9.2.	Babel	13
10.	Security Features	13
10.1.	Security Features in IS-IS	13
10.2.	Security Features in Babel	13
10.3.	Discussion	14
11.	Support for Multicast	14
11.1.	Multicast Routing in IS-IS	14
11.2.	Multicast Routing in Babel	14
11.3.	Discussion	14
12.	Implementation Status	14
13.	Code and State Size	15

13.1.	IS-IS Code and State Size	15
13.2.	Babel Code and State Size	15
13.3.	Comparison	16
14.	Ease of porting	17
14.1.	IS-IS ease of porting	17
14.2.	Babel ease of porting	17
14.3.	Discussion	18
15.	Behaviour upon resource exhaustion	18
15.1.	IS-IS	18
15.2.	Babel	18
16.	Performance and scaling on IEEE 802.11 Wireless Networks . .	18
16.1.	IS-IS Performance on 802.11	19
16.2.	Babel Performance on 802.11	19
16.3.	Discussion	20
17.	Standardization Status	20
17.1.	IS-IS Standardization	20
17.2.	Babel Standardization Status	20
18.	Evaluation of RFC 7368 Criteria	21
19.	Evaluation of RFC 5218 Criteria	22
19.1.	Critical Success Factors	22
19.1.1.	IS-IS Success Factors	22
19.1.2.	Babel Success Factors	23
19.2.	Willing Implementors	24
19.2.1.	IS-IS	24
19.2.2.	Babel	24
19.3.	Willing Customers	24
19.3.1.	IS-IS	24
19.3.2.	Babel	24
19.4.	Potential Niches	25
19.4.1.	IS-IS	25
19.4.2.	Babel	25
19.5.	Complexity Removal	25
19.5.1.	IS-IS	25
19.5.2.	Babel	25
19.6.	Killer App	25
19.6.1.	IS-IS	26
19.6.2.	Babel	26
19.7.	Extensible	26
19.7.1.	IS-IS	26
19.7.2.	Babel	26
19.8.	Success Predictable	27
19.8.1.	IS-IS	27
19.8.2.	Babel	27
20.	Acknowledgments	27
21.	Informative References	27
	Authors' Addresses	29

1. Introduction

This document compares IS-IS (ISO/IEC 10589:2002) [[ISO10589](#)] and Babel [[RFC6126](#)] according to several criteria related to their use in home networks (Homenets), as defined by the Homenet WG.

[There has been a request that this document compare OSPF as well as Babel and IS-IS. If the WG would find that useful, we would need an OSPF expert who is willing to provide text on OSPF similar to the information that has been provided for IS-IS and Babel.]

Please note that this document does not represent the consensus of any group, not even the authors. It is an organized collection of facts and well-informed opinions provided by experts on Babel and IS-IS that may be useful to the Homenet WG in choosing a routing protocol.

The Homenet environment is different from the environment of a professionally administered network. The most obvious difference is that most home networks are not administered: any protocols used must be robust and fully self-configuring, and any tuning knobs that they provide will be unused in the vast majority of deployments. There may be exceptions to the "no configuration" rule in some environments. For instance, Homenet products may be used in environments where network security is important enough to warrant the configuration of security information, such as keys or certificates. However, even in those environments, minimal configuration should be necessary to deploy a Homenet network.

Another difference is that Homenets are usually built out of a specific class of cheap device, the "Plastic Home Router". A Plastic Home Router's firmware is installed at the factory, and is most likely never updated. Additionally, experience shows that home routers are often used way beyond their warranty period, and even after their manufacturer leaves the router business. This, again, argues in favour of simple, robust protocols that are easy to implement and can be expected to keep functioning without software updates.

A Plastic Home Router is usually equipped with a reasonable CPU but fairly small amounts of flash and RAM. At the time of writing, a typical example of the bottom of the range has a 200MHz MIPS 4KEc CPU (8kB data cache), but only 4MB of flash and 8MB of RAM. More expensive multi-purpose products may have a 700MHz MIPS 24Kc with 32MB of flash and as much as 128MB of RAM.

We expect smaller devices to want to participate in the Homenet protocols. In particular, some vendors have expressed interest in producing sensor-class hardware that is able to interoperate with

Homenet (smart thermostats, home automation hardware, etc.). It is therefore desirable to be able to scale down the Homenet protocols down to a few tens of kB, at least in a stub role.

Homenets are built and grow organically, and often end up consisting of multiple link technologies with widely different performance characteristics (twisted-pair Ethernet at gigabit speed, IEEE 802.11 (WiFi) and its multiple variants, Powerline Ethernet, etc.). It is desirable for a Homenet routing protocol to be able to dynamically optimise paths according to the link characteristics.

Experts appear to disagree on the expected size of a Homenet: we have heard estimates ranging from just one router up to 250 routers. In any case, while scaling beyond a few thousand nodes is not likely to be relevant to Homenet in the foreseeable future, the Homenet protocols, if successful, will be repurposed to larger networks, whether we like it or not, and it is desirable to use a protocol that scales well.

2. Protocols and Extensions Included in Comparison

Both the IS-IS and Babel protocols are updated and extended over time. This section lists the extensions that were considered in this comparison. Additional protocol extensions could affect some of the information included in this document.

