www.acuityeducare.com

Acuity Educare

Internet Of Things
SEM : V

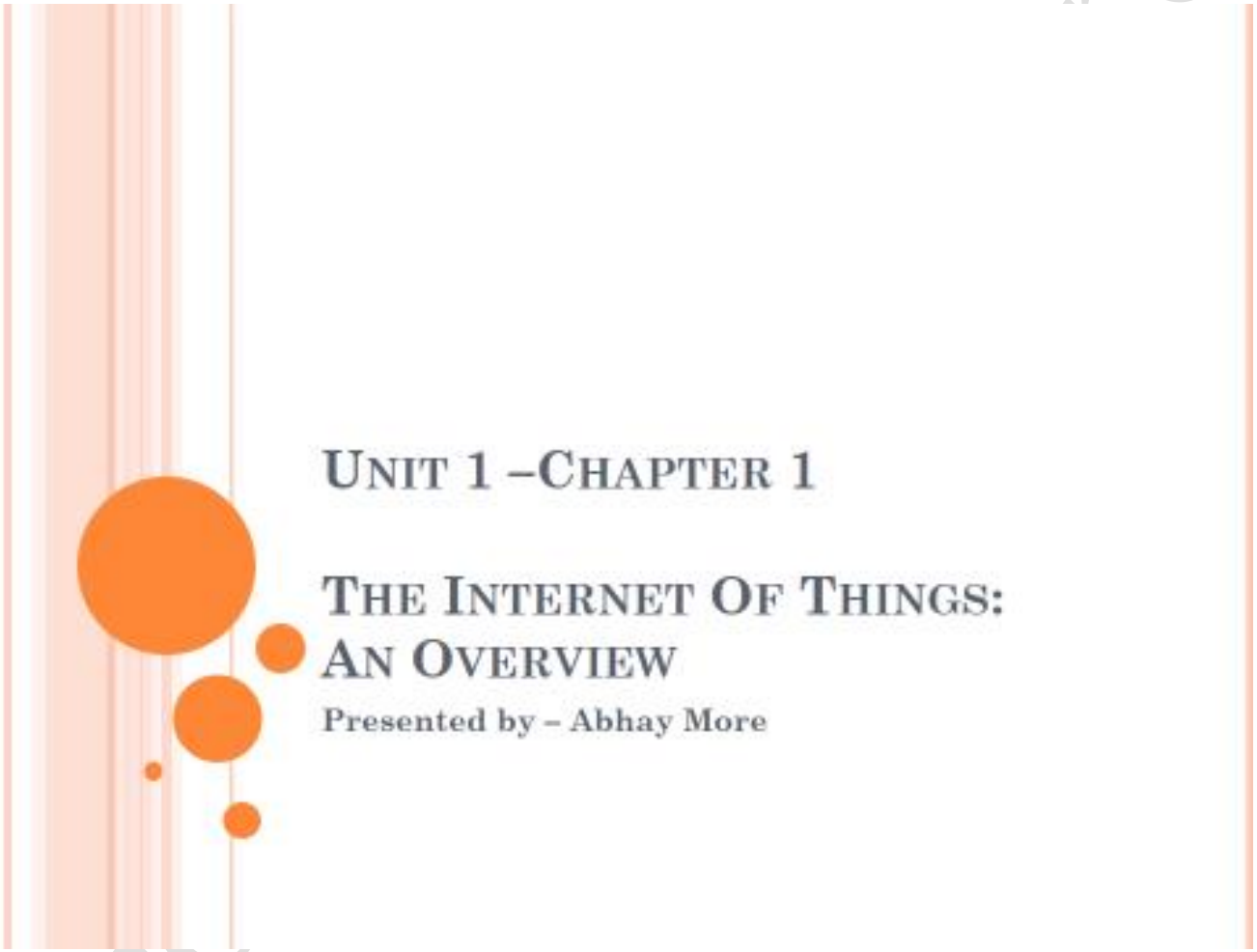
SEM V: UNIT 1

ACUITY EDUCARRE PVT. LTD.
Caring for Education...always!

607A, 6th floor, Ecstasy business park, city of joy, JSD road, mulund (W) | 8591065589/022-25600622

YouTube
Abhay More

Telegram
abhay_more



INTRODUCTION

- THE FIRST QUESTION that we should attempt to answer is, of course, what *is* the Internet of Things? What does the phrase "Internet of Things" mean?

- And how does it relate to the earlier buzzword "ubiquitous computing"? We sketch out some of the exciting projects that give a good flavour of this exciting field.

THE FLAVOUR OF INTERNET OF THINGS

- The alarm rings. As you open your eyes blearily, you see that it's five minutes later than your usual wakeup time. The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer.(See http://makezine.com/magazine/make-11/my-trainschedule-alarm-clock/.)
- In your kitchen, a blinking light reminds you it's time to take your tablets. If you forget, the medicine bottle cap goes online and emails your doctor to let her know. (See www.vitality.net/glowcaps.html.)

THE FLAVOUR OF INTERNET OF THINGS

- The alarm rings. As you open your eyes blearily, you see that it's five minutes later than your usual wakeup time. The clock has checked the train times online, and your train must be delayed, so it lets you sleep in a little longer.(See http://makezine.com/magazine/make-11/my-trainschedule-alarm-clock/.)
- In your kitchen, a blinking light reminds you it's time to take your tablets. If you forget, the medicine bottle cap goes online and emails your doctor to let her know. (See www.vitality.net/glowcaps.html.)

THE FLAVOUR OF INTERNET OF THINGS

- On your way out of the house, you catch a glow in the corner of your eye. Your umbrella handle is lit up, which means that it has checked the BBC weather reports and predicts rain. You sigh and pick it up. (See www.materious.com/#/projects/forecast/.)

THE FLAVOUR OF INTERNET OF THINGS

- As you pass the bus stop on the way to the station, you notice the large LCD display flash that the number 23 is due. It arrives when you turn the next corner. When the bus company first installed those displays, they ran on the expected timetable information only, but now that every bus has GPS tracking its location, they simply connect to the bus company's online service and always give the updated information. Various transport organizations have implemented this. London's TfL has some useful information on their signs at www.tfl.gov.uk/corporate/projectsandschemes/11560.aspx.

THE "INTERNET" OF "THINGS"

- We've looked at a number of examples of the Internet of Things, so what is the common thread that binds them together? And why the name? All the cases we saw used the *Internet* to send, receive, or communicate information.
- And in each case, the gadget that was connected to the Internet wasn't a computer, tablet, or mobile phone but an object, a *Thing*. These Things are designed for a purpose: the umbrella has a retractable canopy and a handle to hold it.

THE "INTERNET" OF "THINGS"

- A bus display has to be readable to public transport users, including the elderly and partially sighted and be able to survive poor weather conditions and the risk of vandalism.
- The sports bracelet is easy to wear while running, has a display that is large enough and bright enough to read even when you are moving, and will survive heat, cold, sweat, and rain.
- Many of the use cases could be fulfilled, and often are, by general-purpose computers. Although we don't carry a desktop PC around with us, many people do carry a laptop or tablet.

THE "INTERNET" OF "THINGS"

- More to the point, in almost every country now, most people do carry a mobile phone, and in many cases this is a smartphone that easily has enough power for any task one could throw at a computer. Let's see how well one could replicate these tasks with a smartphone.
- Viewing your bus provider's timetable with a smartphone web browser seems to fulfill the same function at first glance. But just consider that last phrase, "at first glance". On arriving at the bus stop, one can simply glance at the computerized timetable and see when the next bus is due.

THE "INTERNET" OF "THINGS"

- With a smartphone, if you have one and can afford the data use (which may be prohibitive if you are a foreign tourist), you have to take the phone out of your pocket or bag, unlock it, navigate to the right website (this may be the slowest and most complicated part of the process, whether you have to type the URL or use a QR code), and read the data from a small screen. In this time, you are not able to fully concentrate on the arriving buses and might even miss yours.

THE "INTERNET" OF "THINGS"

- You can track your runs with an app on your smartphone, and many people do: the phone has GPS, many other useful sensors, processing power, an Internet connection, and a great screen.
- But it turns out that such a phone isn't easy to carry on a run without worrying about dropping it or getting it wet. Plenty of carrying options are available, from a waist bag to an arm strap.
- The latter, in theory, enables you to read the device while you are running, but in practice reading details on the screen can be hard while you are jiggling up and down!

THE "INTERNET" OF "THINGS"

- To get around this difficulty, apps such as RunKeeper provide regular audio summaries which can be useful (www. runkeeper.com).
- Ultimately, a phone is a perfectly capable device for tracking your run, and most runners will find it a sufficient, comfortable, and fun way of logging their running data.
- However, others may well prefer a device worn as a watch or wristband, designed to be read on the move, worn in the rain, and connected to peripherals such as heart monitors.

