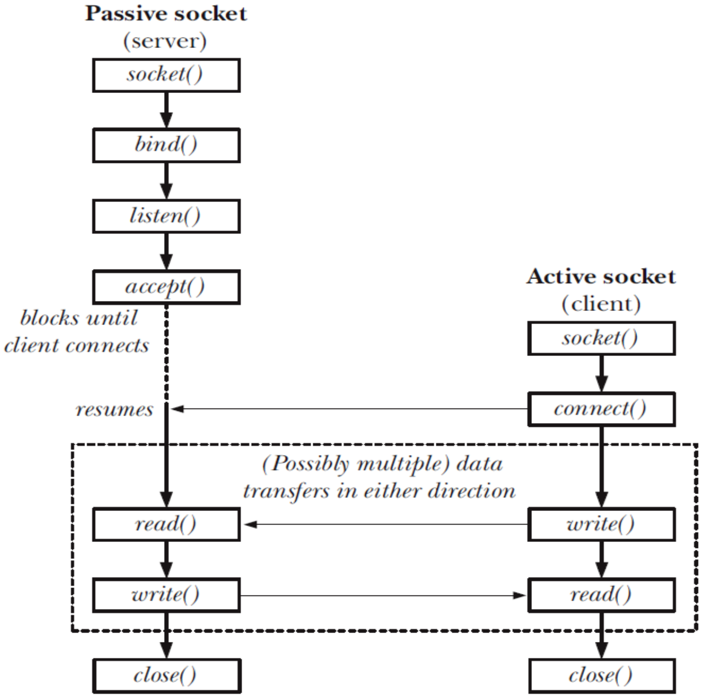
Module 2. Networking

Socket: A method of IPC that allow data to be exchanged between applications, either on the same host or on different hosts connected by a network.

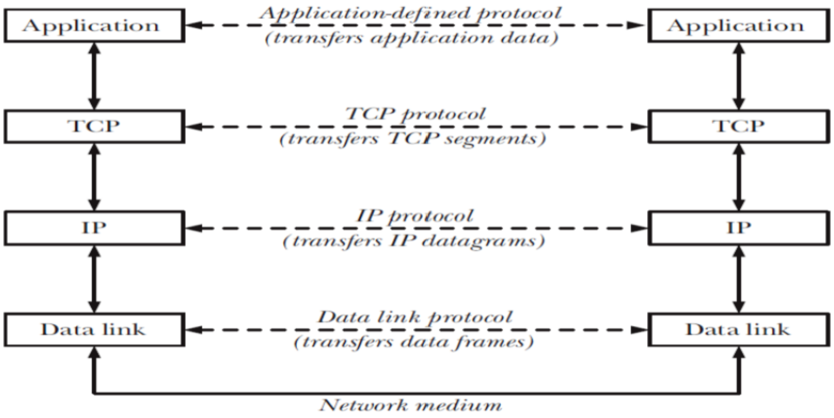
Socket Domains: UNIX(AF_UNIX), IPv4(AF_INET), IPv6(AF_INET6)

Stream Sockets (SOCK_STREAM) : Provide a reliable, bidirectional, byte-stream communication channel.
Diagram Sockets (SOCK_DGRAM) : Allow data to be exchanged in the form of messages called datagrams. With datagram sockets, message boundaries are preserved, but data transmission is not reliable. Messages may arrive out of order, be duplicated, or not arrive at all.

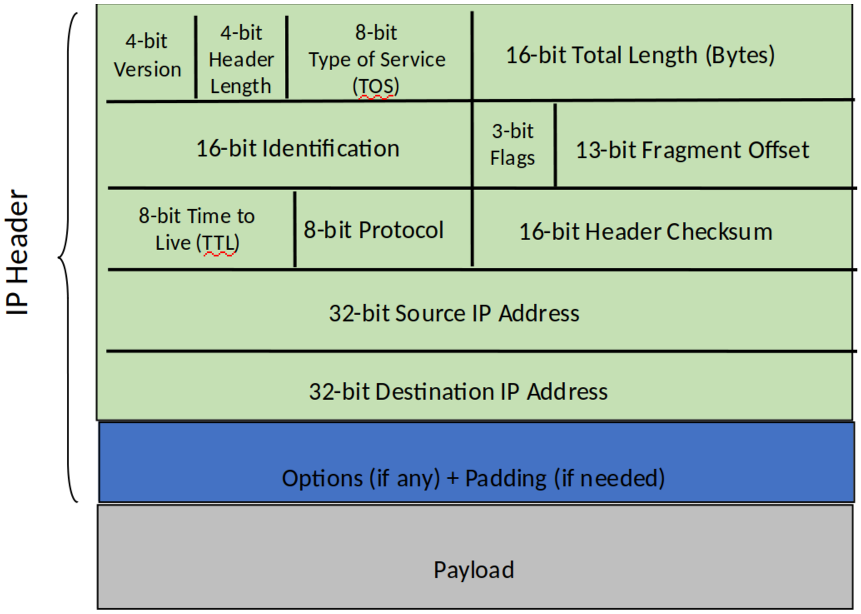
System pipeline:



Layered communication:



IP :



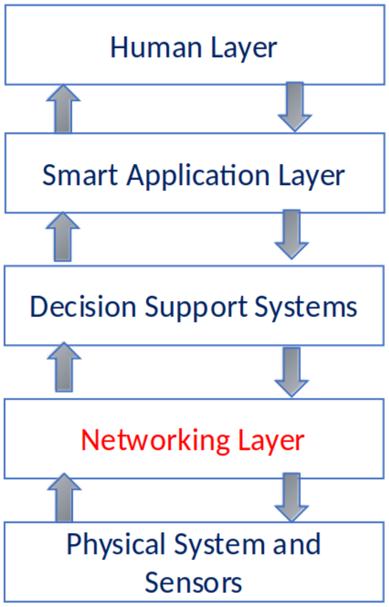
Ports : The task of the transport protocol is to provide an end-to-end communication service to applications residing on different hosts (or sometimes on the same host). In order to do this, the transport layer requires a method of differentiating the applications on a host. In TCP and UDP, this differentiation is provided by a 16-bit port number.

Routers : Router consists of (1) set of input interfaces where packets arrive. (2) set of output interfaces from which packet depart. (3) Some form of interconnection. Router implements forward packet to corresponding output interface.

Broadcast : Set host bits to all 1. That is send the datagram (UDP) to all sockets on all ip addresses in my network.
Multicast : Selective group broadcast, only send messages to a subset of recipients.

Module 3. Middleware and Backend

Networking Layers :



IoT Middleware : Problems to address: (1) Hardwiring target addresses. (2) Typical protocol is A sends data to B, then B replies while A waits and then then A responds back. Data marshalling problem. Middleware as a combination of services [external processes] and libraries that solves the problems.

Communication patterns :

(1) Asynchronous Interaction : The parties do not wait for a response and computation can start without waiting for communication to finish.

(2) Synchronous Interaction : Computation is blocked to ensure that the communication finishes.

Publish/Subscribe : Provides decoupling among producers (called publishers) and consumers (called subscribers along three dimensions).

(1) Time : They do not have to be up and running simultaneously.

(2) Space: They do not need to be aware of each other.

(3) Synchronization : Nobody blocks for the other.

(4) Advantages: Increased scalability. Improved resilience. Reduction in coordination. Suited for asynchronous environments.

Publish/Subscribe Models :

(1) Topic-based : Publishing/Subscription is based on events that are named or have an ID

(2) Content-based : Publishing/Subscription is based on the contents of the event

(3) Type-based : Like topic based but provides a programming language-level type

(4) Data-centric: use topics which are typed and also deals with the content

Quality of Service(QoS) :

(1) Must be provided along the 3 dimensions of decoupling

(2) Persistence of events

(3) Priorities of events may be important

(4) Transactional semantics for group of events

(5) Reliability

Module 4. Reliability

Dependability steps :
(1) Define dependability requirements
(2) Determine all possible faults that might occur
(3) Practice fault avoidance during construction
(4) Practice fault elimination during construction
(5) Practice fault tolerance during operation

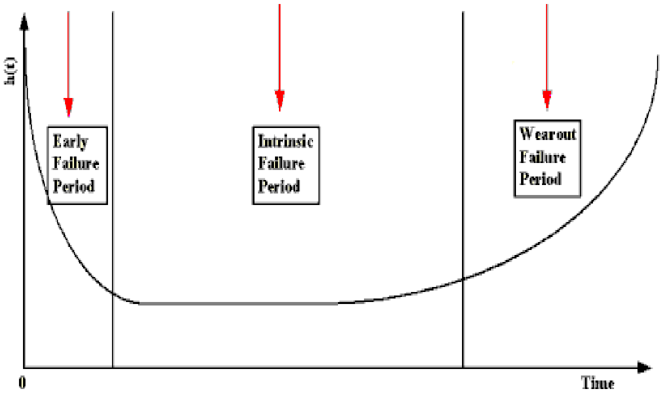
Progression of failures :
(1) Determine consequences of failures: direct / indirect
(2) Define the dependability requirements

Dependability: The dependability of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable.

