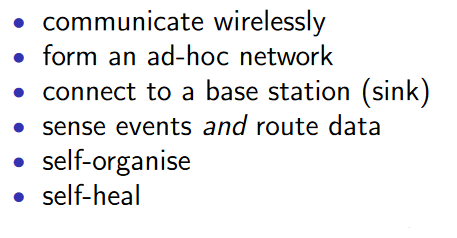
COMP3023J Wireless Sensor Networks

Chap 1 Intro

在一个WSN中的Sensor需要具有的性质：

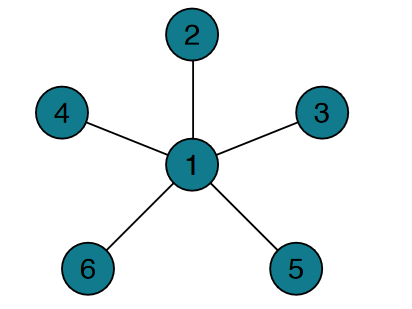


Ad-hoc network: 点对点的网络

形成的network有多种拓扑：

在拓扑中，每一个node就是一个sensor

1 Star Topology：



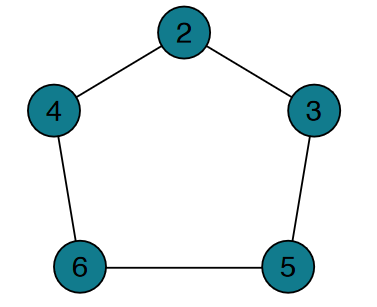
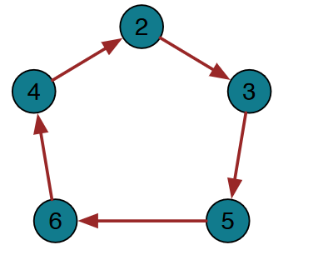
适用于小数量的nodes，不具有scalability

使用一个hub实现node之间的数据传输

在一个Star Topology中需要一个central node即hub，必须为100%reliable

一个node的failure(除了hub之外的node)并不会影响其他的node

2 Ring Topology：



每个node只和相邻的两个node相连

如果只需要整个拓扑中的一个在时间段内active则可以使用Token

一般来讲数据是单向传输的，但是需要利用双向传输解决单点崩溃的问题

3 Bus Topology：

没有central node的Star Topology

通过Broadcast传输数据，因此只有一个node可以同时传输

不需要考虑protocols，IP address等，容易实现（只需要物理层广播进行传输）

适合所有nodes都是instruction node时(指令需要传输给所有别的nodes)，适用Bus Topology

4 Tree Topology：

每个node都可以是Star Topology

可以局部management，device数量多时比较适用

一个node(Sub-root)会影响整个subtree，但对于别的subtree并没有影响

Failure of the root node destroys the whole Tree

如果Tree的Depth太深，则latency会变高

5 Fully Connected Network：

High requirements for each node(Communication, Memory and else)

当有一个新的node进入时，整个拓扑需要被重新连接

6 Mesh Network：

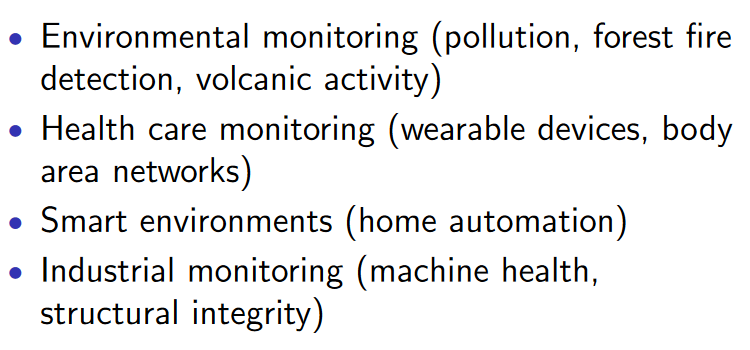
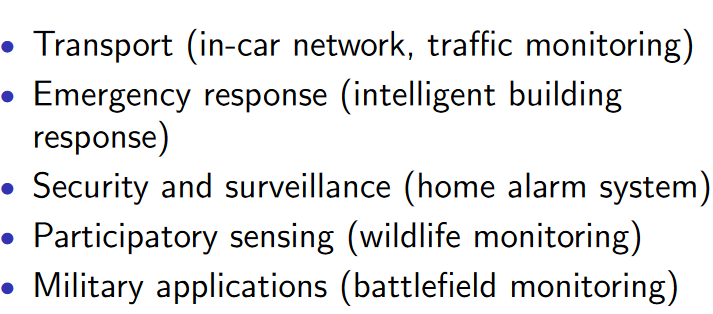
每个node会和其他Neighbor Nodes相互连接

尽量保证每个node在崩溃时都不会影响到数据的传输

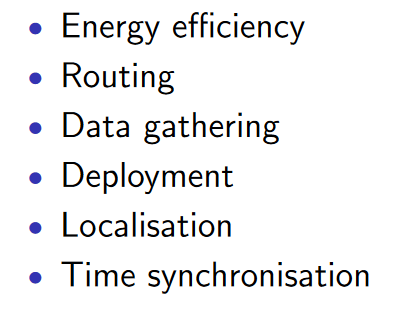
相当于简化版的Fully Connected Network

WSN：

Possible Applications of WSN：



Challenges for WSN：



Energy efficiency：energy的使用决定了硬件的寿命，因此从硬件的层面讲，使用低功耗的部件可以延长整个network的寿命

当一个network有足够的connectivity和coverage时，可以被认为这个network是alive的，因此使用redundancy of nodes可以增加整个network的lifespan

Routing：有众多需要考虑的因素：

For Routing Tables：更新问题

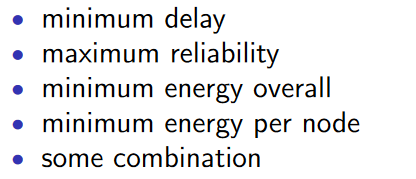
Proactive：一直更新(Constantly up-to-date)

On demand：在需要用时再更新

Hybrid：Proactive + On demand

For Routing Algorithm：如何选择路径

根据哪个方面选择route？



Data Gathering and Processing：如何决定何时发送数据？

Event Detection: 触发特定条件时处理并传输数据

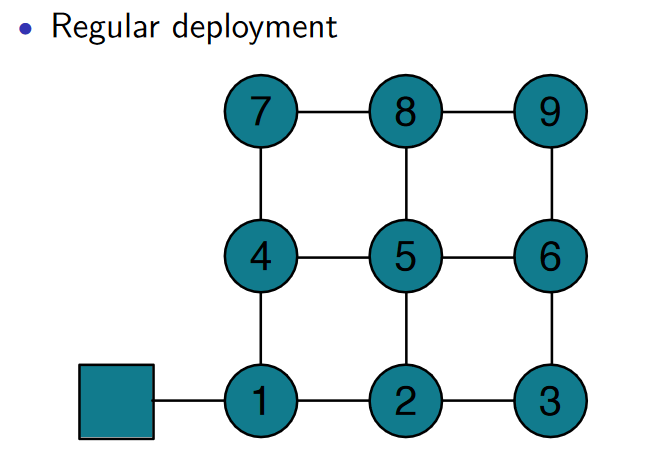
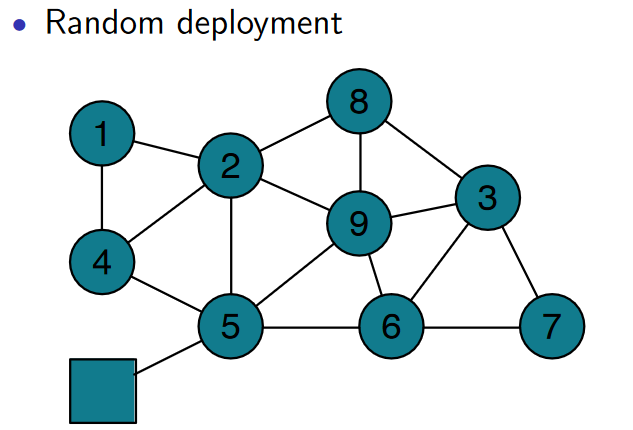
Periodic Measurement: 固定一段时间检测一次，检测的time interval可调节