THE "INTERNET" OF "THINGS"

- Of course, no mobile phone (or even tablet or laptop) is large enough or waterproof enough to use as an umbrella. However, you could pair a smartphone with a normal "dumb" umbrella, by checking an app to see whether it is likely to rain later, before you leave the house.
- Unlike a calm, subtle light in the umbrella stand, glimpsed from the corner of your eye as an ambient piece of information to process subconsciously when you pass it on the way out of your home, an app requires you to perform several actions

THE "INTERNET" OF "THINGS"

- If you are able to establish and maintain the habit of doing this check, it will be just as effective. Rather than having greater capabilities, the smart umbrella simply moves the same intelligence into your environment so that you don't have to change your routine.
- So the idea of the Internet of Things suggests that rather than having a small number of very powerful computing devices in your life (laptop, tablet, phone, music player), you might have a large number of devices which are perhaps less powerful (umbrella, bracelet, mirror, fridge, shoes).

THE "INTERNET" OF "THINGS"

- An earlier buzzword for roughly the same concept was "ubiquitous computing", also known by the ugly portmanteau "ubicomp", and this also reflects the huge number of possible objects that might contain computing technology. Now that the Internet is a central pipe for data, it's hard to imagine, for example, a PC that doesn't have an always on broadband connection. Younger readers may never have seen such a thing.

THE "INTERNET" OF "THINGS"

- As technologist and columnist Russell Davies joked at the 2012 Open Internet of Things Assembly in London:
- ***I can't understand why teddy bears did not have wifi before. A bear without wifi is barely alive, a semi-bear.***
- ***—http://storify.com/PepeBorras/opent-iot-assembly***

THE "INTERNET" OF "THINGS"

- The definition of ubicomp, however, would also include the Glade air fresheners which release scent when they detect movement in the room as part of its domain. That is to say, such a device is an intelligently programmed computer processor, driven by sensors in the real world, and driving output in the real world, all embedded into an everyday object.
- These factors make this ubicomp, and it is only differentiated from the "Internet of Things" by the fact that these days most of the really interesting things done with computing also involve an Internet connection

THE "INTERNET" OF "THINGS"

- But what does it mean to "connect an object to the Internet"? Clearly, sticking an Ethernet socket into a chair or a 3G modem into a sewing machine doesn't suddenly imbue the object with mysterious properties.
- Rather, there has to be some flow of information which connects the defining characteristics of the Thing with the world of data and processing represented by the Internet.
- The Thing is present, physically in the real world, in your home, your work, your car, or worn around your body.

THE "INTERNET" OF "THINGS"

- This means that it can receive inputs from your world and transform those into data which is sent onto the Internet for collection and processing.
- So your chair might collect information about how often you sit on it and for how long, while the sewing machine reports how much thread it has left and how many stitches it has sewn.

THE "INTERNET" OF "THINGS"

- In subsequent chapters, we talk a lot about "sensors". The presence of the Thing also means that it can produce outputs into your world with what we call "actuators".
- Some of these outputs could be triggered by data that has been collected and processed on the Internet. So your chair might vibrate to tell you that you have received email.

THE "INTERNET" OF "THINGS"

- We could summarize these components in the following appealingly simple (though, of course, also simplistic) equation:

Physical Object
+
Controller, Sensor, and Actuators
+
Internet
=
Internet of Things

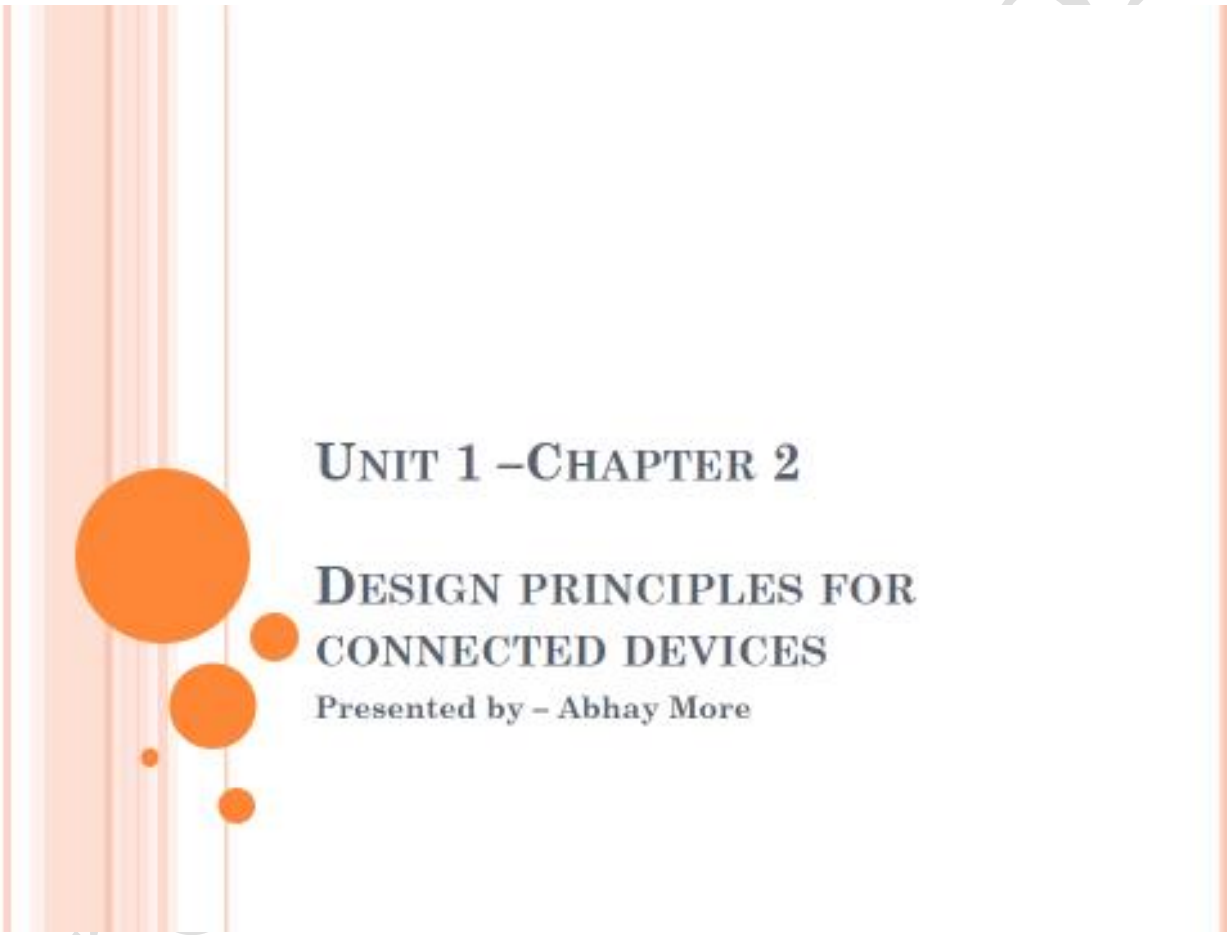
An equation for the Internet of Things.

THE "INTERNET" OF "THINGS"

- Note that in all the cases we've looked at, the form of the object follows the function of the Thing: your chair is designed to sit on, the sewing machine to sew at, and so on.
- The fact of also being connected to the Internet and having general-purpose computing capabilities doesn't necessarily have an impact on the form of the object at all.

THE "INTERNET" OF "THINGS"

- (One might argue that current-generation smartphones and tablets are in forms optimized for use as general-purpose computers, not as portable telephony devices. Certainly, on seeing the number of phones with scratched screens, one could ask whether they are designed to be easy to hold securely and resistant to drops and the impacts of everyday use.)



INTRODUCTION

- THIS BOOK IS called *Designing the Internet of Things, so does that mean we think* design is an important part of building a connected device?
- For some applications, you may think that design isn't important at first glance. Who cares what the box holding a production-line sensor looks like in a factory? Surely form is defined by function?
- If you haven't spent much time with designers, you might be thinking that design is merely concerned with the shape and look of an object—something ornamental, to give it a pleasing appearance. However, design is a much wider field than that.

INTRODUCTION

- Industrial design (sometimes also called *product design) does include the* form and decoration of the item, but it also covers functional aspects.
- The user interface to the object—be that on a screen or the more traditional buttons and switches—is also something of interest to the discipline of experience design. That takes the perspective of the end user of the design as its focus & seeks to create best solution for the user.
- Obviously "best" is subjective and could aim to make using the device as enjoyable as possible, or perhaps as efficient as possible, depending on the priorities of the designer or her team.

