The “mesh-under” versus “route over” debate in IP Smart Objects Networks

JP Vasseur, Cisco Distinguished Engineer (jpv@cisco.com)

Jonathan Hui, Software Engineer (jonhui@cisco.com)

The Internet of Things Workshop – Prague March 2011

Introduction

Routing in IP networks has been a topic of great interest for the last three decades, and a number of routing protocols have been designed and deployed, including IGPs such as IS-IS, OSPF, RIP and EGPs such as BGP. A number of sophisticated routing techniques have been developed to increase routing scalability, decrease convergence times to meet strict SLA requirements, support enhanced functionalities such as Multi-Topology Routing (MTR) to provide different paths optimized for different routing metrics, route tags, etc.

Still IP for Smart Object networks (also referred to as LLN: Low power and Lossy Networks in this paper) exhibit specific requirements and have unique characteristics: large scale networks (with potentially hundreds of thousands of routers), limited link capacity (order hundred kbits/s), constrained nodes (memory, computation, and energy), link and nodes stability, and application requirements for dynamic link/node metric/constraints. This led to the formation of a new IETF WG named ROLL (http://datatracker.ietf.org/wg/roll/) in 2008. The ROLL Working Group produced a new IPv6 routing protocol called RPL (http://datatracker.ietf.org/doc/draft-ietf-roll-rpl/). RPL is fully compliant with the IP architecture and operates at the network layer. The aim of this paper is to spell out the technical reasons for handling routing at the network layer and not at the link layer.

Where should routing take place?

One of the key reasons why the IP architecture has proven to be flexible and robust even while sustaining remarkable growth and innovation over the past three decades lies in its layered architecture, where each layer has evolved independently of other layers (no inter-layer dependency) while still allowing for cross layer optimization thanks to clear demarcations between layers via APIs. IP for Smart Objects networks are no different than “typical” IP networks – they are composed of various link layers technologies that must be tied together with routing functionality at the network layer. For example, although IEEE 802.15.4 was recently seen as *the* link technology for IP for Smart Objects networks, there are now a number of other links technologies used in LLNs, including low-power WiFi and low-power PLC (Power Line Communication) such as IEEE P1901.2. Furthermore, even IEEE 802.15.4 itself represents a number of different link technologies with differing modulations, bit-rates, and channel-coding mechanisms (such as those defined in IEEE 802.15.4g).

Indeed, a new routing protocol could be designed to operate at the link layer. This so-called "mesh-under" approach would imply a multi-layer routing "Architecture" with paths computed at multiple layers. Multiple attempts have been made to utilize different routing protocols at both the link and network layers. A famous example was the attempt to operate a link state routing protocol such as IS-IS or OSPF at the IP layer over ATM VC routed via the PNNI routing protocol. This turned out to be extremely complex, requiring heavy processing on the nodes with a limited success due to a lack of end-to-end path cost visibility, the co-existence of routing protocols with different objective functions and routing metrics, issues related to multi-layer recovery (briefly discussed later in this document), not to mention the complexity to operate and manage two routing protocols.

There are a number of technical challenges of running multiple routing protocols:

- **Lack of end-to-end path cost visibility**: since a path computed at layer N-1 appears as a link at layer N, there is no end-to-end path cost visibility, clearly an issue when trying to compute the shortest (constrained) path end-to-end, and dynamic metric cost update via APIs may be costly and inefficient with unintended interactions and effects.
- **Use of consistent Objective Function**: different routing protocols may make use of different objective functions and/or metrics, making the ability to effectively compute an optimal path end-to-end unachievable.
- **Non-compatible path computation modes of operation**: *what if the routing protocol at layer N-1 uses different set of parameters and triggers to compute its path than the routing protocol at layer N?* For example, the routing protocol at layer N-1 (called routing protocol N-1) may use a reactive mode where the protocol at layer N may be proactive. Ensuring a consistent mode of operation is close to impossible in this case. Even if both routing protocols are pro-active, they may use different triggers to compute a new path/route, different timers to refresh their states, etc.
- **Support of inconsistent functionalities**: MTR (Multi Topology Routing) is a good example. *What if the routing protocol N supports MTR where the routing protocol N-1 does not? What if one of the routing protocol supports constraint-based routing and not the other? What if the routing updates mechanisms make use of low-pass filters using different algorithms?* These are just a few examples where combining two routing protocols operating at different layers support a different set of functionalities makes the network operation extremely difficult or even ***hazardous***.
- **Multi-layer recovery** is yet another challenge with routing protocols operating in parallel at multiple layers. Indeed, when one or both of the routing protocols detect a link or node failure, it is quite challenging to coordinate the rerouting operations at both layers. One may try to use a bottom-up timer-based approach so as to avoid concurrent rerouting; although this approach seems the simplest path to take (by contrast with the use of inter-layer tokens), it becomes difficult to tune timers' values. Setting to T ms the time the routing protocol N should wait before triggering rerouting to provide enough time for routing protocol N-1 to recover is challenging and requires to be able to bound the convergence time at layer N-1. If T is set too large, this does increase the convergence time should routing protocol N-1 not capable of rerouting, and setting T too aggressively may lead to over-reaction and concurrent rerouting.

Of course, this issues discussed so far have been mostly concerned with networks that must operate as transit networks that must forward traffic across different link technologies. But what about situations where the IP for Smart Object network operates as a stub network using a single link technology? Many have argued for supporting the Ethernet abstraction, where a multi-hop mesh network is abstracted as a single broadcast domain. Indeed, there are wildly successful examples of this, including Switched Ethernet and WiFi. In concept, an Ethernet abstraction can simplify the network layer by operating the same IPv6 control protocols such as IPv6 Neighbor Discovery and Stateless Address Autoconfiguration. However, there are a number of challenges extending the Ethernet abstraction model to IP for Smart Object networks.

- **Lack of visibility in to mesh topology:** with a single broadcast domain, any node appears directly connected with all other nodes in the network. Without visibility into the mesh topology, it is impossible to build IP-based protocols that perform in-network processing, data aggregation, and data-aware routing that work in concert with the mesh topology.
- **Link-local scope spans the entire network:** link-local scope is the smallest IPv6 multicast scope beyond interface scope. Communicating with an arbitrary link-local IPv6 address requires the link layer to perform routing functions. In some cases, the application end-point is known to be a node in close physical proximity (e.g. using a handheld field tool) and simply using an IP-based protocol may invoke network-wide route discovery mechanisms even when using link-local scope addresses.
- **Single multicast scope:** any message sent to the link-local all-nodes IPv6 multicast address will necessarily reach all nodes attached to the same link. The inability to utilize smaller IPv6 multicast scopes makes it impossible for IPv6-based protocols to discover immediate neighbors.
- **Variable link characteristics:** unlike Ethernet and WiFi, strict resource constraints often mean that communication between any two arbitrary nodes in the network can have widely varying throughput, delay, and reliability characteristics. Even routine tasks such as IPv6 Neighbor Unreachability Detection and Default Router Selection can experience many of the same issues raised for multi-layer recovery.

In general, many of the issues in supporting the Ethernet abstraction identified above are not problematic in Switched Ethernet and WiFi networks since they are resourceful enough to minimize any visible effects. Furthermore, WiFi networks keep their complexity to a minimum by building shallow star-topology networks.

Conclusion: since the existence of multiple link layers mandates the use of IP routing ("route-over"), designing another link-layer routing protocol ("mesh-under") to find paths that would be "seen" by the IP layer as links seems architecturally the wrong approach and to say the least extremely difficult to operate and manage especially in highly constrained networks experiencing unstable conditions (link flaps, node failures). This paper highlighted some challenging issues experienced in the past with similar attempts in more resourceful environments. Besides the dramatically increased complexity in terms of network management and operation of a multi-layer routing architecture, this may simply not work in IP for Smart Object networks, where resources are scarce and the mode of operation of the routing protocol is critical.