2.1. IS-IS Protocol and Extensions

In addition to the base IS-IS protocol specification [[ISO10589](#)], this comparison considers the following IS-IS extensions:

Homenet related specifications

- o ISIS Auto-Configuration [[ISIS-AUTOCONF](#)];
- o Source-Specific routing in IS-IS [[ISIS-SS](#)].

Base protocol specifications.

- o Use of OSI IS-IS for Routing in TCP/IP and Dual Environments [[RFC1195](#)]
- o IS-IS Extensions for Traffic Engineering [[RFC5305](#)]
- o Routing IPv6 with IS-IS [[RFC5308](#)]

2.2. Babel Protocol and Extensions

In addition to the base Babel Protocol specification ([RFC 6126](#)), this comparison considers the following Babel extensions:

- o Extension Mechanism for the Babel Routing Protocol [[BABEL-EXT](#)];
- o Source-Specific Routing [[BABEL-SS](#)], described in more detail in [[SS-ROUTING](#)].

3. Routing Algorithms

IS-IS is a Link State routing protocol, and Babel is a Loop-Avoiding Distance Vector routing protocol. There are some differences between these algorithms, particularly in terms of scalability, how much information is exchanged when the routing topology changes, and how far topology changes are propagated.

3.1. Link State Algorithm

Link state algorithms distribute information for each node to all other nodes in the network using a flooding algorithm. This database of information is then used by each node to compute the best path to the other nodes in the network.

One benefit of this algorithm is that each node contains the full knowledge of the topology of the network. This information can be used by other applications outside the routing protocol itself.

Additionally the flooding algorithm has been found as an efficient method for other applications to distribute node-specific application data, although some care must be taken with this use so as not to disrupt the fundamental routing function.

3.2. Loop-Avoiding Distance-Vector Algorithm (Babel)

Distance-vector algorithms distribute information about the path length to reach each destination through a given neighbor. Packets are forwarded through the neighbor that is advertising the best path towards the destination.

Babel, like EIGRP, DSDV, and to a certain extent BGP, uses a loop-avoiding distance-vector algorithm: it avoids creating a loop even during reconvergence, there is no "counting to infinity", and even short-lived "microloops" are avoided in most cases.

3.3. Discussion

From a purely algorithmic point of view, both kinds of algorithms are known to scale well beyond the needs of Homenet. Issues of scaling over wireless and lossy links is discussed in [Section 16](#).

4. Convergence Times

Convergence time is defined as the amount of time after a link failure is detected during which the network is not fully operational. It does not include the time necessary to detect a link failure.

4.1. IS-IS

Given fast flooding of any change in the network, IS-IS has been shown to achieve sub 200ms end-to-end convergence even in very large provider networks (single area 900+ nodes). Basically the time for convergence is the time to propagate new link state from one end of the network to the other plus the SPF (tree computation interval) and FIB loading time. The flooding is done without delay and prior to running the SPF (tree-calculation) algorithm. Thus is roughly proportional to propagation delay across the diameter of the network. The tree calculation is sub 20ms on modern CPUs. FIB load time depends on the FIB hardware and design and not the routing protocol choice.

[According to Gredler, "today, 1000-2000 routers in a single area are said to represent the upper boundary of IS-IS [...] The authors do not endorse these optimistic area numbers, since a lot is dependent on other factors than just the raw number of routers." (p. 84). And I'm pretty sure he's not thinking of 200MHz CPUs with 8kB data caches, 802.11 and powerline. -- jch]

We easily should expect sub-second convergence for any change in reachability (addition or subtraction) in any conceivable Homenet deployment that does not use IEEE 802.11 for router-to-router links (see [Section 16](#)).

4.2. Babel

Since Babel maintains a redundant routing table, it is most often able to reconverge almost instantaneously after a link failure (this is similar to e.g. EIGRP). In the case where no feasible routes are available, Babel reconverges in 20ms per hop to the source.

4.3. Discussion

Both protocols enjoy fast convergence. However, the time needed to reconverge is dwarfed by the amount of time needed to detect a link failure, which is the hold time in IS-IS (30s by default), and 2.5 hello intervals on Babel wired links (2.5*4s, i.e. 10s by default). (Babel performs link quality estimation on wireless links, so the delay is a little higher when wireless links are involved.)

This can be mitigated somewhat by using smaller timers, at the cost of higher overhead, especially on wireless links, which tend to suffer from abominable per-frame overhead. IS-IS supports hold times as small as 1s, while Babel supports hello intervals as low as 10ms (leading to a 25ms hold time). (Either protocol could of course be combined with BFD, which supports arbitrarily small hold times.)

It has been suggested that physical layer carrier sense could be used to detect broken links in a timely manner. Unfortunately, this technique is not useful in Homenet, since most Plastic Home Routers put their Ethernet ports behind an internal switch in order to provide 5 or more Ethernet ports using a single NIC: carrier sense indicates the status of the internal link to the switch, not that of the user-visible Ethernet link. Carrier sense has also been found to be unreliable on wireless links.

5. Autoconfiguration

Most home networks are not administered, so a routing protocol needs to be entirely self-configuring in order to be suitable for Homenets.

5.1. IS-IS

The only required configuration for IS-IS is a unique area/system identifier. The Homenet implementation of IS-IS uses an autoconfiguration extension defined in an Internet Draft [[ISIS-AUTOCONF](#)], to set this value.

