SECTION 1: INTRODUCING THE ABSTRACTED NETWORKS CONCEPT

Widespread adoption of TCI/IP protocols over the last two decades appears on the surface to have created a *lingua franca* for computer networking. And with the emergence of IPv6 removing the addressing restrictions of earlier versions, it would appear that now every device in the world may easily be connected with a common protocol.

But three emerging factors are requiring a fresh look at this worldview. The first is the coming wave of sensors, actuators, and devices making up the Internet of Things (IOT). Although not yet widely recognized, it is beginning to be understood that a majority of these devices will be too small, too cheap, too dumb, and too copious to run the hegemonic IPv6 protocol. Instead, much simpler protocols will predominate (see below), which must somehow be incorporated into the IP networks of Enterprises and the Internet.

At the other end of the scale from these tiny devices are huge Enterprise networks, increasing movingly to the cloud for computing and communication resources. An important requirement of these Enterprises is the capacity to manage, control, and tune their networks using a variety of Software Defined Networking (SDN) technologies and protocols. These depend on computing resource at the edges of the network to manage the interactions.

The third element is a conundrum presented by the first two: Enterprises will be struggling with the need to bring vast numbers of simple IOT devices into their networks. Though many of these devices will lack computing and protocol smarts, the requirement will still remain to manage everything via SDN. Along with this, many legacy Machine-to-Machine (M2M) networks (such as those on the factory floor) present the same challenges as the IOT: simple and/or proprietary protocols operating in operational silos today that Enterprises desire to manage and tune with SDN techniques.

And by the way, any new networking solution to address these factors must be tolerant of disruption, mobility, and change, with operation continuing despite perturbation.

The Abstracted Network

The solution to these seemingly contradictory requirements is an idea that is simple in concept, but demanding in implementation: The Abstracted Network. Conceptually, the MeshDynamics Abstracted Network replaces topologically independent networks with a single network made up of a new class of devices called propagators.

As seen in Figure 1a below, traditional networks are still often separate today based on whether they are handling human-oriented traffic (smartphones, tablets, computers, etc.) or M2M traffic (sensors, actuators, robots, etc.). In some cases, such as shop floor environments, legacy protocols and response requirements may even have kept these devices as “islands” or “silos” disconnected from the rest of the Enterprise. These isolated networks have remained despite the rise of IP because of simplicity of the end devices or their peculiar communications and control requirements.

Besides the obvious networking inefficiencies, these isolated networks may not be managed and tuned via SDN techniques. More importantly, potentially important data flows and status indications are hidden from the primary Big Data servers managing the rest of the Enterprise’s business. Indeed, the potential power of the publish/discover/subscribe model for integrating IOT and legacy traffic into business intelligence processes is the leading benefit of the Abstracted Network concept (Figure 1b below).

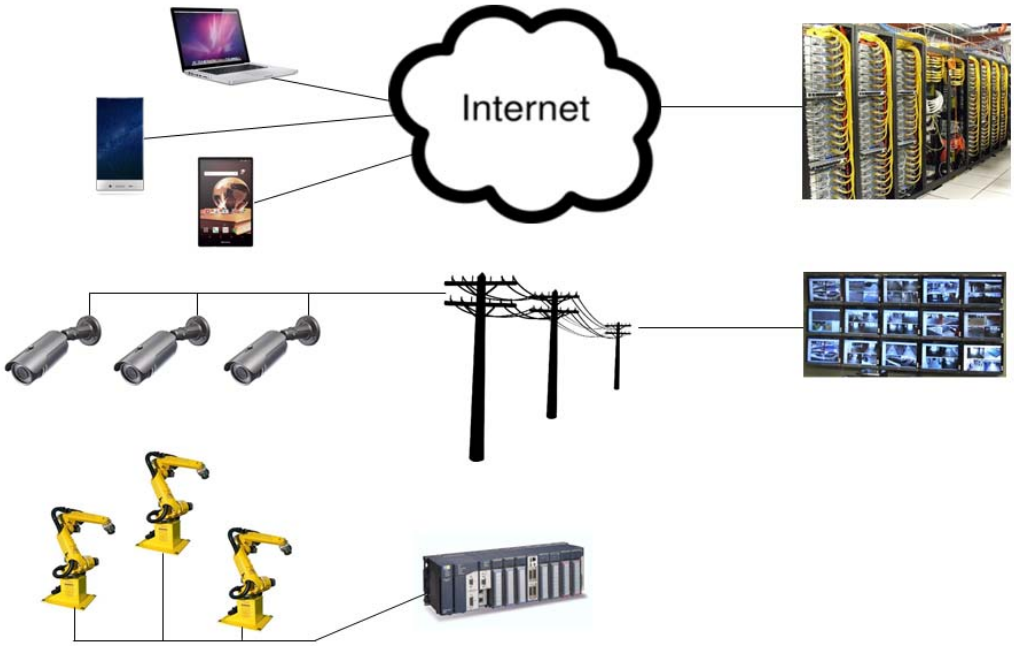


FIG 1a: traditionally separate networks for human-oriented and machine-to-machine data

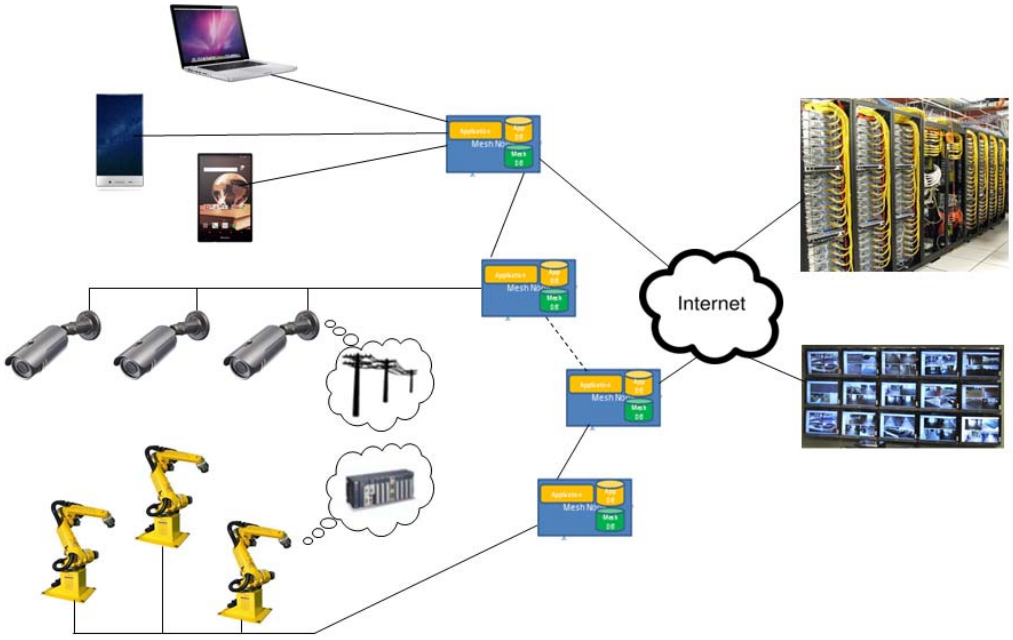


Figure 1b: The Abstracted Network based on propagator nodes emulates separate networks but allows Enterprise control and publish/discover/subscribe

The Abstracted Network model is based on MeshDynamics propagator node communications devices (blue rectangles in Figure 1b) that take advantage of processing power and memory density advancements as well as exploiting emerging Open Source operating systems and communications protocols. Propagator nodes emulate the previously separate networks' protocols, timing, and control interactions such that legacy and IOT devices need not incorporate higher level-protocols themselves but may still become part of the overall Enterprise operational structure.

Database, not topology, constructs network

The underlying technology making MeshDynamics Abstracted Networks possible is a sophisticated real-time database that creates a model of the logical

network connections and requirements (Figure 2 below). A variety of internal processes monitor and update this model based on network traffic flows to create an efficient virtual network structure no matter what the physical topology may be. This includes such details as latency, protocol translation, multicast pruning and forwarding, and even control loop management.

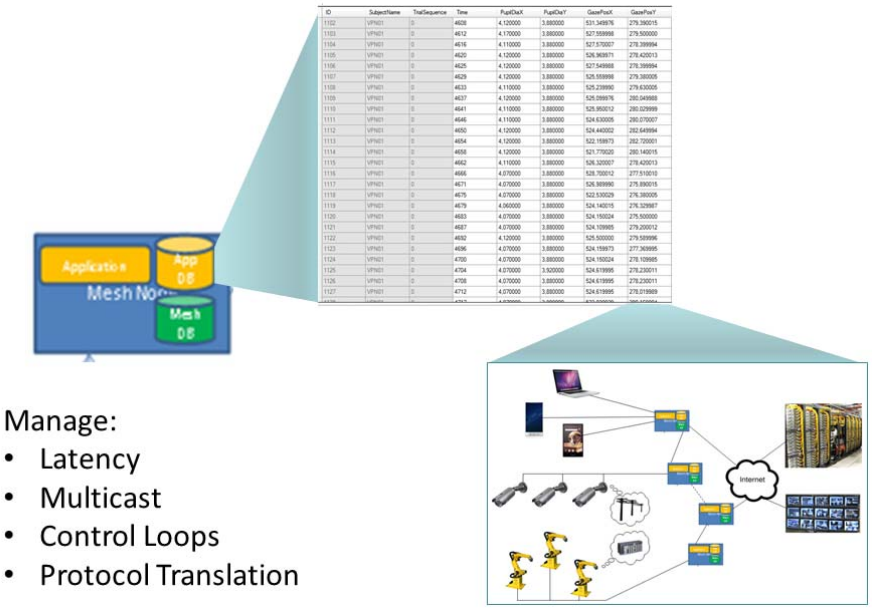


Figure 2: A database driven approach to managing networked devices.

Critically, as this Abstracted Network model is autonomously constructed, it may then be further refined, restructured, monitored, and tuned by the Enterprise using SDN techniques and protocols. This is accomplished in two ways. First, internal processes maintain deterministic performance between different network applications through a variety of scheduling and modeling algorithms within the propagator node network itself.

Secondly, applications agents onboard each of the propagator nodes may participate with Enterprise-wide SDN messaging to optimize network performance based on rules and other parameters (Figure 3 below).

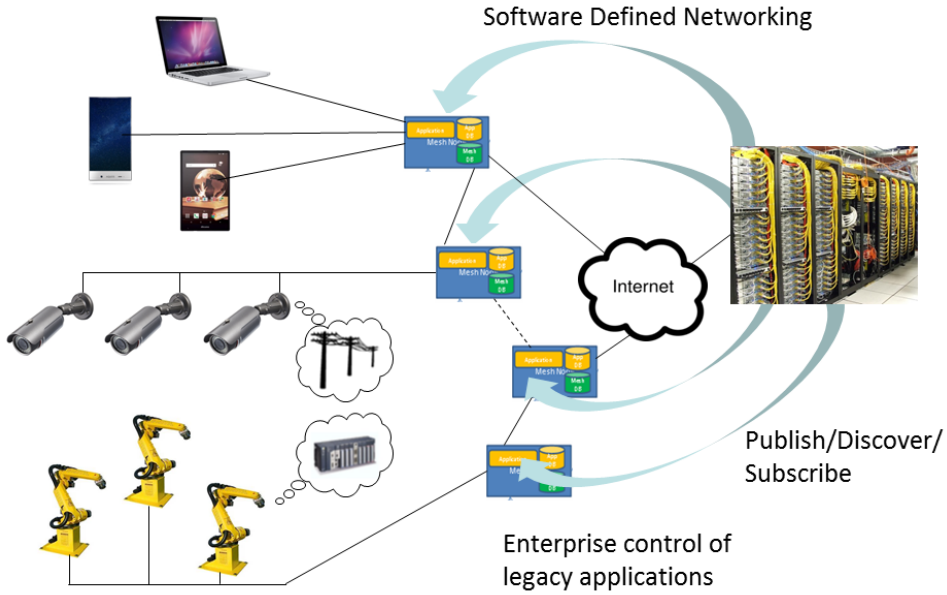


Figure 3: The Abstracted Network allows Enterprise tools and publish/subscribe relationships everywhere, even to legacy and IOT devices

Physically, propagator nodes may incorporate a wide variety of wired and wireless interfaces to facilitate connection to many legacy and IOT devices.

