Smart Objects Workshop Meeting Minutes

Scribe: Spencer Dawkins

Links:
- Main Workshop Page: http://www.iab.org/about/workshops/smartobjects/
- Agenda: http://www.iab.org/about/workshops/smartobjects/agenda.html
- Papers: http://www.iab.org/about/workshops/smartobjects/papers/
- Tutorial: http://www.iab.org/about/workshops/smartobjects/tutorial.html

Location of the meeting venue: http://bit.ly/edQArv
Smart Object Webpage: http://www.iab.org/about/workshops/smartobjects/agenda.html

9:00 - 9:15 Opening Remarks and Logistics

Thanking our sponsors, Ericsson, Nokia Siemens Networks and Cisco
Goals are to identify issues where implementers don't have solutions yet, or researchers predict we'll have issues. Identify common interests for possible work in IETF/IRTF
The IETF Note Well statement does not apply to this workshop.

9:15 - 12:00 Architectural Issues

Application Requirements:

Interoperability Challenges in the Internet of Things – Arkko

"Everything that benefits from networking will eventually be networked"…
Tremendous cost and other advantages to using IP, but we have to make sure our technology scales to meet the challenges.
Not a future thing – we're already there. Jari has scales with a wireless interface and his toaster is on Facebook…
Trying to reuse what we've got – may not need more research or standards. Step one is not to redesign the Internet.

Perceived capability mismatch between different devices, mismatches between communication and processing bandwidth. Need to agree on semantics, seeing solutions that are only suitable for some networks.
Are there really capability mismatches, or just mismatches we've created through standardization?
Look beyond "Internet of Things Transport Network" – useful, but not an interoperable Internet of Things – need to agree on what messages mean.
Security as "Authorized Interoperability" – even if two devices talk the same protocols, you can still have problems interoperating.
Specific solutions that work fine in a certain environment, for example an environment with low power requirements. If we have multiple different applications how do we make sure that these work over the same network. Example: RPL using different modes and they do not seem to be interoperable.
Lots of possible actions – standards for applications and data formats, protocols that scale down to IoT devices, architectures, security infrastructures, freedom for innovation. We may need to push back on domain-specific solutions.
Q: – What about management of large numbers of network elements?
Jari: Yes; needs to be added to the list.
Ralph: Gateways and middleware vs. pushing back on domain-specific requirements: Aren't these mutually exclusive?
Spencer: Stress "not optimizing" – didn't NEED to run local-network printers over TCP/IP, but we did it that way…

Application Communications Requirements for 'The Internet of Things' – Baker

Fred is talking about Echelon's view of IoT – paper is from Bob Dolan.
Distributed control – "real time" communications is relative! Normal Internet of communication isn't the same as space command/telemetry, which isn't the same as machine-to-machine control. Sometimes no one dies if things are late, sometimes they do die. This is a network of machines that work together for a system purpose, with real time constraints, and consequences if these constraints don't happen.
Fred showed the fountain in front of the Bellagio hotel in Las Vegas on video, as an example of a distributed control system ("with a conductor for the orchestra"). Link for this video is in Fred's slide deck.
Lots of control points, sometimes addressed as a group. Central controller sets variables, actions are published as status. NACKs may be appropriate when moving large amounts of data, positive ACKs needed for control messages, etc.
Network stack requirements don't all apply to all situations, but these seem to be the common requirements.
Requirements: Retransmission handling (as previously mentioned), network engineering has to meet the application requirements (e.g. bandwidth management), MAC/PHY independence, scaling, assume that the network environment changes over time (10 Mbit OK for today; 100 Mbit may be needed in the future), multicast, duplicate suppression by the network for reliable