INTRODUCTION

- There are no hard boundaries between these differing facets of design, but we think designers of all types would agree that design is about more than just the surface look of the device.

CALM AND AMBIENT TECHNOLOGY

- The Internet of Things has its roots in the work done by Mark Weiser at Xerox PARC in the 1990s. His work didn't assume that there would be network connectivity but was concerned with what happens when computing power becomes cheap enough that it can be embedded into all manner of everyday objects. He coined the term *ubiquitous computing or ubicomp* to describe it.
- With its focus on computing power being embedded everywhere, ubicomp is often also referred to as *ambient computing*.

CALM AND AMBIENT TECHNOLOGY

- *However, the term "ambient"* also has connotations of being merely in the background, not something to which we actively pay attention and in some cases as something which we seek to remove (e.g., ambient noise in a sound recording).
- We prefer, as did Mark Weiser, the term *calm technology—systems which* don't vie for attention yet are ready to provide utility or useful information when we decide to give them some attention.
- Such proliferation of computing devices into the world comes with all manner of new challenges.

CALM AND AMBIENT TECHNOLOGY

- Issues include configuration, how to provide power to all these items, how they talk to each other, and how they communicate with us.
- The power and networking challenges are purely technical and are driving developments such as 6LoWPAN. This is a standards drive to take the next-generation Internet protocol (IPv6) to the simplest and lowest-power networked sensors.
- Configuration and user interaction, however, obviously involve people and so are difficult problems to solve with just technical solutions. This is where good design can aid in adoption and usability.

CALM AND AMBIENT TECHNOLOGY

- You can see this with the introduction of the Apple iPod in 2001. It wasn't the first portable MP3 player, but the combination of the scroll-wheel user interface and the companion iTunes software made it much easier to use and turned them into mass market gadgets.
- For connected devices which are just sensing their world, or generally acting as *inputs, as long as their activity doesn't require them to query the people* around them, there shouldn't be any issues. They will happily collect information and deposit it into some repository online for processing or analysis.

CALM AND AMBIENT TECHNOLOGY

- When the devices start interacting with people, things get more complicated. Already we're seeing the number of notifications, pop-ups. When we scale up this number to include hundreds of new services and applications across the rest of the objects in our world, it will become an attention-seeking cacophony.
- Mark Weiser and John Seely Brown proposed an antidote to such a problem by suggesting we design ubiquitous computing systems to seek to blend into their surroundings.

CALM AND AMBIENT TECHNOLOGY

- A great example of this approach is *Live Wire, one of the first Internet of* Things devices. Created by artist Natalie Jeremijenko when she was in residence at Xerox PARC under the guidance of Mark Weiser, Live Wire (also sometimes called Dangling String) is a simple device: an electric motor connected to an eight-foot long piece of plastic string.
- The power for the motor is provided by the data transmissions on the Ethernet network to which it is connected, so it twitches whenever a packet of information is sent across the network.

CALM AND AMBIENT TECHNOLOGY

- Under normal, light network load, the string twitches occasionally. If the network is overloaded, the string whirls madly, accompanied by a distinctive noise from the motor's activity.
- Conversely, if no network activity is occurring, an unusual stillness comes over the string.
- Both extremes of activity therefore alert the nearby human (who is used to the normal behaviour) that something is amiss and lets him investigate further.
- Like Live Wire, Bubblino—Adrian's Internet of Things bubble machine which searches Twitter and blows bubbles.

CALM AND AMBIENT TECHNOLOGY

- It blows bubbles when it finds new tweets matching a search phrase is a good example in which the side effect of motor is to generate an audible notification that something is happening.
- These noisy "side effects" are something that we should also be wary of losing with a move to "better" technology. Years ago all airport and railway arrival and departure boards were built using split-flap displays. They consisted of a number of flaps on a roll—sometimes with full place names printed onto the flap, and in other times as individually controllable characters—which could be rotated until they showed the correct item.

CALM AND AMBIENT TECHNOLOGY

- In most locations these split-flap displays have been phased out in preference for dot-matrix LED displays. The newer displays are much easier to update with new destinations. They also have capabilities of horizontally scrolling messages.
- Sadly, in doing so they have lost one important characteristic: the flurry of clacking as the display updates. As a result, passengers waiting in a station terminal must stare endlessly up at the display waiting for their train to be announced, rather than attending to other tasks and checking the departures board only when a change occurs.

CALM AND AMBIENT TECHNOLOGY

- Chris is an interaction designer and realised that every morning he would check a few different apps on his phone to find out things like the weather forecast, his appointments for the day, and how the trains on the London Underground were running.
- He didn't have any power sockets near to his front door, and so settled on a bedside information display instead. Given that it would be always on and next to where he sleeps, a standard monitor or other LCD display, with its persistent glow, wouldn't be suitable.

CALM AND AMBIENT TECHNOLOGY

- The e-ink display on a Kindle, however, was ideal. He took advantage of the WiFi connectivity and computing power in the Kindle to make it a selfcontained device and configured it to just display a web page, which refreshed every few minutes. The resultant device, which Chris called the Kindleframe which will display the up to date information everyday.

MAGIC AS METAPHOR

- One of the main issues with introducing any new technology or service that is radically different from the norm is getting people to understand and accept it.
- There are many examples when the main difference between a failed technology and a wildly successful one is that the successful one arrived a few years later, when people were more receptive to what was offered.
- Technology blogger Venkatesh Rao came up with a good term to help explain how new technology becomes adopted.

MAGIC AS METAPHOR

- He posits that we don't see the present, the world that we live in now, as something that is changing. If we step back for a second, we do *know that it has changed, although the* big advances sneak up on us over time, hidden in plain sight. Rao called this concept the *manufactured normalcy field.*
- As a result, the successful user-experience designer is the one who presents users with an experience which doesn't stretch the boundaries of their particular normalcy field too far, even if the underlying technology being employed is a huge leap ahead of the norm.

MAGIC AS METAPHOR

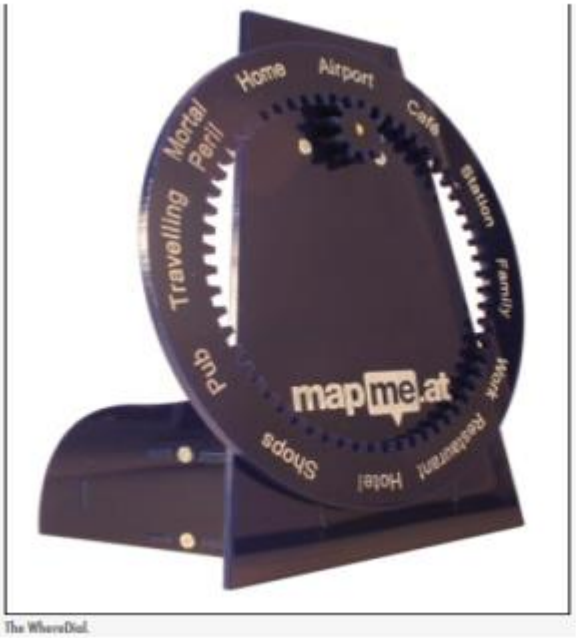
- For example, the mobile phone was first introduced as a phone that wasn't tethered to a particular location. Now broadly the same technology is used to provide a portable Internet terminal, which can play movies, carry your entire music collection, and (every now and then) make phone calls.
- Introducing technology to people in terms of something they already understand is a tried and tested effect: computers started off as glorified typewriters; graphical user interfaces as desktops....

MAGIC AS METAPHOR

- So, what of the Internet of Things? As we saw in the last chapter, Arthur C. Clarke has claimed that "any sufficiently advanced technology is indistinguishable from magic," and given that the Internet of Things commonly bestows semi-hidden capabilities onto everyday objects, maybe the enchanted objects of magic and fairy tale are a good metaphor to help people grasp the possibilities.
- Some Internet of Things projects draw their inspiration directly from magic.

MAGIC AS METAPHOR

- For example, John McKerrell's WhereDial takes its lead from the clock in *Harry Potter which tracked the location of the members of the Weasley* family. The Weasley clock could use magic to divine the whereabouts of each family member and was therefore also aware of when they were in mortal peril. The WhereDial, by comparison, has to rely on mere technology for its capabilities; however, with the GPS chipsets in smartphones and location check-in services like FourSquare, it isn't much of a leap to also own an ornament which updates to show when you are at work, or travelling, or at a restaurant.



The WhereDial

MAGIC AS METAPHOR

- Other projects have a less obvious influence.
- Enchanted mirrors seem a popular choice in design research, although they haven't quite reached the capabilities of the evil queen's "Mirror, mirror on the wall" in the Snow White tale. *They tend to show information which is* useful as you start or end your day, in line with the expected times when you'd use a bathroom mirror.
- You get the time and can check appointments and traffic and weather information while having your morning shower.