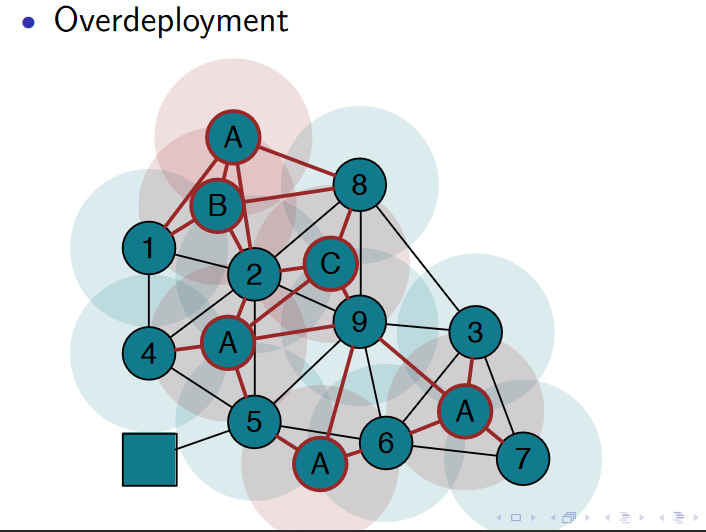
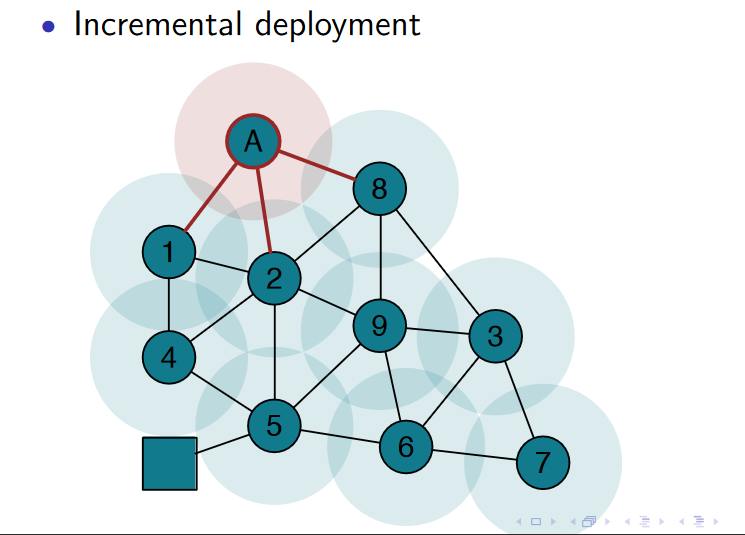
Request-Based: 请求时才会发送数据

Deployment：How is Topology Implemented?

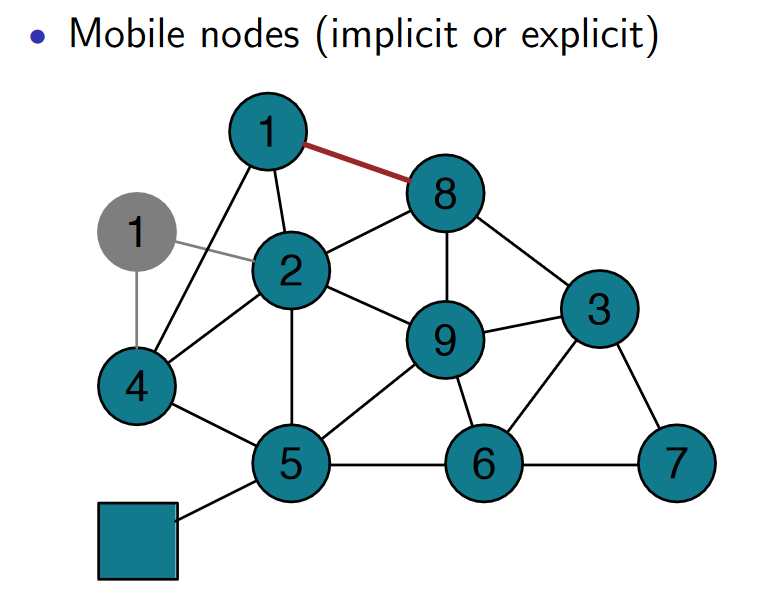
Random or Regular? Incremental deployment or Over deployment?

e.g. Mesh Network incremental---大小不定，可以增加node; over---保证robust





Mobile nodes: 在智能汽车中应用较多，nodes在移动的同时实现data gathering



Localization：定位，使用固定的node来确定位置，一般多用于室内的网络中

GPS在室内网络中的效果并不理想，因此要用到localization

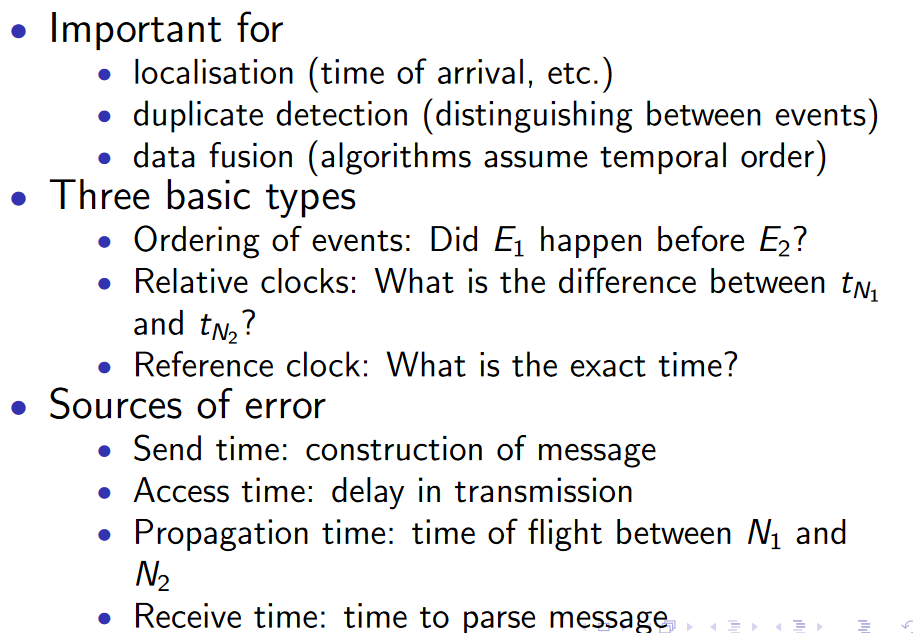
Coarse-grained: proximity-based

Fine-grained: measurement-based

通过信号接收的角度，强度，时间等等进行位置的测算

Reference-less localization：没有参考系的情况下可以测算出relative-map

Time Synchronization：



WSN Requirements：

QoS: Reliability; responsiveness---delay; priority; fairness---确保没有垄断;

Fault tolerance: 处理loss

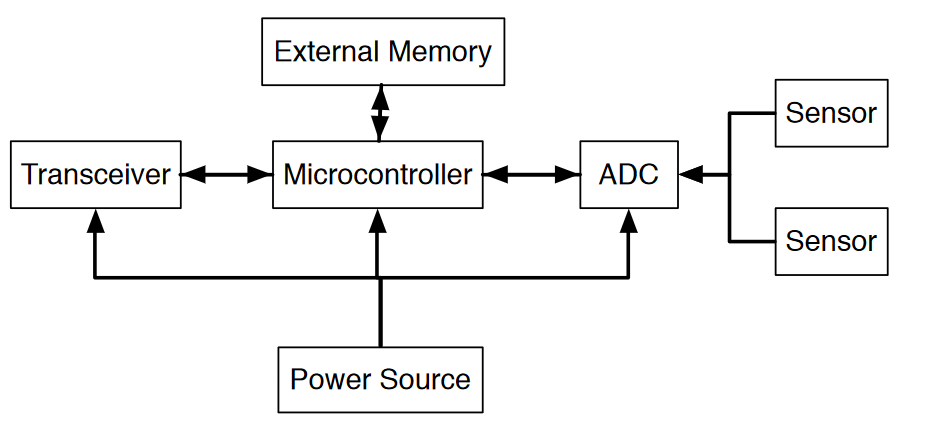
Self-configuration and adaption: 应对变换的环境

Scalability: 在不需要降低别的性能的前提下be scalable

Heterogeneity: 不同种类的sensor, different capabilities(e.g. 不同种类的通信)

Privacy and security: 敏感的数据，如何确保安全

Wireless Sensor Mote: 芯片



Chap 2 WSN-Hardware(Sensor Hardware)

Sensor Mote Design(相当于将整个系统的design拆分为每个sensor的design): different requirements---Lifetime, Coverage, Robustness, Communication, Time Synchronization, Security, Cost & Size---tradeoff and balancing

Lifetime：通过sensor management实现optimal use of power---sensors work in a group using a collaborative manner: Clustering Technique，通过集群的方式将相同功能的sensors组合在一起，并且管理使用，实现power management以延长network lifetime，这是从整体的层面上讲。从individual component的角度来讲，需要每个component实现minimal power consumption。 可以增加energy harvesting capabilities: 不只是通过reduce power consumption，也可以通过增加power的方法来实现，比如太阳能, etc.

Coverage: sensing range---增加sensing range也会导致更多的power consumption, 也会对communication有更高的要求

Robustness: sensor可能被部署在harsh condition中，因此对于每个individual sensor来说健壮性很重要，虽然可能会通过网络拓扑结构等上层方式弥补(因此整个WSN需要对individual sensor failure有一定的tolerance)，但是仍然会对整个network system带来影响。

Communication: 尽可能地减少通信所使用的power consumption和data rate需求；当能量需求或者节点数目提升时需要用transmission power control protocols进行管控，比如调整适合的power进行不同距离的传输。通过使用scalable network infrastructure可以实现更加scalable的传输，比如使用第三方提供的5G蜂窝或者Dust-Fog-Cloud infrastructure

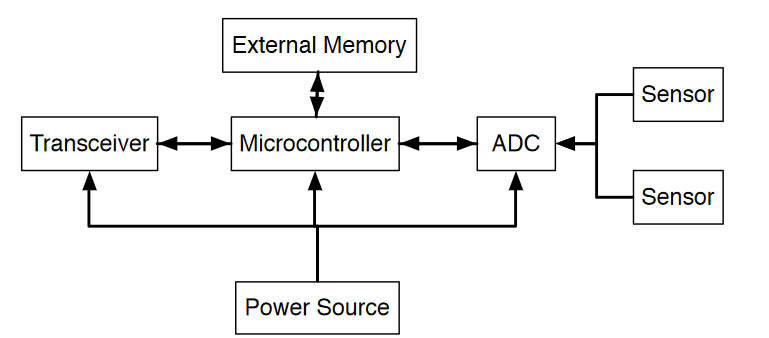
Time Synchronization: 通过sleep/wake之间的synchronization实现better efficiency，但是即使是休眠状态下仍然要保持时钟同步

Security: confidentiality, availability and integrity; 但是并不是所有的node都适合运行加密和认证算法，因此对于IoT devices来说security是一个很大的问题。可能的解决办法是通过边缘计算，依赖于infrastructure与端系统进行合作以得到安全方面的保证

Cost & Size: 取决于application---需要的sensor数量，种类，环境等

WSN节点的主要任务：computation, communication, storage and sensing

WSN节点的结构：



Controller Tasks:

Make decisions of when and how(interaction and scheduling), process data from sensors, store and read data, initiate sending data and process received data from other nodes, control and configure other components in the node.