As discussed below for the simplest IOT devices, propagator nodes permit publish/subscribe data flows even for the simplest or proprietary devices. These capabilities are discussed in more detail below. Similarly, a variety of network connections may be used as the links between propagator nodes themselves, the traditional Internet, and Big Data integrators.

Disruption tolerance, mobility, and changeability

With “more eggs in one basket”, survivability and non-stop operation of the Abstracted Network becomes vitally important. But because the interconnected propagators maintain databases for both their own topology and for the logical network configuration, disruption tolerance is provided inherently.

As a first level of disruption tolerance, the propagator network maintains awareness of adjacent nodes and alternate paths, automatically making use of the best links as primary connections and maintaining awareness of any changes or failures. This allows near-instantaneous re-routing around failures as seen in Figure 4 below and described in further detail later in this paper.

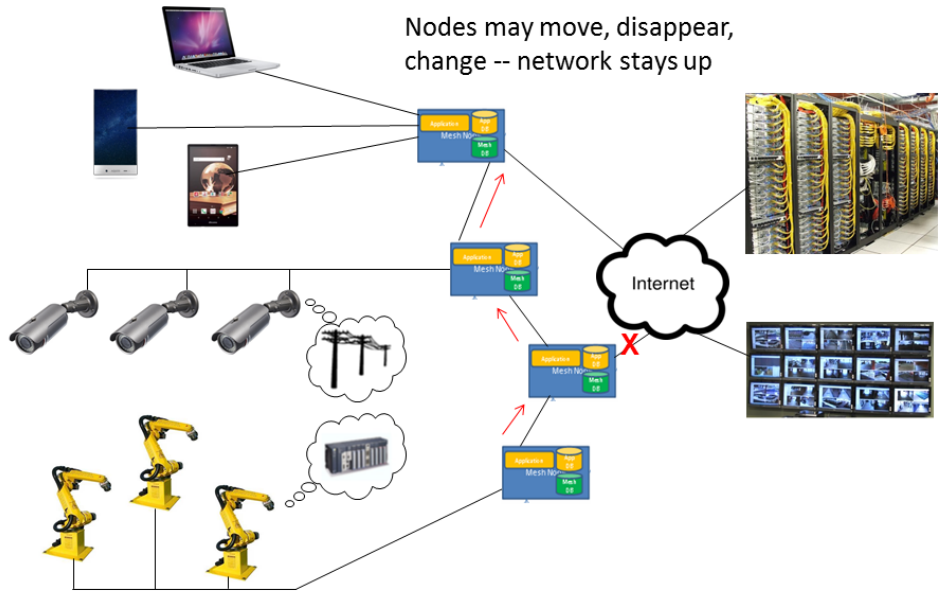


Figure 4: Disruption tolerance between propagator nodes is provided autonomously through an adjacency database and routing table exchanges

A by-product of this capability, propagator nodes may be in motion relative to one another, to end devices, and to other network elements. Because the internal database of node adjacencies and link qualities is constantly updated, individual nodes may shift to new connections as old ones are fading, maintaining network operation.

Disruption tolerance also extends to individual applications and families of connected devices. The applications agents in each propagator node may be equipped with software that actually performs many functions on behalf of connected devices such as data collection, alarms and status, control loop management, even spoofing of network acknowledgement and requests. This may be maintained through a complete failure of the upstream connection and interruption of round-trip communications – the propagator node keeps the local operation intact until the upstream connection is restored (Figure 5).

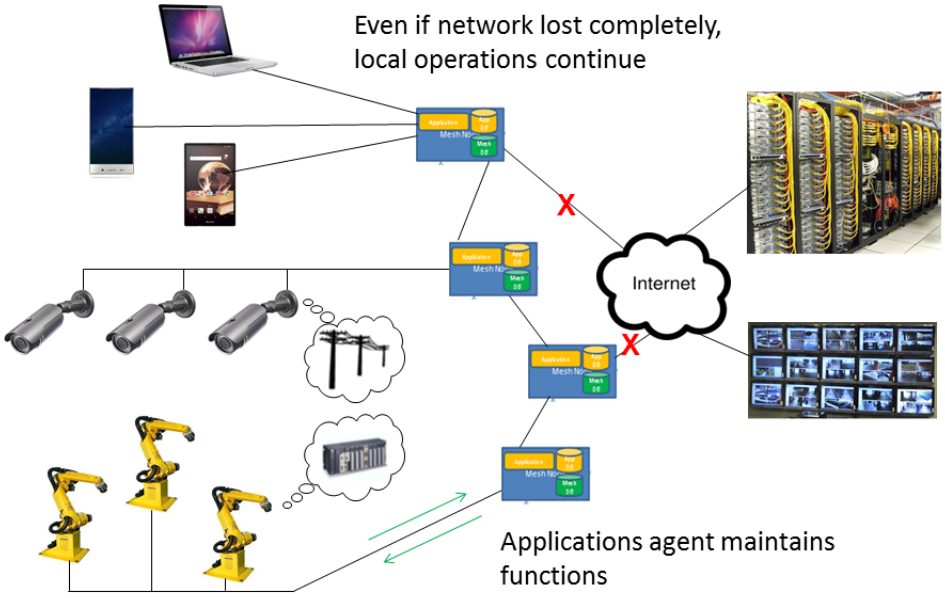


Figure 5: Applications agents in propagator nodes maintain local operation for connected devices despite upstream communications failures.

The capabilities of the propagator node and the Abstracted Network Concept will be especially important for integrating the coming explosion of Internet of Things devices and their terse M2M communications schemes, discussed in the next section.

SECTION 2: IMPACT OF THE INTERNET OF THINGS

Today companies view the IOT as an extension of current networking protocols and practices. But those on the front lines of the Industrial Internet of Things are seeing problems already:

> *"While much of the ink spilled today is about evolutionary improvements using modern IT technologies to address traditional operational technology concerns, the real business impact will be to expand our horizon of addressable concerns. Traditional operational technology has focused on process correctness and safety; traditional IT has focused on time to market and, as a recent concern, security. Both disciplines have developed in a world of relative scarcity, with perhaps hundreds of devices interconnected to perform specific tasks. The future, however, points toward billions of devices and tasks that change by the millisecond under autonomous control, and are so distributed they cannot be tracked by any individual. Our existing processes for ensuring safety, security and management break down when faced with such scale. Stimulating the redevelopment of our technologies for this new world is a focal point for the Industrial Internet Consortium."*
>
> Industrial Internet Consortium *Quarterly Report February 2016*

Over the next decade, billions of interconnected devices will be monitoring and responding to transportation systems, factories, farms, forests, utilities, soil and weather conditions, oceans, and other resources.

The unique characteristic that the majority of these otherwise incredibly diverse IOT devices will share is that they will be too *small*, too *dumb*, too *cheap*, and too *copious* to use traditional networking protocols such as IPv6.

For the same reasons, this tidal wave of IOT devices cannot be controlled by existing operational techniques and tools. Instead, lessons from Nature's massive scale will guide a new architecture for the IOT.

Taking cues from Nature, and in collaboration with our OEM licensees, MeshDynamics is extending concepts outlined in the book "Rethinking the Internet of Things" to real-world problems of supporting "smart: secure and scalable" IOT M2M communities at the edge.

Simple devices, speaking simply

At the edge of the network are simple **Devices**.

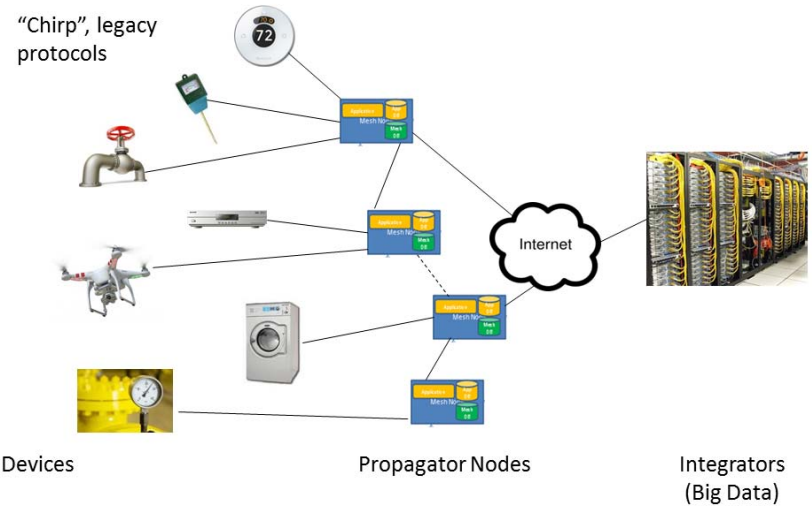


Figure 6a: A three-tier architecture and the Abstracted Networks concept unlock the power of the IOT

These edge devices simply "chirp" their bits of data or listen for chirps directed toward them. The *vast numerical majority of devices* will simply speak and listen in tiny bits of data ("small" data). In an IOT universe, these small and seemingly unsecure (covered later) chirps will propagate upstream and up the food chain to cloud-based integration points, a.k.a Big Data subscribers, see Figure 6a.

As noted above, the key to integrating simple devices at the edge of the network is off-loading networking complexity to the MeshDynamics propagator nodes in an Abstracted Network. Propagator nodes are the "gasket" between simple devices and higher-level Enterprise computing (Figure 6b below).

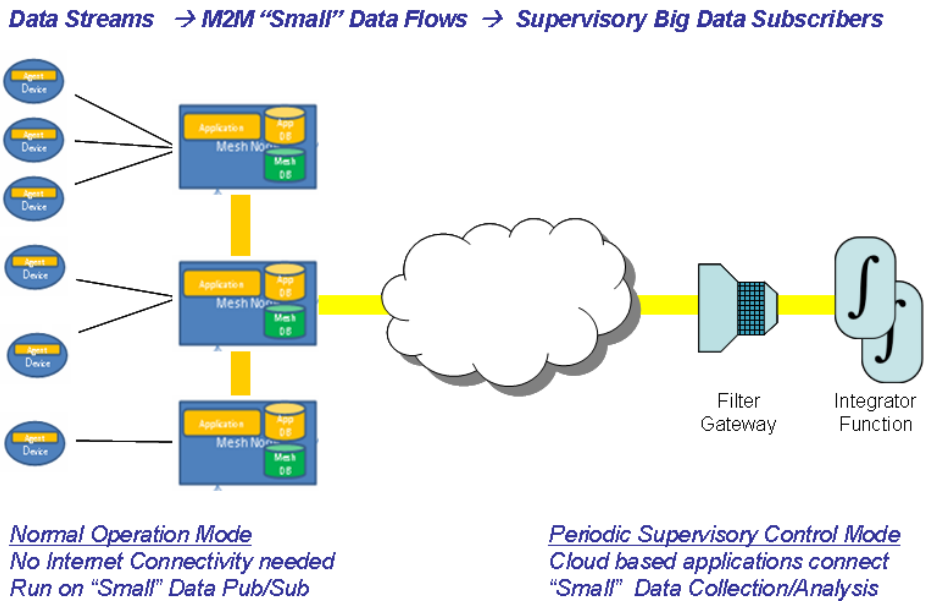


FIG 6b: Scalable and Secure Basic Architecture for IOT/M2M

Like Nature treats pollen, the (scalable) IOT must treat any single chirp as truly "best effort" – so heavy broadcast storms caused by an external event will die out pretty quickly. IOT chirps are digital pollen – lightweight, broadly "published", with meaning only to "interested" receivers/subscribers.

Pollen is lightweight because it is receiver-oriented. Security is inherent in its "packet" structure – only intended flowers can decrypt the payload. For the rest of us, it's just allergy season. Also, since no individual chirp message is critical, there's no need for error-recovery or even integrity-checking overhead (except for basic checksums).

Each IOT chirp message has some short and simple markers, a short data field, and a checksum. As described in my book, the simplest chirps may be only 5 bytes (contrast this with 40 bytes for the smallest sender-oriented IPv6 packet). [Slides]

Chirps are what IP Datagrams were meant to be. The bandwidth savings are immediately obvious, but pale in comparison to the reduction in memory, processing, and power consumption at each of myriad end devices compared to running an IP stack. The cost and complexity burden on the end devices will be very low, as it must be in the IOT.

In contrast to the traditional Internet, error-checking, routing, higher-level addressing, or anything of the sort *are not needed*. Edge devices are fairly mindless "worker bees" existing on a minimum of data flow. This will suffice for the *overwhelming majority* of devices connected to the IOT. (And for those more-sophisticated applications where higher-level protocols are still needed and justified by human interaction, IPv6 will do nicely.)