Attributes of dependability: reliability, availability, safety, confidentiality, integrity, maintainability

Failure viewpoints :
(1) Failure domain: Content failure/ Timing failure
(2) Detectability of failures: signaled, unsignaled, false alarm
(3) Consistency of failures:
consistent: incorrect service seen identically by all users
inconsistent: incorrect service seen differently by different users, Byzantine failures
(4) Consequences of failures

Reliability Theory :
Failure rate (bathtub curve) has three regions:
(1) Early-life failures : No probability evaluation for stages
(2) Random failures : Failure probability represented using an exponential distribution (MTTF)
(3) Wear-out failures : Failure probability represented using Weibull distribution



Random Processes :
(1) Binomial Distribution: Probability of exact number of successes in n independent trials
(2) Poisson Distribution: Probability of rare events (special case of binomial distribution)
(3) Normal Distribution
(4) Exponential Distribution

Reliability block diagram (RBD) : Reliability Block Diagram (RBD) is a graphical representation of how the components of a system are connected from reliability point of view. RBD is used to model the various series-parallel and complex block combinations (paths) that result in system success.
(step 1) Define boundary of the system for analysis
(step 2) Break system into functional components
(step 3) Determine serial-parallel combinations
(step 4) Represent each components as a separate block in the diagram
(step 5) Draw lines connecting the block in a logical order for mission success

Hazard analysis :
Goal:
(1) Identify events that may eventually lead to accidents
(2) Identify individual elements/operations within a system that render it vulnerable (single point failures)
(3) Determine impact on system
Techniques:
(1) FMEA: Failure Models and Effects Analysis
(2) FMECA: Failure Models, Effect and Critically Analysis
(3) ETA: Event Tree Analysis
(4) FTA: Fault Tree Analysis
(5) HAZOP: HAZard and Operability studies

Module 5. Testing-Monitoring

Test type :
(1.1) Functional testing (Blackbox)

-identify the the functions which software is expected to perform

-create test data which will check whether these functions are performed by the software

-no consideration is given how the program performs these functions, program is treated as a black-box: black-box testing

-need an oracle: oracle states precisely what the outcome of a program execution will be for a particular test case. This may not always be possible, oracle may give a range of plausible values

(1.2) Structural testing (Whitebox)

-the test data is derived from the structure of the software

-white-box testing: the internal structure of the software is taken into account to derive the test cases

(2.1) Module testing
(2.2) Integration testing

Test case : A test case is a partial specification of a run through the naming of its direct input variables and their values.

Control Flow Graphs(CFGs) : Nodes in the control flow graph re basic blocks. A basic block is a sequence of statements always entered at the beginning of the block and exited at the end.

Statement Coverage : Choose a test set T s.t by executing program P for each test case in T, each basic statement of P is executed at least once.

Branch Coverage : After constructing a control flow graph, select a test set T s.t by executing program P for each test case d in T, each edge of P's control flow graph is traversed at least once.

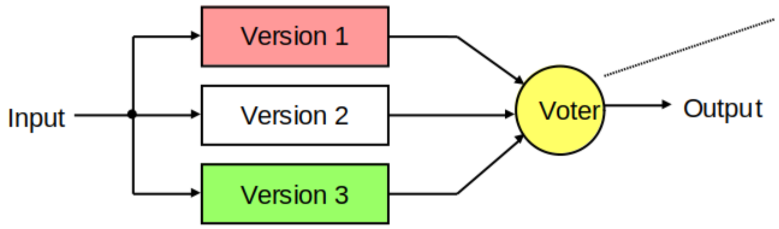
Path Coverage : Select a test set T s.t by executing program P for each test case d in T, all paths leading from the initial to the final node of P's control flow graph are traversed.

