Fault-Detection in Cloud Computing Systems

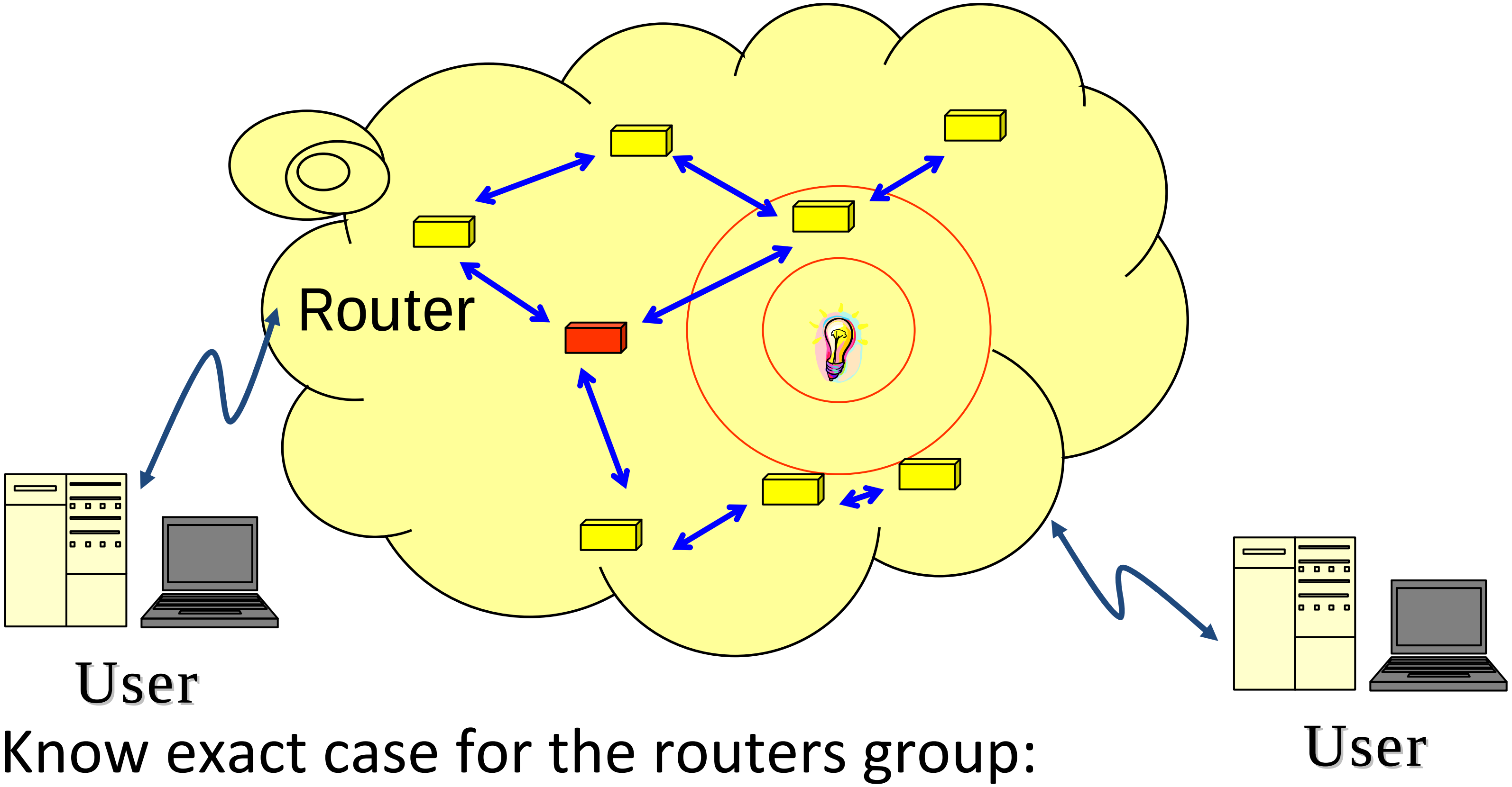
Yi Pan

Department of Computer Science

Georgia State University, Atlanta, Georgia

In collaboration with N. Xiong, A. Vandenberg, and Andy Rindos

Traditional network application

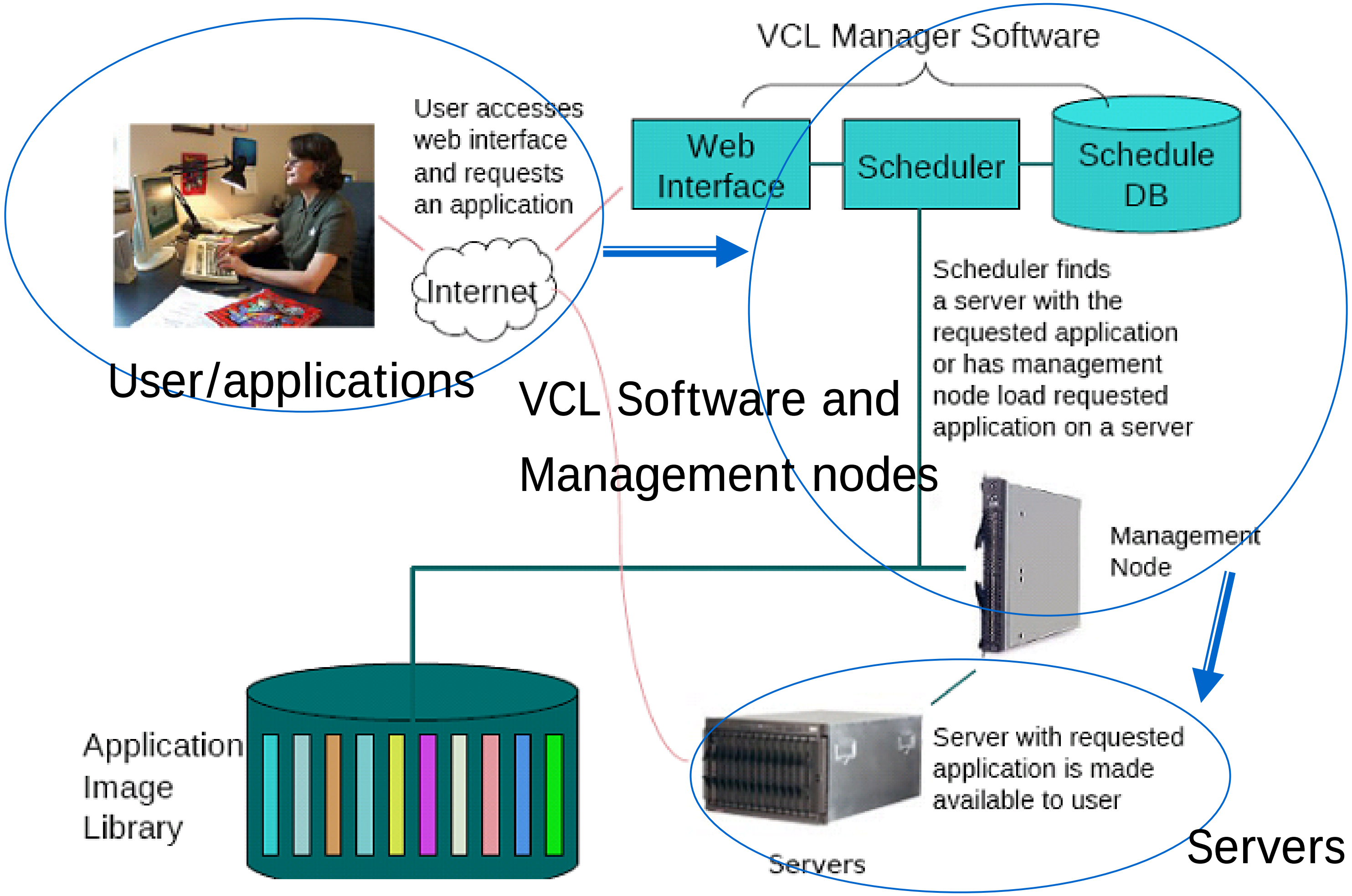


- Know exact case for the routers group:
  - If, better for packets transmission
  - Otherwise, miss packets, reduce QoS of packets transmission
  - Networks resource are not extensive shared (partly shared)

What is a cloud?

- Definition [Abadi 2009]
  - shift of computer processing, storage, and software delivery away from the desktop and local servers
  - across the network and into next generation data centers
  - hosted by large infrastructure companies, such as Amazon, Google, Yahoo, Microsoft, or Sun