MAGIC AS METAPHOR

- Presumably, it is merely a matter of time before one shows the number of "Likes" you have on Facebook, thus turning it into the modern equivalent of the evil queen's query to know "who is the fairest of them all?"
- Ambient Devices then took the idea one step further and built an enchanted umbrella. It can read the weather forecast, and the handle glows gently if rain is expected, alerting you to the fact that you may need to pick it up as you head out of the house.

PRIVACY

- The Internet of Things devices that we own aren't the only ones that should concern us when it comes to matters of trust. With more sensors and device watching us and reporting data to the Internet, the privacy of third parties who cross our sensors' paths (either by accident or design) is an important consideration. Designers of an IoT service will need to balance carefully.
- For certain realms, such as health care, privacy concerns are an obvious issue. However, even seemingly innocuous applications can leak personal information, so you should be alert to the danger and take measures to avoid it.

PRIVACY – KEEPING SECRETS

- This advice is perfectly illustrated with an example from an early instrumented car park in a Westfield shopping mall in Australia. Each parking bay is overlooked by a small sensor from Park Assist, which uses a cheap camera to tell whether the space is occupied.
- The sensors are all networked and presumably can provide analytics to the owner of the car park as to its usage. A light on the sensor can help guide drivers to a free space. All useful and harmless stuff.
- The problem came with a more advanced feature of the system.

PRIVACY – KEEPING SECRETS

- The shopping mall provided a smartphone app for visitors to download so that they could find out more information about the facilities. One of the features of the app was a Find My Car option.
- Choosing that, you were prompted to enter the first few characters of your licence plate, and app would return four small photos of potential matches—from OCR software processing the sensor data on mall's server.
- The returned images were only thumbnails—good enough to recognise which was your car, but not much else, and the licence plates were blurry and hard to see.

PRIVACY – KEEPING SECRETS

- However, security professional Troy Hunt found that implementation method left a lot to be desired. With a fairly simple, off-the-shelf bit of software, Troy was able to watch what information app was requesting from server & found that it was an unencrypted web request.
- The initial request URL had a number of parameters, including the search string, but also included number of results to return.
- That request returned a chunk of data, which included the URLs for the four images to download, but also included a raft of additional pieces of information.

PRIVACY – KEEPING SECRETS

- Presumably, it was easier for the developer of web service to just return all the available data than to restrict it.
- The extra data included, for example, the IP addresses of each of the sensor units, but more importantly, it also included the full licence plate for each vehicle and the length of time it had been parked in the space.
- By altering the search parameters, Troy found that he could request many more than the four matches, and it was also possible to omit the licence plate search string.

PRIVACY – KEEPING SECRETS

- That meant he could download a full list of licence plates from all 2550 parking spaces in a single web request, whenever he liked.
- Once alerted to the problem, Westfield and Park Assist were quick to disable the feature and then work with Troy to build a better solution. However, that situation came about only because Troy was generous enough to bring it to their attention.
- As founder of WikiLeaks, Julian Assange, has said, "The best way to keep a secret is to never have it"

PRIVACY – KEEPING SECRETS

- If you can avoid gathering and/or storing the data in the first place, you need not worry about disclosing it accidentally.
- In this day and age, it is standard practice to never store passwords as cleartext. You could also consider applying the standard mechanisms for password encryption, such as the one-way hash, to other pieces of data. This technique was suggested by Peter Wayner.
- Rather than storing identifying data in the database, if you don't need to return it to its original form, use a one-way hashed version of the information instead.

PRIVACY – KEEPING SECRETS

- Doing so still lets the originators of the data find their data (as they can provide it to be hashed again) and allows statistics gathering and reports, and the like, without storing the data in a recoverable form.

PRIVACY – HASHES

- One-way hashing is a cryptographic technique used to condense an arbitrarily sized chunk of data into a fixed-sized piece, called the hash. It's called one-way hashing because there isn't an easy way, given the resultant hash, to work out what the original data was. Hashing algorithms are so designed such that even a small difference in the input data leads to a huge difference in the output hash.
- This makes them very useful for times when you want to verify that two pieces of data are identical without having to store them for comparison.

PRIVACY – HASHES

- That's useful when the data you want to compare is either very large or something you don't want to store in its original form.
- The most common use of cryptographic hashes is in password verification. Rather than store the user's password, the service provider stores a hash of the password. When the user wants to authenticate himself in the future, the hash can be recalculated; if it matches the stored one, the service can be reasonably sure that the user has provided the correct password.

PRIVACY – HASHES

- That's useful when the data you want to compare is either very large or something you don't want to store in its original form.
- The most common use of cryptographic hashes is in password verification. Rather than store the user's password, the service provider stores a hash of the password. When the user wants to authenticate himself in the future, the hash can be recalculated; if it matches the stored one, the service can be reasonably sure that the user has provided the correct password.

PRIVACY – WHOSE DATA IS IT ANYWAY?

- On private property, you can more easily claim that the members of the public don't have such a right, but perhaps the property owner might assert rights to the data rather than whoever installed the camera.
- And there are many places such as shopping malls which, to all intents and purposes, look and feel like public spaces, despite being privately owned. How do things stand in those areas?

WEB THINKING FOR CONNECTED DEVICES

- When you are thinking of the networked aspect of Internet of Things objects, it might help to draw on experiences and design guidelines from existing network deployments.
- The obvious choice would be to look at design guides to the World Wide Web and the Internet itself; after all, the term *Internet of Things will look quaint in the future when we accept that it* is completely natural for the Internet to be full of Things as well as computers and phones.

WEB THINKING FOR CONNECTED DEVICES

- In an early version of the specification for TCP Jon Postel wrote: "Be conservative in what you do, be liberal in what you accept from others". Since then, that *robustness principle has become so well known that it is* commonly referred to as *Postel's Law.*
- *It is good to bear this in mind when* designing or building anything which must interact with other services— particularly when you aren't the one building the other components with which your system interacts.

WEB THINKING FOR CONNECTED DEVICES – SMALL PIECES LOOSELY JOINED

- Even if you are building all the components of your service, it makes sense not to couple them too tightly together. The Internet flourished not because it is neatly controlled from a central location, but because it isn't; it is a collection of services and machines following the maxim of *small pieces, loosely joined.*
- What this means for the architects of a service is that each piece should be designed to do one thing well and not rely too much on tight integration with the separate components it uses.

WEB THINKING FOR CONNECTED DEVICES – SMALL PIECES LOOSELY JOINED

- Strive also to make the components more generalised and able to serve other systems which require a similar function. That will help you, and others, to reuse and repurpose the components to build new capabilities unimagined when the initial system was commissioned.
- Where possible, use existing standards and protocols rather than inventing your own. Any loss of elegance or efficiency of code size or electronics will be outweighed by the availability of standard libraries and skills for people to interact with, and build on, your system.

WEB THINKING FOR CONNECTED DEVICES – SMALL PIECES LOOSELY JOINED

- For example, when the designers at Twitter implemented the search feature, they chose to include a more machine-readable option in the form of a feed of results in the standard Atom syndication format. I'm fairly certain that they didn't expect it to be consumed by an Arduino, which would then use it to activate a bubble machine to blow bubbles for each new tweet.

WEB THINKING FOR CONNECTED DEVICES – FIRST CLASS CITIZENS ON THE INTERNET

- An extension of the concept of loose coupling is to strive to make your devices first-class citizens on the Internet. What do we mean by that? Where possible, you should use the same protocols and conventions that the rest of the Internet uses.
- In the early days of any new development, it is tempting to compromise and choose protocols which are easier to implement. Indeed, many middleware providers encourage that practice, claiming that such low-powered end-points are not capable enough (in processing power or RAM or in the networks they use).

WEB THINKING FOR CONNECTED DEVICES – FIRST CLASS CITIZENS ON THE INTERNET

- There is an element of truth to this—though not as much as you will be led to believe—but a good rule of thumb for the past 20 years or more has been to expect the IP protocol to penetrate everywhere. We see no reason for it not to continue into the Internet of Things.
- In the few cases where the existing protocols don't work, such as in extremely low-powered sensors, a better solution is to work with your peers to amend existing standards or create new open standards which address the issue within the conventional standards groups.

WEB THINKING FOR CONNECTED DEVICES – FIRST CLASS CITIZENS ON THE INTERNET

- The evolution of the mobile web serves as a good cautionary example. When mobile phones were first being connected to the Internet, it was deemed too difficult for them to talk to web servers directly, and a whole suite of new protocols, Wireless Application Protocol (WAP), were developed.
- Handsets accessed either bespoke WAP sites or standard websites via a WAP/web gateway server. As they needed site developers to learn a whole new set of skills, the new protocols gained little traction; without the sites to visit, user adoption of the mobile web was slow.