The basic concept of chirps is not new. Terse M2M messaging is prevalent in all of our purpose-built end devices and products that communicate – your TV remote, your car subsystems, networked factories etc. Terse M2M messaging is how machines have communicated since 8-bit microcontroller days. Challenges lie in scaling securely that which already works, but not reinventing it.

SECTION 3: TECHNICAL STRATEGIES AND IMPLEMENTATION

The Abstracted Networks concept has one basic premise:

If simple devices aren't capable of protocol intelligence, it must reside somewhere and operate on behalf of the end devices. The major elements of that somewhere are the Level II **Propagator** nodes (Figure 6b above and Figure 7 below). As introduced above, these are like familiar networking equipment such as routers and access points, but they operate in a different way in with IOT and legacy devices.

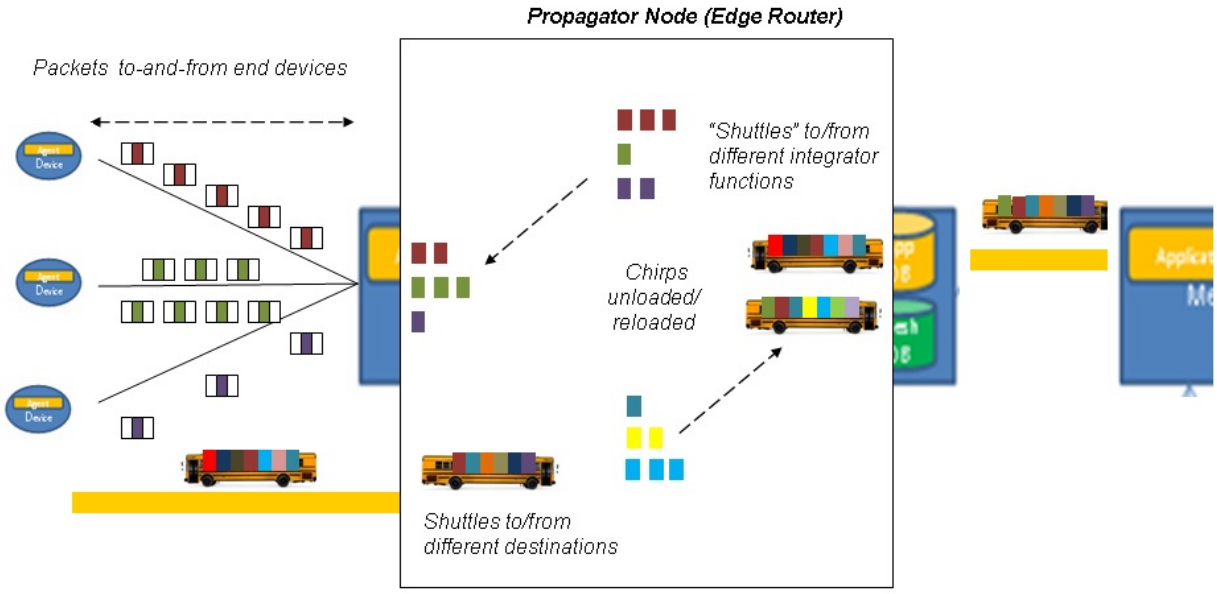
To explore the functions of the propagator nodes in more detail, we can examine their function in an IOT application – but of course this may be extended to legacy and IP networking as well.

Publish, Subscribe and Discovery for the Edge

Propagators listen for data "chirping" from any device. Based on a simple set of “markers” in the chirps (described below), propagator nodes decide how to broadcast these chirps to other propagator nodes and on to the higher-level **Integrator** (Big Data) subscribers.

In order to scale to the immense size of the Internet of Things, these propagator nodes must be capable of a great deal of *discovery and self-organization*. They must recognize other propagator nodes within range, set up simple routing tables of adjacencies, and discover likely paths to the appropriate integrators.

The key is building a logical tree-like topology from physically meshed propagators. The topology algorithms have been tested and described in (more geeky) MeshDynamics patents. [More]



Application: Real Time Publishing of applications/devices data flows to Subscribers/Applications

. Pub/Sub framework with periodic, timed, “shuttle” service between publishers/subscriber apps.
. MAC80211 “radio” abstractions for proprietary devices supported (every interface is port based)
. Applications ingress and egress ports monitored by supervisory audit/management subscribers.

FIG 7: Applications for Aggregation, Pruning and Real Time M2M “Shuttles”

Pub Sub aware Applications on the Propagator nodes serve as aggregation and pruning “hubs”, see Figure 7 above. Chirps passing from-and-to end devices may be combined with other traffic for forwarding. Applications provide this networking on behalf of devices and integrators at levels "above" and "below"

themselves. Any of the standard networking protocols may be used, and propagator nodes will perform important translation functions between different networks (power line or Bluetooth to ZigBee or WiFi, for example), essentially creating small data "flows" from chirp data streams.

Other trusted applications and agents, many residing inside the Propagators, coordinate the function and control of dumb small, cheap, and copious IOT devices through Software Defined Networking (SDN) paradigms for the edge.

MeshDynamics has been developing an open-source propagator platform for disruption tolerant networking for the US Navy and US Department of Energy. Propagator nodes support User Space Application Layer within an OpenWRT architecture for deep packet inspection, SDN based routing, Video, IFTTT (conditional "If This Then That" rules), etc. These propagator nodes provide autonomous, robust machine control with no assurance of internet connectivity through the built-in applications agents.

The end result is a Publish/Subscribe (Pub Sub) network that can be extended from Big Data servers all the way to the edge of the network while still maintaining a degree of responsive local autonomy. A variety of standards-based SDN protocols may be implemented on the distributed applications agents.

> *"MeshDynamics Scalable and Open Pub Sub enables us to rapidly integrate with Enterprise Class, OMG (Object Management Group)-approved, industry-standard messaging systems from RTI (Real-Time Innovations), PRISMTECH, OpenDDS, and others to provide assured real time end to end performance, even if we scale to billions of devices at the edge."* said Curtis Wright, Sr. Research Systems Engineer, Space and Navy Warfare Center.

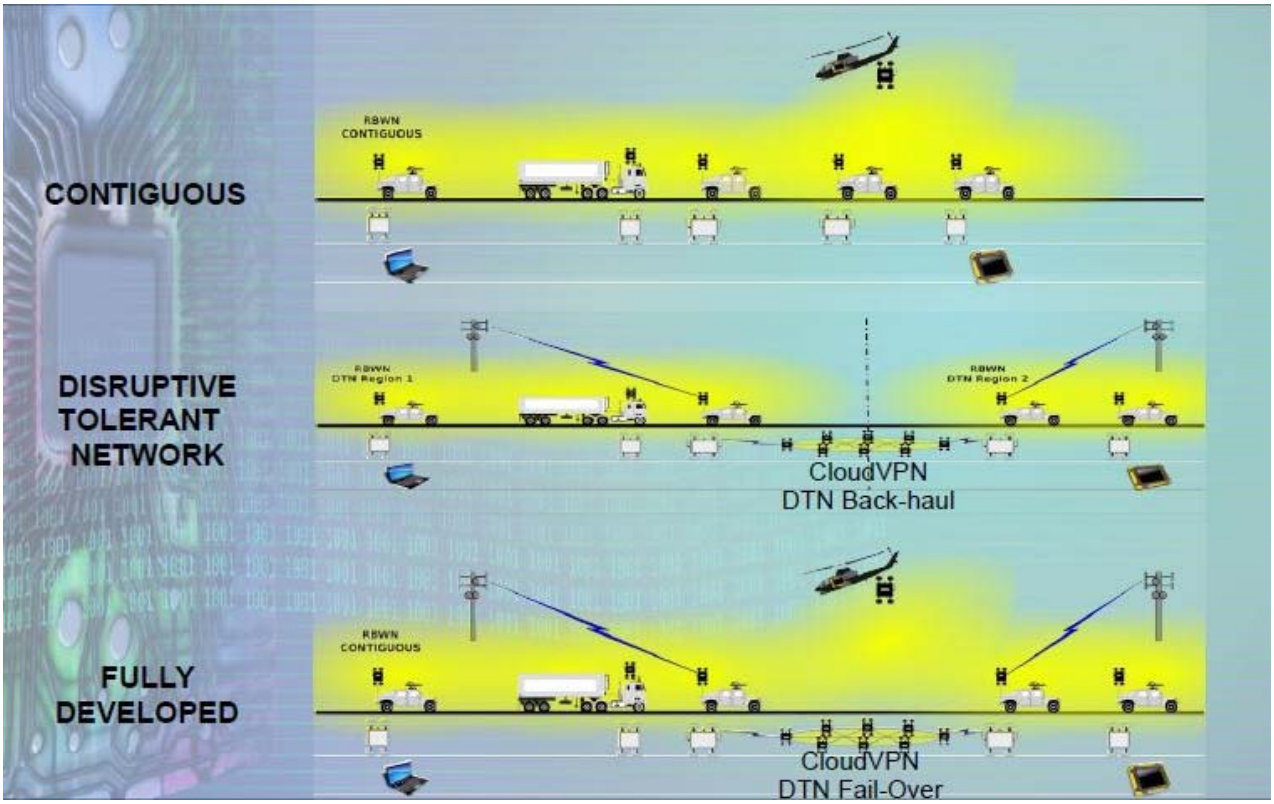


Figure 8: Disruption Tolerant, Semi-autonomous networks (SPAWAR)

Nearly everything described above takes place autonomously and automatically within the *Disruption Tolerant Mesh Network* (Figure 8 above). Propagators and their applications even route around failures of links or nodes and deal with issues created by mobility of network elements. But some of the most important capabilities of propagators are distributed application

intelligence agents that can allow higher level functions to “tune” the propagator network as part of an overall publication/discover/subscribe infrastructure and/or create application instantiations (e.g. machine controllers) within the nodes themselves. These permit connected devices to continue to operate when the broader network connection is lost. [More]

Extending network reach

Along with the ability to manage change and disruption, the unique multi-radio architecture for propagator nodes allows the extension of the network along many “hops” (node-to-node links) without additional wired connections. This “string of pearls” capability allows large outdoor Enterprises to be connected as well as extending services to new remote locations. These extended propagator node links are simply treated as another element of the Abstracted Network and may be managed by Enterprise SDN tools and techniques.

Internal databases observe and accommodate the additional delays experienced by traffic traversing long strings of propagator nodes with minimal additional latency or jitter compared to other network connections.

Integrating current and future networks

Propagator nodes also act as traditional switches or routers for IP traffic, while translating and packaging chirp and legacy traffic into IP packets for forwarding. Because the propagator nodes incorporate both chirp-based and traditional protocols, they are the natural point of integration for emerging IOT, legacy, and IP networks. Sharp Corporation recently announced the QX-C300 series of networking devices that acts as a traditional WiFi Access Point and provides connectivity for IOT devices such as cameras to deploy what they call “smart networks.”

> *“MeshDynamics’ propagator node software allows us to deploy WiFi networks today with minimal additional wiring and also incorporate emerging Internet of Things devices on the same infrastructure today and in the future.”* said Mr. Arai Yuji, GM, Communication Division, Sharp Electronics, Japan.