5.2. Babel

Babel is a fully self-configuring protocol -- the standard implementation of Babel only requires a list of interfaces in order to start routing.

5.3. Discussion

When combined with HNCP, both protocols are able to run in an unadministered network (but see also [Section 7](#)).

6. Support for Source-Specific Routing

Source-Specific Routing is a hard requirement for Homenets, as it will allow traffic to be routed to the correct outbound network based on host source address selection. Routing packets to the wrong outbound network could result in packets being dropped due to ISP ingress filtering rules.

Both Babel and IS-IS have extensions for source-specific routing.

6.1. Source-Specific Routing in IS-IS

IS-IS support for source specific routing is implemented with the addition of a sub-TLV to a reachability (prefix) TLV. The implementation assumes that all IS-IS routers have support for the sub-TLV. This assumption is safe to make due to the requirement that all homenet IS-IS routers also use a homenet specific area ID and cleartext password. Mixing in IS-IS routers without support for source specific routing is not supported as it may cause routing loops.

6.2. Source-Specific Routing in Babel

The Source-specific extension to the Babel routing protocol [[BABEL-SS](#)] has been implemented for over a year, has been made widely available and has seen a fair amount deployment as part of OpenWRT and CeroWRT. The source-specific code is currently in the process of being merged into the standard Babel implementation, and is scheduled to be included in version 1.6 (planned for March 2015).

Babel's source-specific extensions were carefully designed so that source-specific and ordinary (non-specific) routers can coexist in a single routing domain, without causing routing loops. However, unless there is a connected backbone of source-specific routers, any non-specific routers present in the Homenet may experience blackholes. Interoperability between plain Babel and Source-Specific Babel is described in detail in Section VI.A of [[SS-ROUTING](#)].

6.3. Discussion

Both protocols have been extended with support for source-specific routing. The IS-IS extension is not able to coexist with non-extended implementations, but this is probably not a serious problem for Homenet.

7. Support for Link Metrics

Typical Homenets are built out of multiple link technologies with very different performance characteristics -- Gigabit Ethernet can easily have three orders of magnitude higher throughput than a marginal wireless link. Both IS-IS and Babel model the notion of greater or lesser desirability of a link by a value called a metric; the smaller the metric, the more desirable the link.

The following example illustrates the importance of assigning suitable metrics to links in a home network:

```

Internet --- A --- B....C          --- is Ethernet
              .           .          ... is WiFi
              .....

```

A is the CPE (edge router), and lives in a storeroom. B is the home's main router, lives in the living room, and is connected to A over Ethernet. C is a wireless router in the guest room, or perhaps in the garden shed. All the routers are equipped with wireless interfaces.

Suppose that the wireless link B..C is solid, but the longer A..C link has very high packet loss. Murphy's law dictates that the link A..C will be just good enough for the routing instances on A and C to become neighbours. If the routing protocol doesn't take link quality into account in its metric computation, even if it is smart enough to prefer wired links to wireless ones, it will prefer the lossy route A..C to the solid route A-B..C.

7.1. Link Metrics in IS-IS

The Homenet implementation of IS-IS uses the wide-metric (24-bit) link metric. Additionally IS-IS includes multi-topology support allowing for a variable number of metrics per link, as well as per-link and per-prefix tags allowing for coloring of links and reachability, and finally per-link and per-prefix sub-tlv's allowing for any future additional extensions.

7.2. Link Metrics in Babel

Since Babel was originally designed for heterogeneous networks, the protocol is able to automatically include a number of sources of information in its metric computation. The current implementation is able to automatically take the following criteria into account:

- o whether a link is wired or wireless;

- o packet loss;
- o link latency [[BABEL-RTT](#)];
- o intra-route radio interference [[BABEL-Z](#)].

This wealth of information can produce conflicting data in edge cases. Babel's loop-avoidance mechanisms ensure that the network remains in a consistent state in all cases, and a hysteresis mechanism ensures that, should a feedback loop occur, the frequency of oscillations remains bounded [[DELAY-BASED](#)].

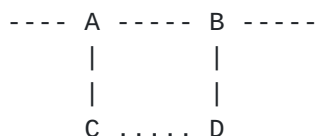
7.3. Discussion

Babel has reasonably good support for dynamically computed metrics for IEEE 802.11 links and high-latency tunnels. The current implementation doesn't have explicit support for variable-speed Ethernet and Powerline Ethernet, both technologies are bundled into a single "good quality link" category.

Current implementations of IS-IS will supply a default metric for links if not configured. We are unaware of any implementation that computes this default metric dynamically, rather it is static and supplied by the vendor. While support for dynamically computed metrics could potentially be added to IS-IS, this is not completely trivial to do right, since a naive approach would cause unacceptable churn, potentially leading to repeated SPF computations and transient microloops. Suitable mitigation techniques could probably be developed, but they would likely be different from the mitigation techniques that work well in Babel, since the link-state algorithm used by IS-IS and the triggered updates used by Babel have fundamentally different dynamics.

8. Support for Stub Networks and Stub Routers

A stub network is one that is attached to a Homenet, possibly through multiple Homenet routers, but must not be used for transit. In the following example, if the dotted link between C and D is a stub network, then it must not be used for transit even if the link between A and B fails:



Similarly a stub router is a router which may advertise and be a destination for stub networks but should never be used for transit traffic.