Transceiver: State and Energy Harvesting

States can be receive / transmit / idle(准备完成但当前无task, listening) / sleep

一般来说sensor mote的transceiver只能发送或者接受，不能同时进行(MIMO可以实现同时进行接受，甚至从不同的channel)

Power source：Energy Harvesting

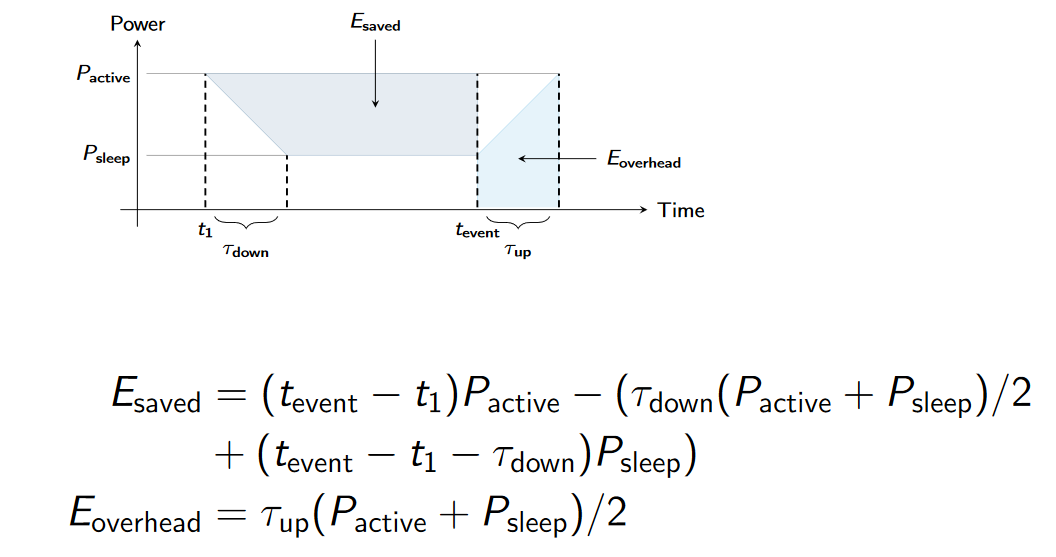
Photovoltaics(光学生电)，RF power distribution，temperature gradients，mechanical energy

Power Management:

Monitor the voltage of power supply, current consumption and running time(working & idle)

Power Management Strategies: Operational state switching & Dynamic voltage scaling

Operational state switching---取决于各个hardware component的state数量，切换state的过程是需要时间和energy的，被称为overhead，一般从low power state更高的state的transition的latency更高



Dynamic voltage scaling---根据CPU的负载进行调节，并不是所有的task都需要用满载的速度进行处理的。Power consumption取决于frequency和voltage supply：

对于周期性的task和可以预测的workload可以进行动态调节

Chap 3 WSN-protocols

LR-WPAN intro：

WSN的协议栈和TCP/IP是类似的，并没有采用OSI的7层model

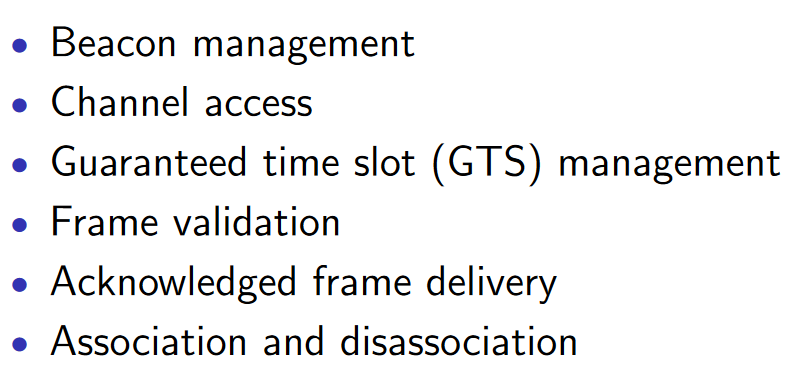
WSN-Physical Layer: 管理的是physical specification of data connection---frequency selection，Carrier frequency generation，signal detection，signal modulation，data encryption

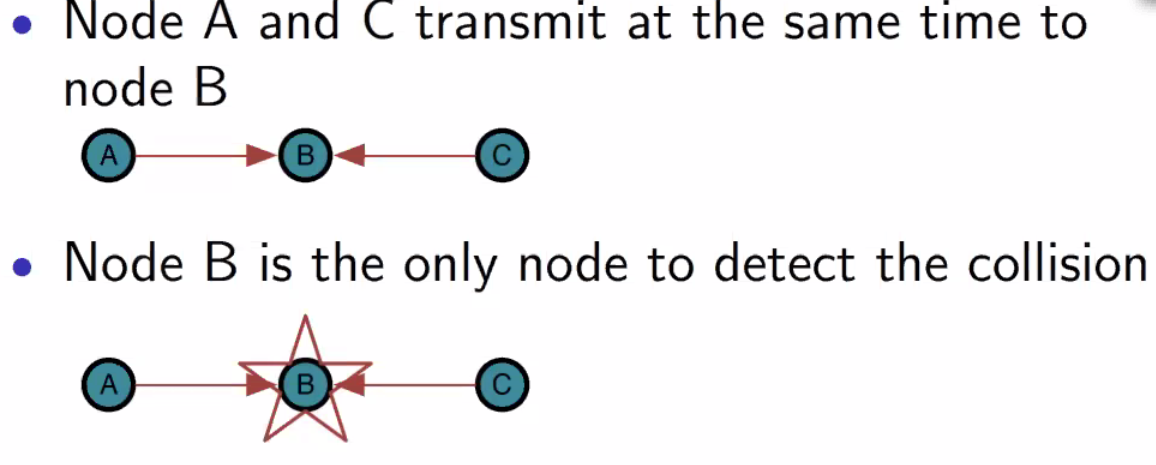
WSN-Data Link Layer：分为两个sublayer：MAC + LLC

Medium Access Control：

Addressing，Channel Access Control，Flow Control，Error Detection & Correction，Reliable Delivery

Tasks of Medium Access Layer：



Collision avoidance：通过CSMA/CA来避免，传输前先进行监听，只在channel idle时传输。如果collision发生了，则在随机的时间之后尝试重传。但是会有hidden node problem：如下图所示，AC不在同一通信范围内，但同时尝试给B传输信息，则会发生collision  


Logical Link Control：

Frame synchronization, flow control and error checking, Multiplexing

WSN-Network Layer：route

Network Address translation，message routing，packet segmentation，IP level

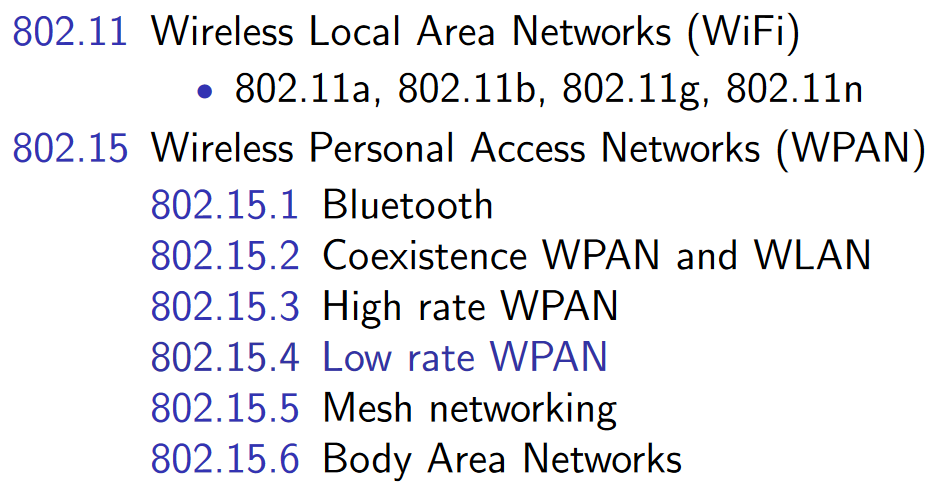
WSN-Data Transport layer：end-to-end communication