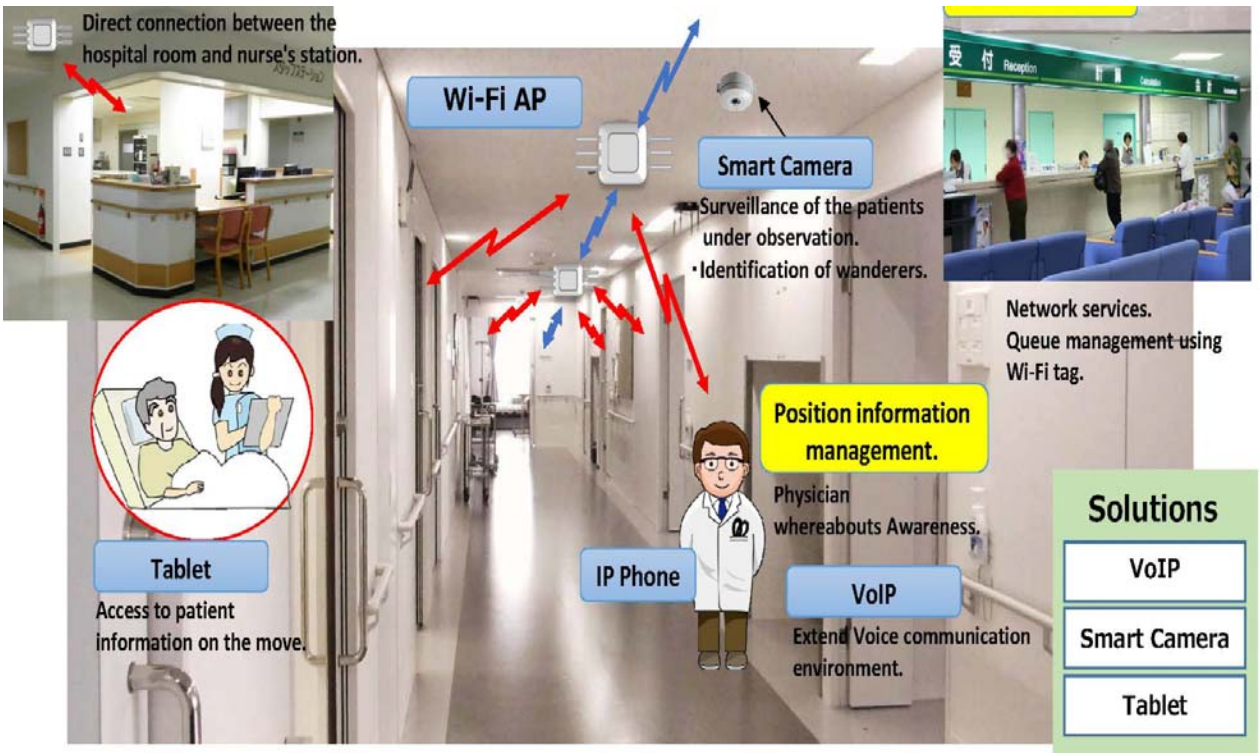


Figure 9: “Smart” Networks supporting Real time Situation Assessment (SHARP)

Chirps enable discovery

As in Nature's "messages" such as pollen, the chirp structure lacks a unique device identification, but does provide a classification of the device type within the public markers. In a hierarchical fashion, devices may be classified as being sensors or actuators, then the type of sensor, and other further defining characteristics, e.g., model number, etc. It is anticipated that there will be an industry-specific open-source registries of chirp identifications that OEM manufacturers utilize and extend.

Individual OEMS may create a new chirp genre or add private extensions to existing chirps to allow more end-to-end capabilities and control. It is expected that a number of industry working groups and SIGs will join together to refine sub-classifications to suit their needs. Importantly, this data structure allows a "start fast and accommodate change" evolutionary approach that will speed deployment of the IOT versus waiting for a conventional standards process. Nature didn't.

Chirps marked with a type ID open a truly powerful opportunity within the IOT. In many cases, an Enterprise IOT network may be "closed", using the private markers within the IOT packets to secure the data within. (Chirp data security is discussed in more detail in "What about security?" below.) But in many other cases, individuals and organizations will open their chirp data streams to the public, allowing anyone to make use of the published data. (This is somewhat analogous to the streaming webcams that are made available on the Internet today)

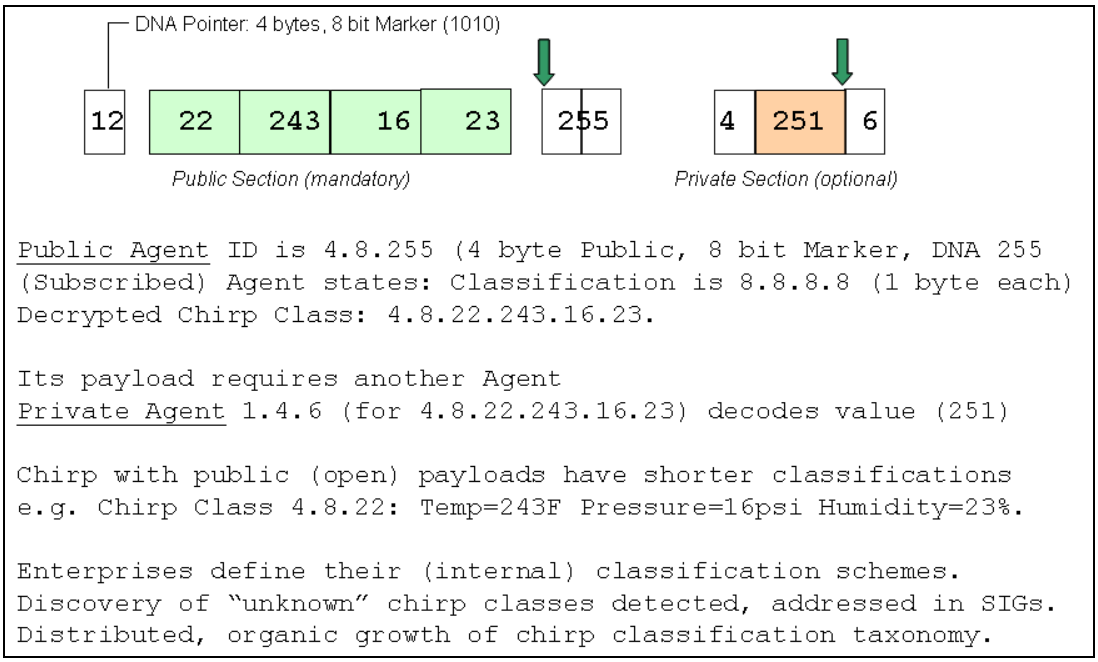


Figure 10: Incremental Extensions to Chirp Grammar supports OEM customizations

Because these chirp streams are tagged with device type, "interested" integrator functions may "discover" potentially useful chirp streams based on geographic location, device type, or data patterns. Thus, the architecture is *receiver-based*, with integrator functions seeking out and subscribing to data streams of interest.

While the chirp data structure is very different from traditional networking protocols, it will be all that is needed for the majority of sensors, actuators, and devices on the IOT. And type-marked chirp data streams open tremendous opportunity for leveraging the expected tsunami of data.

What about security?

In a chirp-based IOT, huge packets, security at the publisher, and assured delivery of any single message are passé. Chirps instead mirror nature with massive publish and subscribe networks based on “light” pollen. As with nature’s pollen, pheromones, and birdsong, many may recognize that there is some data being published, but only the *correct* receiver will have the key to fully unlock the meaning.

Chirp IOT is "female" (receiver-oriented) versus the "male" structure of IP (sender-oriented). When messaging is receiver-oriented, networks survive the relentless broadcast storms of spring. IP-based networks would collapse within days.

The security threat of billions of (conventional IP based) IOT devices is very real. IP based messaging (from Server to Device) simply won’t scale. IP is a sender-oriented form of messaging – thus, it mandates Encryption. That is a losing battle. Moore’s Law is slowing down and any way Metcalfe’s law is exponential e.g. O(n*n). There is a good reason why Nature uses open, extensible, subscription-based (receiver-oriented) “messaging.”

Further security is achieved through the applications agents in propagators (see Figure 6b). Secure data may be flowing through the propagator node network alongside open data, but is unintelligible without the encryption keys provided to the application agents. This is similar to receiver-oriented schemes found in nature, such as when air transports both proprietary (e.g., pollen) and open “signals” (e.g., birdsong). Individual propagators may be biased to transport or discard secured or open data.

One of the hidden security benefits of the chirp architecture is that there is no end-to-end direct connectivity to end devices – *the propagator is always in the data path*. With the potential for sophisticated security applications within the propagator, end devices are invisible to hackers and vandals. This approach is far simpler and cheaper than managing encryption and security at millions (or billions) of end devices – which further need not be burdened themselves with the processing power and memory needed for security applications. Small, dumb, cheap, copious – *and* secure. Security is obviously a key concern for MeshDynamics Military OEM licensees.

What about Standards?

The “Standards conundrum” suffers from the same misleading logic as requiring unique MAC IDs to address an IOT device. I alluded to this fallacy in my book where I describe how there are many John Smiths in the world, but the ones I have in my rolodex are sufficiently distinctive (to me, based on context) to be "uniquely" addressable. Local Uniqueness is enough. Nature concurs. In combination with receiver-oriented messaging, it is even exploited in how prolonged “broadcast” storms of spring disseminate pollen. The winds that carry the pollen are not “global” and time to live is inherently constrained. The same sort of broadcast over IP networks would be crippling, but each propagator effectively and automatically segments its local end devices from the network and vice-versa.

Propagators play a further role in managing standards and accelerating the proliferation of the IOT. Because each may contain applications agents tuned or defined by elements “higher up” in the network, they may serve as a translator for a wide variety of end devices. Chirp-based or IP-based, to name two, but also any variety of ad hoc, standards-based, or proprietary protocols found in older M2M networks. The propagator node effectively isolates and “spoofs” addressing, control timing, and other characteristics of the data stream. Again, addressing need not be globally unique – or even globally *understood* – application agents in the propagators host the necessary intelligence to handle all conversions.

Along with the possibility of a very wide array of physical interfaces on

propagator nodes (wired, optical, and wireless, for example), these conversion capabilities ease standards issues and allow rapid migration of legacy networks to the IOT.

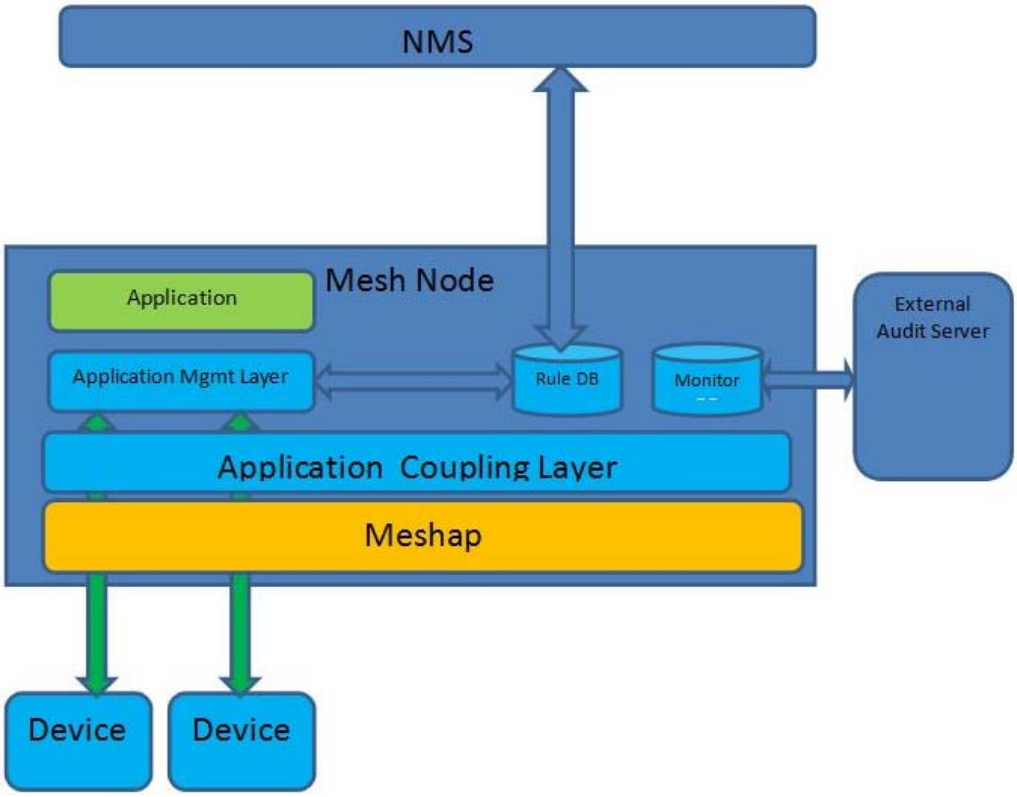


Figure 11: Applications for low level Device protocol translation services.

Deterministic-time IOT Test Bed

The Internet of Things is about purpose-built machines and their interchange. Machines behave according to “states” as Deterministic Finite Automata (DFA). The Internet “Machine” is one such, in addition to other familiar “things”.

Machines connect to other machines via “networking” machines. Propagators connect machines into one ecosystem, from Big Data servers to lawn sprinklers. They make no assumptions about machine protocols or network topology. But machines do expect to be “fed” regularly – most of them need isochronous delivery for the control systems to function. The new IOT trains (shuttles) have to run on time.