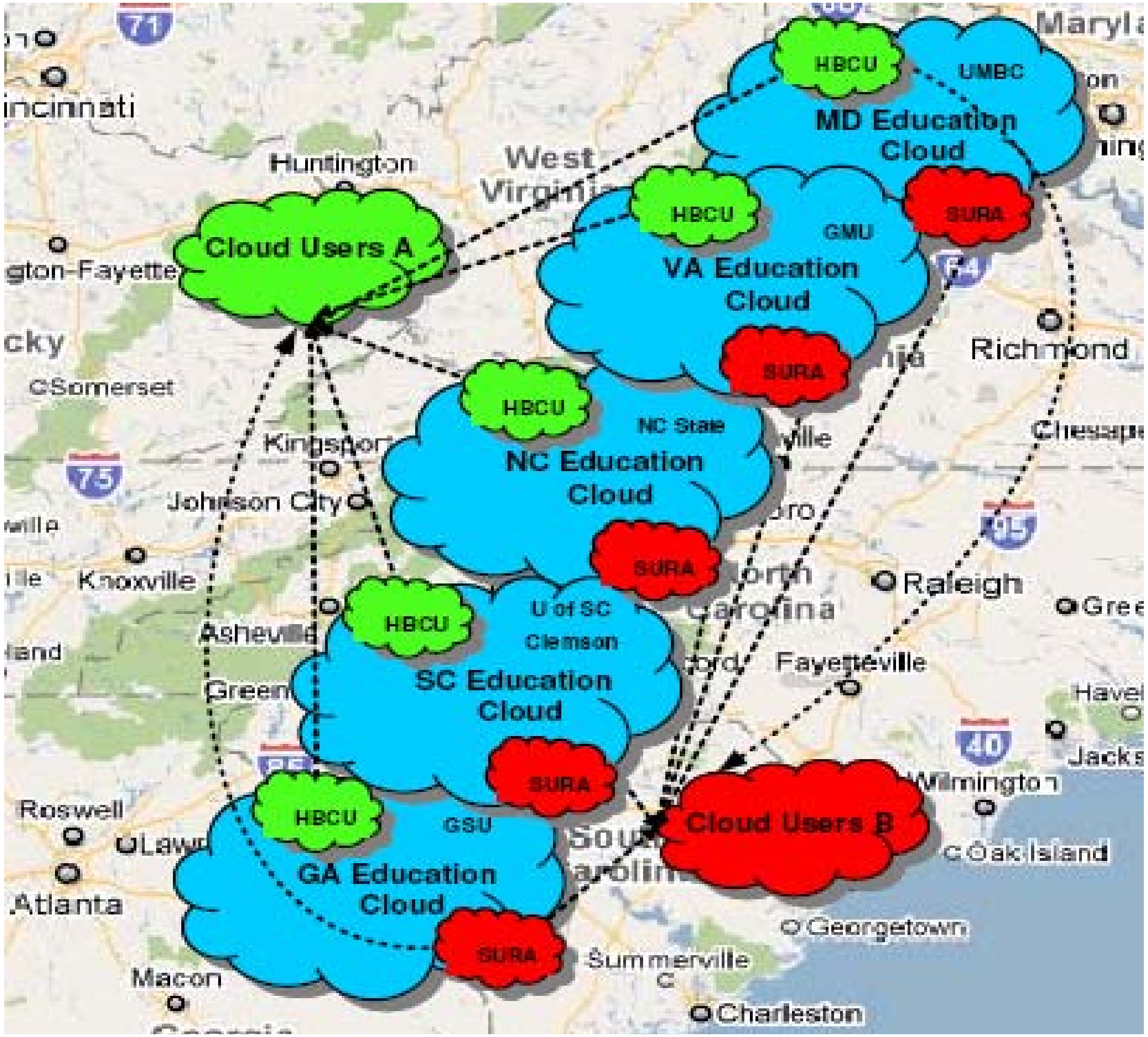
Dynamic cloud-based network model



North Carolina State University VCL model

http://vcl.ncsu.edu/

Dynamic cloud-based network model



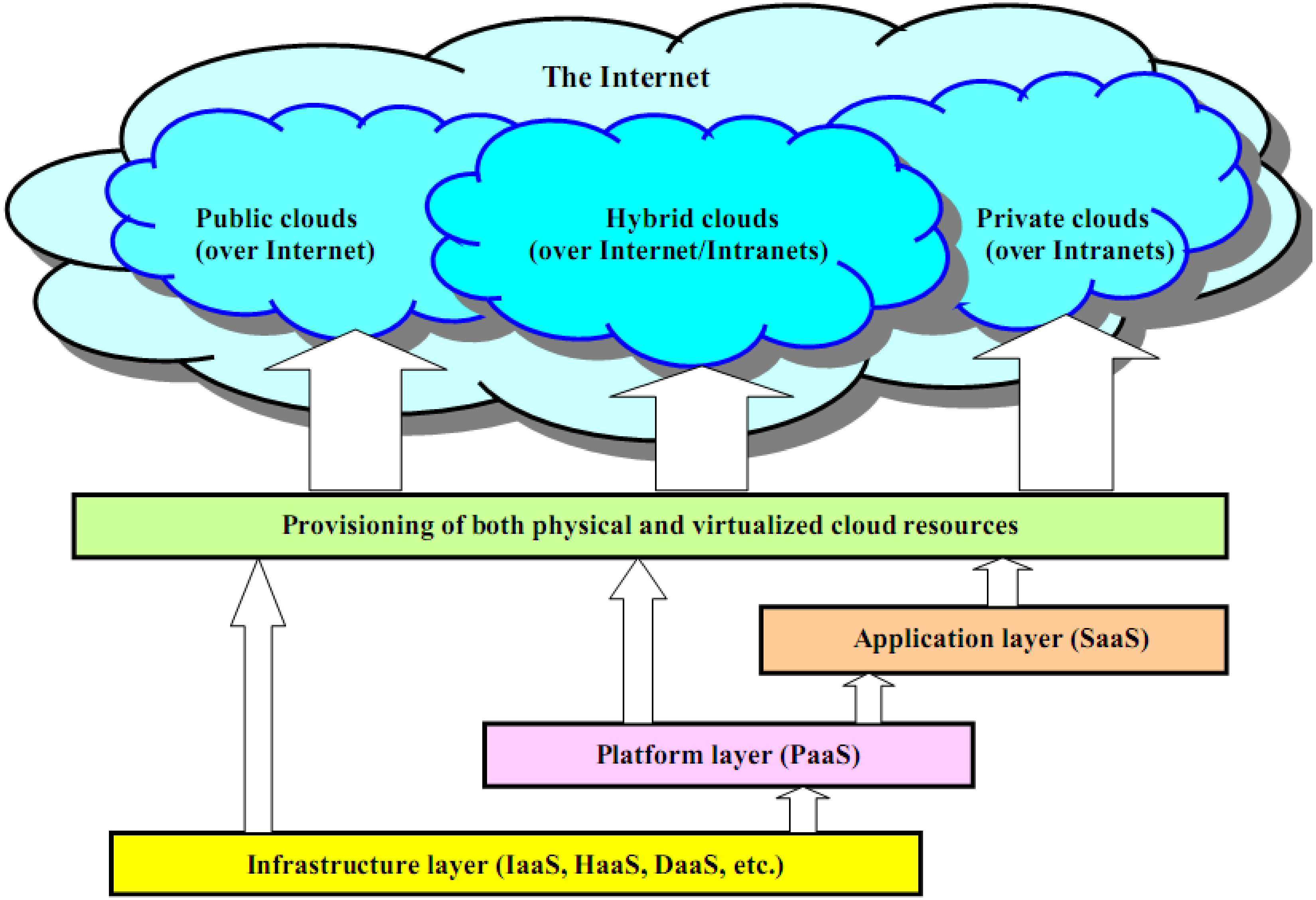
U.S. southern state education Cloud, sponsored By IBM, SURA & TTP/ELC

Types of Cloud Service

- According to architectural structure [Sun 2009]
  - Platform as a Service (PaaS)
  - Infrastructure as a Service (IaaS)
  - Software as a Service (SaaS)

- Database solution
  - Database as a Service (DaaS)

Cloud Computing as A Service



Cloud Services Stack

Application Cloud Services

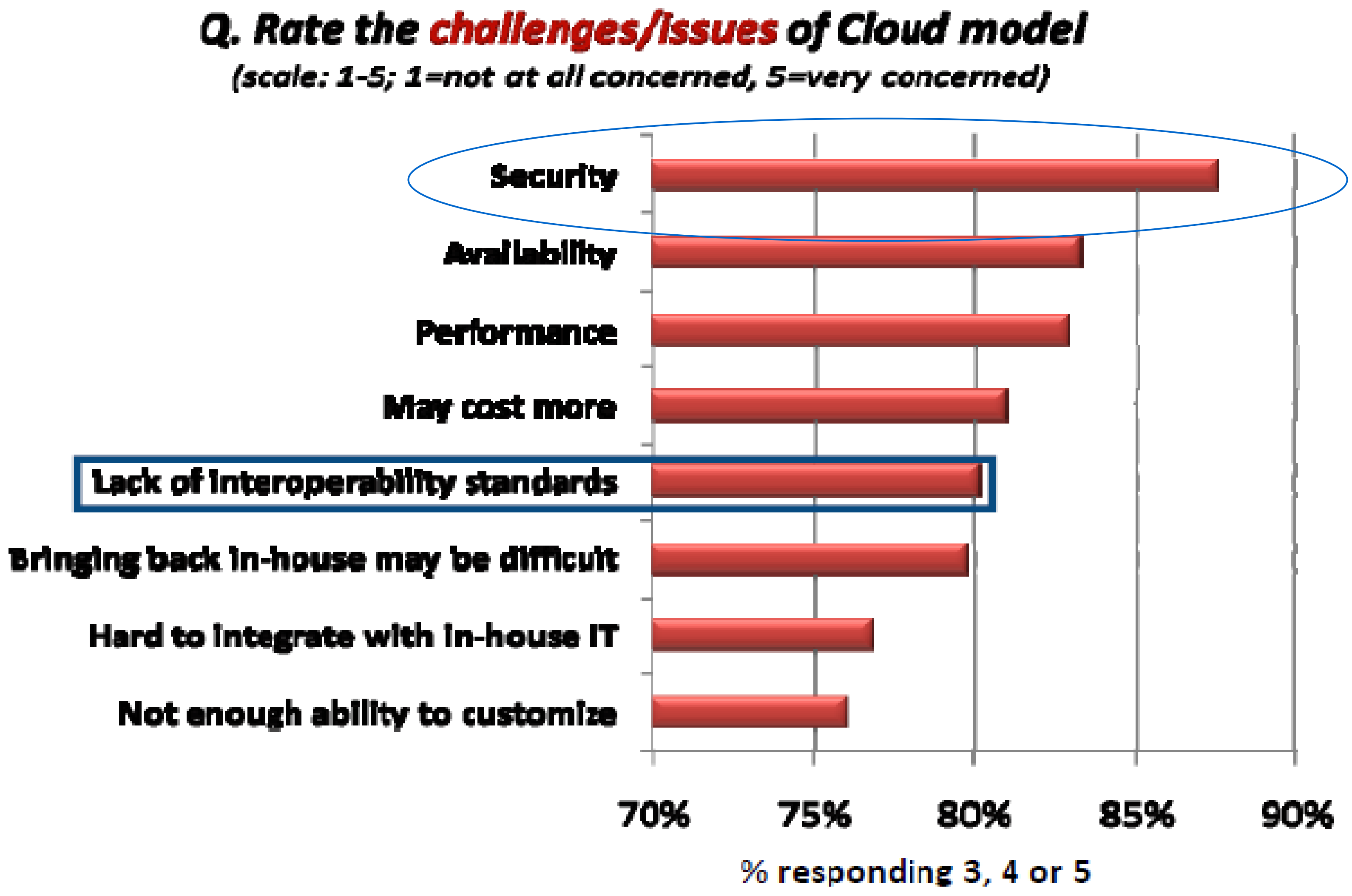
Platform Cloud Services

Compute & Storage Cloud Services

Co-Location Cloud Services

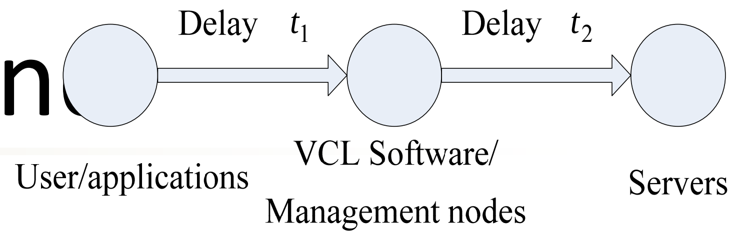
Network Cloud Services

What's Worrisome about Cloud?