WEB THINKING FOR CONNECTED DEVICES – FIRST CLASS CITIZENS ON THE INTERNET

- As the handsets evolved to talk the standard protocols of the web, even though the display wasn't perfect, it was good enough for users to begin using it.
- With the increase in usage directly to the websites developers could then see the demand for mobile-friendly features.
- Given that those features could be developed with the tools with which they were already familiar, over time the mobile web has just become a facet of the web in general.

WEB THINKING FOR CONNECTED DEVICES – GRACEFUL DEGRADATION

- Because the Internet is so welcoming and tolerant of all sorts of devices and services, the endpoints have a massively disparate and diverse range of capabilities. As a result, building services which can be used by all of them is a nearly impossible task. However, a number of design patterns have evolved to mitigate the problem.
- The first is to acknowledge that the wealth of different devices is likely to be a problem and design your system to expect it.

WEB THINKING FOR CONNECTED DEVICES – GRACEFUL DEGRADATION

- If you need to come up with a format for some data being transferred between devices, include a way to differentiate between successive versions of the formats—ideally in such a way that older devices can still mostly read newer formats. This is known as *backwards compatibility*.
- The HTML format does this by stating that any client should ignore any tags (the text inside the <>) that it doesn't understand, so newer versions can add new tags without breaking older parsers. The HTTP protocol uses a slightly different technique in which each end specifies the version of protocol that it supports, & other end takes care not to use any of the newer features.

WEB THINKING FOR CONNECTED DEVICES – GRACEFUL DEGRADATION

- The other common technique is to use something called *graceful degradation*.
- This technique involves aiming to provide a fully featured experience if the client is capable of it but then falling back—potentially in a number of levels—to a less feature-rich experience on less capable clients. This capability can even span technologies and does so in its most common usage.
- When trying to implement rich web applications such as Twitter and Gmail, the coder wants to use an assortment of advanced JavaScript features in modern browsers.

WEB THINKING FOR CONNECTED DEVICES – GRACEFUL DEGRADATION

- Well-written apps check that the features are available before using them, but if those features aren't available, the apps might limit themselves to a version using simpler (and more common) JavaScript code—still validating form contents before submitting them, for example, but no longer calling over to the server to provide autocomplete.
- And if JavaScript isn't available at all, they fall back to basic HTML forms. This experience is not as nice as the full one but better than no experience at all!

WEB THINKING FOR CONNECTED DEVICES – GRACEFUL DEGRADATION

- Aside from using the same techniques when designing our connected devices, we might also be able to apply that approach to the devices themselves to give a degree of fault tolerance.
- When your early-adopter Internet Fridge can no longer talk to your WiFi because it's only IPv4 and the world has moved to IPv6, you would still be able to use its touchscreen to write messages and view the photos stored in the USB stick stuck in it. And if the touchscreen breaks, you should still be able to keep the food inside it cold.

AFFORDANCES

- In his book *The Design of Everyday Things, Donald Norman defines affordances as follows:*

  Affordances provide strong clues to the operations of things. Plates are for pushing. Knobs are for turning. Slots are for inserting things into. Balls are for throwing or bouncing. When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction is required. Complex things may require explanation, but simple things should not. When simple things need pictures, labels, or instructions, the design has failed.

- *—The Design of Everyday Things, MIT Press, 1998*

AFFORDANCES

- As adoption of the Internet of Things gathers pace, more and more of our cities, homes, and environment will become suffused with technology. With these additional behaviours and capabilities will come additional complexity—something that successful designers of connected devices and services will need to counter.
- By their very nature, many of the new capabilities bestowed upon objects will be hidden from sight or not immediately apparent from first glance, which makes intuitive design difficult. What are the affordances of digitally enhanced objects?

AFFORDANCES

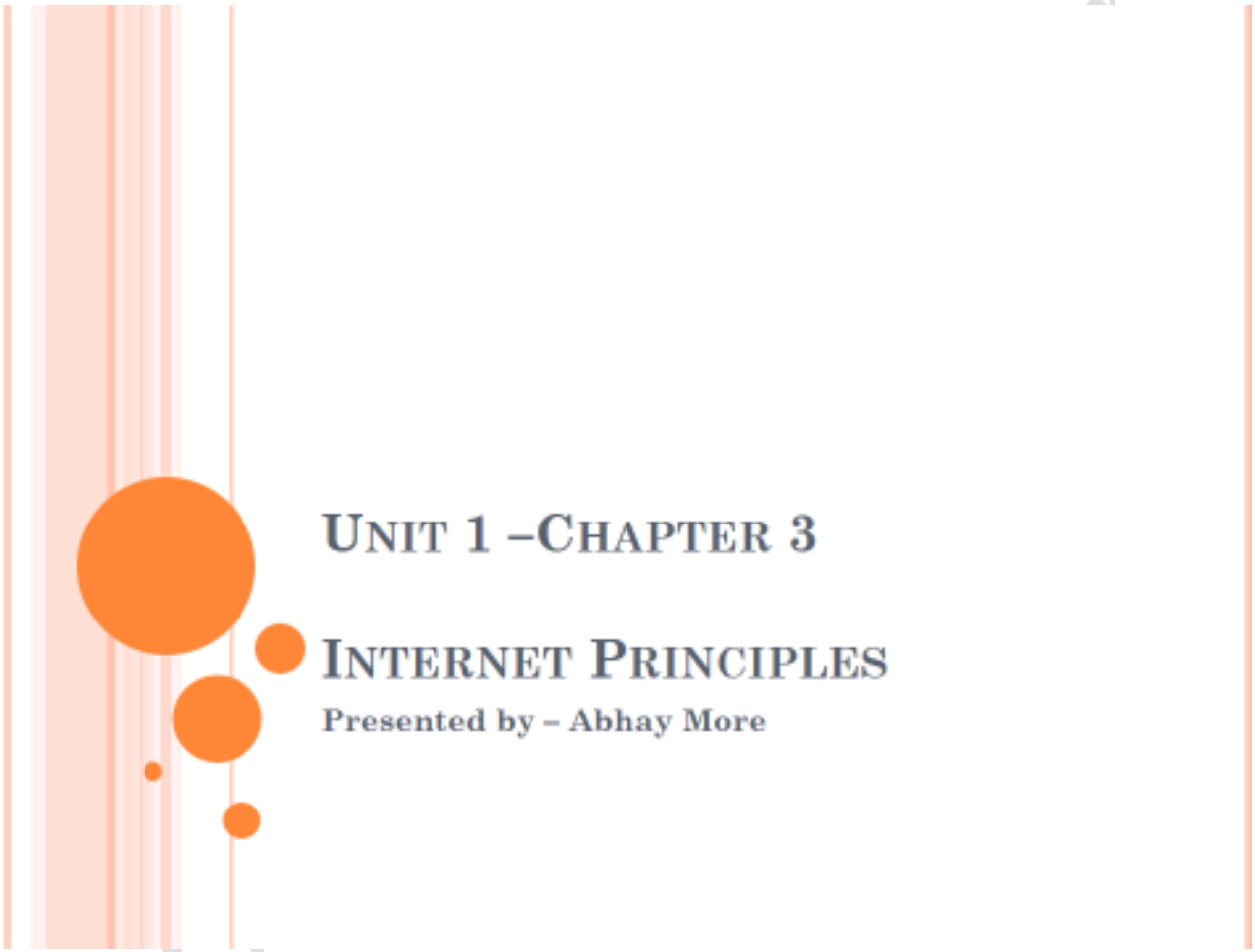
- How do we convey to the user of an object that it can communicate with the cloud? Or that this device is capable of short-range communication such as RFID? What does it mean that a toy knows what the temperature is or when it is shaken?
- An important start is to keep the existing affordances of the object being enhanced. Users who don't realise that a device has any extra capabilities should still be able to use it as if it hasn't.

AFFORDANCES

- For example, a "dumb" light dimmer switch is usually implemented as a rotary knob which gives the user fine-grained control over the brightness.
- When it is hooked up to a home-automation system, the difficulties of synchronising the state of both the knob and the light level, now that the brightness can be controlled remotely or automatically, often leads to the knob being replaced with a couple of buttons.
- As a result, the user loses the ability to make both rapid large changes and smaller, fine-grained adjustments.