MeshDynamics, with our OEM partners, is prototyping an Abstracted Network test bed to test/tune the concepts outlined above. There are applications running inside both propagators and (optionally) devices, communicating with Enterprise Messaging systems. Applications will function as advertised, no changes to their “wiring diagram” required. The Framework ensures timely delivery regardless of network topology. One test case is Delay and Disruption Tolerant networks (DTN).

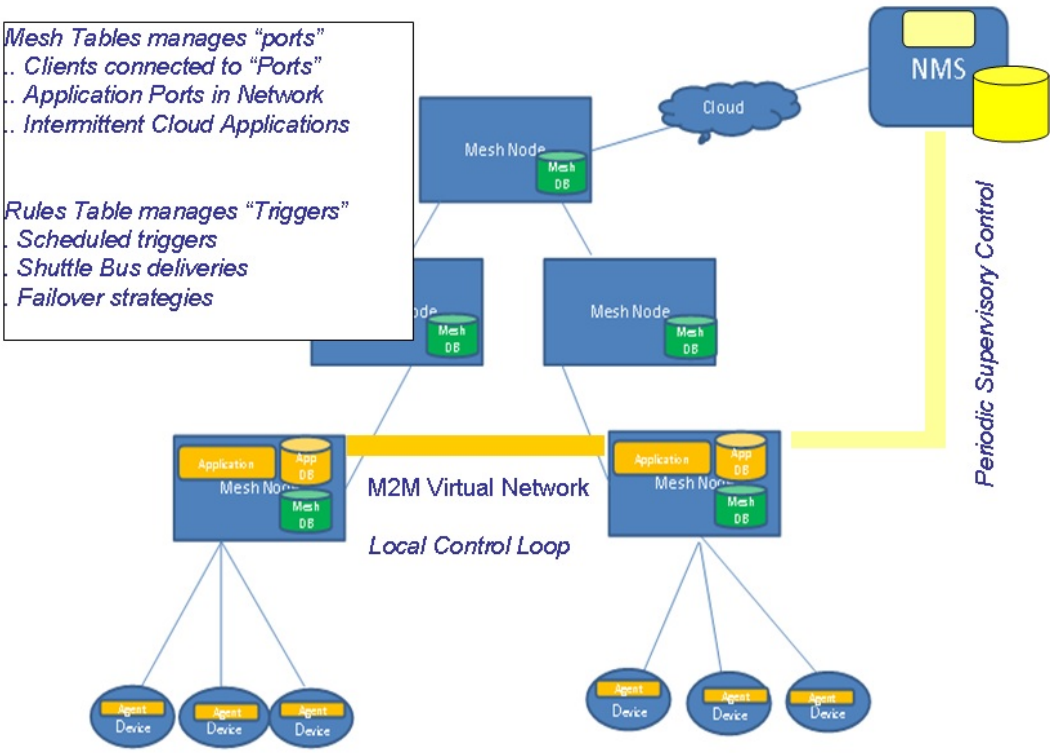


Figure 12: Managing two levels of Control Loops concurrently

From the perspective of the edge devices, there is no apparent disruption of services when cloud connectivity fails. Failover rules trigger applications on the mesh nodes to take over device management seamlessly. When the cloud connectivity is re-established, the applications providing fail over support updates to the cloud and receive updated mission directives.

From a fault tolerant control system perspective there are two control loops in operation at all times, see Figure 12 above. One manages seamless connectivity to mobile end devices (e.g. VOIP phones) by sharing information across meshed propagator nodes using their own protocols and leveraging a periodic, time sensitive pub/sub shuttle service.

From a Cloud Management perspective, there is a distributed real-time database, with portions residing and operating on Propagators.

The database maintains scheduled IF-THEN "triggers" that "fire" applications. The Machines in the network now include end devices, propagators and applications, all part of a scalable pub sub messaging framework.

Big Data subscribers may tune system level behavior with modified triggers, schedules and failover strategies. These include the ability to drill down, to an atomic task being performed by the network and provide suggestions and automated "tuning".

Using an open source framework engenders OEM interests in using it for device to cloud end to end management of all resources in the network.

The Abstracted Network Concept, with onboard applications agents in propagators, can interact with other emerging database-oriented IOT solutions, such as Google's Cloud Platform.

From an IOT Device Manufacturer's perspective, Using DFA-based paradigms to manage the network enables OEMs to model and test where applications "should" reside. Allowing simpler end devices also means that the "smart" of smart networks resides elsewhere. How much intelligence is needed and where within the network it should reside must be modeled, as discussed below.

Dynamic scheduling and simulation

With Dynamic networks, the abstracted network “wiring diagram” is being constantly revised by the meshed propagator network to ensure application and device performance. This constant revision must be automatically provided by tuning/learning algorithms, residing in the network, at various stages of the DFA fan-out.

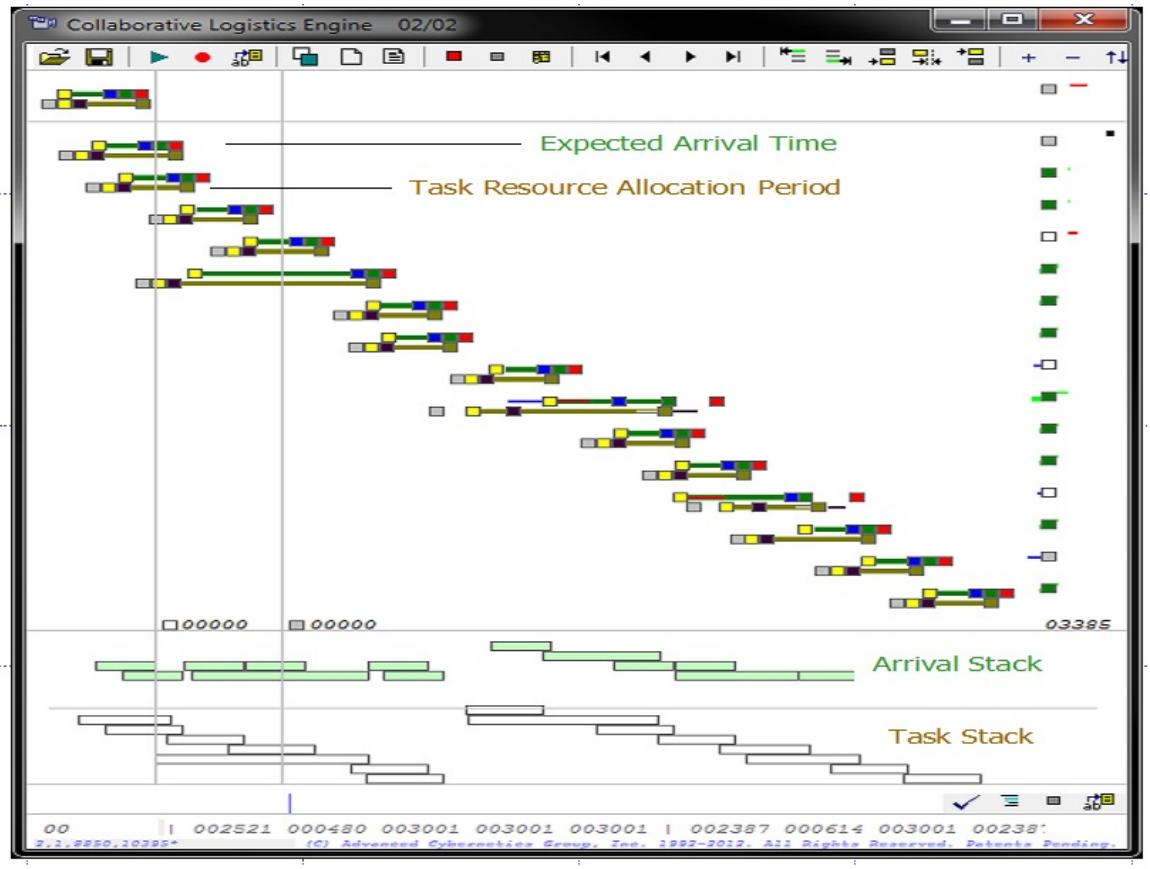


Figure 13: Dynamic Scheduler to meet end-to-end service-level agreements

The Scheduler modifies FIFO queues so some packets shift left/right in time, to accommodate packets that would otherwise be late (Figure 13 above). The illustration on the left indicates a small increase in expected latency, which in FIFO accumulates (see Stack). The illustration on the right alleviates this is real time, based on application delivery stipulations.

The Simulator Scheduler operates on multiple and diverse customer “job” schedule lists. Basic principles used for air traffic control, adapted to application scheduling, where parent apps are awaiting results from child apps before “takeoff”. Algorithms operate periodic shuttles. They re-order shuttle service schedules so no “passenger” misses a “flight”. The Abstracted Network manages all of this internal scheduling and re-scheduling transparently to the end devices and Big Data integrators.

Summary

Far from being a homogenous computing and networking environment, the next generation of Enterprise communications will actually require the integration of an even wider variety of protocols and devices as legacy and IOT applications are finally incorporated into the Enterprise network. Conversely, Enterprises will wish to extend their Software Defined Networking capabilities to encompass these “things” at the edge.

For IOT in particular, the future world of small, dumb, cheap, and copious sensors, actuators, and devices demands rethinking at both ends of the scale. At the far reaches of the network, simplified chirps will minimize lifetime costs for the myriad end points of the Internet of Things.

At the same time, the concept of Abstracted Networks and powerful networking and applications tools concentrated in propagator node devices will allow unprecedented control and flexibility in creating huge Enterprise networks of diverse elements by extending industry-standard SDN tools and techniques.

Fully exploiting the power of the Internet of Things will grow from a total rethinking of network architectures. MeshDynamics, in collaboration with our OEM partners, is exploring a machine-centric Abstracted Networks approach to managing “things”.

About the author

The emerging Internet of Things architecture and MeshDynamics wireless mesh networking propagator technology has been influenced by the Robotics and Machine Control background of founder Francis daCosta. (Early MeshDynamics nodes were installed on mobile robots.)

Francis previously founded Advanced Cybernetics Group, providing robot control system software for mission critical applications. These included local and supervisory real time machine-to-machine control. At MITRE, he served as an advisor to the United States Air Force Robotics and Automation Center of Excellence (RACE).

In 2012, Intel sponsored Francis’ book *Rethinking the Internet Of Things* (Apress, 2013). It was a finalist for the 2014 Dr. Dobbs Jolt Award.

Slide Presentations (Unclassified) Follow

- ***The Abstracted Network***

  - ***Applications of the Abstracted Network:***

    - ***Disruption Tolerant Networks (Military)***

    - ***Smart Networks (Enterprise)***

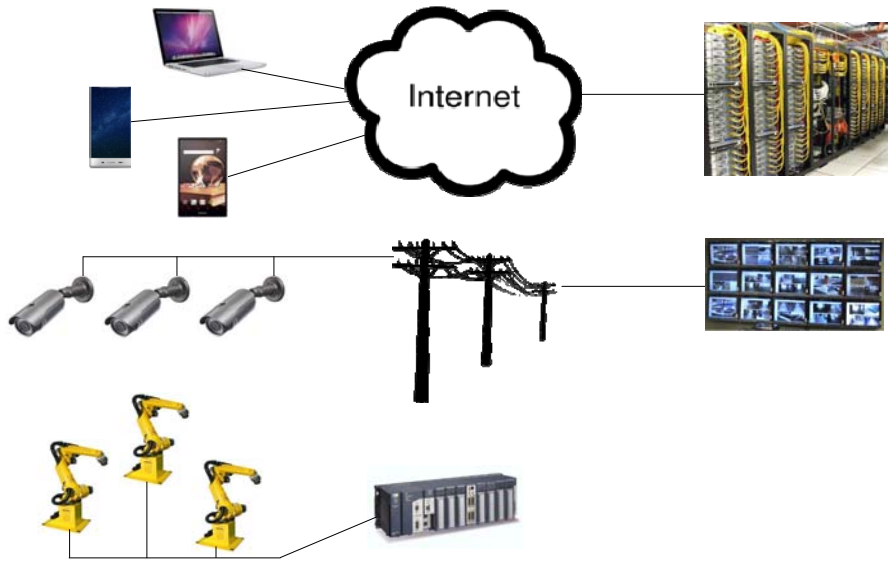
  - ***Sharing Resources and Leveraging Open Source***

    - ***Enterprise Class Messaging with Open Standards***
    - ***Understanding Latency Requirements for Applications***