Source: IDC Enterprise Panel 3Q09 N=263

Background



- GSU is deploying VC as a solution alternative to traditional student computing labs
- VC as a solution to support researchers:
  where researchers request computing environments that may be non-standard configurations not readily available
- Some VCL related areas of interest are:
  Network control and security; dynamic virtual local area networks (VLANS) and VLAN control; support for high-performance computing (HPC); resource allocation between HPC and other services.

An example: PlanetLab

PlanetLab is a global network

supports the development of new network services

consists of 1076 nodes at 494 sites.

While

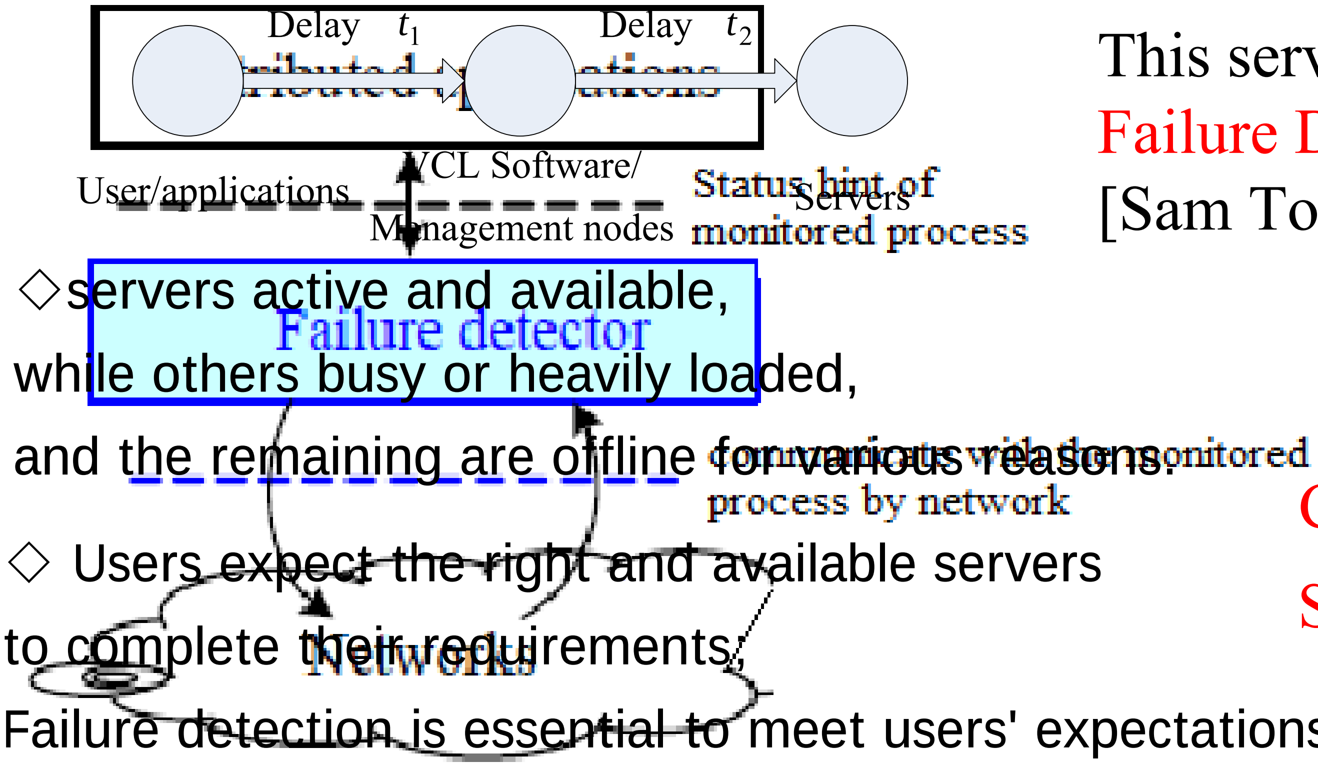
lots of nodes at any time are inactive

do not know the exact status (active, slow, offline, or dead)

impractical to login one by one without any guidance

Dynamic cloud-based network analysis

In distributed systems, applications often need to determine which processes are up (operational) and which are down (crashed)



This service is provided by Failure Detector (FD) [Sam Toueg]

Fast

Accuracy

Connection

Scalable …

◇ servers active and available, while others busy or heavily loaded, and the remaining are offline for various reasons.

◇ Users expect the right and available servers to complete their requirements;

Failure detection is essential to meet users' expectations

Difficulty of designing FD

Arrival time of data becomes unpredictable;

Hard to know if the monitored system works well.

Easy case 1:

- clock synchronous
- reliable communication
- process period and communication delay are bounded.

Actual application 2:

- clock asynchronous
- unreliable communication
- upper bound is unknown

A general application

QoS requirements**:**

- Detect crash within 30 sec
- At most one mistake per month
- Mistake is corrected within 60 s

Network environment:

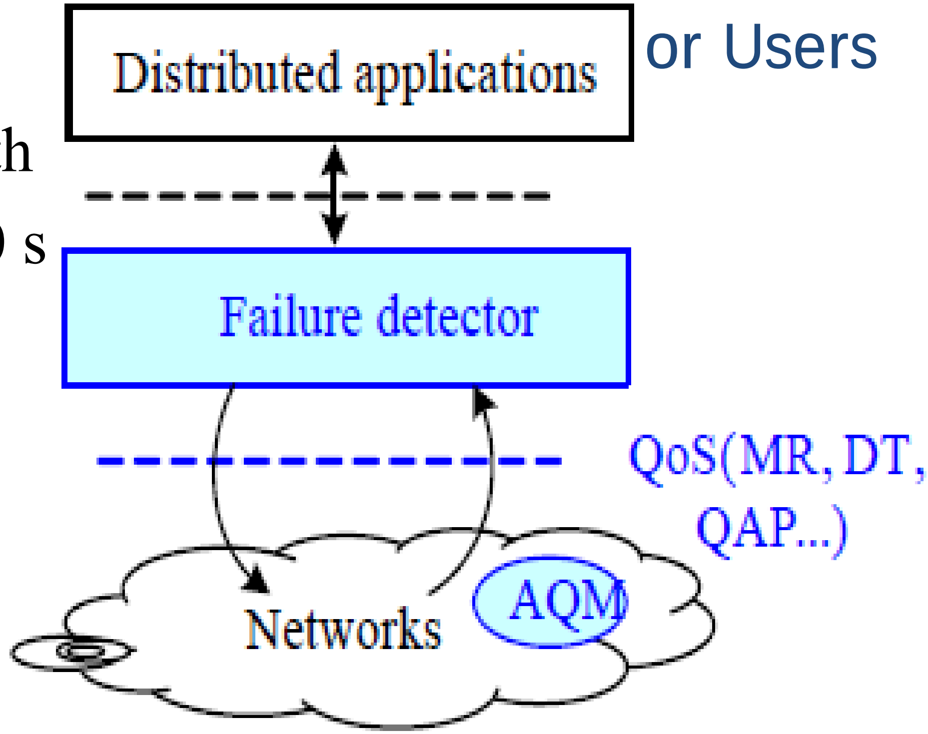
- Probability of heartbeat loss
- Heartbeat delay

Algorithm (parameters):

Detection Time, Mistake Rate

Query Accuracy Probability

Administrator or Users



Important applications of FD

FDs are at core of many fault-tolerant algorithms and applications

- Group Membership
- Group Communication
- Atomic Broadcast
- Primary/Backup systems
- Atomic Commitment
- Consensus
- Leader Election
- …..

FDs are found in many systems: e.g., ISIS, Ensemble, Relacs, Transis, Air Traffic Control Systems, etc.

1. Failure Detectors (FDs)

FD can be viewed as a distributed oracle for giving a hint on the operational status of processes.