multicast, "emergency" messages (in the style of the TCP urgent flag) to flush something aside, in-sequence routine traffic delivery, separate per-peer/per-message timers, polling nodes, and support for peer-to-peer, discovering nodes and node capabilities, record segmentation, versioning, publish/subscribe, and security (for even the smallest and most limited device scenarios).
Q&A:
Steward: You have to make sure that you are able to deal with other paradigms.
Fred: "Execute in the future" model might be useful. We could do this in a 1588 environment.
Pete: Could you talk more about "emergency"? We used to have this control channel - called ICMP. There's a bifurcated channel running over a single network, so how do we do this?
Fred THINKS Bob sees this happening at the transport level, not at the network level. This is built into the transport. You might put a different DiffServ codepoint on each packet.
Stephen: Engineer the network to fit the application. Sounds like there is only one application. Is there more than one application? If so, how do we handle that?
Fred: Everybody either has one application on their network, or they understand what these applications are doing (if they are run in parallel).
Ralph: Does the IETF need to change its operational paradigm to look more at integrated systems across multiple layers instead of looking at individual layers?
Fred: We don't have a good way of doing cross-layer work in the IETF today. We need generalist kind of thinking about what happens where. Fred-as-Bob doesn't think the applications don't care, as long as something lower in the protocol stack does what he needs for his applications.
Q: Is TCP-style retransmission and in-sequence delivery enough for Bob's product?
Fred: We don't have to build this into IP; it is only something that we need to deliver to the application.

Requirements for Building Networks – Nordman

People have been looking at energy and climate applications recently.
Ideal structure (coming from an energy guy) is applications over common IoT infrastructure over standard IETF/IP protocols.
Can't really talk about "Internet of Things" because people have such different assumptions about what a "thing" is. Need to try to subdivide into multiple problem domains – one domain is "building networks".
See two networks – a "Smart Grid", plus a Building Network that just gets electrons and prices from the Grid (and has no additional linkages through the meter, which is a ‘narrow waist’).
Need to support arbitrary communications between two or more devices in a building.
Need location awareness, and identity, so devices know where they are, where other devices are, what the other devices are, and so how to interact.
Need to include people as nodes on the network. Need a common data model. Want "universal interoperability" – a building is a house or an office or a car ….
Location – lights that know they need to dim when a projector turns on, because they are close to the screen.
From Wikipedia – "Identity is whatever makes something the same, or different".

Bruce is looking at EMAN working group for identity needs. An IANA registry for device "species" – so other devices understand what kind of device they are talking to?
"Questions for IAB" is probably somewhat broader than just the IAB…
Q&A:
Cullen: home networks vs very large commercial buildings – can that be "universal"? Don't be separate domains, adapt a common domain for your specific requirements.
Margaret: Would carry the presentation further – same in campus, same in metro region … why do you see borders at all?
Bruce: Probably a question of administrative domains – spanning these changes the problem. Campus could be under one authority, and homes could be part of a condo – need to think here. Want to start with the individual building first.
Margaret: If you talk about an administrative domain then a campus domain is likely to cover multiple buildings.
Another question: Why are people part of the equation in terms of interoperability?
Bruce: Knowing how many people are in the room, knowing what their preferences are. Even if they don't send IP packets … They need to be conceptually be part of the equation.
Q: What about electric vehicles, etc? Part of the premise network, or part of the grid? Vehicle charging is a case for exceptional communication with the grid, but it's exceptional. Vehicle can be thought of as a building.
Q: Distributed energy resources?
Bruce: Try to solve this with a simple pricing mechanisms first. Whatever a building can do internally may not need to be exposed to the grid.
Zach: Is it only about standardizing the data model? What about interfaces? Push back on domain-specific models?
Bruce: I have not looked at other data models. Need standardized terminology for what a 50-percent light level is, what power consumption modes are, etc.
Jari: I said that we have to push back on domain specific things. What I meant is that we need different routing, and different transport protocols for different applications. Of course data models will be different. There are things that need to be different, but the underlying network will be the same.
Antonio: Your slides talk about identity and location. Are you talking about identifiers and location? Have you thought about delegation of authority.
Bruce: Just want universal understanding of what a device is – "I'm a refrigerator, window, light bulb". Haven't thought about delegation yet, although I recognize that it's important.
Q: Small, low-power, battery-powered devices grouped into small networks because you may not want to discover all the light-bulbs in your network, just in the room where you are. You only want to discover the devices in the room and not on the entire campus.
Bruce: Solve the room problem first, and go from there.
Inclusion of pricing so devices can make decisions more intelligently? Doesn't need to be a direct link between grid and devices to make this happen. A given building not have the same pricing priorities that the grid does.
May be worth talking about IP as more than a layer-two protocol.
Some discussion of ASHRAE and OASIS as possible participants in the discussion.