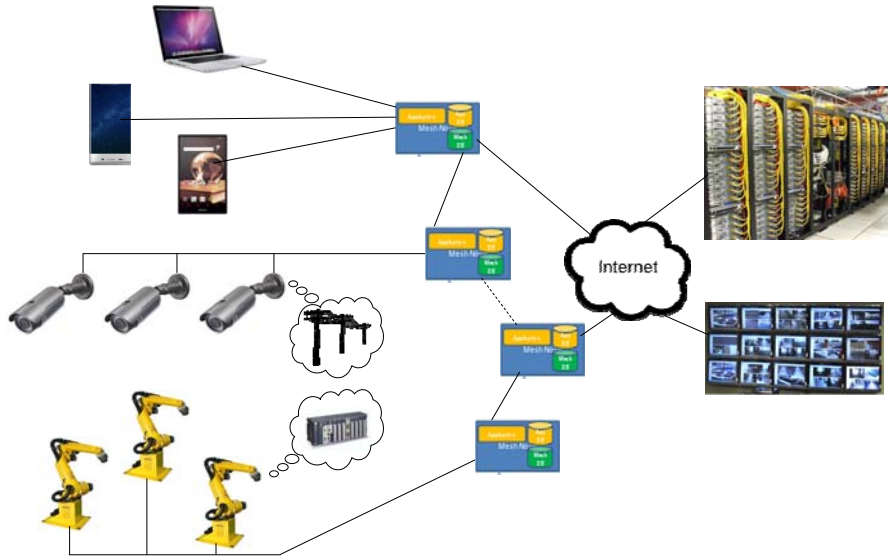
  - ***Develop OEM Specialized Strategies and Templates for:***

    - ***Internet of Things***
    - ***Application-to-Application Networking***
    - ***Real time Machine to Machine Communications***
    - ***Low Cost IC Chips for IOT Chirp Devices***

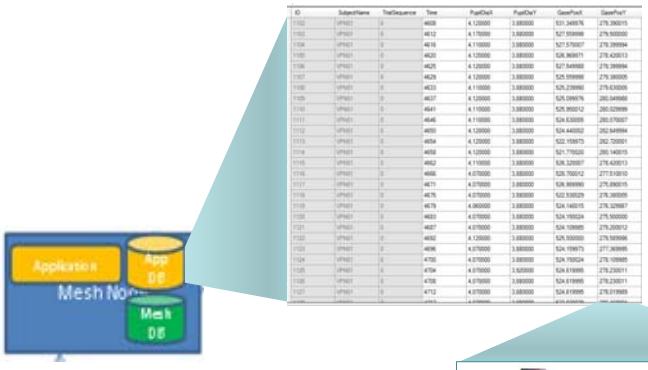
Traditional Networks Separate



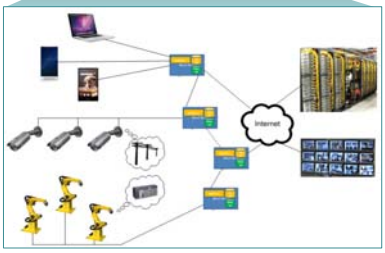
Abstracted Network Emulates Separate Networks



Database Creates all Network Relationships



Manage:
- Latency
- Multicast
- Control Loops
- Protocol Translation



The Abstracted Network

Applications of the Abstracted Network:

- *Disruption Tolerant Networks (Military)*
- *Smart Networks (Enterprise)*

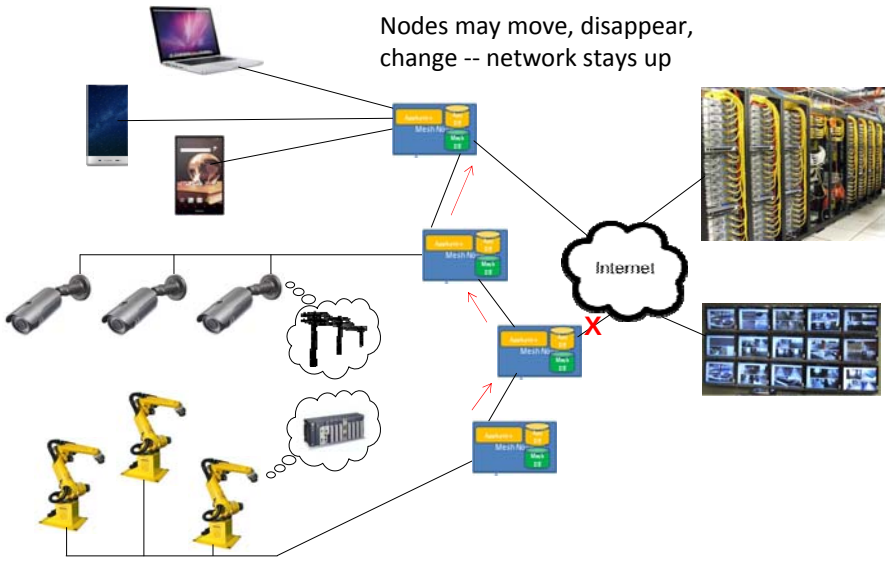
Sharing Resources and Leveraging Open Source

- *Enterprise Class Messaging with Open Standards*
- *Understanding Latency Requirements for Applications*

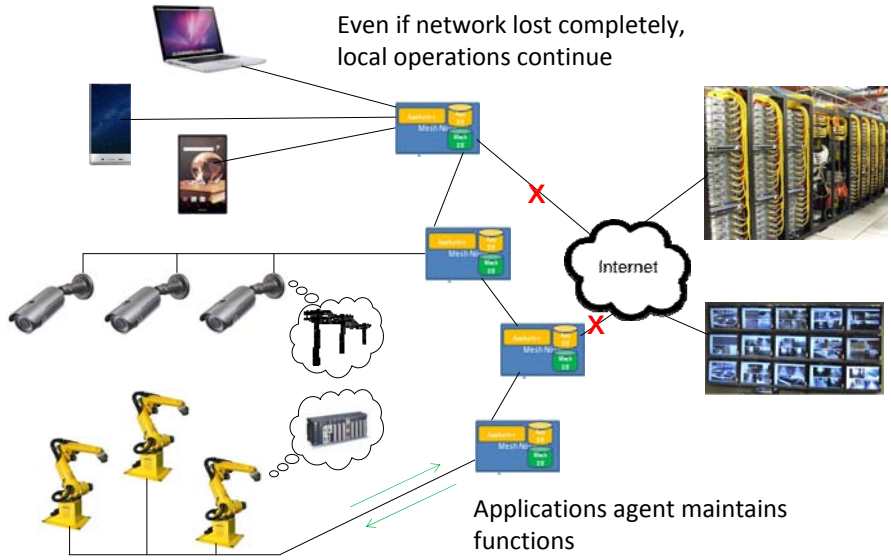
Develop OEM Specialized Strategies and Templates for:

- *Internet of Things*
- *Application-to-Application Networking*
- *Real time Machine to Machine Communications*
- *Low Cost IC Chips for IOT Chirp Devices*

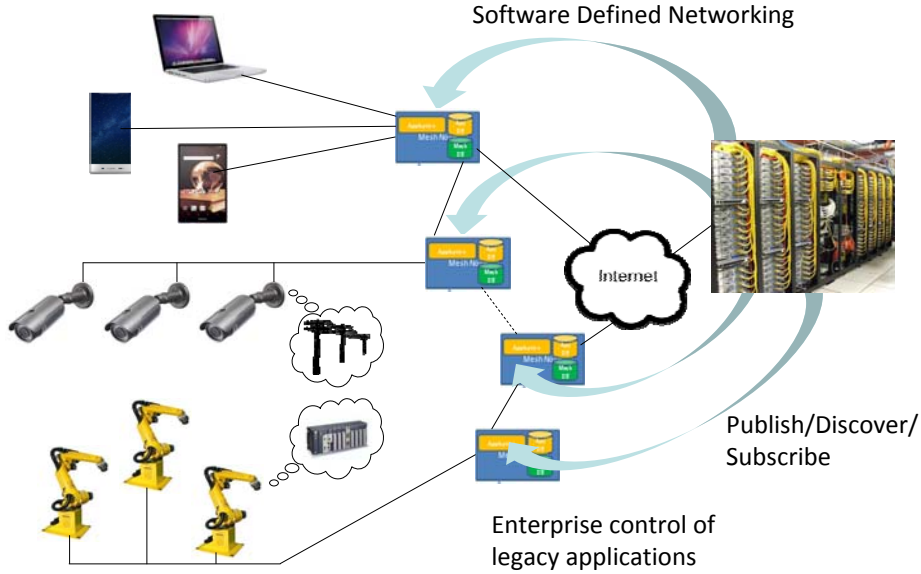
Disruption Tolerance Maintains Connections



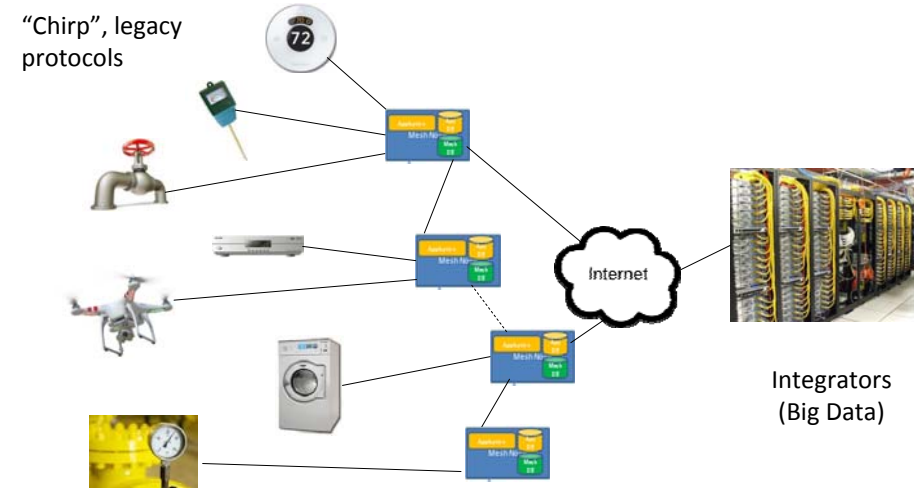
Disruption Tolerance Maintains Operation



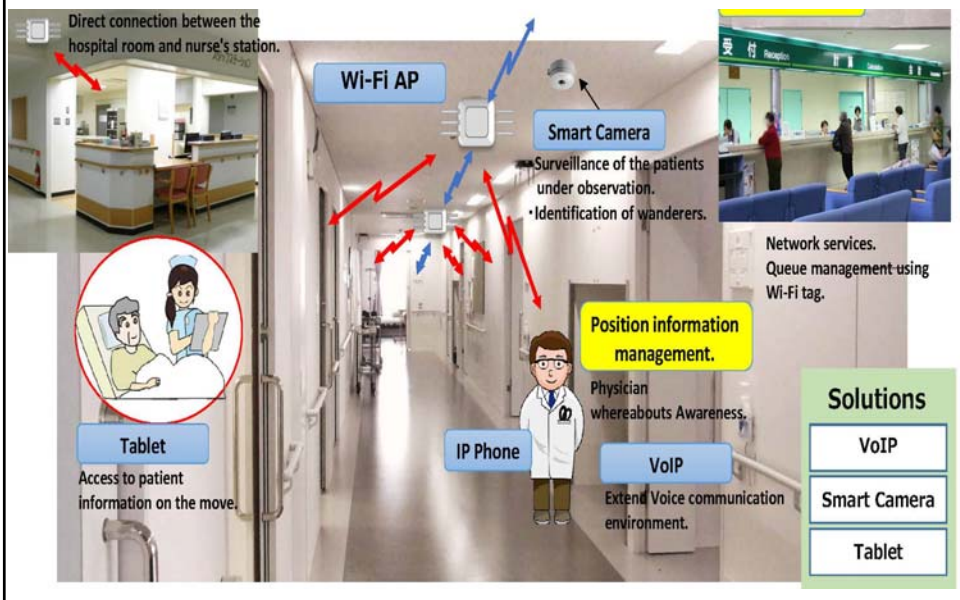
Abstracted Network Offers Enterprise Tools Everywhere