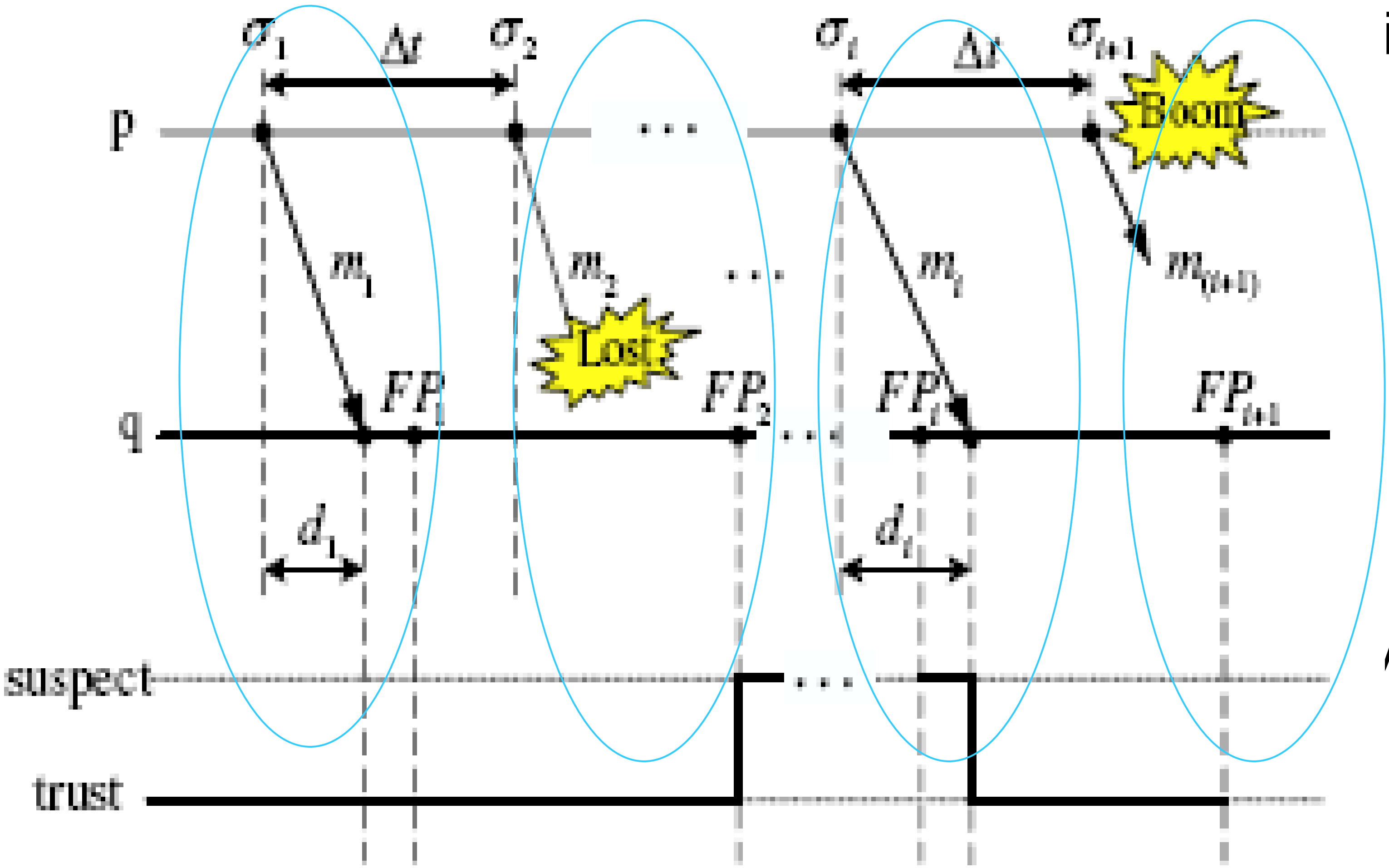
FDs are employed to guarantee continuous operation:

To reduce damage in process groups network systems.

Used to manage the health status, help system reduce fatal accident rate and increase the reliability.

Find crash server, be replaced by other servers

1. Failure Detectors (FDs)



iving

/ ...

1. Failure Detectors (FDs): Outline

- 1 **Problems, Model, QoS of Failure Detectors**
- 2 **Existing Failure Detectors**
- 3 **Tuning adaptive margin FD (TAM FD): JSAC**

  Constant safety margin of Chen FD [30]
- 4 **Exponential distribution FD (ED FD): ToN**

  Normal Distribution in Phi FD [18-19]
- 5 **Self-tuning FD (S FD): Infocom**

  Self-tunes its parameters

1. Outline of failure detectors

- **1 Introduction**
- 2 Existing Failure Detectors
- 3 Tuning adaptive margin FD (TAM FD)
- 4 Exponential distribution FD (ED FD)
- 5 Self-tuning FD (S FD)

1. Failure Detectors (FDs)

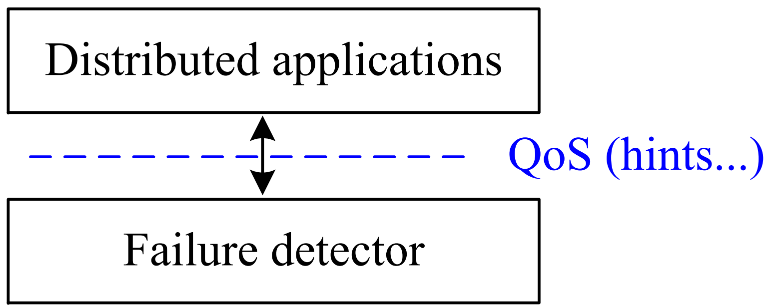
- Importance of FD :
  - Fundamental issue for supporting dependability
  - Bottleneck in providing service in node failure
- Necessity:
  - To find an acceptable and optimized FD

Failure Detectors

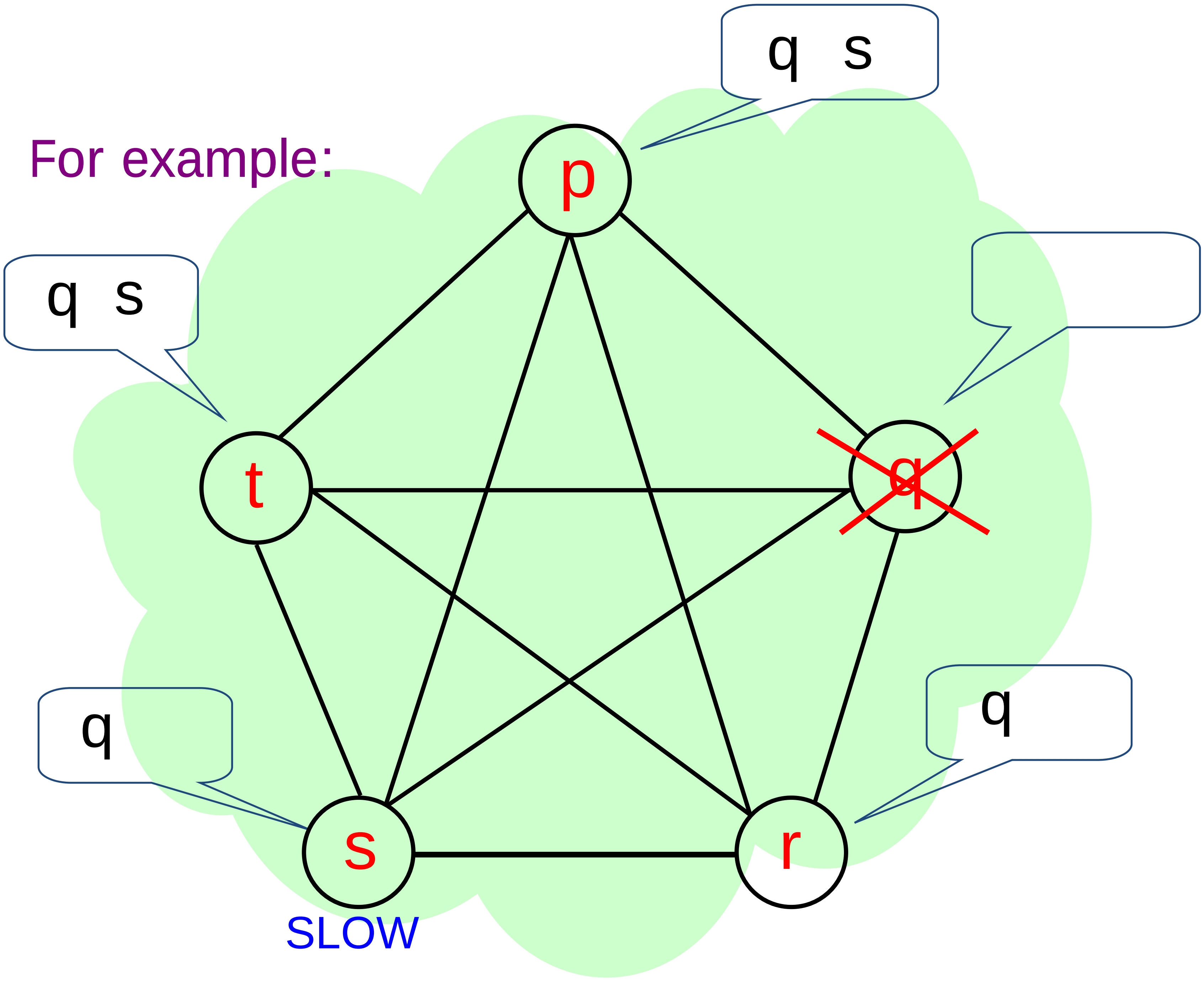
An FD is a distributed oracle that provides hints about the operational status of processes (Chandra-Toueg).

However:

- Hints may be incorrect
- FD may give different hints to different processes
- FD may change its mind (over & over) about the operational status of a process



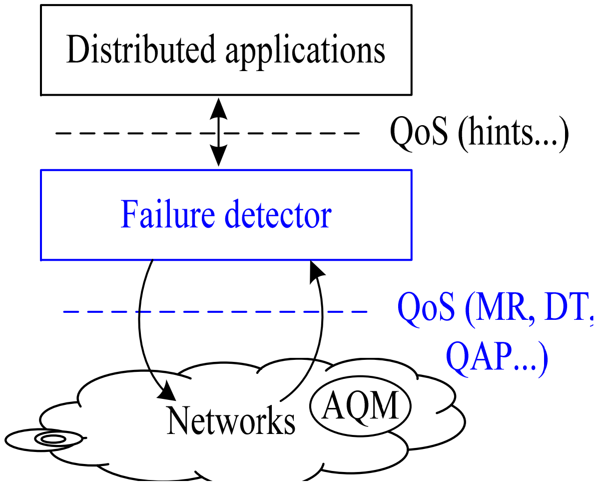
For example:



Quality of Service of FD

- The QoS specification of an FD quantifies [9]:
  - how fast it detects actual crashes
  - how well it avoids mistakes (i.e., false detections)

- Metrics [30]:
  - Detection Time (DT):
    Period from *p* starts crashing to *q* starts suspecting *p*
  - Mistake rate (MR):
    Number of false suspicions in a unit time
  - Query Accuracy Probability (QAP):
    Correct probability that process *p* is up



1. Outline of failure detectors

- 1 Introduction
- **2 Existing Failure Detectors**
- **3 Tuning adaptive margin FD (TAM FD):**
  Constant safety margin of Chen FD [30]
- **4 Exponential distribution FD (ED FD):**
  Normal Distribution in Phi FD [18-19]
- **5 Kappa FD (Kappa FD):**
  Performance evaluation and analysis [3]
- **6 Self-tuning FD (S FD):**
  Self-tunes its parameters

2. Existing FDs: Chen FD [30]

- Major drawbacks:
  a) Probabilistic behavior;
  b) Constant safety margin: quite different delay
  high probability of message loss/topology change
  Dynamic/unpredictable message

[30] W. Chen, S. Toueg, and M. K. Aguilera. On the quality of service of failure detectors. IEEE Trans. on Comp., 51(5):561-580, 2002.