Cullen: You mentioned power pricing. Is the work on power pricing data models the right thing for the IETF to work on ?
Bruce: I don't have an opinion about this. My key point is that building prices aren't grid prices – may prefer non-carbon sources, may have own UPS/generators, etc.
Q: Computer science people have been doing models for years – idea is important, but you'll fail if you try to model everything at once. The idea of the model is important but if you try to model everything then you will fail. Is the job of the IETF to do the modeling? No, it's the IETF's job to develop encodings for models.
Hannes: Depends on how far you go – ubiquitous computing was too big, but we did do location, calendaring and contact address format in the IETF.
Steward: The calendar and location focused on encoding of the specific information in a byte format. This is representation and not modelling.
Hannes: For the location there is an encoding aspect (such as XML encoding), and then you have the data model (such as point, polygon, etc.). OGC on the other hand has gone much further and used these primitives to construct a building.
Bruce: This is my second IETF meeting and there is a great sense of practicality here.
Jari: Three components – the model itself, an encoding, and a way to access the model. Questions about whether IETF needs to do additional work on these independent in these components. I suspect that we are missing the actual model. Who has the expertise? Maybe it is the energy companies?
Bruce: Don't design around energy as a primary goal. People want it to be functional. Design around functionality for people and energy is a nice side benefit.
Q: If you take an abstract model and you represent it in XML then you get something pretty complex. You want simpler models and develop them from bottom up.
Q: You may have elements in the model that are only used for linking other elements, and never send or receive a packet. Need to accommodate this.
Bruce: These are difficult problems. We don't want to rush and do something quickly that we have to live with for decades.
Margaret: Two different things going on. One is about putting small devices, who had been using proprietary networks, now on IP networks is separate from whether we put it on the same IP network as other applications. Controlled applications, like the Bellagio fountains, probably aren't on the same network as the Internet traffic of the hotel guests. In what sense are we talking about one IP network or about using IP technology for a separate, walled garden network that is independent of the Internet?
Jari: We have lots of applications running on a one network today. I am very much of the opinion that we should have one network running multiple applications on it. The point of the whole thing is to have a common technology and a common network: one set of addresses, switches, routers, etc. Want to benefit from one network, not just from using IP.
Fred: I am going to say exactly the opposite. Running many applications over one networks is a good thing, but there are going to be separate networks. I don’t think we should fixate on having one network, fixate on developing technologies that work for both. Smart grid applications will run on a dedicated network, full stop.
Ralph: I agree with both Fred and Jari. Done. I think it depends on which parts of the elephant you are poking around at. Jari's view of the same network is the same underlying technology

and it is possible to put the building sensor's through the same wireless access points or the same fundamental infrastructure, if you want to, or if it make sense to get separation at the physical layer. I agree with Jari to make sure that we don't have different technologies in different networks that forces us to loose interoperability. The reason that this stuff is going to take off is because to have of the interoperablity Jari was talking about. Bright people take an iPhone and build an iPhone app and connect, with the appropriate security controls in place, to the things in my house in ways we have never thought about before. Otherwise we are just building another telephone network.
Bruce: IT devices and energy devices need to talk to each other – turn the fan down when I'm on a Skype call. There really are industrial sensor networks that are fundamentally different and separate – that's not what we're talking about here.
Q: How can we prevent new standards from causing interoperability problems? New standards will have complex interactions with existing standards, requiring complex gateways before they can interoperate.
Jari: You will definitely have new link layer technologies and different routing modes. We have technologies for for bridging between technologies (e.g. IPv4/IPv6). But we have to be careful when we create new standards. We need to make sure we're not constructing islands.
Cullen: Going back to Margaret's question. Think back about how VoIP got deployed. Initially, many vendors recommended to run VoIP over a separate VLAN or a separate infrastructure. Nobody could imagine how to make the type of real-time guarantees, how to debug it, how to get it to work because the Internet is not ideally suited at making any types of guarantees for real-time systems. As time went on people got better at making it work across that type of IP network they suddenly noticed that having it on a separate virtual network than their other applications was a disaster. They couldn't decide if a PC is running a softphone and whether it went on a voice or a data network. Suddenly it became to converge and all moved to one. I wouldn't be surprised to see the same happens here. Initially, we will see very separated networks. Then, those will be running over the same hardware to take advantage of the cost benefits of not having to deploy multiple sets of wires around buildings. Over time there will be strong needs to directly communicate with each other. We need to be designing the system for the long run. I assume everything will end up on the same network even if we wish they wouldn't.
Bruce: On the notation of the constrained devices we have two types: First, there are devices that are less capable than standard Internet hosts. Second, we have devices with application requirements exceeding the standard requirements. Can we expose constraints through the stack (for example, latency requirements) up to the application? Is that something IETF could do?
Q: In the stages of the early requirements steps for RPL we identified the need to constraints and the need for establishing different topologies over the same medium in order to route different flows who have different constraints. We realized that our link layer media do not have the concept of VLANs at all. We really need layer 3 abstractions that would replace the VLAN. We have MPLS but we cannot run it there.
Q: We will have different networks for different functionalities and capabilities. VLANs or overlay technologies could satisfy these requirements, as mentioned by the previous speaker.