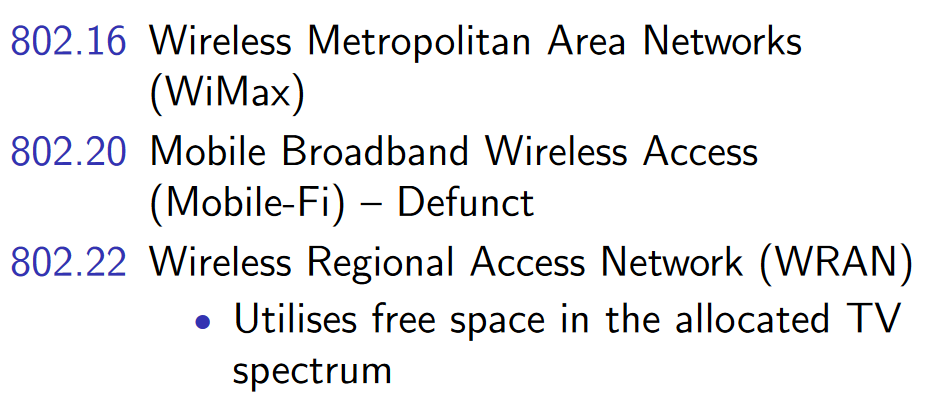
Flow control，packet segmentation，error control，TCP/UDP

WSN-Data Application layer：interface between application & network

Identification of communication partner，resource availability，Synchronization，Authentication and encryption

IEEE standard：





802.15.4---Low-rate wireless personal area networks(LR-WPANs): low rate & low power

一般在几十到几百kbps的bit rate。适用于real-time application，因为guaranteed time slots(GTS)和collision avoidance(CSMA/CA)。并且通过symmetric encryption实现了secure communications

802.15.4的device types：

FFD---full function devices---完全的实现了整个网络栈，可以初始化网络的行程，也可以作为网络的coordinator(相当于管理者，管控别的网络设备)或者router。

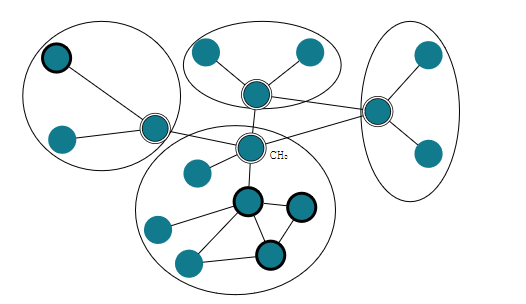
RFD---reduced function devices---并没有完全实现网络栈，只能和FFD进行通信。

Topologies：Star---中间的点是PAN coordinator(必须是FFD)，每个设备都要和coordinator associate，所有的data通过coordinator进行routing

p2p topology：mesh---所有的不是edge的node必须是FFD，不需要route的也可以是RFD。

Cluster Tree Topology: mesh of meshes

每一个subtree算是一个cluster，每个cluster的cluster head必须是FFD，cluster之间通过cluster head进行通信



GTS：Guaranteed time slots

适用于低延迟或者对带宽有要求的应用，可以由coordinator进行分配，最多可以在active period分配7个GTS

ZigBee：built on 802.15.4，定义了MAC layer和Physical layer之上的layer

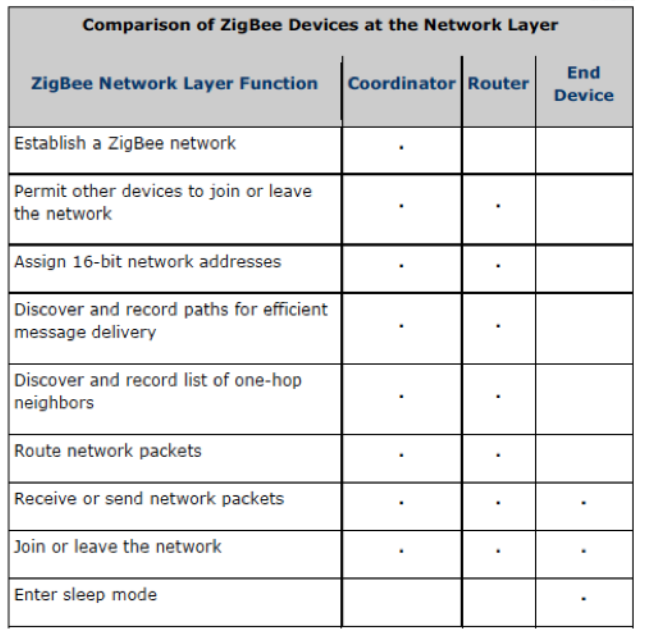
定义了application profiles：message type & format, etc.

Network layer: 提供对MAC layer的interface，通过AODV (Ad hoc On-Demand Distance Vector Routing)进行routing

Application layer: Device management，Security management，Service discovery，Application profiles，Application objects (APOs)

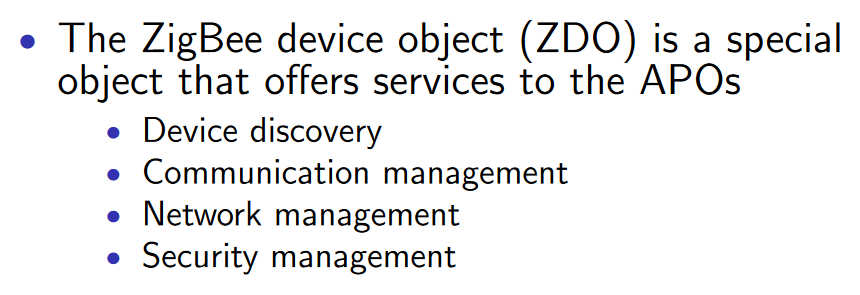
不同的无线传输协议的特性：Wifi的距离最远，ZigBee消耗最低，传输速率Wifi最高ZigBee最低，ZigBee加入所需的时间最短

ZigBee Topologies：star，tree，mesh or hybrid。同样有FFD和RFD的区别：需要注意的是只有end devices可以进入休眠模式，router和coordinator不能进入休眠



ZigBee Address Allocation：ZigBee有独特的scheme，根据每个parent node的children和router children总数，Maximum depth和当前正在计算的层的depth可以得到address数量

APOs：Application Objects，由ZigBee提供给application的object。可以用来管理hardware，可以实分布式的，也可以在一个node中有很多APO。APO之间是互相知道的(i.e. Addressable)，APO之间的message是由ZigBee定义好的。APO依赖于ZDO

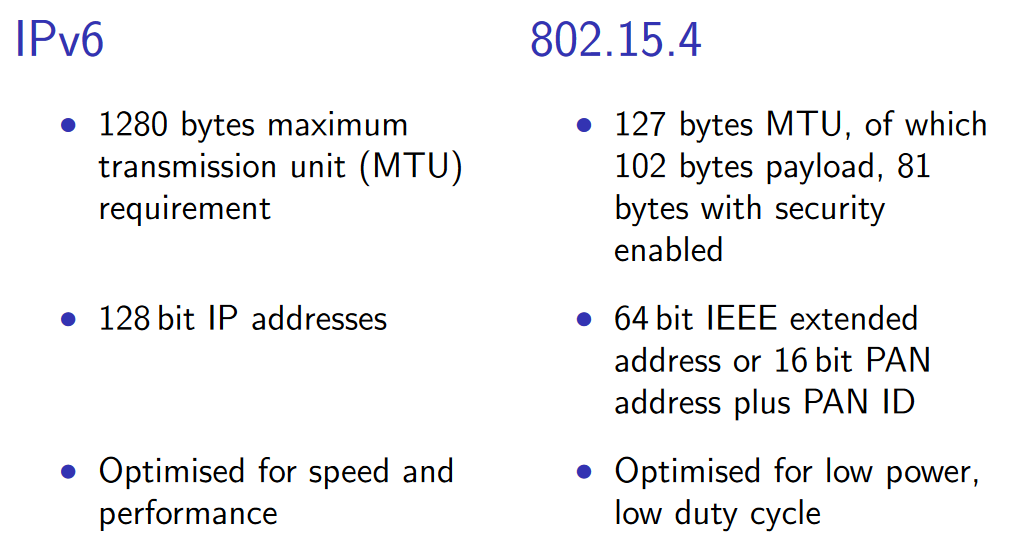


ZigBee security：通过symmetric key加密，使用的是128 bits key的AES，通过一个trusted center分布key。这个trusted center的地址和master key一般是pre-shared，ZDO负责管理security policies和configuration

6LoWPAN：IPv6 over Low power Wireless Personal Area Networks

使得IPV6的包可以通过802.15.4的网络进行传输，这样就可以将WSN接入互联网，使用IPV6地址

IPV6和802.15.4的区别：



在IPV6和6LoWPAN的边界router处要进行一些变换：压缩IPV6包头，将payload拆成更小的块，压缩UDP包头。需要注意防火墙，包的大小，format，注意是IPV4还是IPV6，并且由于接入互联网了还需要注意数据的安全问题

Chap 4 Routing

Routing protocols: 网络层的一部分，决定route of packet & hops

大多数别的wireless protocols对于WSN来说并不适用，因为WSN有physical constraints

在WSN中routing protocols应当更在意power consumption而不是QoS；节点的数量会非常多---global address不一定可行；不同的node可能重复地独立检测到同一事件，需要protocol来减少redundancy；network can be lossy---需要protocol增加一些redundancy来保证resilience(和前面的一点算是trade-off)

WSN constraints：

通过变更topology来应对可能的node failure

不同的applications有不同的需求---power consumption/throughput---如何选择protocol

每一个protocol不一定只能根据一种方式分类，可以同时是QoS based & Hierarchical

Discovery based routing:

Proactive：提前决定routes，当topology发生变化时更新routing table

Reactive：当需要的时候再去discover routes，发送时需要等待discovery converge

Hybrid：proactive和reactive的混合

Topology based routing:

Flat: 认为所有的node都是一样的

Hierarchical: 不同的node有不同的role

Location based: 需要注意不同的physical location

Operation based routing: 根据不同的操作进行分类

Query based: destination node在网络中传输query，有需要的data的node进行reply

Multi-path based: 用多条路径来增加resilience

QoS based: 对多个QoS parameter进行balance(delay, BW, energy, etc.)