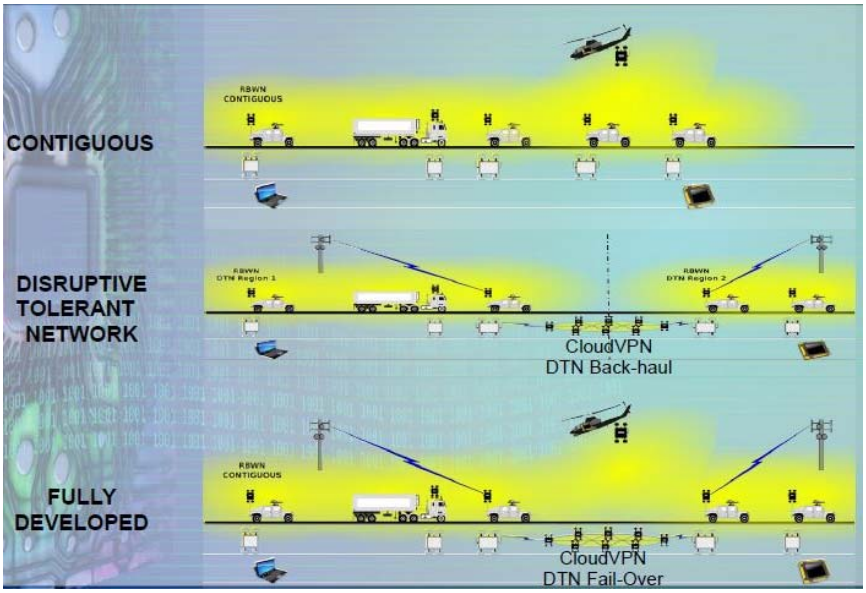
Abstracted Network Connects Old and New IOT Devices



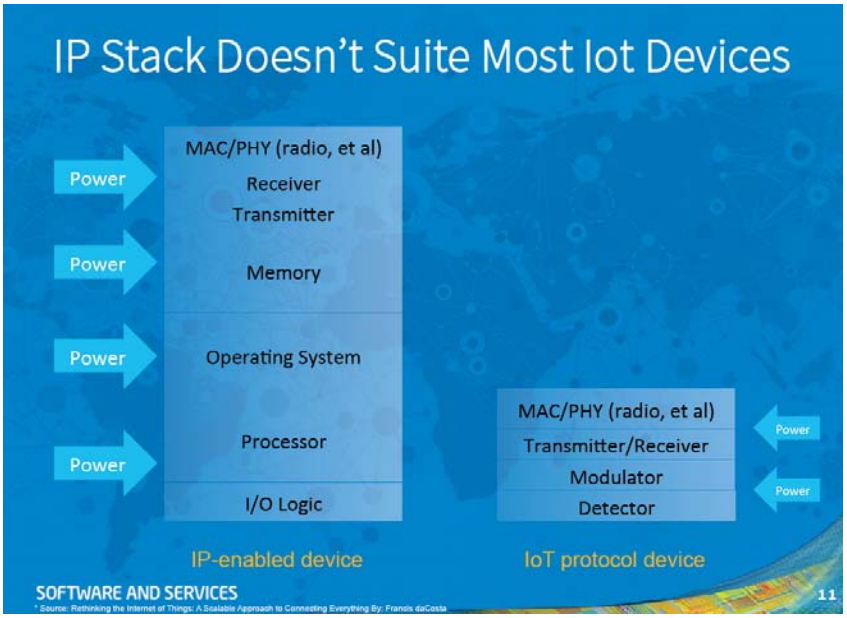
Sharp Smart Network Example



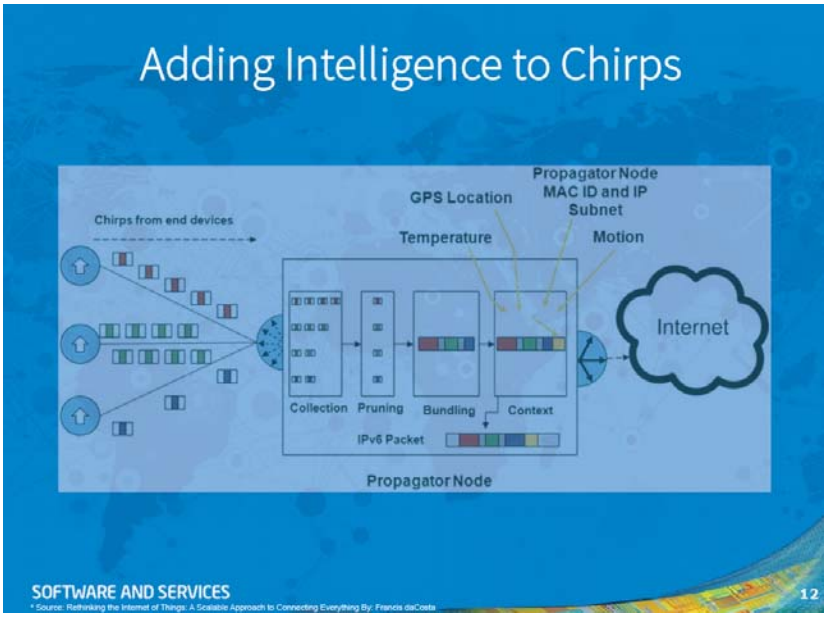
SPAWAR Autonomous Network Example



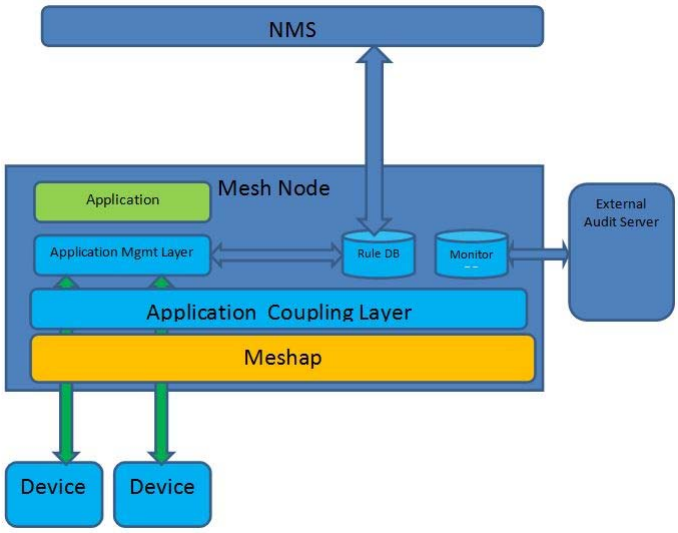
Intel Expects Alternatives to IP in the IOT



Intel Supporting MeshDynamics Propagator Model



Applications Running inside Propagator Network

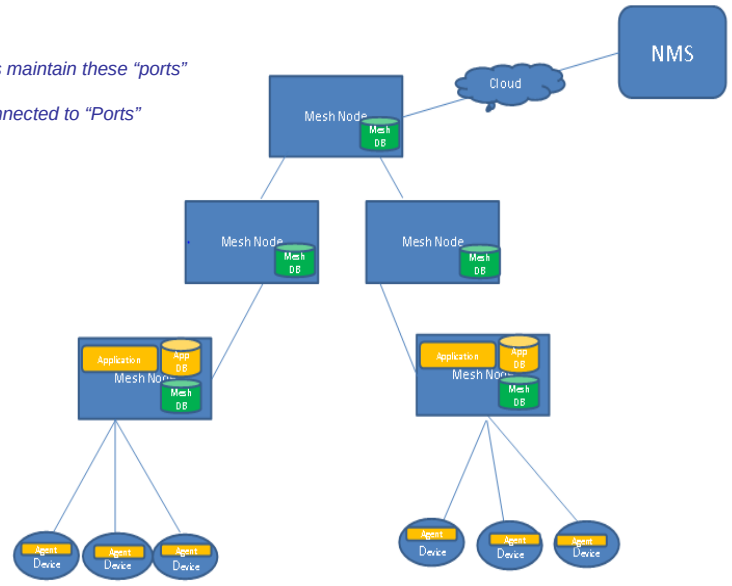


(Autonomous) Applications Running on Mesh Node

Mesh Tables maintain these "ports"

.. Clients connected to "Ports"

..

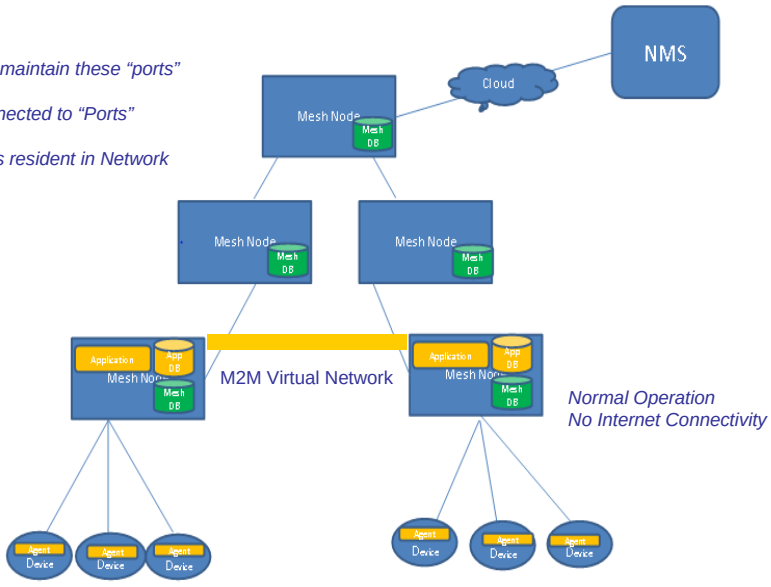


(Autonomous) Applications Running on Mesh Node

Mesh Tables maintain these "ports"

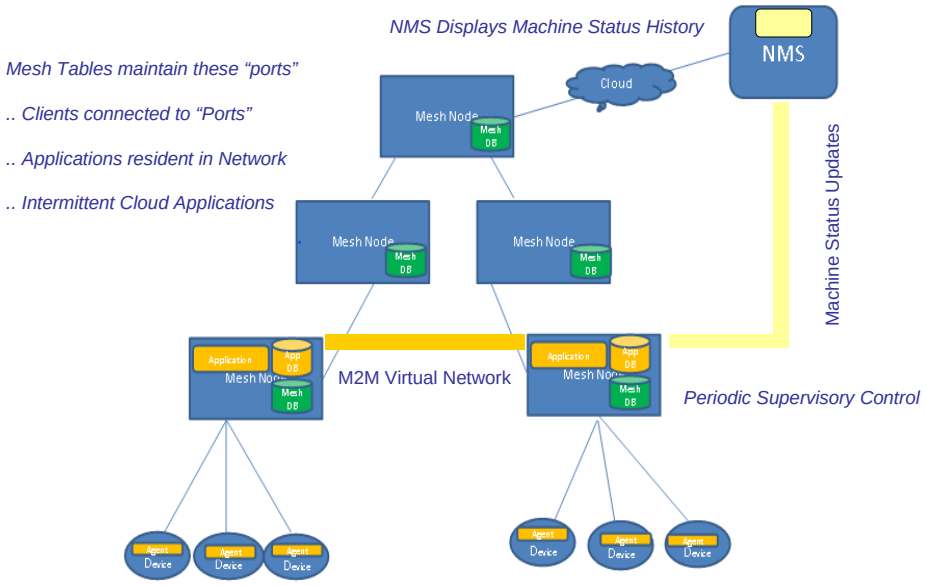
.. Clients connected to "Ports"

.. Applications resident in Network



Normal Operation
No Internet Connectivity

(Autonomous) Applications Running on Mesh Node



The Abstracted Network

Applications of the Abstracted Network:

- ***Disruption Tolerant Networks (Military)***
- ***Smart Networks (Enterprise)***

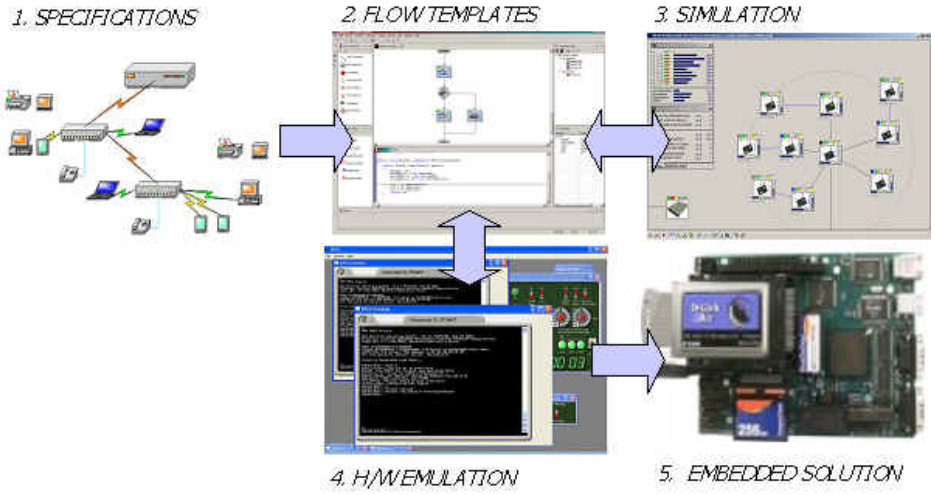
➡ ***Sharing Resources and Leveraging Open Source***