Gateways would play the role to connect these different networks. Would IETF standardize middleboxes and gateways?
Paulo: Doing a ping-pong match here – Internet is about supporting different capabilities now. Would like for my laptop to reach a sensor – have everything on the same network.
We're looking at a group of devices that share constraints.
Bruce: People commonly think that sensors are separate devices. I, however, believe that a lot of sensing is done by devices that are primarily serving some other purpose. Everything is becoming a sensor. For example, Apple computers with cameras could report light levels. Devices may routinely incorporate some temperature sensors. One could just ask all the devices nearby for their temperature status with the benefit of obtaining multiple data points.
Margaret: The notion of location as a network service; the network providing location and proximity. I understand how the network can tell you that certain devices are in the same sub-network or via triangulation that they are within a certain location. But I don't understand how network can tell you that a light sensor and a lamp are in the same room unless you have coded this into the devices or into some central controller. Could you comment?
Bruce: Can come from the network or from the devices itself – either way, but it's a critical piece of information to have. Have lights mapping their own proximity when other lights turn on now – that's not the future. Bruce provides information about a company offering light bulbs that have sensors to determine the light situation. Similar things can be done using sound detection as well.
Theodore: Simplicity is helpful, but important to see the limits of simplicity. We may see cases where this cannot be a unified network. Question whether we need some data aggregation in the network instead of asking each sensor for the temperature. Example: A laptop is one device but it can have many sub-devices (microphones, cameras, etc. of laptop). You typically want to talk to the laptop rather than the individual components. Need to include this concept.
Q: Going onto a separate network is often because of requirements for security and availability. The fountain, for example, will never be connected to the Internet. The same is true with privacy. Q: It's not the network's job to support applications directly. There is a higher-layer of abstractions. Have a service layer that presents APIs that allow devices to cooperate with other devices – and the service layer is not the sensors themselves. For example, the application layer would provide the concept of a room and then you could control the light bulbs in the room.

Data Driven Architectures:

. Pub Sub based Naming – Burke

Stories and experiences are now created with Systems that are built on Architectures.
"Named Data Networking" project, based on content-centric networking (van Jacobson).
Interest packets and data packets.
How do we bridge private control applications to public ("social") networks?
Name-based addressability and control is empowering people who are working with IoT today.
"Named Data Network" – use the names that applications use, as what's actually routed in the network.