Coherent & non-coherent routing: coherent routing中node只会对数据进行简单的处理，比如压缩，然后发送并集中处理(消耗的energy会少，但是增加了traffic)；而non-coherent routing中会在本地先进行处理后在进行传输(消耗更多的energy但减少traffic)

具体的protocols：

Flat routing protocols：

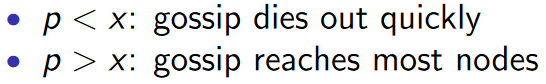
•Flooding:

Node向所有的neighbors进行广播，别的节点在收到后重复广播，直到网络中的所有节点都收到一份copy

产生大量重复的message，并且如果一个node收到同一份data超过一次则会发生implosion

•Gossiping: 改进版的flooding

Source还是广播给所有neighbor，但是其他node会根据一个概率p进行forward，并且只有没收到data的节点才会进行forward。传输的效果取决于概率p，如果p太小则可能永远达不到destination，因此有一个critical value x：

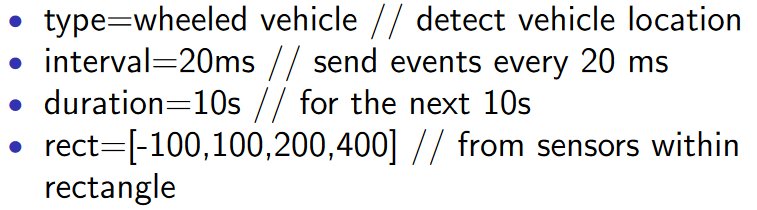


•SPIN: 增加了negotiation的flooding

在发送data之前先发送advertising(ADV)询问，需要的node会回复一个request to receive(REQ)，之后发送data。这个过程持续到destination接收到data为止，已经接收到data的node不会进行回复

•Diffusion: directed diffusion---通过传播兴趣建立梯度场

每个节点都有一个list，其中是很多attribute-value pair，比如：



发出请求的节点被称为sink node，首先通过flooding向网络中interest message，随后每个节点会缓存interest并且通过局部数据融合计算出gradient。如果有节点match interest，则会通过gradient发送data，梯度此时会converge并行成一条通路用于传输(sink会加固一条特定的路径/一个neighbor进行传输)

Gradient可以理解为记录DFS(Depth-first-search)的一个数值，也会用一些其他的参数

Sink在第一次发送interest的时候会将interval设置的比较长，随后再逐渐降低

在确定一条path之后，sink会不断地发送周期性的interest，用来确定当前的路径仍然可用

DD路由协议多采用多路径，健壮性较好;节点只需与邻居节点进行数据通信，从而避免保存全网的信息;节点不需要维护网络的拓扑结构，数据的发送是基于需求的，这样就节省了部分能量。DD路由协议的不足是建立梯度时花销大，多Sink的网络一般不建议使用;时间同步技术在数据融合中的利用，增加了开销

•AODV: Ad hoc On Demand Distance Vector Routing

Source会发送一个route request(RREQ)，通过flooding传送给整个network，每个节点会将回到source的路径缓存

Destination会根据路径发送一个route response(RREP)。如果一个节点已经缓存了前往destination的路径，则会直接发送一个RREP，而不需要继续向其他节点继续传输RREQ

•DSR: Dynamic Source Routing

每个Route Request package中都含有整条路经的信息

每个node会将自己的address加进route中，到destination时会收到完整的路径信息

Route Reply中会包含最短路径的信息，source会将最短的路径缓存并通过这条路径发送后续的packet

Hierarchical Routing Protocols：

LEACH：Low-Energy Adaptive Clustering Hierarchy

一部分节点会自动地成为cluster heads

Cluster heads会向整个网络advertise自己的状态，并增加传输功率(boosting Tx power)

Non-cluster head节点会根据信号强度加入较近的cluster，cluster内的节点通过TDMA向cluster head传输data。Cluster head会将集成的数据(减少transmission)直接发送给基站

Cluster head会被周期性的重新选举来平均energy use

这个算法是基于所有的node都在基站的通信范围之内的情况建立的

TEEN：Threshold-sensitive energy efficient sensor network

Assume a cluster tree topology

为每个传感器接收到的数据设定两个threshold---HT (Hard Threshold) & ST (Soft Threshold)

当传感器接收到的值超过了HT，就会将这个值传输给parent

随后每个node在周期内只会transmit一次，或者传感值同时大于HT和ST

在周期过后，cluster head会rotate，并且会设立新的Threshold

APTEEN：Adaptive Periodic TEEN

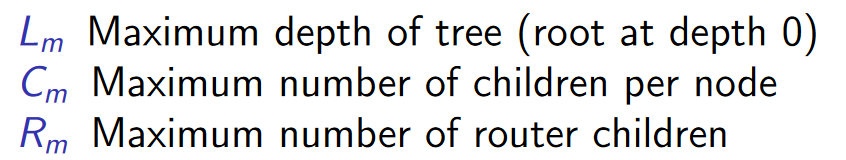
加入了counter和TDMA：如果到了一定时间某个节点还没有传输数据，则强制其通过TDMA发送当前读到的数据

APTEEN从energy dissipation和network lifetime上讲不如TEEN，但是更加flexible

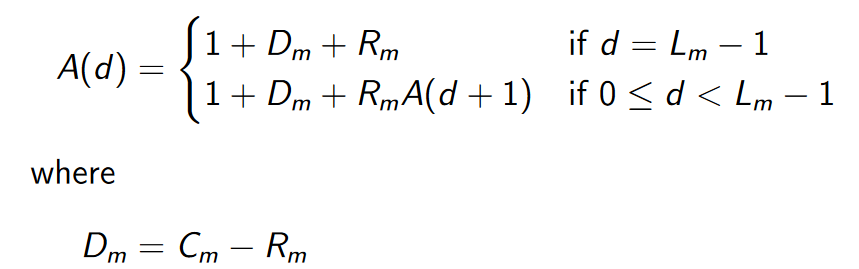
TEEN和APTEEN都要比LEACH要好

CSKIP：ZigBee tree routing

通过提前设定好的address分布式地管理网络

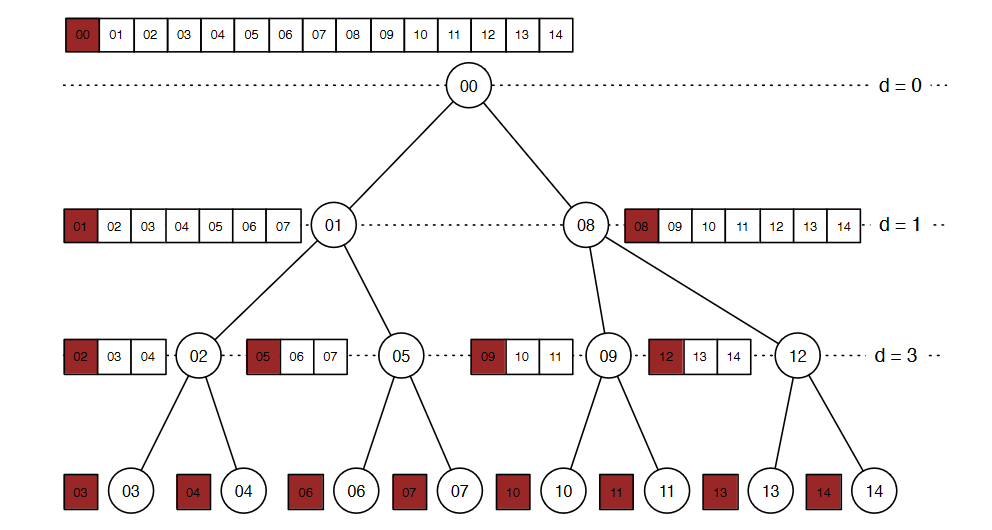


可以通过这三个参数计算出深度为d的节点的可用地址A(d)：



可以看出来每层中maximum end device的数量就是Cm-Rm，也就是Dm就是每层中最大的端系统数量。注意depth是不算external node的。上面式子算出来的结果就是每一层的节点可以使用的address的数量，被记为CSKIP：

比如在这个例子中Rm=2，Cm=2，Lm=3：



因为套用公式可以知道A(2)= 3, A(1)=7, A(0)=15

A(0)的值就相当于是整个tree的所有可用的地址数

CSKIP网络的formation：

首先由coordinator选择参数，coordinator的地址默认为0。节点可以请求加入某个parent的子树并发送请求JREQ，parent会发送一个confirm JCONF并返回三个parameters和给定的address