A number of people have expressed interest in attaching sensor class equipment to Homenets, such as smart thermostats and home automation hardware. Presumably, the sensor-class devices will run over a dedicated low-power link (e.g. IEEE 802.15.4), with a small number of devices acting as gateways between the homenet and the low power network. Ideally, the gateway nodes would not be dedicated devices, but merely sensor nodes that happen to have been equipped with an Ethernet or WiFi interface.

Since low power links have orders of magnitude lower throughput than Homenet links, routing Homenet traffic through the low power link would cause it to collapse. Designating this link as a stub network avoids this issue.

8.1. IS-IS Support for Stub Networks and Stub Routers

In IS-IS reachability (prefixes) and topology (links/adjacencies) are separate things. IS-IS supports stub-networks as defined above simply by advertising the prefix associated with a link, but not the link itself. This is sometimes referred to as a "passive link".

Further an IS-IS router has the ability to set a bit (the overload bit) to indicate that it should not be used for any transit traffic, and that it will only be considered a destination for the prefixes it has advertised, i.e., it is a stub router as defined above. As per the IS-IS standard [[IS010589](#)] an IS-IS router marked as such is not expected to participate in the propagation of link state or the SPF computation.

8.2. Babel Support for Stub Networks

As all distance vector protocols, Babel supports fairly arbitrary route filtering. Designating a stub network is done with two statements in the current implementation's filtering language.

For resource-limited deployments, a minimalistic, stub-only implementation of Babel is available that consists of less than 1000 lines of C code that compile to a dozen kilobytes of text. Just like the standard implementation, the stub implementation is mostly portable code, and should be easily ported to any embedded environment that supports the "sockets" API.

8.3. Discussion

A tiny, stub-only implementation of Babel is available.

A tiny, stub-router-only implementation of IS-IS is available. This is the "tinyisis" referred to later in the document when comparing implementation sizes.

9. Support for non-IEEE 802.2 link layers

Most link technologies employed in a Homenet use the IEEE 802.2 frame format; this is notably the case of Ethernet, IEEE 802.11 and common powerline technologies. However, other framing formats exist, and at least PPP, PPPoE, IEEE 802.15.4, GRE and various VPN technologies are likely to be used in Homenets.

9.1. IS-IS

IS-IS is built directly above the link layer (IS-IS control data is not encapsulated within IP packets), and therefore needs explicit support for every type of lower layer encapsulation. It is not known what particular kinds of framing the Homenet implementation of IS-IS supports.

9.2. Babel

Babel is built above UDP over link-local IPv6, and is able to run with no modifications over any link that supports IPv6.

10. Security Features

10.1. Security Features in IS-IS

IS-IS offers multiple levels of security from none, to simple clear-text (password) authentication, to fully generic cryptographic authentication using any number of hashing algorithms [[RFC5304](#)]. Currently, the Homenet implementation of IS-IS uses the cleartext password set to a predefined value for auto-configuration purposes.

10.2. Security Features in Babel

Babel supports symmetric key authentication using an extensible HMAC-based cryptographic authentication mechanism [[RFC7298](#)]. This mechanism is not vulnerable to replay attacks.

10.3. Discussion

Both protocols support password authentication. This probably provides the necessary hooks for the Homenet Configuration Protocol (HNCP) to implement whatever security policies are required by Homenet.

11. Support for Multicast

Although the Homenet WG has not yet determined whether to support multicast in Homenet Networks, it might be desirable to pick a routing protocol that supports multicast, so that it will be easier to add multicast support in the future.

11.1. Multicast Routing in IS-IS

The IS-IS protocol supports multicast routing. However, none of the available implementations include support for multicast.

11.2. Multicast Routing in Babel

There is no support for multicast routing in Babel.

11.3. Discussion

Some experts believe that it may be better to run a dedicated multicast routing protocol rather than extensions to a unicast routing protocol.

12. Implementation Status

There are Homenet implementations of both IS-IS and Babel.

The Homenet implementation of IS-IS is the only IS-IS implementation that supports source-specific routing, which is a hard requirement for Homenet. If source-specific routing is not required, there are several independent, interoperable proprietary implementations of IS-IS (all major router vendors implement IS-IS). We are not aware of any production-quality open-source implementation of IS-IS other than the Homenet one.

There are multiple open source implementations of Babel, two of which support source-specific routing. All implementations (except the stub-only version) were originally derived from the same codebase.

13. Code and State Size

13.1. IS-IS Code and State Size

The Homenet implementation of IS-IS consists of 7000 lines of Erlang code and has an installed size of over 11MB. Its initial memory usage (as reported by the operating system) is 22MB, and its working set increases by XXX bytes for each new edge in the network graph. To put these numbers into perspective, in a network with XXX nodes each of which has XXX neighbours, the Homenet implementation of IS-IS requires XXX bytes for its data structures.

The code size of IS-IS depends greatly on what aspects of the protocol have been implemented. IS-IS supports multiple address families as well as completely different protocol stacks (OSI and IP), multiple area hierarchical operation with automatic virtual link support for repairing area partitions, and multiple link types. Additionally many other protocol features have been added over time to augment the protocol or replace previous behavior. The protocol lends itself well to not only extension, but pairing down of features.