- $EA_{i+1} = i \cdot \Delta(t) + \bar{d}_i$
- $\tau_{i+1} = EA_{i+1} + \gamma$

Not applicable for the actual network to obtain good QoS

Variables: EA_{i+1}: theoretical arrival; τ_{i+1}: timeout delay;
$\Delta(t)$: sending interval; γ: a constant;
$\bar{d}_i$: average delay;

2. Existing FDs: Bertier FD [16]

$$\tau_{(k+1)} = EA_{(k+1)} + \alpha_{(k+1)}$$

[16] M. Bertier, O. Marin, P. Sens. Implementation and performance evaluation of an adaptable failure detector. In Proc. Intl. Conf. on Dependable Systems and Networks (DSN'02), pages 354-363, Washington DC, USA, Jun. 2002.

$\alpha_{(k+1)}$ safety margin dynamically based on Jacobson's estimation of the round-trip time; based on the variable error in the last estimation.

Major drawbacks:

a) No adjustable parameters;

b) Large Mistake Rate and Query Accuracy Probability.

Variables: EA_{k+1}: theoretical arrival; τ_{k+1}: timeout delay;

2. Existing FDs: Phi FD [18-19]

[18] N. Hayashibara, X. Defago, R. Yared, and T. Katayama. The phi accrual failure detector. In Proc. 23rd IEEE Intl. Symp. on Reliable Distributed Systems (SRDS'04),pages 66-78, Florianpolis, Brazil, Oct. 2004.

[19] X. Defago, P. Urban, N. Hayashibara, T. Katayama. Definition and specification of accrual failure detectors. In Proc. Intl. Conf. on Dependable Systems and Networks (DSN'05), pages 206 - 215, Yokohama, Japan, Jun. 2005.

$$\varphi(t_{now}) = -log_{10}(P_{later}(t_{now} - T_{last}))$$

$$P_{later}(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_t^{+\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 - F(t)$$

φ: suspicion level, t_{now} current time; T_{last} is the time for most recent received heartbeat.

Major drawbacks:

a) Normal distribution isn't good enough for ...

b) Improvement for better performance

Outline of failure detectors

- 1 **Introduction**
- **2 Existing Failure Detectors**
- **3 Tuning adaptive margin FD (TAM FD)**
- **4 Exponential distribution FD (ED FD):**

  Normal Distribution in Phi FD [18-19]
- **5 Self-tuning FD (S FD):** Self-tunes its parameters

3. Our TAM-FD Motivation

- Basic Chen-FD scheme [1]:
  - Probabilistic behavior;
  - Constant safety margin problem;

[1] W. Chen, S. Toueg, and M. K. Aguilera. On the quality of service of failure detectors. IEEE Trans. on Comp., 51(5):561-580, 2002.

- Tuning adaptive margin FD is presented :
  - $\hat{d}_{i+1} = \alpha \cdot \hat{d}_i + (1-\alpha) \cdot d_i$
  - $EA_{i+1} = i \cdot \Delta(t) + \bar{d}_i$
  - $\tau_{i+1} = EA_{i+1} + \beta \cdot (|\hat{d}_{i+1} - \bar{d}_i| + \varepsilon)$ Bertier FD: Jacobson's estimation

Variables: $\hat{d}_{i+1}$: predictive delay; α , β : a variable; ε : a constant, EA_{i+1}: theoretical arrival

3. TAM-FD Experiment 1

- Exp. settings: All FDs are compared with the same experiment conditions:
  - the **same** network model,
  - the **same** heartbeat traffic,
  - the **same** experiment parameters (sending interval time, slide window size (1000), and communication delay, etc.).

Small WS means: Save memory and CPU resources, it's imp. for scalability.

- TAM FD, Phi FD [18-19], Chen FD [30], and Bertier FD [16-17]
- Environments: **Cluster, WiFi, LAN, WAN**

3. TAM-FD Experiment 1

Experiment setting:

- Two computers: p & q
- Without network breaking down
- Heartbeats UDP
- CPU below the full capacity
- Logged heartbeat time
- Replayed the receiving time

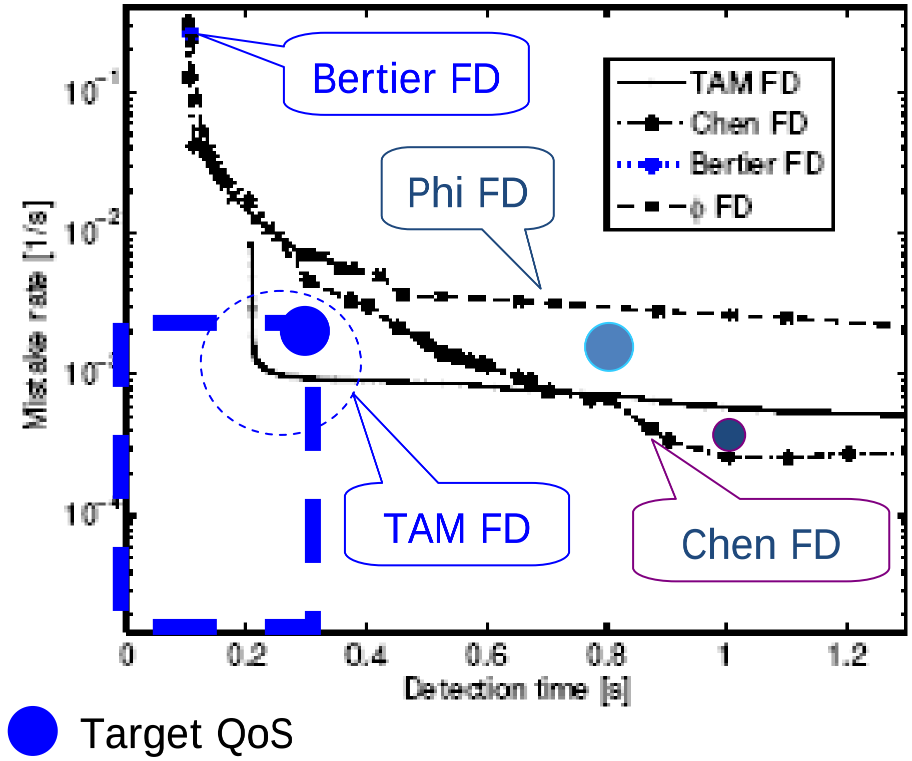
……

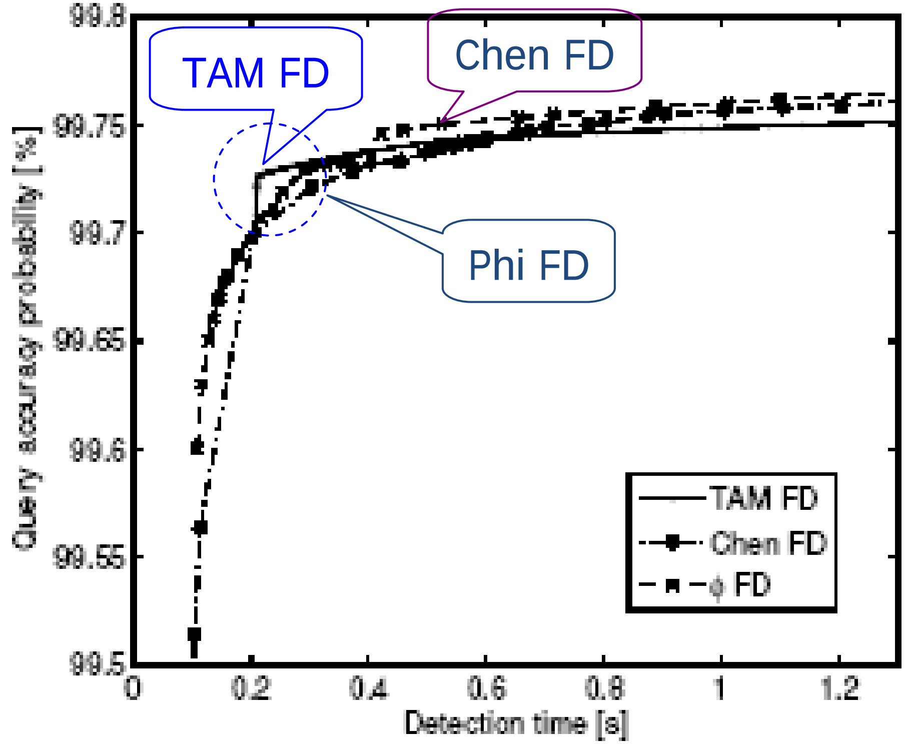
3. TAM-FD Exp. WAN (example)

- WAN exp. settings:
  - Swiss Federal Institute of Technology in Lausanne (EPFL), in Switzerland---JAIST;
  - HB sampling (over one week)
    - Sending 5,845,712 samples;
    - Receiving 5,822,521 samples;
    - Ave. sending rate: 103.501ms;
    - Ave. RTT: 283.338ms;

    …

3. TAM-FD Exp. WAN





MR and QAP comparison of FDs in WAN:

WS=1000 (logarithmic, aggressive, conservative).

