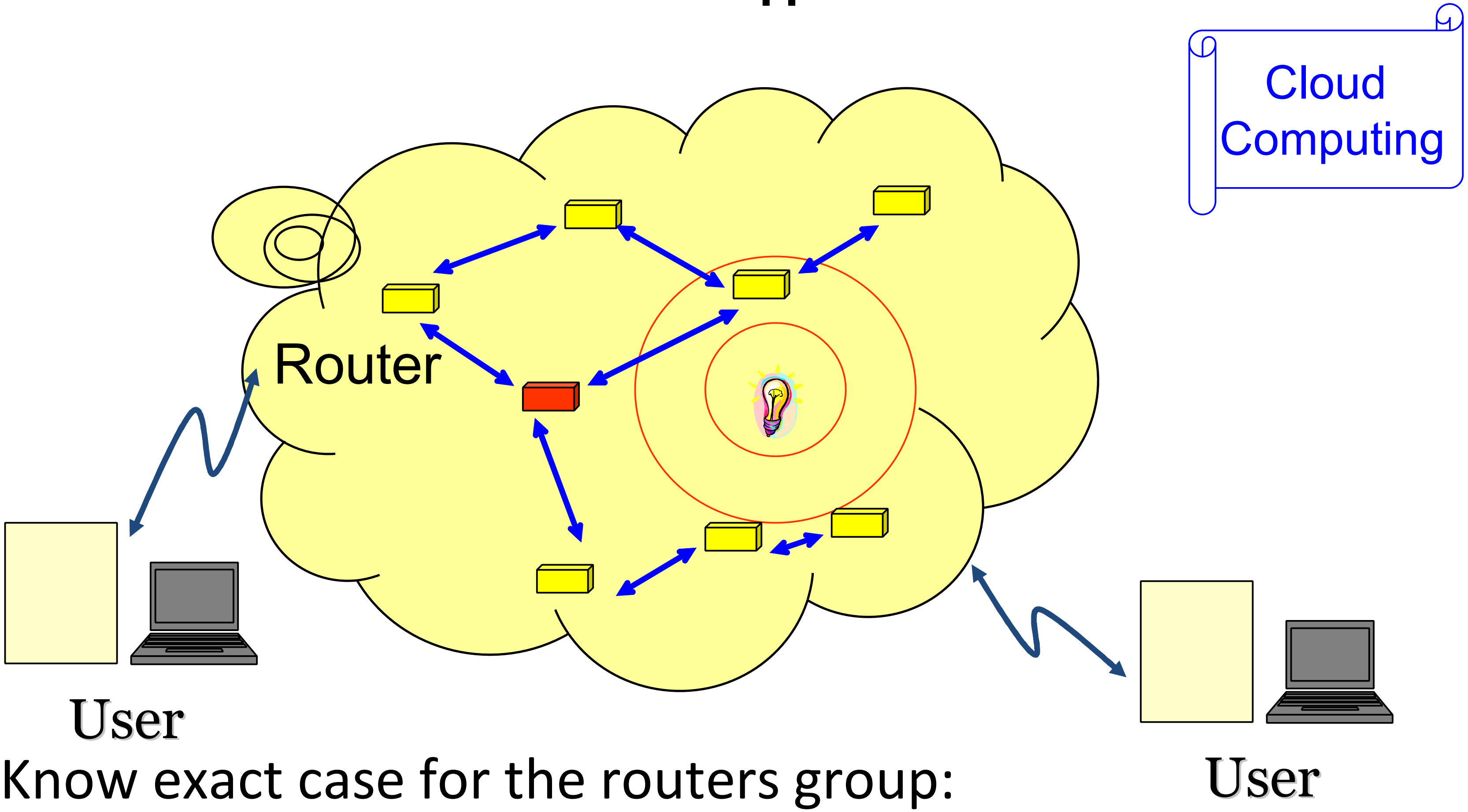
A Class of Practical Self-tuning Failure Detection Schemes for Distributed Networks