For Homenet we need a level-1 only implementation supporting a common topology for IPv4 and IPv6 over broadcast (i.e., ethernet) link types. Additionally, we only require support of the latest extended metric TLVs (i.e., not implement legacy metric support).

[Does that mean that we don't care about PPP, PPPeE, GRE etc?]

The operational state required by IS-IS is proportional to the number of routers, links, and prefixes in the network. Each router in the network generates and advertises a Link State Protocol Data Unit (LSP) that describes it's attached links and prefixes. A copy of each of these LSPs is stored by each router in the network. IS-IS uses these LSPs to construct a shortest-path-first (SPF) tree with attached prefix information from which routes to the prefixes are created.

Concrete numbers lacking.

13.2. Babel Code and State Size

The source-specific implementation of Babel, which implements many non-Homenet extensions to the protocol, consists of roughly 10000 lines of C and has an installed size of less than 130kB on AMD-64. Its initial memory usage (as reported by the operating system) is 300kB.

The amount of state stored by a Babel router is at worst one routing table entry for each destination through each neighbour. In the source-specific implementation, one routing entry occupies roughly 100 bytes of memory. To put these figures into perspective, in a network with 1000 nodes, a Babel router with 10 neighbours needs roughly a megabyte of memory to store its routing table (not counting malloc overhead).

The stub-only implementation of Babel consists of 900 lines of C and compiles to 13kB (dynamically linked). Its memory usage (as reported by the operating system) is 200kB, and remains constant (it doesn't perform any dynamic memory allocation).

The stub-only implementation of IS-IS consists of 1300 lines of C and compiles to 18kB (stripped and dynamically linked). Its base memory usage (as reported by 32-bit linux) is 330kB.

13.3. Comparison

Table 1 summarises the sizes of the available Homenet routing protocol implementations. (Babel data courtesy of Steven Barth and Markus Stenberg.)

	babeld (SS)	sbabeld (stub)	AutoISIS (SS)	tinyisis (stub)
Version	2598774	cc7d681	[update]0.8.0	3ed2068
Date	2014-09-08	2014-11-21	2014-08-26	2015-02-22
License	MIT	MIT	Apache 2.0	Apache 2.0
Lines of Code	10000 (C)	1000 (C)	7000 (Erlang)	1300 (C)
Installed size (AMD64)	129kB	13kB	732kB	17kB
Total installed size	129kB	13kB	7MB	17kB
Baseline RSS	~300kB	~200kB	11MB	330kB

Table 1: Comparison of Homenet implementation size

In this table, "Installed size" is the size reported by the package manager for the routing daemon's package(s), while "Total installed size" is the sum of the size of the daemon's packages and all its dependencies, excluding the C library.

The erlang AutoISIS has not been optimized for space or features (i.e., it is a full IS-IS multilevel multi-address family implementation). One example of how this affects the reported numbers, is that the erlang logging package is taking up 4MB of ram.

Additionally, as erlang is interpreted and not compiled as with the C implementations, there's not much use in directly comparing the numbers themselves.

[We should add the size of the native Quagga isisd. While this implementation is incomplete and doesn't support the Homenet extensions, it should give us an idea how much we can hope to scale IS-IS down. It's roughly 20000 lines of C, not counting the additional 20000 for zebra. -- jch]

14. Ease of porting

If successful, the Homenet protocols will be implemented on exotic devices, such as sensor-class devices, set-top boxes and mobile phones. Rather than a full Unix system, such devices sometimes provide a more exotic environment, such as an embedded operating system or one designed for mobile phones.

14.1. IS-IS ease of porting

As IS-IS runs directly over layer 2, an implementation needs to be able to send and receive layer 2 frames itself. On Unix systems, this can be done using raw sockets or by hooking into the Berkeley Packet Filter. Such interfaces might not be available on non-Unix systems, and even on Unix are restricted to privileged processes.

14.2. Babel ease of porting

Babel is layered above UDP. Except for a thin layer interfacing to the kernel, the reference implementation of Babel consists of portable code using the familiar "sockets" API ([RFC 3493](#)). Enabling forwarding and manipulating the kernel's routing tables is the only privileged operation it performs.

Babel has been successfully ported to Android. Android does not currently allow unprivileged code to manipulate the kernel's routing table, so Babel can only run on "rooted" phones. Should a future version of Android provide the ability to manipulate the kernel's routing table, Babel could conceivably be packaged as an installable "app".

14.3. Discussion

IS-IS is somewhat difficult to port, due to the requirement for raw sockets. Except for the requirement to install routes, Babel is portable code, and has been successfully ported to Android.

15. Behaviour upon resource exhaustion

15.1. IS-IS

The IS-IS protocol has an overload indicator. When the protocol is unable to acquire a needed resource, it enters overloaded state. This state is signaled to the other routers and the overloaded router is considered to be destination only (i.e., it becomes a stub router).

15.2. Babel

Babel degrades gracefully. Since Babel is a distance-vector protocol, a Babel implementation can simply drop excess routes when it runs out of memory. As long as the default route is not discarded, connectivity will be degraded but not completely lost.

The current implementation of Babel discards redundant routes before it starts dropping announcements, but doesn't yet prioritise the default route. (This behaviour was tested when somebody redistributed the entire IPv6 default-free zone into a Babel network. We had a lot of fun that day.)