Condition Coverage : Select a test set T such that by executing program P for each test case d in T, (1) each edge of P's control flow graph is traversed at least once and (2) each boolean term that appears in a branch condition takes the value TRUE at least once and the value FALSE at least once.

Service Level Objectives (SLO) : Target level of reliability for the service's customers.
Service Level Indicators (SLI) : The number of good events divided by the total number of events

Module 6. Software Fault Tolerance

Software Fault Tolerance : The point of fault tolerance is to absorb faults and prevent them from becoming failures without an active mitigation. The key is redundancy.
(1) Masking redundancy: N-version programming

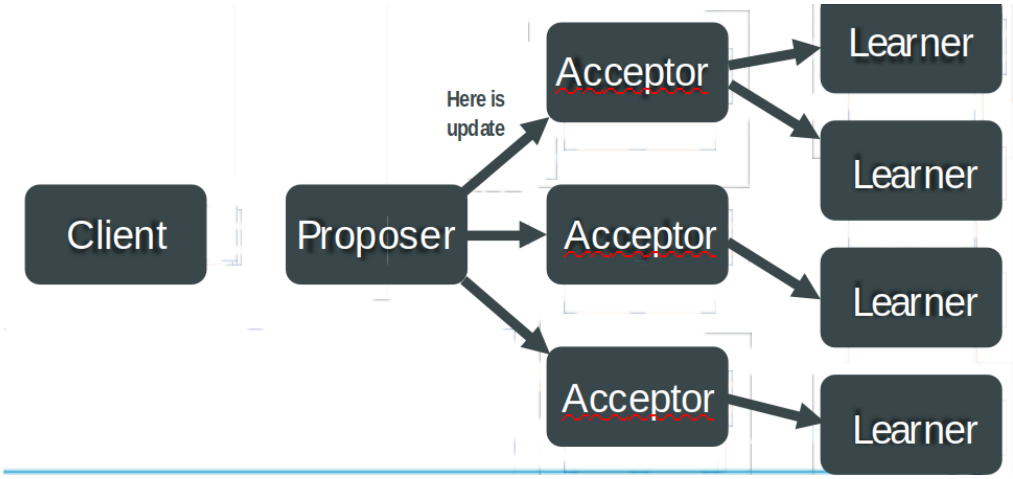


(2) Standby redundancy: the recovery-block scheme
(3) Self-checking design: N-self-checking programming

CAP Theorem :
(1) Consistency: Allnodes should see the same data at the same time
(2) Availability: Node failures do not present survivors from continuing to operate
(3) Partition-tolerance: The system continues to operate despite network partitions
Two of these can be guaranteed at the same time but not all three.

Three kinds of distributed systems:
(1) CP
(2) AP
(3) CA

Paxos :



Byzantine Fault Tolerance :
Assumptions:
(1) Every message sent is delivered correctly
(2) The receiver knows who sent the message
(3) Message delivery time is bounded

Problem statement:
An imaginary General who makes a decision to attack or retreat and must communicate the decision to his lieutenants. A given number of these actors are traitors. Traitors cannot be relied on to properly communicate orders, they may actively alter messages in an attempt tp subvert the process.

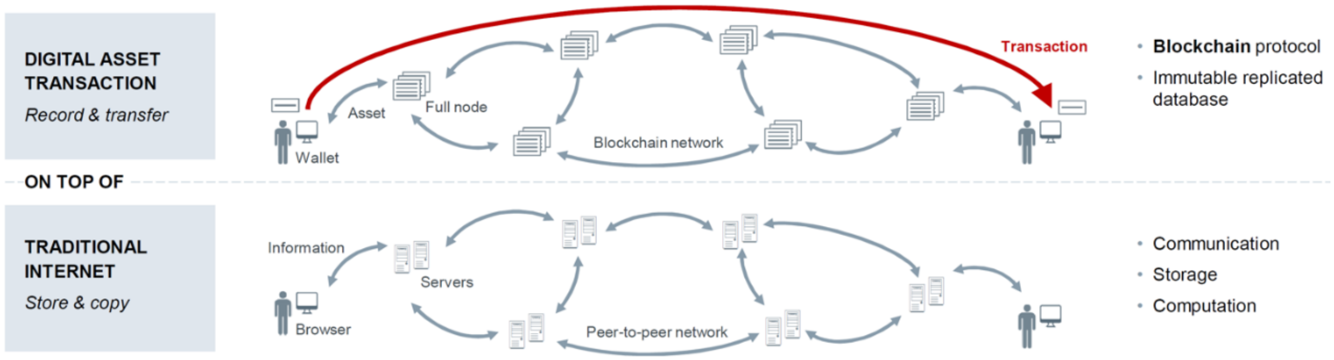
Analogies:
General – source process (P1)
orders – messages
traitors – faulty processes

Basic result: if m are faulty then we need 3m+1 total processes. The algorithm runs for m+1 rounds where m is the number of traitors.