AFFORDANCES

- A better approach would be to adopt the system used on many stereo systems where the volume knob is a motorized potentiometer; the user can still adjust it in the conventional manner, and any changes made by the remote are instantly reflected in the position of the volume knob.
- Similar rules apply when designing physical interfaces. Don't overload familiar connectors with unfamiliar behaviours. For example, you shouldn't use 3.5mm audio jacks to provide power.



UNIT 1 –CHAPTER 3

INTERNET PRINCIPLES

Presented by – Abhay More

YouTube - Abhay More | Telegram - abhay_more
607A, 6th floor, Ecstasy business park, city of joy, JSD road, mulund (W) | 8591065589/022-25600622

INTRODUCTION

- This chapter is a short, high-level survey, designed to be readable rather than complete.
- Let's start by looking at an example of how communication over long distances could work in the real world and then compare it to how information is transmitted across the virtual world of the Internet.

INTERNET COMMUNICATIONS: AN OVERVIEW

- Suppose that you wanted to send a message to the authors of this book, but you didn't have the postal address, and you didn't have any way to look up our phone number (because in this example you don't have the Internet).
- You remember that we're from the UK, and London is the biggest city in the UK. So you send a postcard to your cousin Bob, who lives there. Your cousin sees that the postcard is for some crazy hardware and technology people. So he puts the postcard in an envelope and drops it off at the London Hackspace because the guys there probably know what to do with it.

INTERNET COMMUNICATIONS: AN OVERVIEW

- At the Hackspace, Jonty picks up the envelope and sees that it's for some people in Liverpool. Like all good Londoners, Jonty never goes anywhere to the north of Watford, but he remembers that Manchester is in the north too.
- So he calls up the Manchester Digital Laboratory (MadLab), opens the envelope to read the contents, and says, "Hey, I've got this message for Adrian and Hakim in Liverpool. Can you pass it on?" The guys at MadLab ask whether anyone knows who we are, and it turns out that Hwa Young does. So the next time she comes to Liverpool, she delivers the postcard to us.

IP

- The preceding scenario describes how the Internet Protocol (IP) works. Data is sent from one machine to another in a packet, with a destination address and a source address in a standardised format (a "protocol").
- Just like the original sender of the message in the example, the sending machine doesn't always know the best route to the destination in advance. Most of the time, the packets of data have to go through a number of intermediary machines, called *routers, to reach their destination.*

IP

- In our example, a postcard was placed in an envelope before getting passed onwards. This happens with Internet packets, too. So, an *IP packet is a block* of data along with the same kind of information you would write on a physical envelope: the name and address of the server, and so on.
- But if an IP packet ever gets transmitted across your local wired network via an Ethernet cable—the cable that connects your home broadband router or your office local area network (LAN) to a desktop PC—then whole packet will get bundled up into another type of envelope, an *Ethernet Frame, which adds extra information.*

IP

- Of course, it's possible that your cousin Bob didn't know about the London Hackspace, and then maybe the message would have got stuck with him.
- You would have had no way to know whether it got there.
- This is how IP works. There is no guarantee, and you can send only what will fit in a single packet.

TCP

- What if you wanted to send longer messages than fit on a postcard? Or wanted to make sure your messages got through?
- What if everyone agreed that postcards written in green ink meant that we cared about whether they arrived. And that we would always number them, so if we wanted to send longer messages, we could. The person at the other end would be able to put the messages in order, even if they got delivered in the wrong order.
- We would send back postcard notifications that jus told you which postcards we had received, so you could resend any that went missing.

TCP

- That is basically how the Transmission Control Protocol (TCP) works. The simplest transport protocol on the Internet, TCP is built on top of the basic IP protocol and adds sequence numbers, acknowledgements, and retransmissions.
- This means that a message sent with TCP can be arbitrarily long and give the sender some assurance that it actually arrived at the destination intact.
- Because the combination of TCP and IP is so useful, many services are built on it in turn, such as email and the HTTP protocol that transmits information across the World Wide Web.

THE IP PROTOCOL SUITE (TCP/IP)

- The combination of TCP and IP is so ubiquitous that we often refer simply to "TCP/IP" to describe a whole suite or stack of protocols layered on top of each other, each layer building on the capabilities of the one below.
- The low-level protocols at the *link layer manage the transfer of bits of* information across a network link. This could be by an Ethernet cable, by WiFi, or across a telephone network designed to carry data over the Personal Area Network (PAN),
- The *Internet layer then sits on top of these various links and abstracts* away the gory details in favour of a simple destination address.

THE IP PROTOCOL SUITE (TCP/IP)

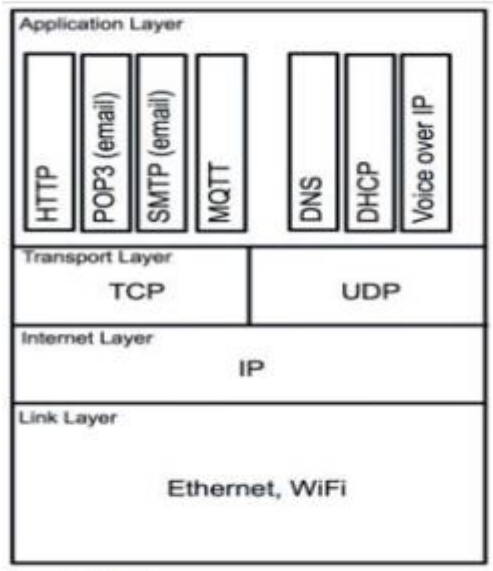
- Then TCP, which lives in the *transport layer, sits on top of IP and* extends it with more sophisticated control of the messages passed.
- Finally, the *application layer contains the protocols that deal with* fetching web pages, sending emails, and Internet telephony. Of these, HTTP is the most ubiquitous for the web, and indeed for communication between Internet of Things devices.

UDP

- As you can see, TCP is not the only protocol in the transport layer. Unlike TCP, but as with IP itself, in UDP each message may or may not arrive. No handshake or retransmission occurs, nor is there any delay to wait for messages in sequence. These limitations make TCP preferable for many of the tasks that Internet of Things devices will be used for.
- The lack of overhead, however, makes UDP useful for applications such as streaming data, which can cope with minor errors but doesn't like delays.

UDP

- Voice over IP (VoIP)—computer-based telephony, such as Skype—is an example of this: missing one packet might cause a tiny glitch in the sound quality, but waiting for several packets to arrive in the right order could make the speech too jittery to be easy to understand.



The Internet Protocol suite.

IP ADDRESSES

- We mentioned earlier that the Internet Protocol knows the addresses of the destination and source devices. But what does an "address" consist of? Here is a typical human (or in this case, hobbit) address:

Bilbo Baggins
"Bag End", Bagshot Row
Hobbiton
The Shire
Middle Earth

IP ADDRESSES

- In the world of low-level computer networking, however, numbers are much easier to deal with. So, IP addresses are numbers. In Internet Protocol version 4 (IPv4), almost 4.3 billion IP addresses are possible—4,294,967,296 to be precise, or 232.
- Though that is convenient for computers, it's tough for humans to read, so IP addresses are usually written as four 8-bit numbers separated by dots (from 0.0.0.0 to 255.255.255.255)—for example, 192.168.0.1 (which is often the address of your home router) or 8.8.8.8 (which is the address of one of Google's DNS servers).

IP ADDRESSES

- This "dotted quad" is still exactly equivalent to the 32-bit number. As well as being simply easier for humans to remember, it is also easier to infer information about the address by grouping certain blocks of addresses together. For example,
- 8.8.8.x — One of several IP ranges assigned to Google.
- 192.168.x.x — A range assigned for private networks. Your home or office network router may well assign IP addresses in this range.
- 10.x.x.x — Another private range.

IP ADDRESSES

- Every machine on the Internet has at least one IP address. That means every computer, every network-connected printer, every smartphone, and every Internet of Things device has one. If you already have a Raspberry Pi, an Arduino board, or any of the other microcontrollers they will expect to get their own IP address, too.
- When you consider this fact, those 4 billion addresses suddenly look as if they might not be enough.

IP ADDRESSES

- The private ranges such as 192.168.x.x offer one mitigation to this problem. Your home or office network might have only one publicly visible IP address. However, you could have all the IP addresses in the range 192.168.0.0 to 192.168.255.255 (2^{16} = 65,536 addresses) assigned to distinct devices.
- A better solution to this problem is the next generation of Internet Protocol, IPv6.

DNS

- Although computers can easily handle 32-bit numbers, even formatted as dotted quads they are easy for most humans to forget. The Domain Name System (DNS) helps our feeble brains navigate the Internet. Domain names, such as the following, are familiar to us from the web, or perhaps from email or other services:

google.com

bbc.co.uk

wiley.com

arduino.cc

DNS

- Each domain name has a top-level domain (TLD), like .com or.uk, which further subdivides into .co.uk and .gov.uk, and so on. This top-level domain knows where to find more information about the domains within it; for example, .com knows where to find google.com and wiley.com.
- The preceding examples are all instantly recognizable as website names, which is to say you could enter them into your web browser as, for example, http://www.google.com.