Q&A – how do you turn a light off and on? There's an interest packet that says "I'm interested in turning this light on", and the light replies "yes, I turned on".
Can you accept a changed IP address? It's an IP overlay now, but one goal is to hide IP changes – not hard to do in networking, but authentication is hard to get right.
You're IP-independent, but what's the advantage of the name-based approach compared to just translating it? A good question for Van J but this helps with broadcasting channels for discovery/configuration. You end up with a disconnect between the way you configure your network and the way people address and understand devices.
Experience on scalability? NSF project is interested in precisely this. We're exploding the namespace and routing at scale. We're past the building on scalability, though.
What's the human interface between these different namespaces? Today, I'm interested in … tomorrow, that's different … Dirk assumes that we'll have local names that aren't globally routable. Can have a variety of names (some globally routable, others not). But you need to think about the relationship between all these namespaces. "This lightswitch" isn't a globally unique name, but you need to resolve it to turn on the light!
You're not just passing messages on – you're transforming what comes in, before it goes out! This is aggregation and republishing – where that happens is an ongoing discussion. This project thinks that doesn't happen in routers because it would be too application-specific.
Previous question - What is special about the theater environment? Is there a reason why name-based networking works better there? If this isn't IP-based, there are a lot of problems that pop up (how you ensure unique names, etc.). But we don't have to think about IPv4 addressing, and we can use names that are semantically meaningful. But if everything doesn't do name-based networking, you have ecosystem problems.
The author wants to work with names that are meaningful in your application – that's cool, but you're saying that there's an advantage when you route based on those names – why? Actually, a designer could rename lights for every scene in the play, but that's at one extreme. Every time you cross networking domains, you build up scaffolding that handles the underlying networking, and that means you experiment less because each experiment is harder. Sometimes we even write code with people on stage (nerve-wracking!).
It's different when you have devices. When something changes, there's a network configuration step, and networking configurations are brittle – tend to be statically configured, and that makes experimentation harder.
Also relying on content caching in their project (but that's not name-based networking).
Do you use attributes in the name, or is the name an opaque identifier with associated attributes? That's a conversation that's still happening.

. Information Centric Networking – Kutscher

Does anyone remember the Wireless Application Protocol (WAP)? They were special, too … J
"Internet of Things" is about funding – the Internet has always had "things". The question is "how many things".
The definition of "challenged" changes over time. We're more concerned about battery technology evolution than about storage and processing evolution.
Want to extend the diameter of the Internet to include "networks of things".

Using RFC 5050/DTN architecture. Don't want to have application knowledge at network domain boundaries – should be end-to-end.
Evolving RFC 5050 transport. Developing a naming format for object identification.
Want to mix capable and constrained nodes at the application layer, without paying attention to which is which.
Don't develop the NGN/IMS of Things!
Q&A: draft on names? Do you know how routing will work? Actually, there's not a relationship between this naming draft and routing. We plan to do more with name resolution.
Object transport – could you describe differences? Want to support local/hop-by-hop transport when that makes sense.
Domain-specific routing – is there inter-domain routing? Sure, if you want something like Internet-style connectivity. But research on how to do this is still ongoing.
If you have new transport, naming and routing, is this still the Internet of Things? We're still using the same fundamental underlying mechanisms…

. Discussion

13:00 - 14:00 Lightweight Implementations, Sleep Modes, Data Formats

Sleep Compatible Protocols – Wasserman

Focus of the presentation is on small low cost battery powered devices that sleep 99%-99,5% of their time. Idea is to extend battery life of a watch or single AAA battery for months. "Every bit is precious" … and every round trip …
For example, a light sensor regularly goes online to check whether it is day-time or night-time. Most part of the system is off-line most of the time. The communication components are off (e.g. WLAN) and Only very few components of an embedded system are on periodically

A number of issue arise from these always-off nodes. Many IETF protocols assume that nodes are always on and respond to messages. Examples: Neighbour discovery and duplicate address detection, DNA, default gateway. Typically you have to do a lot of stuff before you can send a packet. Protocols also get really chatty. These protocols often do no care how large the messages are.

View is that the "low-power/battery" principles are widely applicable – laptop/cell phone batteries would live longer, even wired nodes would use less energy.
Need to use less power to turn off your lights than you save by turning them off!
Q&A: Is LWIG a possible place for this? They're doing implementation guidance; we're probably about at least removing MUSTs.
Cullen – we've always been against profiling, but there are cases where we should do profiles, especially if things still work.
But making things optional adds complexity – need to really cut stuff out.