16. Performance and scaling on IEEE 802.11 Wireless Networks

We expect wireless networks, and notably IEEE 802.11 wireless networks, to be in wide use in Homenets, not only for client connections but also for router-to-router connections. In fact, some of us believe that the ability to cheaply and easily extend wireless coverage by adding a wireless router is a "killer feature" of Homenet, one that is easy to explain both to one's boss and to end users.

IEEE 802.11 has some rather unpleasant performance characteristics. First, wireless links are of very variable quality. Second, multicast traffic is sent at a lowest common denominator speed, typically 2Mbit/s, which makes multicast outrageously expensive (13ms for a full-size frame). Third, multicast traffic is not protected by link-layer ARQ, which makes multicast packet drops very common, especially in the presence of collisions, which are particularly likely when the routing protocol is in the process of reconverging.

This has some consequences for routing protocols:

- o the routing protocol must be able to deal with varying link qualities (see [Section 7](#));
- o if the routing protocol uses multicast for transmitting control data, it must deal gracefully with packet loss during reconvergence;
- o the routing protocol must avoid sending large bursts of multicast traffic from multiple nodes simultaneously during reconvergence.

IEEE 802.11 has two main modes of operation. In "infrastructure mode", a small number of "access points" (APs), often just one, communicate with a number of "stations" (STAs); communication between two stations transits through one or at most two APs. In "IBSS mode" (also called "ad hoc mode"), communication is unrestricted, and any node can directly communicate with any other node in range; as there is no link-layer forwarding, IBSS mode does not guarantee that communication is transitive. Virtually all home network deployments today use infrastructure mode, with the router acting as AP and client devices acting as stations. We believe that IBSS may be out of scope for Homenet, but we expect that people will attempt to use the Homenet protocols in IBSS mode, whether we like it or not.

[16.1.](#) IS-IS Performance on 802.11

IS-IS is in active use in the Internet in large non-hierarchical (i.e., level-2 or single area level-1) deployments with hundreds of nodes. The protocol has proven to be very scalable.

We are not aware of any results in the open literature about the behaviour of IS-IS over IEEE 802.11. In particular, we do not know how IS-IS's reliable flooding mechanism behaves over IEEE 802.11.

[16.2.](#) Babel Performance on 802.11

Babel has been carefully optimised for IEEE 802.11 networks. While Babel transmits most control data over multicast, it applies large amounts of random jitter (on the order of seconds) to all but its most urgent control messages. Roughly speaking, Babel sends in a timely manner those control messages that are likely to repair a broken route, but delays sending those messages that merely announce a slightly better route than one that is already available. This mechanism has been shown to work well in dense wireless deployments, with recent versions of Babel being able to avoid link-layer meltdowns even when a whole network is being rebooted simultaneously.

As explained in [Section 7](#), Babel performs link quality estimation on wireless links in a manner that works relatively well with the 802.11 MAC. In addition, Babel has provisions for estimating radio interference [[BABEL-Z](#)], which is necessary in order to provide decent throughput on multi-hop radio routes.

Babel has been designed to work well in IBSS mode, where communication is not transitive and therefore a packet may be routed through the same interface it was received on.

[16.3.](#) Discussion

Babel has been designed with wireless networks in mind, and has been shown to work reasonably well even in the extremely hostile environment of wireless mesh networks built over IEEE 802.11 in IBSS ("ad hoc") mode.

We are not aware of any results about the behaviour of IS-IS's reliable flooding sub-protocol over IEEE 802.11 multicast. IS-IS will not work over IEEE 802.11 IBSS mode without changes, since both the pseudonode optimisation and DIS election assume that communication is transitive.

[17.](#) Standardization Status

[17.1.](#) IS-IS Standardization

IS-IS is an ISO Standard documented in ISO/IEC 10589:2002 [[ISO10589](#)]. There is an active IETF IS-IS Working Group (isis-wg) that maintains and extends the IS-IS protocol, and the IS-IS protocol has been extended in several IS-IS Working Group documents.

The autoconfiguration and source-specific extensions to IS-IS, which are both hard requirements for Homenet, are documented in (non-WG) Internet Drafts [[ISIS-AUTOCONF](#)] [[ISIS-SS](#)].

[17.2.](#) Babel Standardization Status

Babel is documented in an Experimental RFC ([RFC 6126](#)) published in 2011, and it has been updated in several individual-submission RFCs and Internet Drafts. An Internet Draft establishing an IANA registry of Babel extensions has been submitted for publication as an RFC [[BABEL-EXT](#)].

The use of Babel in a Standards Track Homenet RFC would require a "downref" to non-Standards Track documents. It would also be necessary to finish publishing the extensions that are needed for the Homenet use case as RFCs.

18. Evaluation of [RFC 7368](#) Criteria

[Section 3.5 of \[RFC7368\]](#) lists a set of criteria that the Homenet routing protocol must satisfy.