Location based protocols：

LEACH-C: LEACH centralized

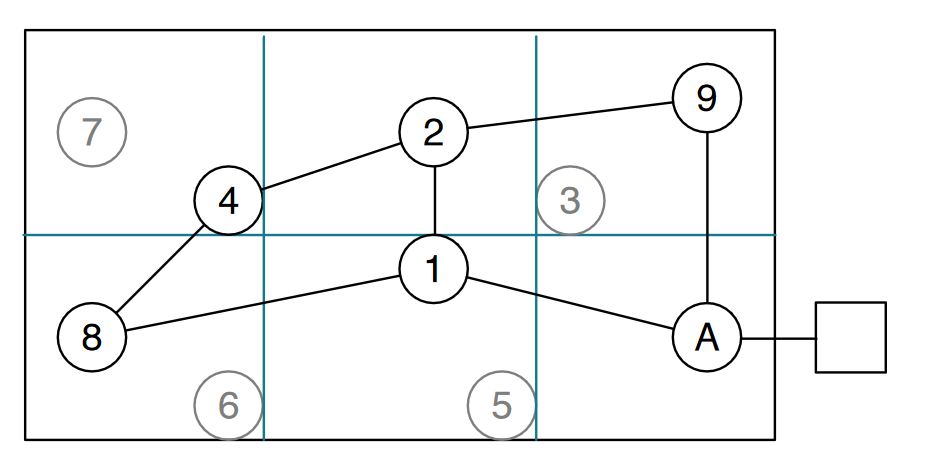
在setup过程中，所有节点会将当前的location和energy level发送给sink，sink会丢弃掉所有在平均energy level之下的node。Sink会根据最小距离平方之和选择若干个cluster heads，随后的传输等同于LEACH

K-Mean algorithm：K-Means clustering是一种unsupervised learning，适用于有unlabeled data的情况下。这个算法的目的是通过组的数量K(变量)给数据分组，通过不断迭代一个算法：将一组数据通过某个指定的feature归类给K组中的一组

具体的方法是在空间中设置一个初始的centroid，根据和中心点的距离将所有点放进最近的cluster中。不断重复前面两步，直到所有的centroid不再发生变化。

GAF：Geographic Adaptive Fidelity

将区域分成几个虚拟的矩形块(固定)，一个区域中的节点在network formation的过程中发现处在同一区域中的其他节点。假设通过同一区域内的任意节点的routing都是相同的，因此在一个区域内除了一个节点之外的其他节点可以进入休眠模式，剩下的节点负责区域之间的routing：需要在维持区域间联通的前提条件下进行



GEAR：Geographical and Energy-Aware Routing

尝试给一个区域内的所有节点都传送数据

在发送时从neighbor中选择离目标较近的消耗最低的next hop---f(E,s)

如果没有neighbor更近的话network holes就会发生

在network holes附近寻找routes并更新cost

在目标区内使用flooding或者RGF

RGF：Recursive geographic forwarding

在到达目标区域(Target region)后激活，将目标区域分为四个小区域并向每个小区域forward packets。重复进行此步骤直到每个node都是一个独立的region并都有了一份copy为止

Chap 5 Data Processing

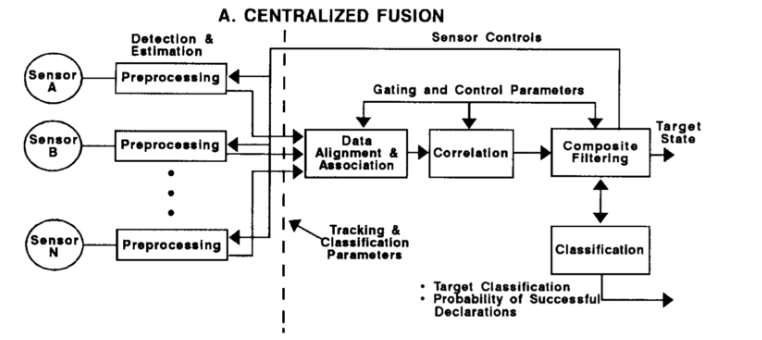
Data fusion, analysis and visualization: 之前的所有讲的都是如何获取以及传输数据，现在在已经获取到了需要的数据的情况下需要考虑如何处理这些数据，也就是data processing：三个步骤---data fusion(相当于预处理)➡data analysis➡data visualization

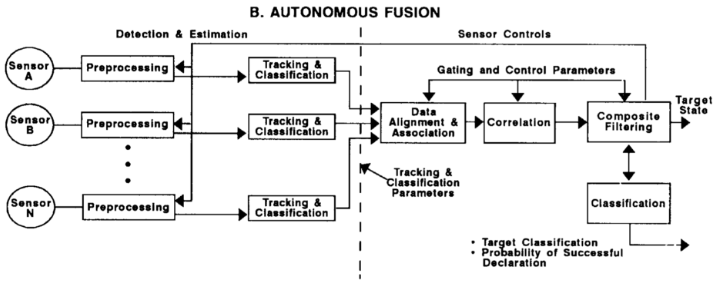
WSN是非常庞大的分布式系统，因此会收到来自于各种地方的数据，需要通过data fusion将这些数据整合在一起。这个过程可以在edge或者是central location进行，比如clustering。目的在于获取高质量的数据并且节省储存空间

Analysis同样也可以在end device或者cloud上进行

数据可视化也分为两部分：一部分是在分析之前进行可视化，目的是为了更直观地看到数据；另一部分是在分析之后可视化，目的是为了让客户更容易理解分析的结果

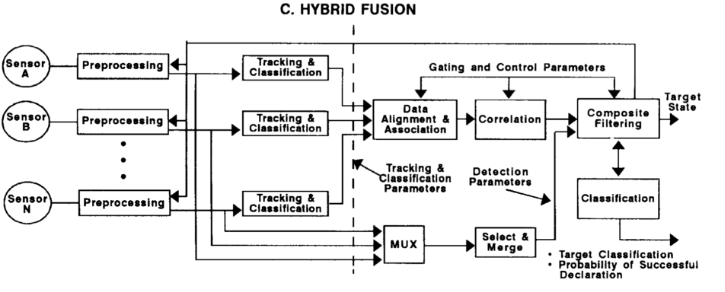
Data Fusion：主要分为两种结构：分布式的和集中式的





在分布式data fusion中，CPU处理的不是raw data，而是经过模型处理后的结果，比如说上图中tracking & classification可能就是某种ML模型

也可以有hybrid fusion：在进行集中处理的同时对raw data使用multiplexer有选择性地将一些原生数据传回集中处理的云



Data Storage：

数据既可以在edge储存，也可以在cloud中储存，一般使用多组服务器进行存储，可以分别存储不同的部分也可以作为RAID使用。由于安全和隐私的问题，现在很多数据都是在edge进行储存的

有一种储存方式是将数据分为两部分储存，一部分在edge一部分在cloud，在edge的一部分相当于key，需要通过这个key才能得到在云中储存的数据。如果只有云中的或者边缘的一部分数据的话没有任何意义

Fog layer：在edge和cloud中间的一层，有一些smart router，可以进行smart network computation/routing；也可以对数据进行data anonymize，这样在云上计算的时候就不知道数据的来源，可以解决一部分的privacy issues

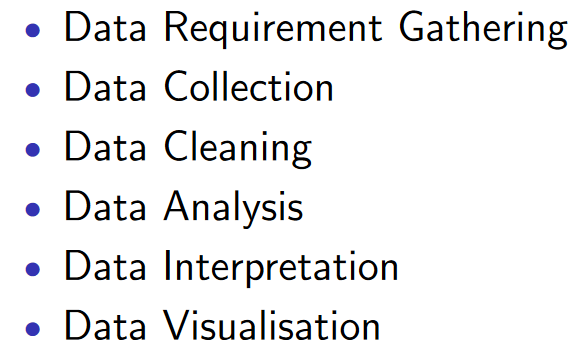
Database Overlap：不同的database strategies，主要分为下面三种

Disjoint Databases：没有common data和overlap的部分

Identical Databases：两个数据库储存完全相同的数据

Overlapping Databases：有部分overlap

General data processing steps：



Interpretation更像是对于现象的一种解释，而analysis是通过分析得到某种现象

Big Data：Considering the quantities of data being stored, and the frequency of that data, large data becomes Big Data. 和传统的large data的主要区别在于数据类型，big data中的数据可能是各种类型的，比如音频或者视频等，而不是像传统large data中的数据可以直接提供信息。

Data Fusion：

首先需要了解sensor data的四个feature：

1. Streaming nature：

Relationship--- Traditionally, data is input via a set of rules with relationships between fields，while streaming data is whatever comes in when it comes in(我对这段话的理解是传统的数据是要收到field之间的约束的，而streaming data如何传入就如何存在)；Volume---随着时间的累加，data的数量会变得非常庞大；Time Variance---数据会随着时间变化(event driving)，根据路由协议的不同会呈现不均匀的分布，数据库必须随时可以接受；Time Ordering---和传统的数据不一样，sensor data在时间上的顺序非常重要

1. high tempo-spatial correlations：

高时空相关性，即时间上和空间上相近的data相关性会较高(因为传感器的部署需要一定的密度)

在一个时间点所读取的数据对于下一时间和附近的节点的数据是indicative的，因此可以利用这点来估计丢失或者不完整的数据，检测异常数据，提升sensor data的quality，减少data transmission

1. generalization of redundant data(可以是时间上或者空间上的冗余)：

和时空相关性是类似的概念，冗余数据在sensor data中的存在非常常见

1. sensor data contains noise：

sensor设计的初衷是低功耗+低价格，所以有时候结果并不会特别准确，环境也会导致一定程度上的interference。Outlier是可以通过判断避免的，但是noise不行

Sensor Data Fusion的三个步骤：

Pre-processing中的各个步骤：

1-data cleaning：

方法包括Bayesian Theory, Neural Networks, Wavelets, Kalman Filter and weighted moving averages，但是很多都过于复杂，一般常用的有Kalman Filter和weighted moving averages：