Tree Storage:
Each process stores all messages it received in a tree structure. Then the decision criteria will be majority voting fold the tree back up from leaves.

Module 7. Bloackchains

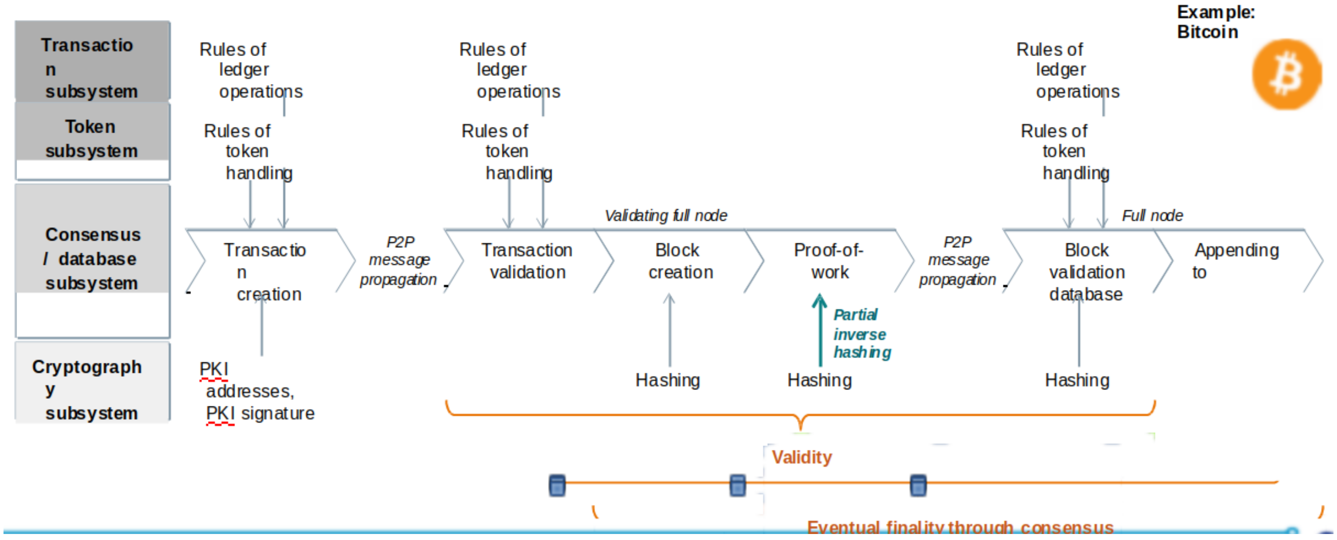
Definition: A blockchain is a decentralized platform that supports transactions with eventual consistensy.



Blockchain middleware enables digital asset transactions on top of the traditional Internet. And it allows anonymous exchange of digital assets without the need of a central authority to verify trust and transfer of value.

Technical Foundations of Blockchain :
Key Problem: Consistent record keeping in distributed systems
(1) Centralized Database System
Central authority, exclusive keeper of “ground truth”. Maybe replicated to a redundant stand-bt system for higher availability.
(2) Distributed Database System
Consistency machanisms for guaranteeing completeness and immutability. Full or partial replication. Distributed queries and transactions.
(3) Peer-to-peer (P2P) Database System
Decentralized control. Full or partial replications. Highest reliability, worst latency.
P2P systems is a network of asysnchronously computing nodes, which means responses are not guaranteed. Communication can occur at any time and at irregular intervals. Communication delay is not bounded.