- ***Enterprise Class Messaging with Open Standards***
- ***Understanding Latency Requirements for Applications***

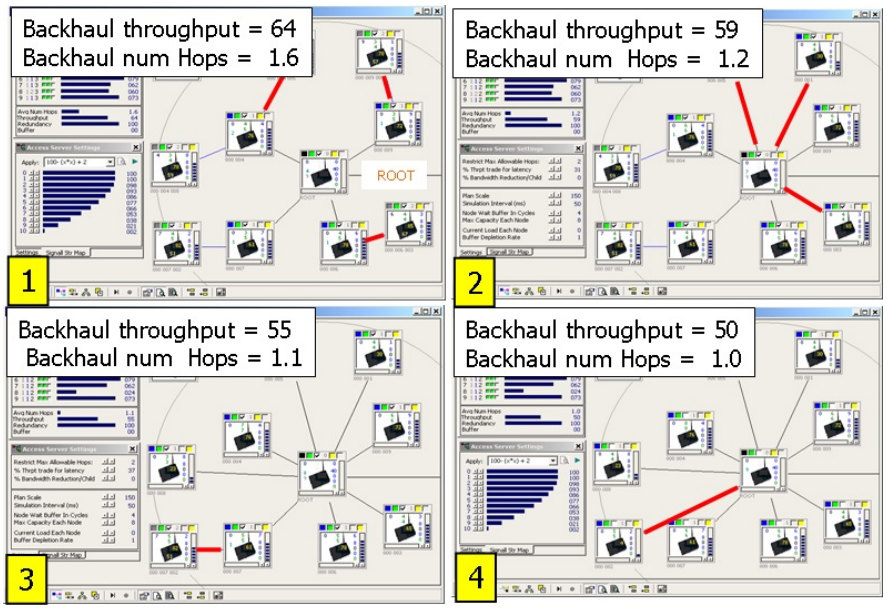
Develop OEM Specialized Strategies and Templates for:

- ***Internet of Things***
- ***Application-to-Application Networking***
- ***Real time Machine to Machine Communications***
- ***Low Cost IC Chips for IOT Chirp Devices***

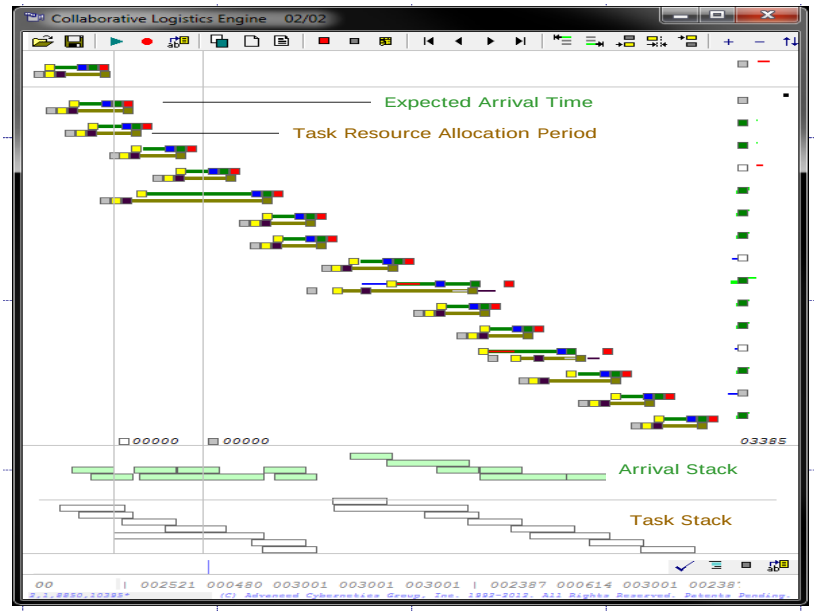
Objectives of Autonomous Network Test Bed



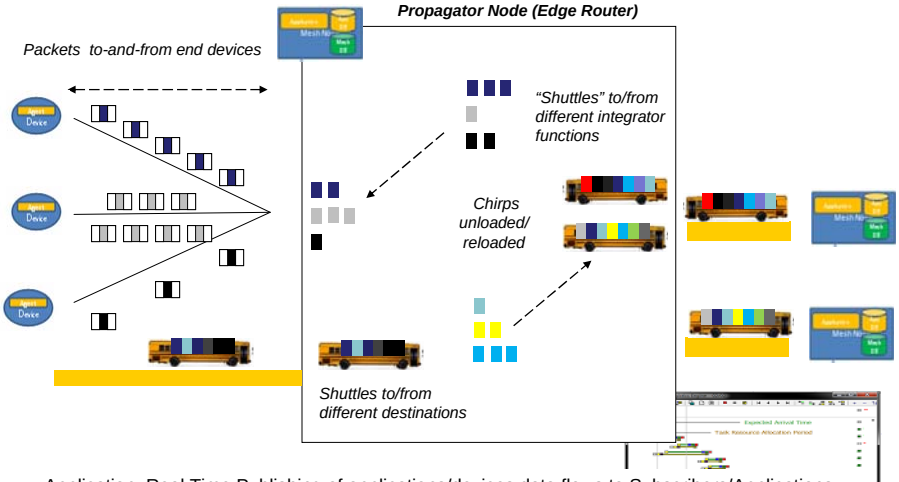
Managing Network Latency (Tree Topology)



Managing Application Latency (Task Scheduling)

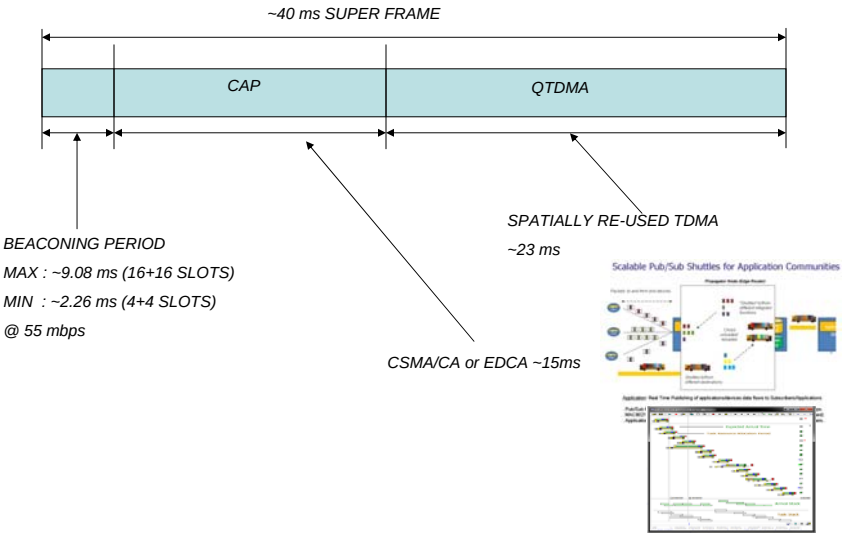


Managing Real Time Enterprise Pub/Sub Messaging

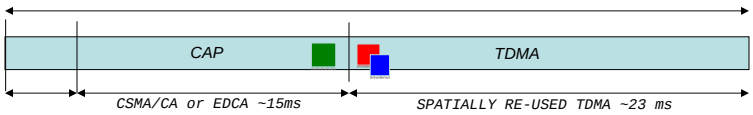


Application: Real Time Publishing of applications/devices data flows to Subscribers/Applications
. Pub/Sub framework with periodic, timed, "shuttle" service between publishers/subscriber apps.
.. Applications ingress and egress ports monitored by supervisory audit/management subscribers.

Managing Latency at the 802.11 MAC Radio Level



Quasi-TDMA within IEEE 802.11 Protocols



BEACONING PERIOD

MAX : ~9.08 ms (16+16 SLOTS)

MIN : ~2.26 ms (4+4 SLOTS)

@ 55 mbps

Every 20 ms

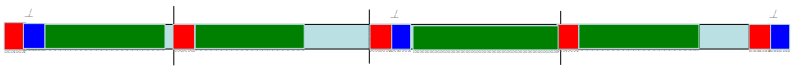
Every 10 ms

Assume

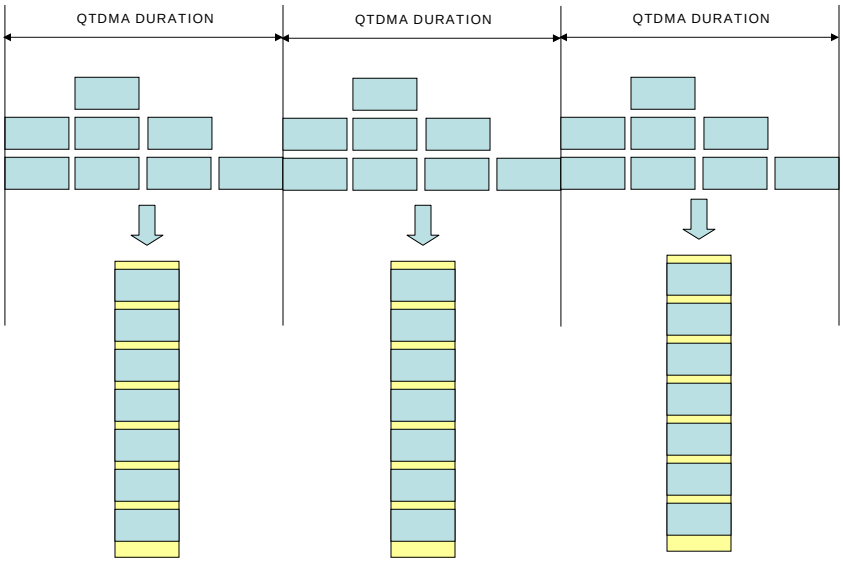
1. we use QTDMA and CSM/RSCM.
2. Queues unloaded/loaded on 10 ms hard real time clock
3. Allocations are made within each Class of Service COS.
4. Within each COS there is a QOS to sort within COS Queue.

Then

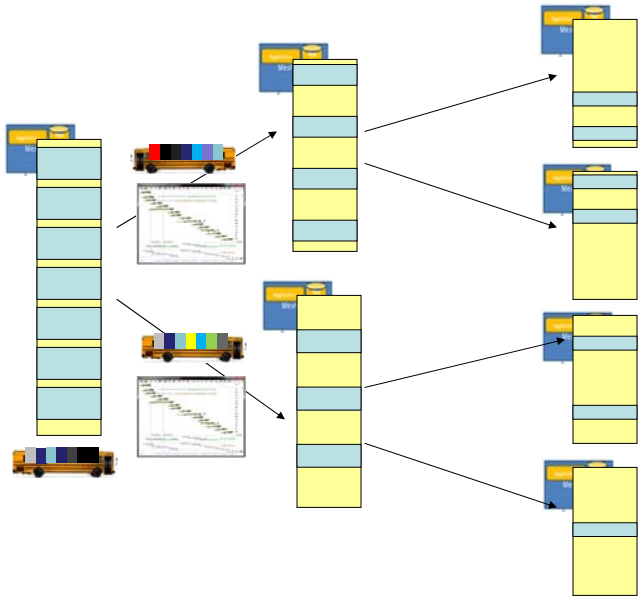
1.Reconstruct Super Frame to order Sort inside Super Frame
2.In side super frame of any size, always 10 ms intervals
3.In example below 40 ms -> 4 10 ms intervals.



Timed Concatenation State Machine (QTDMA)



Timed Concatenation State Machine (QTDMA)



The Abstracted Network

Applications of the Abstracted Network:

- *Disruption Tolerant Networks (Military)*
- *Smart Networks (Enterprise)*

Sharing Resources and Leveraging Open Source

- *Enterprise Class Messaging with Open Standards*
- *Understanding Latency Requirements for Applications*

Develop OEM Specialized Strategies and Templates for:

- *Internet of Things*
- *Application-to-Application Networking*
- *Real time Machine to Machine Communications*
- *Low Cost IC Chips for IOT Chirp Devices*