3. TAM-FD Exp. WAN

- Results analysis:

  - In aggressive range: TAM FD behaves a little better than the other three FDs (short DT);

  - In conservative range, Chen FD behaves a little better than the other three FDs (long DT).

Outline of failure detectors

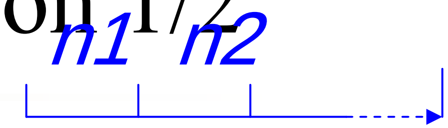
- 1 Introduction
- 2 Existing Failure Detectors
- 3 Tuning adaptive margin FD (TAM FD)
- **4 Exponential distribution FD (ED FD)**
- **5 Self-tuning FD (S FD):** Self-tunes its parameters

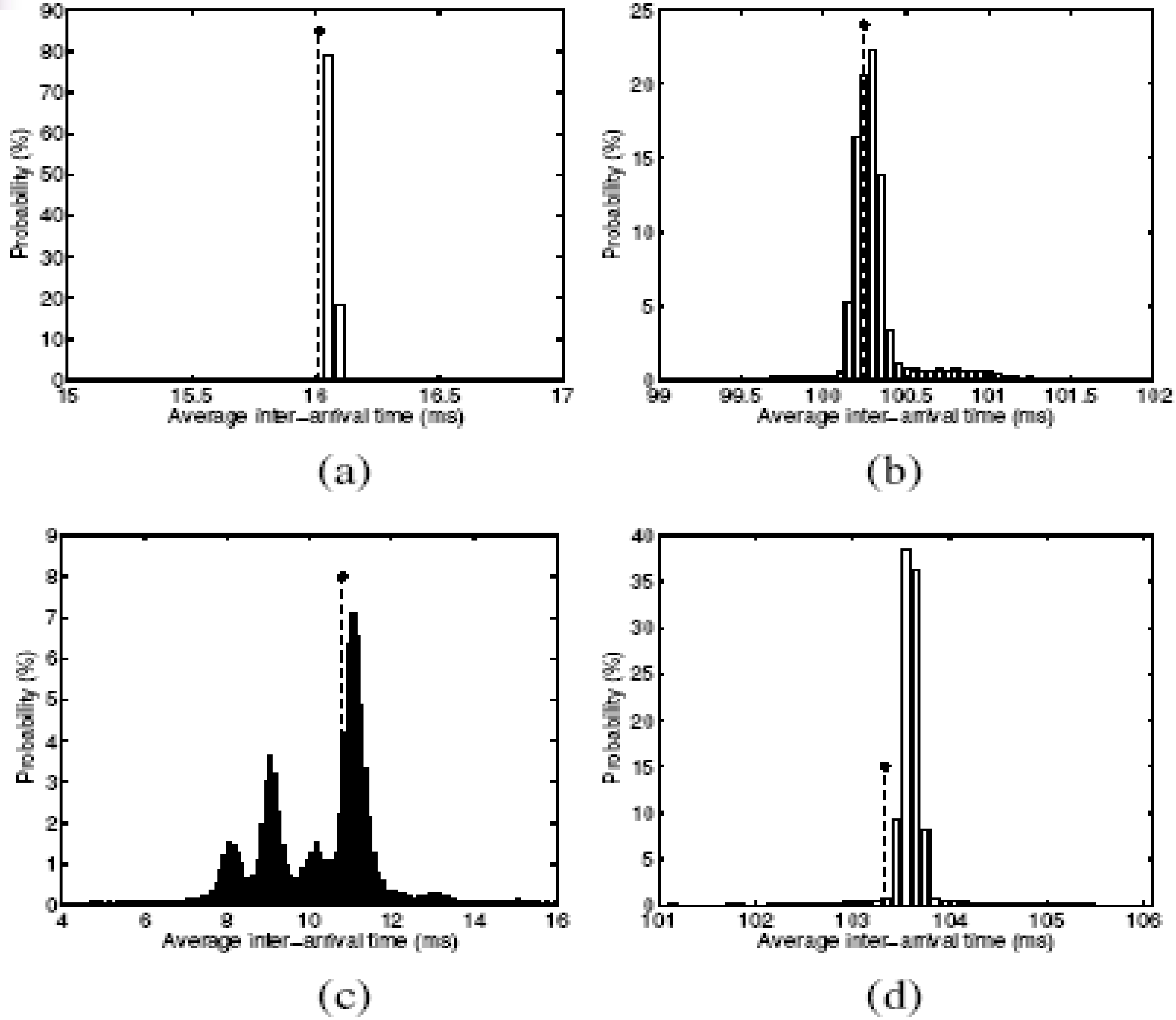
4. ED FD: Motivation

- Major drawbacks of Phi FD by… [18-19]:
  a) Normal distribution isn't good enough for…
  b) ED FD has higher slope than Phi FD;

- Our ED FD:
  - One implementation of an accrual FD
  - Inter-arrival time – Exponential distribution

4. ED-FD Motivation 1/2





Min~Max:

50 µs~time unit

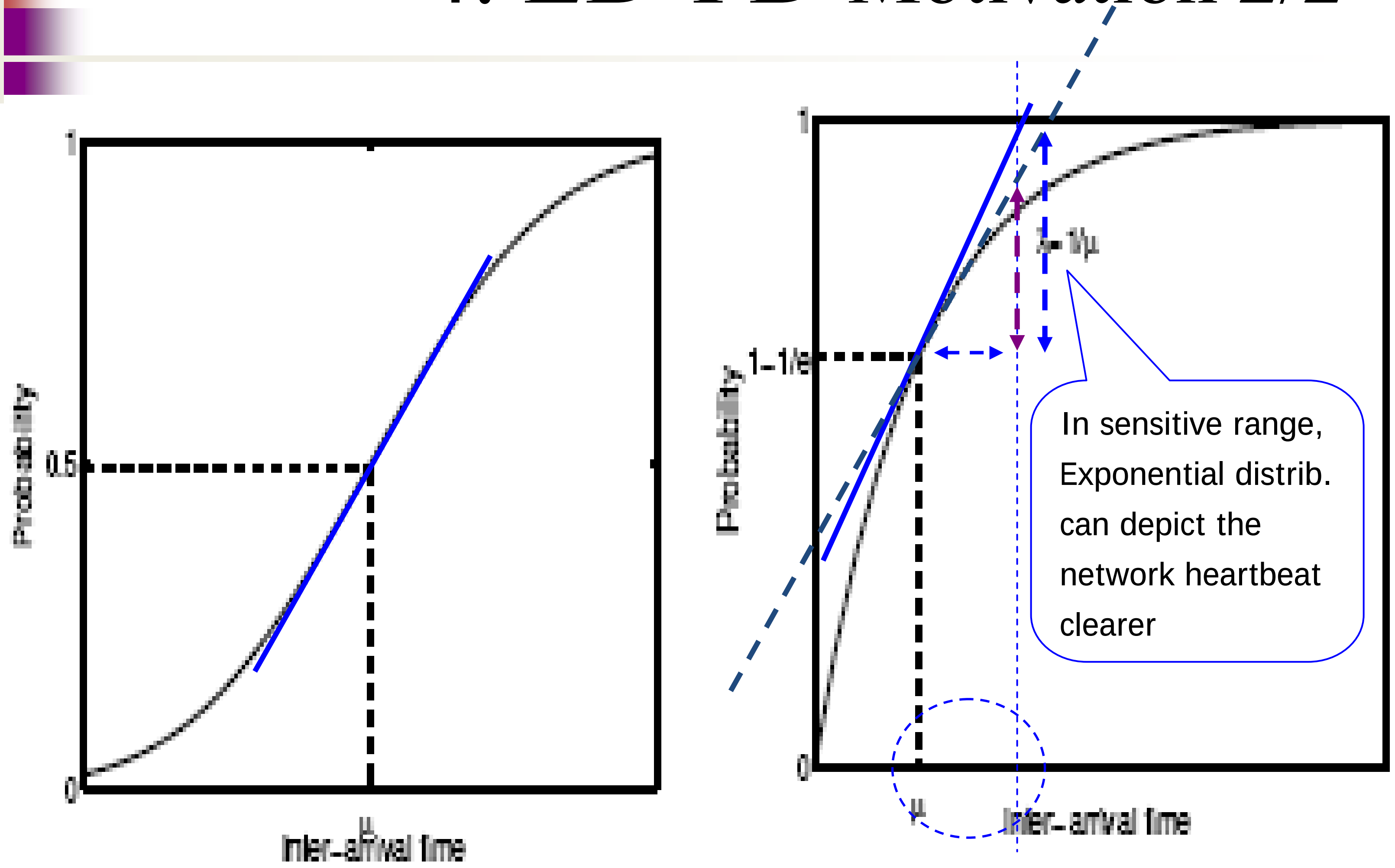
$n_1, n_2, \ldots, n_k$

$P_i = n_i / N_{sum}$

$P_i \sim i$

Statistics: (a) Cluster; (b) WiFi; (c) Wired LAN; (d) WAN (N_{unit}/N_{all})

4. ED-FD Motivation 2/2



Probability distribution vs. inter-arrival time: Phi FD [18]; ED FD (Normal distribution~ Exponential distribution, slope)