Blockchain Protocol :



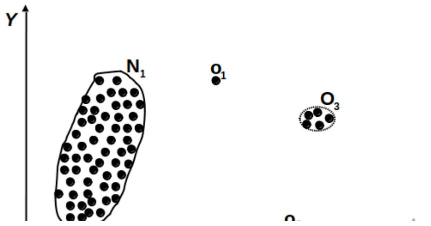
Module 8. Anomaly

<u>Definitions</u> :
(1) Precision: Proportion of events detected that were significant.--------P = TP/(TP+FP)
(2) Recall: Proportion of significant events detected. -----------------------R = TP/(TP+FN)
(3) Detection time: How long it takes to send notifications in various conditions.
(4) Reset time: How long alerts fire after an issue is resolved.
(5) Error budget: Number of allowed bad events.
(6) Error rate: Ratio of bad events to total events.
(7) F-value: Evaluation of anomaly detection 2*R*P/(R+P)
(8)ROC curve: Trade-off between detection rate and false alarm rate

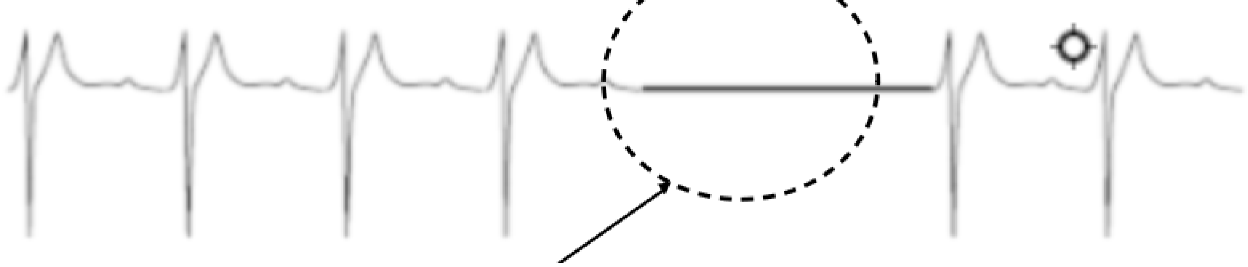
<u>Pre-configured threshold</u> :
Example: SLO is 99.9% over 30 days (pre-configured 0.1%), alert if the error rate over the previous 10 minutes (time window) is > 0.1%.

<u>More General Anomaly Detection</u> :
(1) Supervised Anomaly Detection: Label available for both normal and anomalies
(2) Semi-supervised Anomaly Detection : Labels available only for normal data
(3) Unsupervised Anomaly Detection: No labels assumed. Assume anomalies are very rare compared to normal data.

<u>Types of Anomaly</u> :
(1) Point Anomalies : Simplest, can occur in dataset
(2) Contextual Anomalies : Individual instance, anomalous in a specific context
(3) Collective Anomalies : A collection of data instances, their occurrence together as a collection is anomalous



Monthly Temp
Anomaly



Techniques for Point Anomaly Detection :
(1) Statistical : Outliers are objects that are fit poorly by a statistical model.
(2) Proximity-based :
(3) Density-based
(4) Clustering-based

Control Charts : Provide a mechanism for recognizing situation where assignable causes may be adversely affecting product quality. A basic element is that samples have been selected from the process of interest at sequence of time.

CUSUM :
Definition: A cumulative sum (CUSUM) chart is a type of control chart used to monitor small shifts in the process mean. It uses the cumulative sum of deviation from target.

Classification-Based Approaches :
Main idea: Build a classification model for normal (and anomalous) events based on labeled training data, and use it to classify each new unseen event.
(1) Supervised classification techniques : Require knowledge of both normal and anomaly class.
(2) Semi-supervised classification techniques : Require knowledge of normal class only.

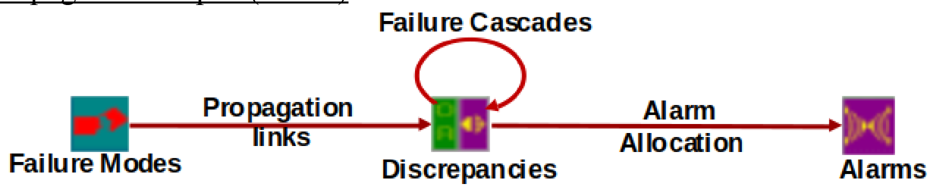
Proximity Based Approaches :
(1) Nearest Neighbor based techniques: Distance based / Density based
(2) Clustering based techniques: Semi-supervised / Unsupervised