- o border discovery: out of scope, as this is done by HNCP.
- o self configuring: both protocols are self-configuring, see [Section 5](#).
- o behaviour during reconfiguration and reconvergence:
 - * both protocols are able to establish adjacencies and to route IPv6 before they even receive an IPv6 address due to their use of stable neighbour identifiers;
 - * Babel will avoid creating loops even during reconvergence, but might create temporary blackholes;
 - * IS-IS may create short-lived loops during reconvergence;
 - * since HNCP doesn't depend on unicast, in principle it is not impacted by the routing protocol's behaviour during reconvergence; however, on a wireless network, the interference caused by routing loops may delay HNCP's reconvergence.
- o the Homenet routing protocol should be based on a previously deployed protocol:
 - * IS-IS is more widely deployed in carrier networks,
 - * there is more experience with running Babel on Plastic Home Routers
 - * Only the Babel implementation of source-specific routing has seen any deployment.
- o support for IPv4: both protocols are intrinsically double-stack -- a single routing instance is able to carry both IPv6 and IPv4.
- o multiple types of physical interfaces must be accounted for: only Babel is able to differentiate between different link technologies, see [Section 7](#); nothing is known about IS-IS's behaviour on wireless links.
- o minimising convergence time: both protocols converge fast, see [Section 4](#).

- o support the generic use of multiple Internet connections: both protocols support source-specific routing, see [Section 6](#).
- o scalable to higher hop counts: both protocols have reasonably wide metrics (24 bits in IS-IS, 16 bits in Babel).
- o support for attached LLNs and other non-transit networks: both protocols are able to designate a link as non-transit. Tiny stub implementations of Babel and IS-IS are available. See [Section 8](#).
- o support for Multicast: IS-IS is able to carry multicast routes, Babel is not.

[19.](#) Evaluation of [RFC 5218](#) Criteria

[19.1.](#) Critical Success Factors

Does the protocol exhibit one or more of the critical initial success factors as defined in [RFC 5218](#)?

[19.1.1.](#) IS-IS Success Factors

IS-IS exhibits the following critical initial success factors:

- o Positive Net Value:
 - * Hardware cost: None.
 - * Operational interface: Existing and extensive.
 - * Retraining: None.
 - * Business dependencies: None.
- o Incremental Deployment: Yes.
- o Open Code Availability: Yes. One implementation of the Homenet extensions, multiple proprietary implementations of the base protocol.
- o Freedom from Usage Restrictions: Yes.
- o Open Specification Availability: Yes.
- o Open Maintenance Processes: Yes.

- o Good Technical Design: Proven with extensive deployment and experience with the base protocol, little deployment of the Homenet extensions.
- o Extensible: Yes.
- o No Hard Scalability bound: Yes.
- o Threats Sufficiently Mitigated: Yes.

19.1.2. Babel Success Factors

Babel exhibits the following critical initial success factors:

- o Positive Net Value:
 - * Hardware cost: None.
 - * Operational interface: tcpdump and wireshark support, dedicated monitoring software.
 - * Retraining: None.
 - * Business dependencies: None.
- o Incremental Deployment: Yes.
- o Open Code Availability: Yes. Multiple implementations derived from a common source.
- o Freedom from Usage Restrictions: Yes.
- o Open Specification Availability: Yes.
- o Open Maintenance Processes: IANA registry in the process of being created.
- o Good Technical Design: Yes.
- o Extensible: Yes.
- o No Hard Scalability bound: Yes.
- o Threats Sufficiently Mitigated: probably.

19.2. Willing Implementors

Are there implementers who are ready to implement the technology in ways that are likely to be deployed?

19.2.1. IS-IS

There is only one implementation of autoconfiguration and source-specific routing for IS-IS. There are some other open source implementations of the base protocol, but they are incomplete (as of February 2015).

As all major routing vendors have (proprietary) IS-IS implementations, the barrier for implementing IS-IS for Homenet use is probably manageable, assuming that the willingness to implement modifications needed for Homenet use is present.

19.2.2. Babel

The Babel implementation is open source software (MIT licensed), and the codebase has proven of sufficiently high quality to be easily extended by people who were not in direct contact with the author [[RFC7298](#)].

With the exception of a small number of functions used for interfacing with the kernel's routing tables, Babel is portable code, and is easily ported to any environment that supports the familiar "sockets" API.

19.3. Willing Customers

Are there customers (especially high-profile customers) who are ready to deploy the technology?

19.3.1. IS-IS

Yes. IS-IS is already widely deployed in operational networks.

19.3.2. Babel

Source-Specific Babel is currently deployed as part of the OpenWRT and CeroWRT operating systems. Additionally, the current version is used as a testbed for the Homenet configuration protocol.

19.4. Potential Niches

Are there potential niches where the technology is compelling?

19.4.1. IS-IS

[TBD]

19.4.2. Babel

Babel is a simple and flexible routing protocol. Like most distance-vector protocols, it requires little to no configuration in most topologies, and has proved popular in scenarios where competent network administration was not available. In addition, it has been shown to be particularly useful in scenarios where non-standard dynamically computed metrics are beneficial, notably wireless mesh networks and overlay networks.

19.5. Complexity Removal

If so, can complexity be removed to reduce cost?

19.5.1. IS-IS

As mentioned previously IS-IS can be significantly and easily pared down to fit the more limited scope of homenet use. However, no such pared down implementation exists, and the subset of the protocol that needs to be implemented has never been formally defined.

19.5.2. Babel

Babel is a fairly simple protocol -- [RFC 6126](#) is just 40 pages long (not counting informative appendices), and it has been successfully explained to fourth year university students in less than two hours.

The stub-only implementation of Babel consists of 900 lines of C code, and has deliberately been kept as simple as possible. We expect a competent engineer to get up to speed with it within hours.