Profiling can be harmful – sometimes you really do break things.
IETF already has a non-working group mailing list on using less energy in protocols.
What about buffering until a node wakes up? Don't want to receive all the multicast/broadcast packets that were sent since the last time you woke up!
Can you signal over the radio to wake up the node? No, the radio is completely shut off when the system is sleeping. Receiving consumes more power than sending (however counter-intuitive that is).
Need to look at sleeping nodes systemically – not just one protocol. Can't solve all the problems at one level of the protocol stack.
For a lot of sensor networks, we have really long duty cycles – maybe once per hour. You really can do more than you think, if you're only staying awake once an hour. But Moore's Law isn't your friend if people add computing power at a constant cost and burn through all the improvements in battery time. We're talking about people wanting to run for a year on a 1-watt battery.
You can't send a message and wake up for the response, because you don't know when the response will come.
Limitations of proxies? Not clear how to age out the entries that you're proxying for – can be telling more and more nodes about devices that don't exist anymore.
Can do very low-level radio capabilities to wake up devices, but if there's really a network with traffic, you'll wake up a lot.

. Home Control in a Consumer's Perspective – Brandt

Customers have products now. If the next generation gets half the battery life because we went with IP, they'll notice!
Many application protocols, millions of devices, limited integration …
Just going to IP isn't enough – you still don't have interoperability between protocols.
Don't try to solve the problem for every possible application – 85 percent would be an improvement!
Proposing incremental steps…
Need a discovery protocol to advertise local resources, resources in other subnets, legacy devices in oother subnets, and sleeping nodes. Don't want to rely on multicast for this (battery-powered devices won't respond anyway), but want to support multicast if it's supported.
Dreaming of application compatibility for M2M style command sets.
Need a sufficient subset, and need input from industry alliances.
Want backbone routing by default (and whatever happened to HomeGate?) and ICMP "Destination Responding Slowly" message ("so don't keep retransmitting, because it will probably still be asleep when you retransmit").
Q&A: There is an MDNS extension to discover on multiple subnets.
We're creating mirrors for sleeping nodes – battery-powered device simply reports to a powered "mirror" device, and anyone else can pick it up from there.
ICMP message is interesting, but we need to look at the semantics of this, and how it will propagate up and down the stack.

. Discussion

14:00 - 15:00 Security

. Security Challenges for the Internet of Things – Polk

Need "mandatory to implement" security. Yes, these devices will be initially deployed with no security, but when people see problems, there needs to be security that you can turn on!
We do see some plausible current algorithms (AES-GCM, ECC, maybe SHA-3), but we need crypto agility, and we may need to develop new algorithms (with at least 112 bits of security strength, since attackers aren't resource-constrained).
Automated key management is hard, and credentialing is harder. Pre-shared keys aren't a realistic option. May need to be innovative (leap of faith or pairing protocols to efficiently introduce new devices).
SmartGrid encountered privacy concerns serious enough to impede deployment. One European country rolled their SmartGrid deployment back because of privacy concerns. The IoT will be worse.
Useable credential management is the challenge.
Q&A: Mandatory-to-implement, optional-to-use – is this reasonable to implement in practice? Tim thinks possible but painful (security already is). Design for security needs to happen early. Footprint of multiple security mechanisms may be a problem – can't support both old and new mechanisms on one device.
Is privacy the same as security? Is it a byproduct? Some mechanisms are about opt-in/opt-out and helping users make good choices.
Alyssa quote here … need to check.
“Privacy by design” concept should be considered along with “Mandatory to implement, but optional to use security”. Privacy per se is NOT a byproduct of security. For this reason, it should be treated separately (similar to security, considered already at system design time).
Jari - concerns about premature optimization apply here. Need to be aware of IPR concerns about optimized mechanisms – knocks out open source, difficult for commercial vendors. Need actual implementation experience (hearing "can't do it" and "can do it, I've implemented it" at the same time).
Two purposes of security, and one is locking yourself out of your own devices – this is a bigger problem when you deploy lots of low-end systems. Things have to plug and play (pairing protocols in Bluetooth, etc.). Networks are going to be dynamic – you're going to replace devices and you have to be able to configure the new devices. Can't be harder than pairing your phone with your earpiece.
We have no stick – "mandatory to implement" isn't mandatory!
What about AES-CCM? Other systems use that … NIST likes AES-GCM, and includes it in Suite D, but if AES-CCM wins, that wouldn't be a problem for them. AES-GCM works better for pipelining – not clear how much that matters for IoT.
See security and privacy differently – security protects your data, privacy protects your identity. Tim would have to think about that.