Contextual Anomaly Detection :
Assumption: All normal instances within a context will be similar (in terms of behavioral attributes), while the anomalies will be different.
General Approach: Identify a context around a data instance (using a set of contextual attributes). Determine if the data instance is anomalous w.r.t the context(using a set of behavioral attributes)

Collective Anomaly Detection :
(1) Sequential anomaly detection: Detect anomalous sequences
(2) Spatial anomaly detection : Detect anomalous sub-regions within a spatial data set
(3) Graph anomaly detection : Detect anomalous sub-graphs in graph data

Module 9. Diagnosis

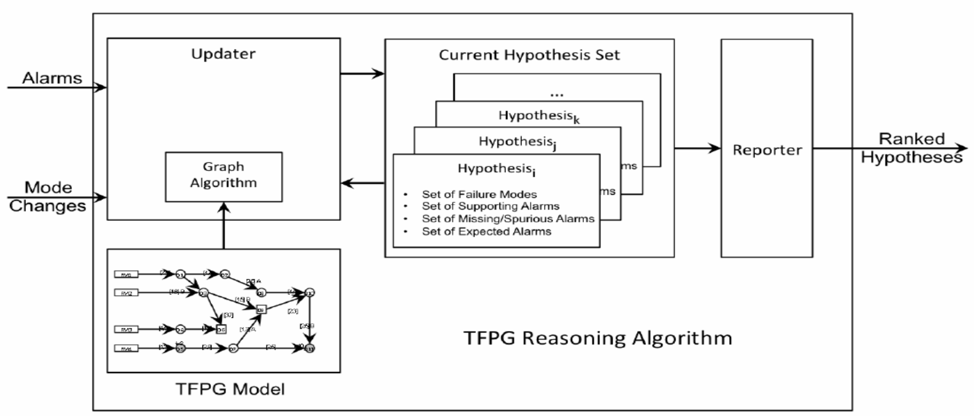
Timed Failure Propagation Graphs (TFPG) :



Cause model that describe the system behavior in presence of faults.
Model is a labeled directed graph where:

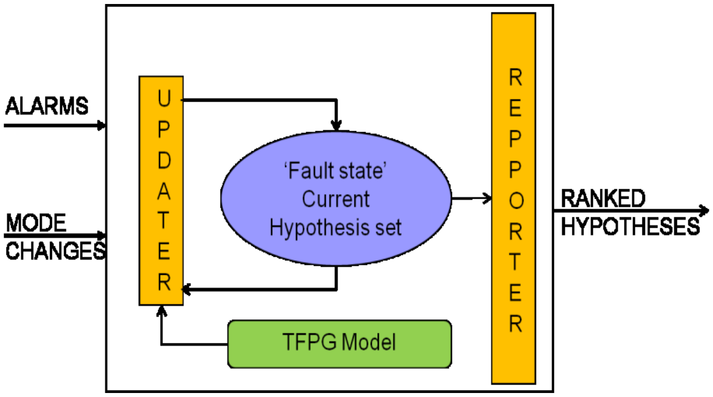
(1) Nodes represent either failure modes or discrepencies
(2) Edges between nodes in the graph represent causality
(3) Edge are attributed with timing and mode constrains on failure propagation.

TFPG Reasoning :
Input: Sequence of alarms and mode changes
Output: Sequence of sorted and ranked hypotheses containing failure mode(s) that explain the observations (alarms, mode changes)



TFPG H
-Directl s.
-Indirec istent with the hypothesized state of the failure modes.

TFPG Diagnostics :
Algorithm outline:
– Check if new evidence is explained by current hypothesis.
– If not, create a new hypothesis that assumes a hypothetical state of the system consistent with observations.
– Rank hypotheses for plausibility and robustness.
– Discard low-rank hypotheses, keep plausible ones



Hypotheses Evaluation Metrics :
Plausibility: Reflects the support of a hypothesis based on the current observed alarm state.
Robustness: Reflects the potential of a hypothesis(evidence) to change based on remaining alarms.
Failure Rate: Failure rate is a measure of how often a particular failure mode has occurred in the past.

Module 10. Time-Synchronization and Logical Ordering

Synchronous networks : Messages always arrive, with propagation delay at most D. Sender send time T in a message. Receiver set clock to T+D/2. Synchronization error is at most D/2.