4. ED-FD basic principle

- **Basic principle:**

  Suspicion level is defined for accrual:

$$e_d(t_{now}) \stackrel{\text{def}}{=} F(t_{now} - t_{last}),$$

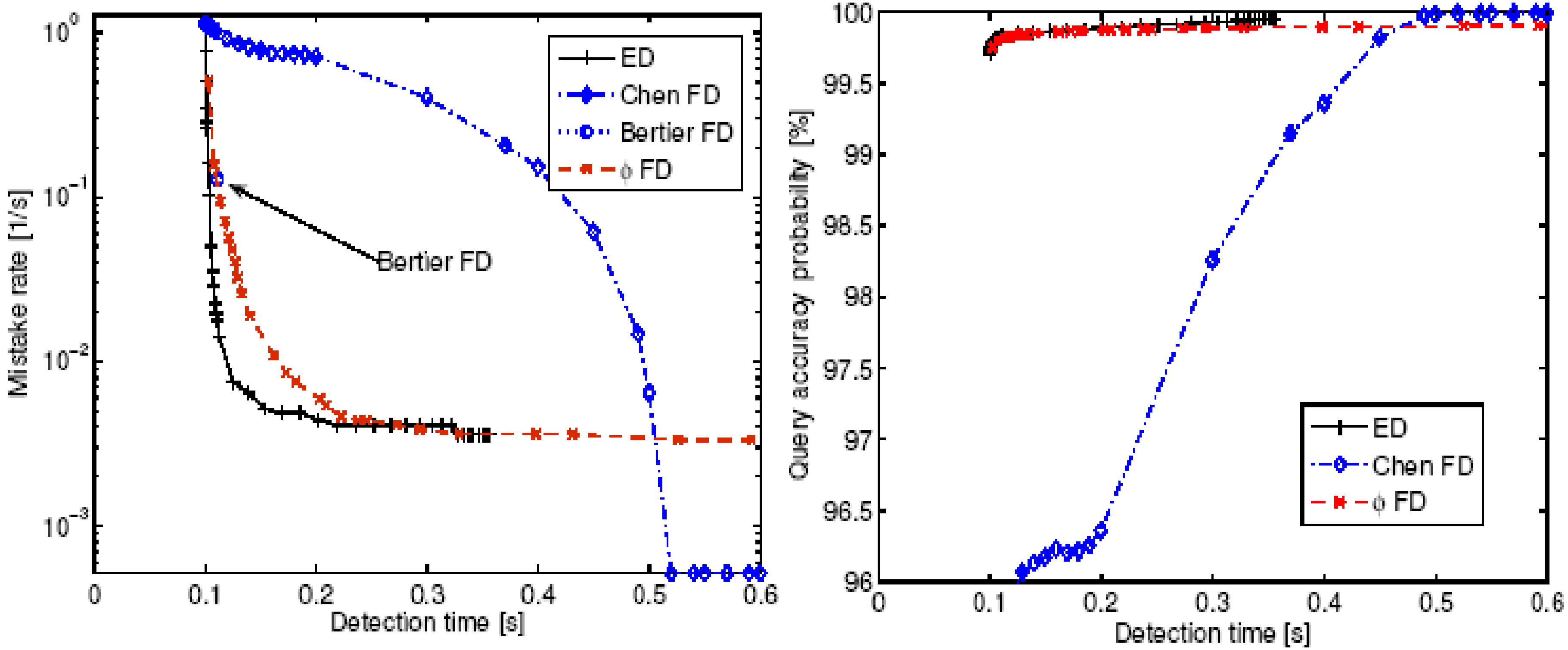
where the $F(t)$ is an exponential distribution function, and one has

$$F(t) = 1 - e^{-\lambda t},$$

where $t > 0$, and $\lambda = 1/\mu$

4. ED-FD Exp. Wireless1

- Experiment 1:

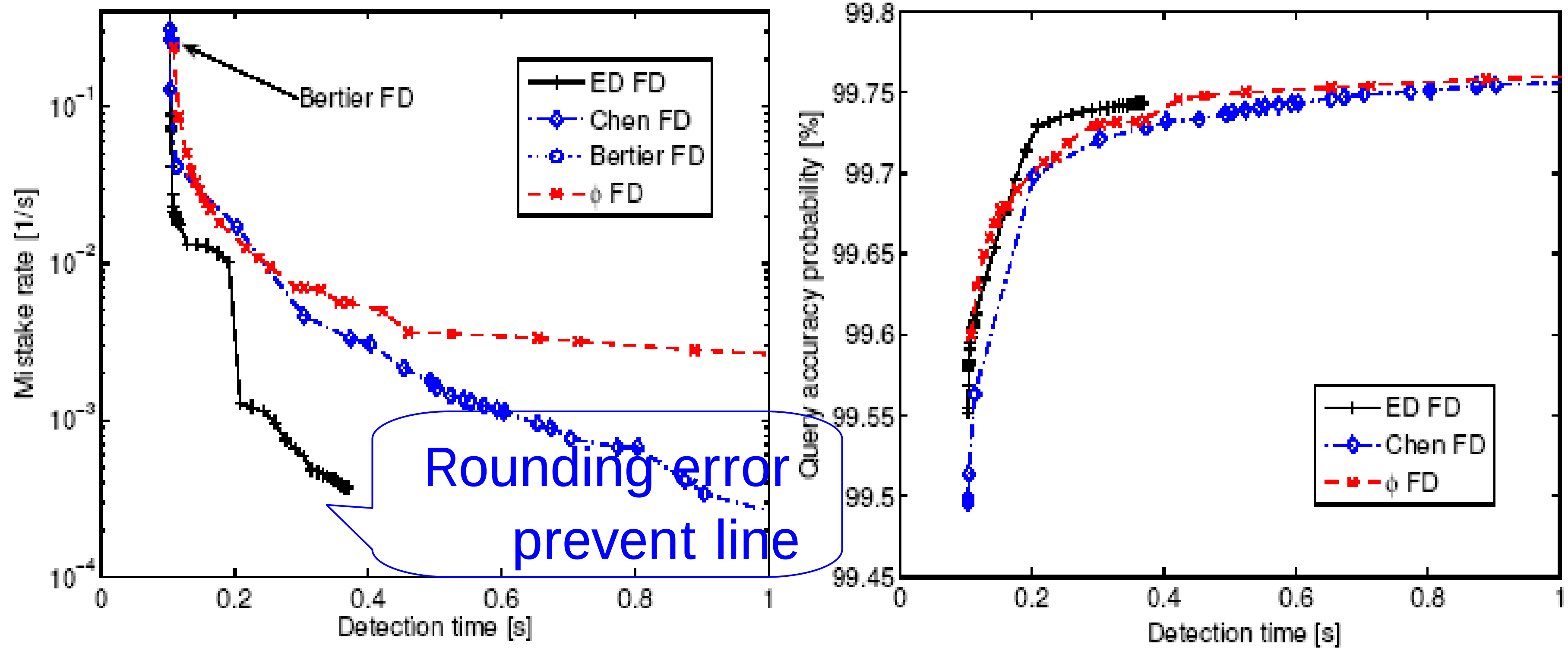


MR and QAP vs. DT comparison of FDs

in Wireless (logarithmic).

4. ED-FD Exp. WAN2

- Experiment 2:



MR and QAP comparison of FDs in WAN.

4. ED-FD Exp. WAN4

- **Results:**

- In the aggressive range of FD: ED FD behaves a little better than the other three FDs.

  (short DT, low MR and high QAP)

- It is obvious that the ED FD is more aggressive than Phi FD, and Phi FD is more aggressive than Chen FD.

Outline of failure detectors

- 1 Introduction
- 2 Existing Failure Detectors
- 3 Tuning adaptive margin FD (TAM FD)
- 4 Exponential distribution FD (ED FD)
- **5 Self-tuning FD (SFD)**

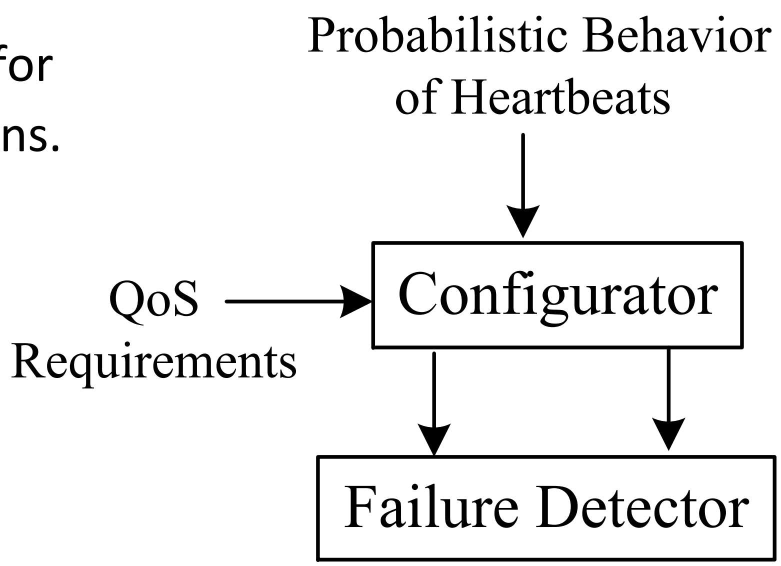
5. Self-tuning FD

- Users give target QoS, How to provide corresponding QoS?

Chen FD [30]

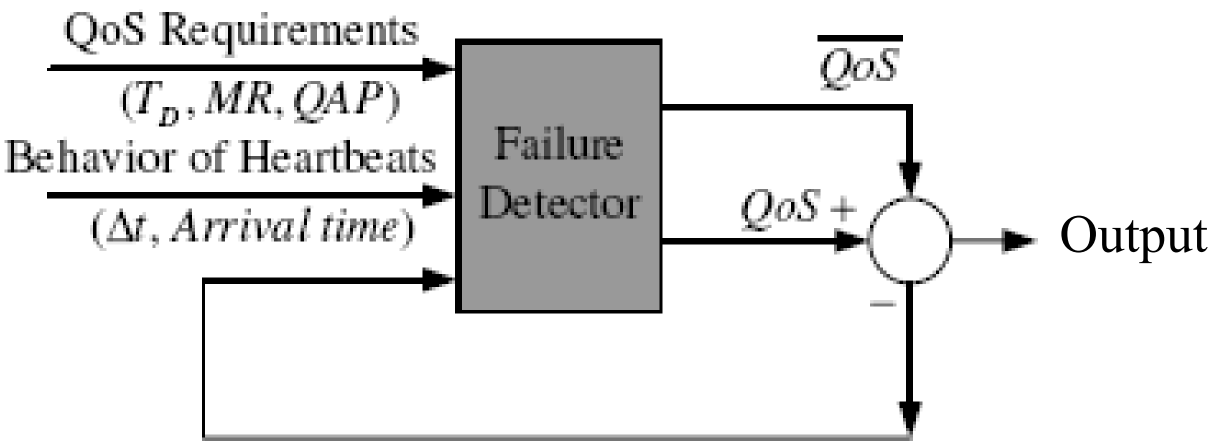
- Gives a list QoS services for users -- different parameters
- For certain QoS service -- match the QoS requirement
- Choose the corresponding parameters -- by hand.