Utility customers that absolutely refuse to use ECC because of vague vendor claims about IPR, others won't use RSA because of key sizes, and you don't like shared keys – what do I do?
RFC 6090 was supposed to clarify the ECC problems.
What about trust on the network side as part of IoT – do you trust the next network, etc?
Security of routing is an issue, need to keep this in mind, but expecting ROLL to think about this.
ZIGBY pushing through AES-CCM – want to use the same building blocks as much as possible.
Remember that we need to hide complexity from the user – if we don't do this, we're already through.

. IKEv2 and Smart Objects – Kivinen

Everything needs security – even garage door openers!
Protocol is simple, and device only wakes up when a human pushes the button.
Would be an IKEv2 Initiator, never a Responder. One IKEv2 SA, and one IPsec SA. Don't need to manage SAs, do NAT-T, EAP authentication, Cookies, etc.
Preshared keys? Raw RSA keys?
Implementation experience has been that IKEv2 minimal subset is a very small protocol. 44K lines of code for full IKEv2, 1K lines of code for minimal subset.
Q&A: What does the server side look like? Used their full IKEv2 protocol implementation for that.
If you use preset keys, and use your garage door opener at home and at work, you're handing out preset keys in both cases – that's not good!
What's the non-volatile memory size for the minimal subset? Don't know, used PERL. Can't make an estimate here.
X.509 – is there an opportunity to do something much simpler? We're talking about maybe 12 useful fields …
Would you recommend IKEv2/IPsec? DTLS? Security in the application? Don't do proprietary!
Robert reported implementation size very close to what Temo is reporting, thinking that code size was low 10s of Kbytes.

15:30 - 16:30 Routing

. IP (RPL) vs. Link Layer Routing – Vasseur

"Mesh-under" versus "Route-over", where every PHY hop is an IP hop.
"Mesh-under" can be any link layer that includes a routing function.
IPv6 ND assumes deterministic link characteristics – these aren't.
Neighbor unreachability detection has to work over multiple LLN link hops. What timeout to use?
Latency and reliability can vary greatly.
Do we expose link-layer path cost when selecting a router?
Any IP traffic can invoke costly operations (any link-local traffic can invoke L2 routing functions, link-local multicast can span the entire LLN).
"Support of multiple PHY/MAC is a must" = IP routing …

We've been here before – IP over ATM. Routing had no clue what the underlying network topology looked like, which made shortest-path routing very difficult.
Combine "Mesh-under" and "Route-over"? Multi-layer recovery is really hard.
We see no advantages of "Mesh-under", only drawbacks…
Q&A:
We've been doing routing at multiple layers for a while – are you saying there's no use for a tunnel? Once you've built a tunnel, you've constructed a virtual link, with no indication if it's multi-hop.
Robert – NO advantages of "mesh-under"? Small MTUs = forcing fragmentation reassembly and re-fragmentation at every IP hop, but not at every L2 intermediate node.
"Routing" has become a bit vague in recent years. Do you mean "running a routing protocol"? Yes.
Does 802.11S need RPL? Trying to run a proactive routing protocol on top of a reactive routing protocol on a highly constrained device is problematic.
We don't always reassemble at every hop…
ROLL charter has different scenarios – what about devices in different administrative domains (that vary in reachability)? RPL is very modular, so now we want to go back to applicability statements for RPL in different environments.
Jari – if at all possible, you should have one-hop network topology – that's even better than Route-over…
Route-over would be straightforward if it didn't try to implement a multi-hop scenario. The problem is, that causes mobility problems, and we don't have any good solutions for those problems.
Report that implementation of mesh-under protocol is significantly less complex than RPL, measured in lines of code?
Biggest complexity of RPL is in the spec – lots of options.
In 6LOWPAN we support both mesh-under and route-over, and we're not tied to a single routing protocol.
Architecture is independent of routing protocols, and it needs to be that way.

. Some Considerations When Routing in Particular and Lossy Environments – Clausen

"RPL is doing some things really well" (gasp!)…
People are taking the Objective Function out, replacing it with something naïve that doesn't work, and conclude that the routing protocol is broken.
Unidirectional links are common in the wireless world – this breaks Neighbor Unreachability Detection.
Upward traffic to the controller is common, downward traffic is rare, P2P is esoteric, broadcast is non-existent. But if you're doing an application with bi-directional traffic, you're either doing source routing or unbounded state.
Can do aggregation, but we can't adapt to changing conditions and still do aggregation that works.