Cristian's Algorithm :
Request time, get reply: Measure actual round-trip time d
Sender's time was T between t1 and t2.
Receiver sets time to T+d/2, synchronization error is at most d/2

The Berkeley Algorithm : Master uses Cristian's algorithm to get time from many clients. Sends time adjust back to all clients.

The Network Time Protocol(NTP) :
(1) Uses a hierarchy of time servers
(2) Synchronization similar to Cristian's Algorithm
(3) Accuracy: Local~1ms, Global~10ms

Ordering of Events (Logical Clocks) :
Lamport logical clocks
Total-order Lamport clocks: use Lamport's algorithm, but break ties using the process ID

Global snapshots : A global snapshot includes the state of each local process and information in the channels at the time the snapshot was taken.

Snapshots :
Checkpointing: Restart if the application fails
Collecting garbage: Remove onjects that don't have any references
Detecting deadlocks: can examine the current application state
Other debugging: a little easier to work with than printf

Chandy-Lamport Algorithm

Module 12. Streaming Middleware

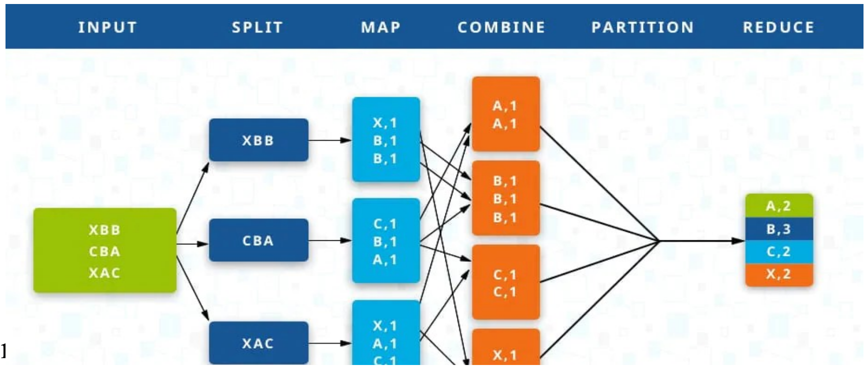
Client-Server Model :
(1) Request : User triggers communication with server through web browser
(2) Routes the request to resource / application logic to process request
(3) Application logic communicates with data model to query, access or update data as needed

SQL Data Model :
(1) Rooted in relational logic
(2) Support complex data retrieval-joins
(3) Find individual records, combine result and return to application

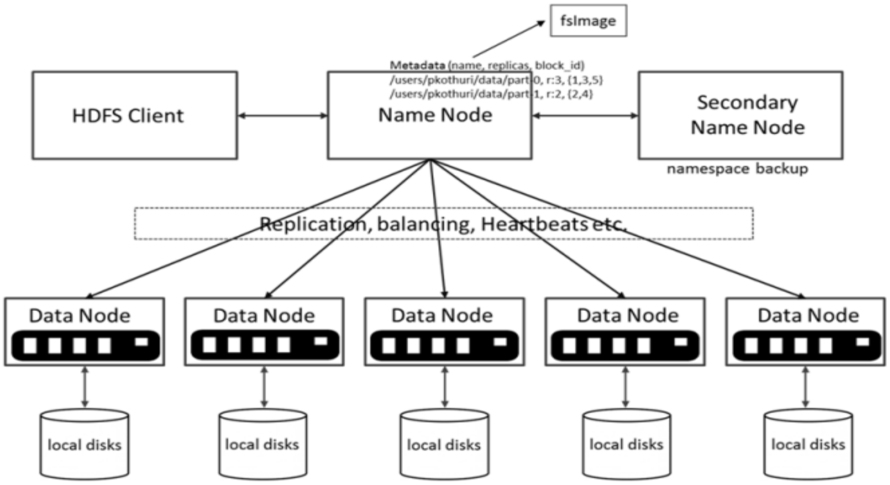
Batch and Stream Processing :
(1) Jobs are periodically scheduled to run on large batches of data
(2) Outputs aggregated analysis

(3) Common for data analysis

MapReduce and Distributed File Systems(HDFS) :
MapReduce is to split data, apply mapping functions to each subset and finally the processed outputs are combined and returned.



HDFS is used to sto



Stream Processing :
(1) Near real-time processing
(2) Apply lightweight functions to transform data as it arrives
(3) Spark is a common processing framework for real-time streaming applications

Spark Streaming