DNS

- But DNS can also point to other services on the Internet—for example:

  pop3.google.com — For receiving email from Gmail

  smtp.google.com — For sending email to Gmail

  ns1.google.com — The address of one of Google's many DNS servers

STATIC IP ADDRESS ASSIGNMENT

- How do you get assigned an IP address? If you have bought a server-hosting package from an Internet service provider (ISP), you might typically be given a single IP address. But the company itself has been given a block of addresses to assign. Historically, these were ranges of different sizes, typically separated into "classes" of 8 bits, 16 bits, or 24 bits:

- Class A — From 0.x.x.x
- Class B — From 128.0.x.x
- Class C — From 192.0.0.x

STATIC IP ADDRESS ASSIGNMENT

- The class C ranges had a mere 8 bits (256 addresses) assigned to them, while the class A ranges had many more addresses and would therefore be given only to the very largest of Internet organisations.
- The rigid separation of address ranges into classes was not very efficient; every entity would want to keep enough spare addresses for future expansion, but this means that many addresses would remain unused.
- With the explosion of the number of devices connecting to the Internet, the scheme has been superceded since 1993 by Classless Inter-Domain Routing (CIDR).

STATIC IP ADDRESS ASSIGNMENT

- CIDR which allows you to specify exactly how many bits of the address are fixed. So, the class A addresses we mentioned above would be equivalent to 0.0.0.0/8, while a class C might be 208.215.179.0/24.
- The admininstrator makes a note of the addresses and updates DNS records and so on to point to these addresses. We call this kind of address *static because once assigned it won't change again without human* intervention.

STATIC IP ADDRESS ASSIGNMENT

- Now consider your home network: every time you plug a desktop PC to your router, connect your laptop or phone to the wireless, or switch on your network-enabled printer, this device has to get an IP address (often in the range 192.168.0.0/16).
- You *could assign an address sequentially yourself,* but the typical person at home isn't a system administrator and may not keep thorough records. If your brother, who used to use the address 192.168.0.5 but hasn't been home for ages, comes back to find that your new laser printer now has that address, he won't be able to connect to the Internet.

DYNAMIC IP ADDRESS ASSIGNMENT

- Thankfully, we don't typically have to choose an IP address for every device we connect to a network. Instead, when you connect a laptop, a printer, or even a Twitter-following bubble machine, it can request an IP address from the network itself using the Dynamic Host Configuration Protocol (DHCP).
- When the device tries to connect, instead of checking its internal configuration for its address, it sends a message to the router asking for an address. The router assigns it an address.

DYNAMIC IP ADDRESS ASSIGNMENT

- This is not a static IP address which belongs to the device indefinitely; rather, it is a temporary "lease" which is selected *dynamically according to which addresses are currently available. If the* router is rebooted, the lease expires, or the device is switched off, some other device may end up with that IP address.
- Even the simplest computing devices such as the Arduino board, can use DHCP. Although the Arduino's Ethernet library allows you to configure a static IP address, you can also request one via DHCP.

DYNAMIC IP ADDRESS ASSIGNMENT

- Using a static address may be fine for development (if you are the only person connected to it with that address), but for working in groups or preparing a device to be distributed to other people on arbitrary networks, you almost certainly want a dynamic IP address.

IPv6

- When IP was standardised, few could have predicted how quickly the 4.3 billion addresses that IPv4 allowed for would be allocated. The expected growth of the Internet of Things can only speed up this trend.
- If your mobile phone, watch, MP3 player, augmented reality sunglasses, and telehealth or sports-monitoring devices are all connected to the Internet, then you personally are carrying half a dozen IP addresses already. Perhaps rather than a single health monitoring device, you have several distributed across your person, with sensors for temperature, heart rate, insulin levels, and any number of other stimuli.

IPv6

- At home you would start with all your electronic devices being connected. But beyond that, you might also have sensors at every door and window for security. More sensitive sound sensors to detect the presence of mice or beetles. Other sensors to check temperature, moisture, and airflow levels for efficiency. It is hard to predict what order of number of Internet connected devices a household might have in the near future. Tens? Hundreds? Thousands?

IPv6

- Enter IPv6, which uses 128-bit addresses, usually displayed to users as eight groups of four hexadecimal digits—for example, 2001:0db8:85a3:0042:0000:8a2e:0370:7334. The address space of 2^{128} is too large when compared to IPv4.
- The new standard was discussed during the 1980s and finally released in 1996. In 2013, it is still less popular than IPv4.
- It was originally expected that mobile phones connected to the Internet would push this technology over the tipping point.

IPv6 and Powering Devices

- We can see that an explosion in the number of Internet of Things devices will almost certainly need IPv6 in the future. But we also have to consider the power consumption of all these devices.
- We know that we can regularly charge and maintain a small handful of devices. At any one moment, we might have a laptop, a tablet, a phone, a camera, and a music player plugged in to charge. The constant juggling of power sockets, chargers, and cables is feasible but fiddly.

IPv6 and Powering Devices

- The requirements for large numbers of devices, however, are very different. The devices should be low power and very reliable, while still being capable of connecting to the Internet.
- Perhaps to accomplish this, these devices will team together in a mesh network. This is the vision of 6LoWPAN, an IETF working group proposing solutions for "IPv6 over Low power Wireless Personal Area Networks"

Conclusion on IPv6

- Although IPv6 is, or will be, big news. In 2013, you can find more libraries, more hardware, and more people that can support IPv4, and this is what will be most helpful when you are moving from prototype to production on an Internet of Things device.
- Even though we are getting close to the tipping point, existing IPv4 services will be able to migrate to IPv6 networks with minimal or possibly no rewriting.

MAC ADDRESSES

- As well as an IP address, every network-connected device also has a MAC address, which is like the final address on a physical envelope in our analogy.
- It is used to differentiate different machines on the same physical network so that they can exchange packets. This relates to the lowest-level "link layer" of the TCP/IP stack.
- Though MAC addresses are globally unique, they don't typically get used outside of one Ethernet network.

MAC ADDRESSES

- So, when an IP message is routed, it hops from node to node, and when it finally reaches a node which knows where the *physical* machine is, that node passes the message to the device associated with that MAC address.
- MAC stands for *Media Access Control. It is a 48-bit number, usually written* as six groups of hexadecimal digits, separated by colons—for example:

01:23:45:67:89:ab

MAC ADDRESSES

- Most devices, such as your laptop, come with the MAC address burned into their Ethernet chips. Some chips, such as the Arduino Ethernet's WizNet, don't have a hard-coded MAC address, though. Alternatively, one could provide a simple data chip which stores just the MAC address and have the WizNet chip read that.
- Yet it does come with a sticker with a MAC address printed on it. Although this might seem a bit odd, there is a good reason for it: that MAC address is reserved and therefore is guaranteed unique if you want to use it. For development purposes, you can simply choose a MAC address that is known not to exist in your network.

TCP AND UDP PORTS

- A messenger with a formal invitation for a wealthy family of the Italian Renaissance would go straight to the front entrance to deliver it. A grocer delivering a crate of the first artichokes of the season would go instead to a service entrance, where the crate could be taken quickly to the kitchen without getting in the way of the masters.
- Similarly, when you send a TCP/IP message over the Internet, you have to end it to the right port. TCP ports, unlike entrances to the Capulet house, are referred to by numbers (from 0 to 65535).

TCP AND UDP PORTS

An Example: HTTP Ports:

- If your browser requests an HTTP page, it usually sends that request to port 80. The web server is "listening" to that port and therefore replies to it. If you send an HTTP message to a different port, one of several things will happen:
- Nothing is listening to that port, and the machine replies with an "RST" packet (a control sequence resetting the TCP/IP connection) to complain about this.

TCP AND UDP PORTS

- Nothing is listening to that port, but the firewall lets the request simply hang instead of replying. The purpose of this (lack of) response is to discourage attackers from trying to find information about the machine by scanning every port.
- The message arrives at a port that is expecting something other than an HTTP message. The server reads the client's response, decides that it is garbage, and then terminates the connection (or, worse, does a nonsensical operation based on the message).

TCP AND UDP PORTS

- Ports 0–1023 are "well-known ports", and only a system process or an administrator can connect to them.
- Ports 1024–49151 are "registered", so that common applications can have a usual port number. However, most services are able to bind any port number in this range.
- The Internet Assigned Numbers Authority (IANA) is responsible for registering the numbers in these ranges. People can and do abuse them, especially in the range 1024–49151, but unless you know what you're doing, you are better off using either the correct assigned port or (for an entirely custom application) a port above 49151.