Kalman Filter：对计算需求很小，simple form，工作原理是maximize likelihood

适用的情景：当想得到的变量无法被直接测量时，可以通过测量别的数据来进行估算 & 可以直接进行测量但是测量值对noise非常敏感

Algorithm：在知道任何具体数据之前，先用一种分布模型预估，然后采集数据并且整理，用于更新这个distribution，不断重复更新

Moving Average：设置一个窗口，根据窗口内的大小求均值记为这个点的值

Kernel Average：加权的Moving Average，通常来说使用的是Gaussian Kernel，概率就是权重

2-outlier detection：和之前提到的一样，会规定一个normal的区间，超出区间即为outlier

3-missing value recovery：传统的方法是请求重传，但是会增加功耗和latency，因此可以使用相关程度较高的sensor data来进行估算

4-data reduction(由于redundancy)：如果不去掉这些冗余的数据，会增加能耗和traffic flow并减缓knowledge discovery process；这个过程也可以帮助减少noise和outlier

通过sensor data aggregation实现：最简单的方法就是使用一个predefined threshold---通过90%置信区间，一段time window内的平均值，一定范围内的最大最小值以及更复杂的随机/加权采样来实现

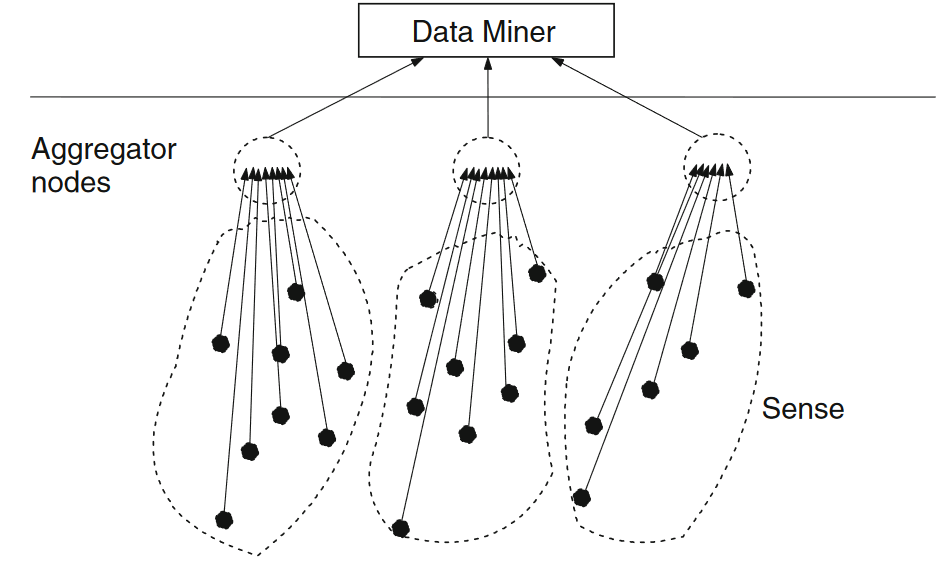
5-data prediction……

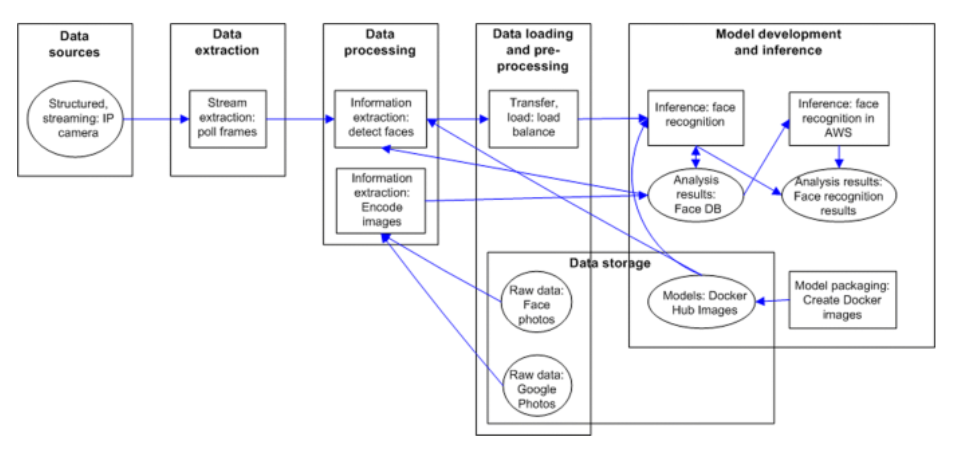
Data Mining-pre：从数据中提取pattern

传统的数据挖掘技术包括decision trees，rule-based classifiers，ANN等，但是一般应用在集中式的数据中心

对于sensor data mining---希望实现in-network data mining，所以需要用到层级结构：

云上的data miner针对aggregated data，aggregator用来收集和处理





Post-Processing：

在sensor data mining之后进行，根据之前的得到的model来进行evaluation和data visualization等等

Event Detection：Identify(试别event) & Characterize(extract details of event)

Threshold-based Event Detection: 设置阀值进行检测，这个threshold一般由过去的数据统计得出

同理可以有Multi-Threshold Detection，也就是使用多个阀值

Tempo-Spatial Pattern Based Event Detection：

假设某个事件会导致多个node的读取值呈现某种tempo-spatial pattern

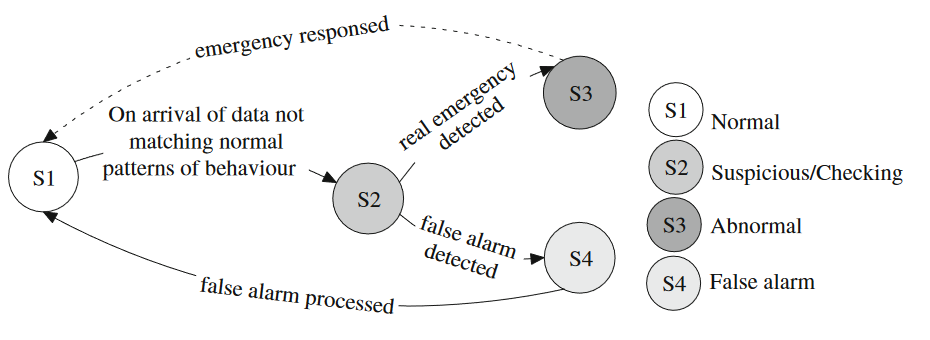
Spatial Pattern是需要提前定义的，并且很难定义所有的event，一般用contour map来实现：能够读取到相似的轮廓(包括event pattern的一部分)时，就认为event发生。Spatial Contour Map就是读数的直观体现

对于Time series data一般先转换为图像之后再检测pattern，转换之后可能就不是直接使用原先的读数进行分析了，而是使用一些ML model进行分析，pattern可能对人是不可见的

Tempo-Spatial Pattern Based Event Detection的优点是结合了时空相关性，提升了准确度；缺点是比较复杂，因为要用到整个网络的data，并且定义合适的pattern很难

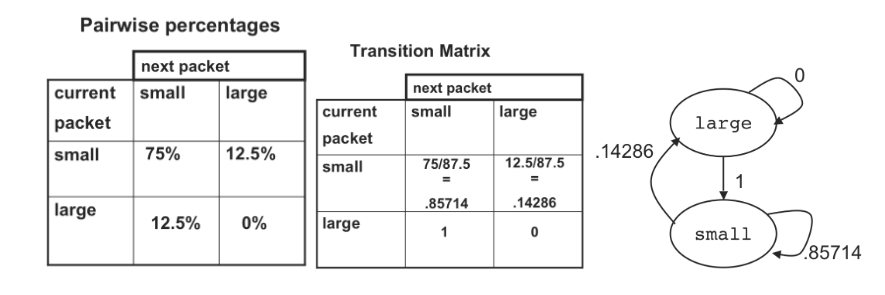
Generic Sensor State Model：

无论什么情况下sensor都可以被归类为四种状态之一：normal, suspicious, abnormal, and false alarm---四种状态之间的转换如下图所示：



First Order Markov Model：用于modeling packets on the network

比如这样一个例子：网络中所有的包被分为大包和小包，每8个包里一个是大包，大包之后一定会跟着一个小包，所以可以画出下面的图：transition matrix就是计算的条件概率



First Order Markov Model的性质：

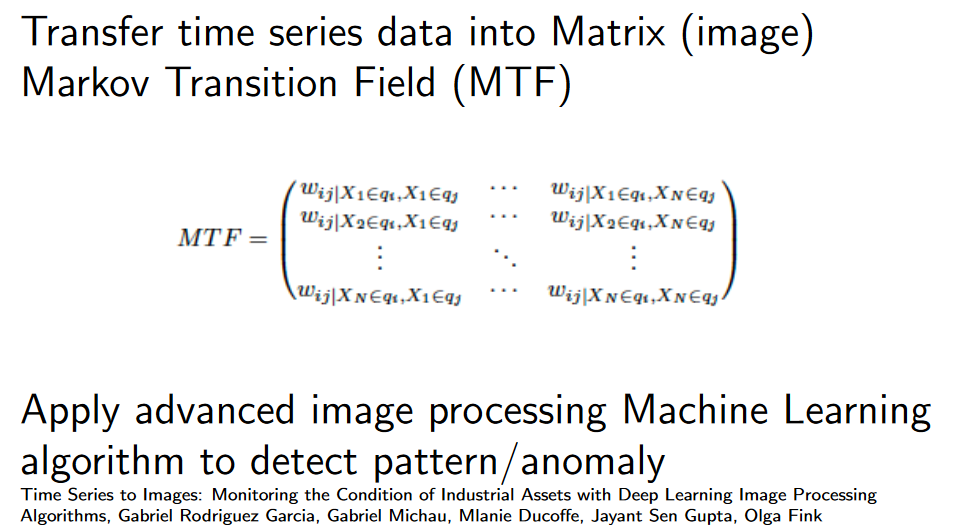
Finite---有限的状态数

Memory-less---下一个state的状态只由现在的state决定，与过去的无关

Absorbing state---在主对角线上概率为1的state记为absorbing state，进入这个state就无法离开了

Time Independent---transition matrix并不会随着时间变化

可以将matrix当成一个image来处理，就可以使用image processing ML algorithm



Data Analysis：多数情况下，Data Analysis指的主要就是Data Mining

Data Mining：

主要任务：

Anomaly detection---异常数据值的检测

Association Rule Learning---寻找variable之间的relationship

Clustering---在不通过已知的structure的前提下在数据中寻找有共同点的group/structure

Classification---generalizing known structure to apply to new data（比如通过模型训练之后，可以对将要到来的数据进行预测，将其标注为可能性最大的label）