19.6. Killer App

Is there a potential killer app? Or can the technology work underneath existing unmodified applications?

[19.6.1.](#) IS-IS

As IS-IS already qualifies as successful (bordering on wildly) a killer app is not particularly relevant.

[19.6.2.](#) Babel

Since Babel requires virtually no configuration, it is particularly suitable to scenarios where a dedicated network administrator is not available. Additionally, its support for dynamically computed non-standard metrics makes it particularly appealing in highly heterogeneous networks, (networks built on multiple link-layer technologies with widely varying performance characteristics).

[19.7.](#) Extensible

Is the protocol sufficiently extensible to allow potential deficiencies to be addressed in the future?

[19.7.1.](#) IS-IS

IS-IS has been shown to be incredibly extensible, originally designed for a completely different protocol stack (OSI) it was easily adapted for IP use, then to multiple address families (IPv4, IPv6) and multi-topology. Indeed one of the major drivers of IS-IS's success is its extensibility and adaptability.

[19.7.2.](#) Babel

The extension mechanisms built into the Babel protocol [[BABEL-EXT](#)] have been used to build a number of extensions, including one that significantly extends the structure of announcements [[BABEL-SS](#)] and one that needs a non-trivial extension to the space of metrics [[BABEL-Z](#)].

All Babel extensions that have been defined interoperate with the original protocol, as defined in [RFC 6126](#), to the extent possible. For example, a router that doesn't implement the radio interference extension will just ignore the frequency information attached to route updates, which may lead to slightly sub-optimal routing but will cause neither blackholes nor routing loops. To the contrary, a router that doesn't support the source-specific extensions will silently ignore any source-specific update, and therefore route according to the non-specific subset of available routes, which might cause blackholes, but under no circumstances will cause routing loops. Backwards compatibility provisions are described in Section 4 of [[BABEL-EXT](#)].

19.8. Success Predictable

If it is not known whether the protocol will be successful, should the market decide first? Or should the IETF work on multiple alternatives and let the market decide among them? Are there factors listed in this document that may predict which is more likely to succeed?

19.8.1. IS-IS

For IS-IS the market has already decided that the protocol is successful in a fairly wide variety of deployments.

19.8.2. Babel

Plain (non-source-specific) Babel has seen a modest amount of deployment, most notably for routing over wireless mesh networks and intercontinental overlay networks. Babel is included as an installable package in many Linux distributions, which makes it easily available to interested parties.

Source-specific Babel is probably the only source-specific routing protocol that has seen any deployment. Source-specific Babel is available as an installable package for OpenWRT, and is used by default by CeroWRT.

20. Acknowledgments

The authors are grateful for the input of Steven Barth, Denis Ovsienko, Mark Townsley, and Curtis Villamizar.

21. Informative References

[BABEL-EXT]

Chroboczek, J., "Extension Mechanism for the Babel Routing Protocol", [draft-chroboczek-babel-extension-mechanism-03](#) (work in progress), June 2013.

[BABEL-RTT]

Jonglez, B. and J. Chroboczek, "Delay-based Metric Extension for the Babel Routing Protocol", Internet Draft [draft-jonglez-babel-rtt-extension-00](#), July 2014.

[BABEL-SS]

Boutier, M. and J. Chroboczek, "Source-Specific Routing in Babel", [draft-boutier-babel-source-specific-00](#) (work in progress), November 2014.

- [BABEL-Z] Chroboczek, J., "Diversity Routing for the Babel Routing Protocol", [draft-chroboczek-babel-diversity-routing-00](#) (work in progress), July 2014.
- [DELAY-BASED]
Jonglez, B. and M. Boutier, "A delay-based routing metric", March 2014, <<http://arxiv.org/abs/1403.3488>>.
- [ISIS-AUTOCONF]
Liu, B., "ISIS Auto-Configuration", [draft-liu-isis-auto-conf-03](#) (work in progress), October 2014.
- [ISIS-SS] Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", [draft-baker-ipv6-isis-dst-src-routing-02](#) (work in progress), October 2014.
- [ISO10589]
ISO/IEC, "Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473) Second Edition", November 2002.

ISO 10589:2002 Second Edition
- [RFC1195] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", [RFC 1195](#), December 1990.
- [RFC5304] Li, T. and R. Atkinson, "IS-IS Cryptographic Authentication", [RFC 5304](#), October 2008.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), October 2008.
- [RFC5308] Hopps, C., "Routing IPv6 with IS-IS", [RFC 5308](#), October 2008.
- [RFC6126] Chroboczek, J., "The Babel Routing Protocol", [RFC 6126](#), April 2011.
- [RFC7298] Ovsienko, D., "Babel Hashed Message Authentication Code (HMAC) Cryptographic Authentication", [RFC 7298](#), July 2014.
- [RFC7368] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", [RFC 7368](#), October 2014.

[SS-ROUTING]

Boutier, M. and J. Chroboczek, "Source-sensitive routing",
December 2014, <<http://arxiv.org/abs/1403.0445>>.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405-7464
Email: mrw@painless-security.com
URI: <http://www.painless-security.com>

Christian E. Hopps
Deutsche Telekom

Email: chopps@chopps.org

Juliusz Chroboczek
University of Paris-Diderot (Paris 7)

Email: jch@pps.univ-paris-diderot.fr