TCP AND UDP PORTS

- You see custom port numbers if a machine has more than one web server; for example, in development you might have another server, bound to port 8080:

  http://www.example.com:8080

- The secure (encrypted) HTTPS usually runs on port 443. So these two URLs are equivalent:

  https://www.example.com

  https://www.example.com:443

TCP AND UDP PORTS

Other Common Ports:

You will very likely come across the following ports regularly:

- 80 HTTP
- 8080 HTTP (for testing servers)
- 443 HTTPS
- 22 SSH (Secure Shell)
- 23 Telnet
- 25 SMTP (outbound email)
- 110 POP3 (inbound email)
- 220 IMAP (inbound email)

APPLICATION LAYER PROTOCOLS

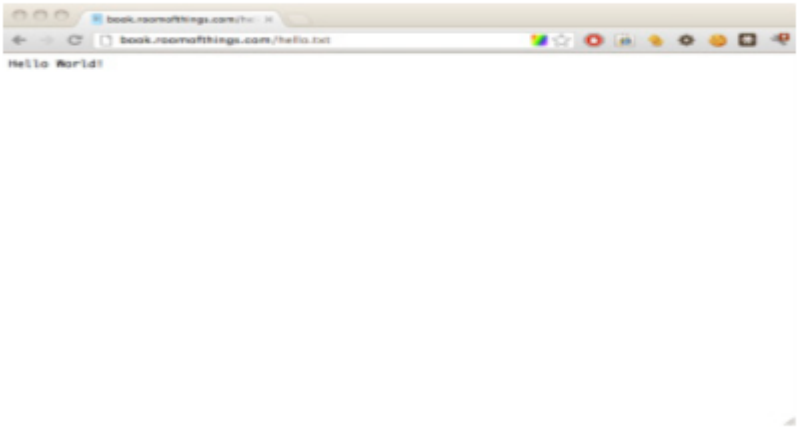
- This is the layer you are most likely to interact with while prototyping an Internet of Things. A *protocol* is a set of rules for communication between computers.
- It includes rules about how to initiate the conversation and what format the messages should be in. It determines what inputs are understood and what output is transmitted.
- It also specifies how the messages are sent and authenticated and how to handle (and maybe correct) errors caused by transmission. Bearing this definition in mind, we are ready to look in more detail at some application layer protocols, starting with HTTP.

Application Layer Protocols

HTTP:

- The Internet is much more than just "the web", but inevitably web services carried over HTTP hold a large part of our attention when looking at the Internet of Things.
- HTTP is, at its core, a simple protocol. The client requests a resource by sending a command to a URL, with some headers. We use the current version of HTTP, 1.1, in these examples. Let's try to get a simple document at http://book.roomofthings.com/hello.txt. You can see the result if you open the URL in your web browser.

Application Layer Protocols



A browser showing "Hello World!"

Application Layer Protocols

- But let's look at what the browser is actually sending to the server to do this. The basic structure of the request would look like this:

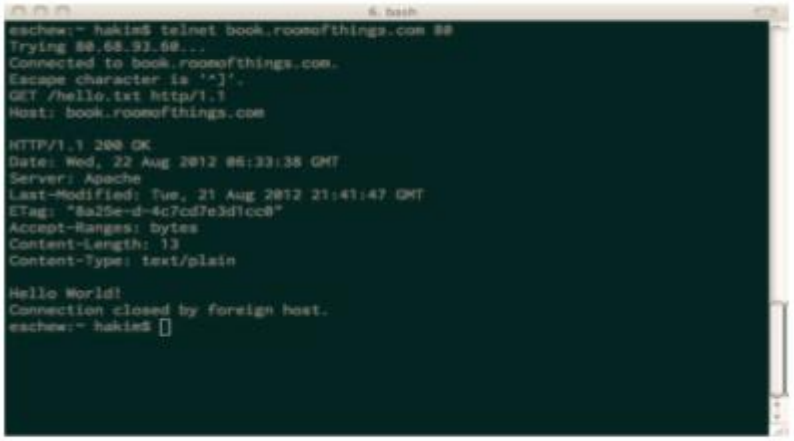
```
GET /hello.txt HTTP/1.1
Host: book.roomofthings.com
```

- We specified the GET method because we're simply getting the page. We then tell the server which resource we want (/hello.txt) and what version of the protocol we're using.

Application Layer Protocols

- We already saw what that looked like in the browser, but now let's look at what the full request/response looks like if we speak the HTTP protocol directly. (Obviously, you rarely have to do this in real life. Even if you are programming an Internet of Things device, you usually have access to code libraries that make the request, and reading of the response, easier.)

Application Layer Protocols



The request/response cycle.

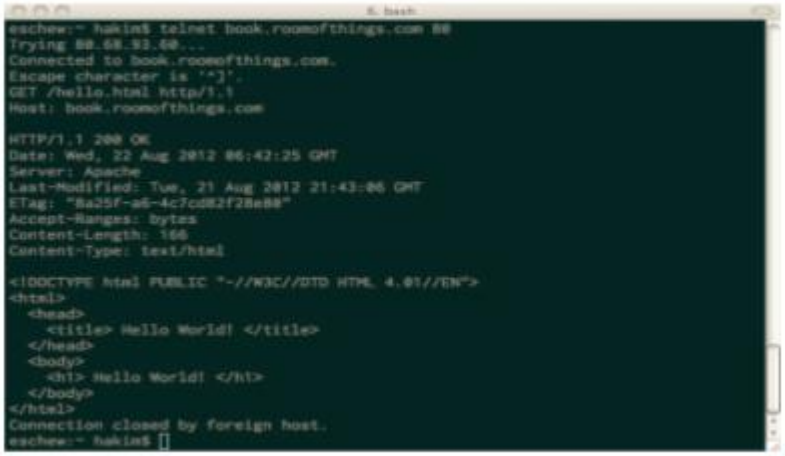
Application Layer Protocols

- Notice how we connect using the telnet command to access port 80 directly. The server replies, giving us a 200 status code (which it summarizes as "OK"; that is, the request was successful).
- It also identifies itself as an Apache server, tells us the type of content is text/plain, and returns information to help the client cache the content to make future access to the resource more efficient.
- You may be wondering where the *Hypertext* part of the protocol is. All we've had back so far is text, so shouldn't we be talking HTML to the server?

Application Layer Protocols

- Of course, HTML documents are text documents too, and they're just as easy to request.
- Notice how, for the server, replying with a text file or an HTML document is exactly the same process! The only difference is that the Content-Type is now text/html. It's up to the client to read that markup and display it appropriately.
- We look at more features of HTTP over the course of this book, but everything is based around this simple request/response cycle!

Application Layer Protocols



The request/response cycle with HTML.

Application Layer Protocols

HTTPS: Encrypted HTTP:

- We have seen how the request and response are created in a simple text format. If someone eavesdropped your connection (easy to do with tools such as Wireshark if you have access to the network at either end), that person can easily read the conversation.
- In fact, it isn't the *format* of the protocol that is the problem: even if the conversation happened in binary, an attacker could write a tool to translate the format into something readable. Rather, the problem is that the conversation isn't encrypted.

Application Layer Protocols

- The HTTPS protocol is actually just a mix-up of plain old HTTP over the Secure Socket Layer (SSL) protocol. An HTTPS server listens to a different port (usually 443) and on connection sets up a secure, encrypted connection with the client (using some fascinating mathematics and clever tricks such as the "Diffie–Hellman key exchange").
- When that's established, both sides just speak HTTP to each other as before!

APPLICATION LAYER PROTOCOLS

Published by Whitfield Diffie and Martin Hellman in 1976, Diffie–Hellman (D-H) key exchange is a way for two people to exchange cryptographic keys in public, without an eavesdropper being able to decode their subsequent conversation. This is done by each side performing mathematical calculations which are simple to do but not to undo. For example multiplying two prime numbers together is easily done, but if they are sufficiently large numbers, an attacker cannot factor the result back into the original primes given current computing power. Neither side ever sends their own secret key unencrypted, but only the result of multiplying it with a shared piece of information. By performing the calculation again on the key sent to them over the network, both parties end up with a "shared secret".

APPLICATION LAYER PROTOCOLS

- This means that a network snooper can find out only the IP address and port number of the request (because both of these are public information in the envelope of the underlying TCP message, there's no way around that). After that, all it can see is that packets of data are being sent in a request and packets are returned for the response.

YouTube - Abhay More | **Telegram** - abhay_more
607A, 6th floor, Ecstasy business park, city of joy, JSD road, mulund (W) | 8591065589/022-25600622