RPL is optimized for upward traffic – anything else uses source routing or potentially stores full network topology. RPL hasn't addressed this yet, but we could …
Zigby went to non-stored mode because they didn't want arbitrary increases in storage requirements.
Why not do AODV-like RREQ/RREP?
Vendors should be making choices that are best for the vendors, but they can be pressured to use what other vendors are already using.
RPL has a reactive flavor under development now – you'll end up with something like AODV anyway.

. Discussion

16:30 - 17:30 Take-away, Conclusions

We were exposed to interesting applications, radically different architectures (and their issues), existing technologies from new angles, focused on details of the protocol stack, and (what was last point?)
Possible conclusions:

- Plan for the case where all applications live in the same network.
- Implementation constraints relax over time.
- "One Internet" is important!
- Useful to build abstractions where data and names are in a central role.
- Still don't know how to build deployable security, but we know we need it.
- Prefer route-over instead of mesh-under (and one-hop over multi-hop).

Possible IETF Actions

- Implementation guidance -> LWIG
- Data models (energy, pricing),
- Device identity and attribute discovery,
- Networking beyond subnet boundaries.
- Support for sleeping nodes.
- Named Data Networking

There are possible IRTF actions as well – clearly some of the security stuff is research. Option to start up an IRTF RG. We're smarter than we were a year ago.
Lars – talking to Aaron about possible new research groups. Content-centric networking is one of the possibilities.
From Pascal:

- Need to recognize that IPv6 is a necessary component, and we're doing multilink subnets.
- Need to recognize that we need Layer 3 VLANs – we're doing VLAN-in-VLAN now.
- We have microflows and this doesn't match what we're seeing in standardization work.
- We see local mobility and global mobility.

All of these are engineering, not research.
Data modeling as protocol/interface definition, naming and identification, and the data model itself.

Why isn't security on this list? Mostly Jari thinking we don't know what to do! There were several position papers on security. IETF looking at existing recommendations for crypto algorithms/cyphersuites on constrained devices, IRTF research on deployable security for smart objects.
We need to think about transport – that's a very different model from the Internet today. COAP is the plan of the moment, but it has open questions. Multiple uncoordinated ZIGBEE networks would interfere with each other at the radio level – need to figure out how to put multiple applications on one network, to fix that.
Credential management and provisioning? Is this covered by security bullet above?
Is the existing IETF routing protocol one-size-fits-all? Is that an open question, or closed? Start with routing protocol applicability statements in ROLL and see what's left over.
Address aggregation mechanisms? Is this IETF or IRTF?
Pricing/charging/billing is application-specific – not usually something IETF works on? CDNi also thinking about what to do on billing.
Distributed autonomous self-management? This is probably research. Could NMRG work on this?
Topics for further discussions

- Network management
- Location – going to cross many applications
- Application architecture
- Guidelines for Gateways and Middleboxes (how to ensure innovation can still happen)

IETF is bad at doing architecture and worse at educating about architecture. We need to start educating people about "one Internet". Is this IAB? Fred did a great document educating SmartGrid people about IETF protocols – we need something like that.
The people in this room are the ones who need to do the education about Internet of Things – the IAB isn't going to hand down a statement on Monday. When we have a story, we need to articulate it.
We need some idea of a service layer for this, so applications can coexist on the same network.
A little concerned that we're declaring that the architecture is done – RPL is at Last Call, COAP isn't that far yet. That's premature.
We'll always have some stuff under construction. The architecture will keep changing as we go.
Energy design considerations – is this mostly control and routing protocols?
There are some "possible IETF actions" that we don't actually want to do ("ICMP host responding slowly") – can we take this off the list?
Need to look at addressing and subnetting considerations that affects how big your routing table is – that's what "address aggregation" means.
Security handshake protocols and key exchange – may need this for IPsec payload.
One participant thinks IPsec can be used for Internet of Things – we've been able to fit this in a very small node.
Carsten – we actually know more about RPL and CoAP than you think – we have implementation experience, etc.
Hannes – this is a very long list – if we do three of these things, I'll be thrilled!
Jari – IETF works best when people come with real problems – please bring real problems forward!