Regression---寻找error rate最低的方式来model data

Summarization---对数据集进行准确的表示，包括visualization和report generation

Input of Data Mining：

Concept---The thing to be learned, i.e. the goal to achieve

Instances---现有的information

Attributes---instances的不同方面的measurement

Output of Data Mining：

可以是多种形式的，比如table，线性模型(attributes的组合及权重)，Trees，Rules(指rules for prediction，一般是一组测试和基于测试结果可以针对被测试的instance得出的结论)

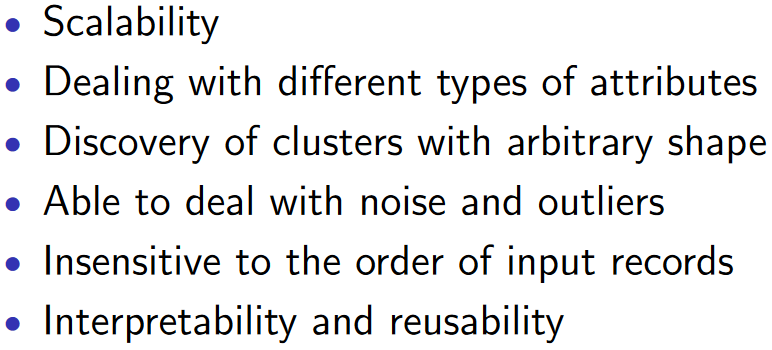
Clustering：寻找pattern的一种方法，可以寻找某方面相似的data之间的pattern；可以在pre-processing中作为分类使用，也可以在exploratory data analysis，pattern recognition，summarization中使用

Clustering is the unsupervised mining function for finding natural groupings in data

Clustering被用来执行data的segmentation，这些segmentation并不是之前被定义好的

Clustering的目标是分成多个cluster，inter-cluster similarity很低但intra-cluster similarity很高，并且能够检测hidden patterns

Usability of Clustering：

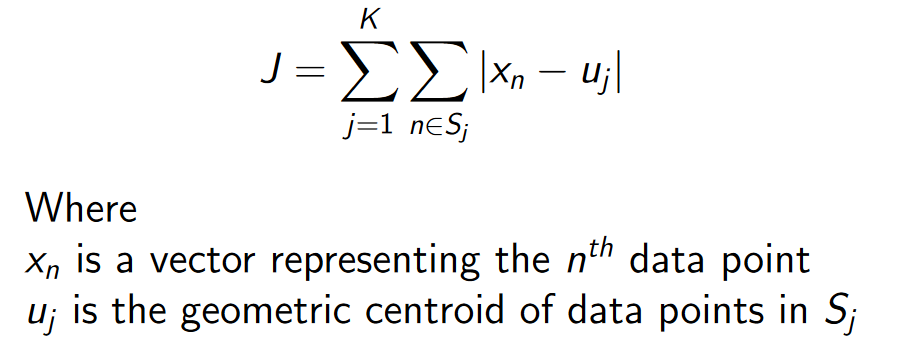


Clustering Algorithms：可以分为很多类

Partitioning/Centroid-Based---创建partition并根据某些criteria进行评估，这其中就包括K-Mean algorithm：K-Means clustering是一种unsupervised learning，适用于有unlabeled data的情况下。这个算法的目的是通过组的数量K(变量)给数据分组，通过不断迭代一个算法：将一组数据通过某个指定的feature归类给K组中的一组

具体的方法是在空间中设置一个初始的centroid，根据和中心点的距离将所有点放进最近的cluster中。不断重复前面两步，直到所有的centroid不再发生变化。

总的来说就是使得下面这个式子的值最小



Centroid-based的优缺点：

优点是可控，由于是不断迭代的算法，可以随时停止；比较容易实现

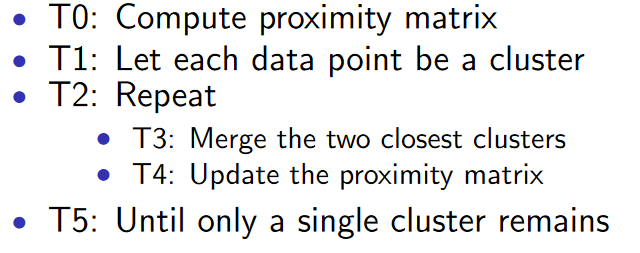
缺点是partition number需要提前知道；计算上是NP-Hard问题；比较倾向于寻找空间相似度高的集群；由于是heuristic algorithm，所以不一定会收敛，最终结果可能取决于最初的cluster的选择

Hierarchy---基础思想是通过层级结构分解数据集

距离较近一般相似度会比较高，通过连接Object来形成cluster，因此在这种方式下cluster可以被定义为连接一个点的最大距离，根据不同的距离可以形成不同的cluster---这里的距离没有明确的定义，可以是Euclidean distance，也可以是Manhattan distance

主要分为两种方式：agglomerative和divisive

Agglomerative approach：被用的比较多



Divisive approach：

T0---把整个dataset分成两个cluster：首先找出最不同的一个点，然后将这个点分离成一个单独的cluster，然后在原来的cluster中的点可以选择留在原来的集群中或者加入这个新的cluster，直到所有点完成选择

T1---选择一个最大的cluster(根据数量或者直径决定)，重复T0

T2---重复T1

Hierarchical method的优缺点：

优点是不需要预先知道cluster的数量；The clusters may represent meaningful taxonomies，也就是有分类学意义；实现方式比较straight-forward；在任何情况下都是时间复杂度为O(N3)空间复杂度为O(N2)---可以接受的范围，不需要很多的pre-processing

缺点是对于noise和outlier很敏感；很难处理不同大小的cluster和convex shapes；Tends to break large clusters(容易丢失一部分信息)；整合或者分离两个cluster的过程是不可逆的

Density-Based---基于density和connectivity functions

在Density-Based方式中，cluster被定义为point density高于周围环境的区域，在很空旷的区域内的点被认为是noise或者outlier

需要某种density drop off来决定集群的边缘，也就是设置threshold

DBSCAN Algorithm：

一些术语：Eps---给定的半径；Density---半径为Eps之内的点的数量；Core Point---在半径Eps之内点的数量多于一定的值(MinPts)的点；Border Point---内侧的点是Core Point的点，点的数量略少于MinPts；Noise Point---不是Border Point和Core Point的点

Density-Based method的优缺点：

优点是不需要提前设定cluster的数量；can find clusters of any shape；has resilience to noise

缺点是假设clusters有相近的density；需要通过density drop off来区分cluster和outlier

Grid-Based---Based on multiple level granularity

Model-Based---对每个cluster都有一个hypothesized model，目标是寻找最优的model

在完成data mining之后还有一步result validation：即在更大的数据集中检测结果中的pattern确实存在，用来检测的数据集就是test dataset，和用来寻找pattern的数据集会有一定差异。Result validation被用来调整算法，解决overfitting problems, etc.

准确率可以通过identifications/verifications来得出

如果Result validation的结果和test set并不匹配，那么就要重新进行不同的pre-processing和data mining过程

在data analysis中，如何提取信息通常取决于想要获取的信息及目的：

Descriptive information---不寻找新的信息，一般是总结dataset：并不是基于概率论的分析和统计，一般的测量包括central tendency (mean, mode, median)和variability (standard deviation, min/max)。output is typically simple in nature，一般用来做决定

Exploratory data---目的是为了寻找新的information和pattern，在数据集很大的时候可能在data collection刚开始的时候就已经开始了，而traditional approach一般会先处理data collection & format。This analysis goes beyond formal modelling or hypothesis testing

Confirmatory analysis---目的是confirm或者reject之前的hypothesis，一般用的是之前的pattern generation中没用过的test dataset。这种方式通常需要human intervention

Data Visualization：通过软件或者编程语言实现

通常针对basic dataset的表示形式有：Line Graph，Bar Chart，Scatter Plots(离散的点的图，某些情况下还可能有regression line)，Bubble Plot(也是画点，通过点的大小来表示某些信息，相当于用二维的图来标识三维/种信息)，Pie Charts

但是有的过大的数据集无法用这些方式来呈现，所以就需要进行sample：采样的时候需要考虑很多问题，比如volume---取多大的样本 & 单独的点还是整合的数据；variety---are all data comparable；Velocity---数据变化的频率/速度(在针对单次分析时问题不大，但是对于整合的数据会有一定影响)；Access---是否能够及时更新并且提供最近版本的数据给使用的人