Problem: it is not applicable for actual engineering applications.



5. Self-tuning FD

- Output QoS of FD does not satisfy target, the feedback information is returned to FD； -- parameters
- Eventually, FD can satisfy the target, if there is a certain field for FD, where FD can satisfy target
- Otherwise, FD give a response:

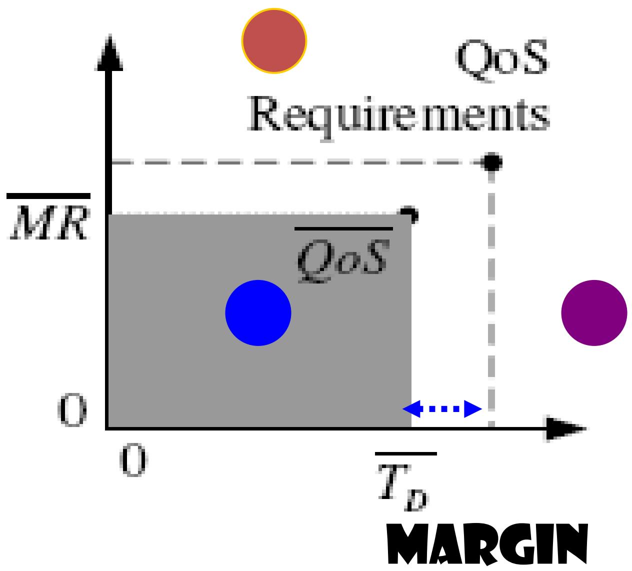


5. Self-tuning FD

- Basic scheme:

$$\tau_{(k+1)} = SM + EA_{(k+1)},$$

$$SM_{(k+1)} = SM_k + Sat_k\{QoS, \overline{QoS}\} \cdot \alpha,$$



Variables:

EA_{k+1}: theoretical arrival;

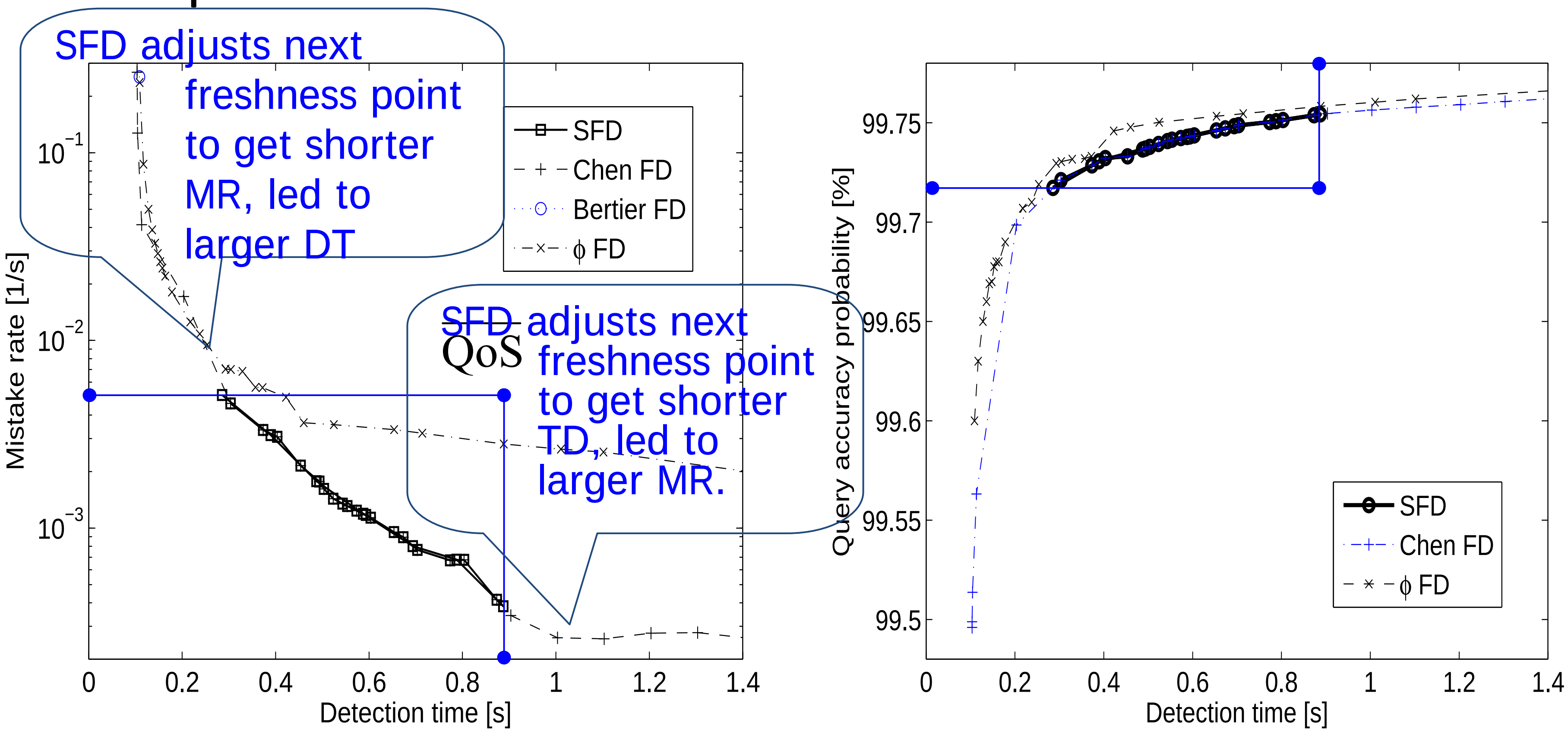
SM: safety margin;

τ_{k+1}: timeout delay;

α : a constant;

5. Self-tuning FD

- Experimental Results: WAN



MR and QAP comparison of FDs (logarithmic).

5. Self-tuning FD

- Experimental Results: WAN

- TD > 0.9, Chen-FD and Bertier-FD have longer TD and smaller MR.
- TD< 0.25, Chen-FD and Bertier-FD have shorter TD and larger MR.

- While, SFD adjusts the next freshness point $\tau_{(k+1)}$ to get shorter TD gradually --- it led to a little larger MR.

- So, SFD adjusts its parameters by itself to satisfy the target QoS.

Contributions

- **For FD (failure detector):**
  - 1 **Problems, Model, QoS of Failure Detectors**
  - 2 **Existing Failure Detectors**
  - 3 **Tuning adaptive margin FD (TAM FD, JSAC):**

    Constant safety margin of Chen FD [30]
  - 4 **Exponential distribution FD (ED FD, JSAC):**

    Normal Distribution in Phi FD [18-19]
  - 5 **Self-tuning FD (S FD, Sigcom10):**

    Self-tunes its parameters

Future Work

- Self-tuning FD;
- Indirection FD;
- New schemes: different Probability Distribution;
- New schemes: different architectures;
- FD-Network: dependable network software in cloud;
- Combining Scheduling and Fault-Detection

Q & A

Thank You!

$Ex' = \overline{X}$

Security and Trust Crisis in Cloud Computing

- Protecting datacenters must first secure cloud resources and uphold user privacy and data integrity.
- Trust overlay networks could be applied to build reputation systems for establishing the trust among interactive datacenters.
- A FD technique is suggested to protect shared data objects and massively distributed software modules.
- The new approach could be more cost-effective than using the traditional encryption and firewalls to secure the clouds.

Security and Trust Crisis in Cloud Computing

- Computing clouds are changing the whole IT , service industry, and global economy. Clearly, cloud computing demands ubiquity, efficiency, security, and trustworthiness.
- **Cloud computing has become a common practice in business, government, education, and entertainment leveraging 50 millions of servers** globally **installed at thousands of datacenters today.**
- **Private clouds will become widespread in addition to using a few public clouds, that are under heavy competition among Google, MS, Amazon, Intel, EMC, IBM, SGI, VMWare, Saleforce.com, etc.**
- **Effective reliable management, guaranteed security, user privacy, data integrity, mobility support, and copyright protection are crucial to the universal acceptance of cloud as a ubiquitous service.**

Content:

- Reliable, Performance Distributed file system
- Bandwidth to Data
  - Scan 100TB Datasets on 1000 node cluster
    - Remote storage @ 10MB/s = 165 mins
    - Local storage @ 50-200MB/s = 33-8 mins
  - Moving computation is more efficient than moving data
    - Need visibility into data placement

Scaling Reliably

- • Failure is not an option, it’s a rule !
  - 1000 nodes, MTBF < 1 day
  - 4000 disks, 8000 cores, 25 switches, 1000 NICs, 2000 DIMMS (16TB RAM)
- • Need fault tolerant store with reasonable availability guarantees
  - Handle hardware faults transparently

Hadoop Distributed File System (HDFS)

- • Data is organized into files and directories
- • Files are divided into uniform sized blocks (default 64MB) and distributed across cluster nodes
- • HDFS exposes block placement so that computation can be migrated to data

Problems of CPU-GPU Hybrid Clusters

- Scheduling Map tasks onto CPUs and GPUs efficiently is difficult
- Dependence on computational resource
  - # of CPU cores, GPUs, amount of memory, memory bandwidth, I/O bandwidth to storage
- Dependence on applications
  - GPU computation characteristic
    - Pros. Peak performance, memory bandwidth
    - Cons. Complex instructions

Hybrid Scheduling with CPUs and GPUs to make use of each excellence → Exploit computing resources