N. Xiong, A. V. Vasilakos, J. Wu, Y. R. Yang, A. Rindos, and Y. Par

Traditional network application



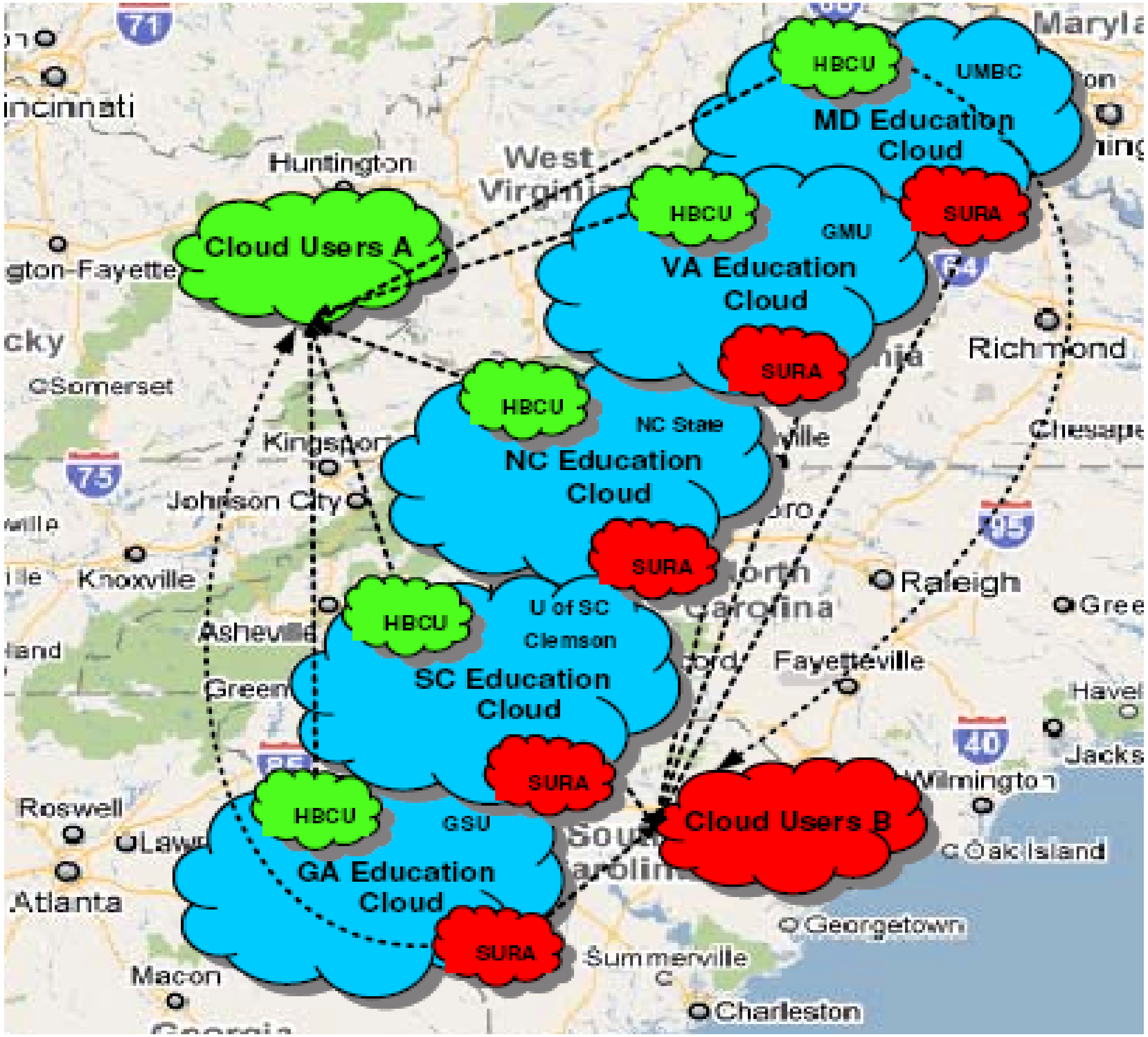
- Know exact case for the routers group:
  - If, good for packets transmission
  - Otherwise, miss packets, reduce QoS of packets transmission
  - Networks resource are not extensive shared (partly shared)

What is a cloud?

- Definition [Abadi 2009]
  - shift computer processing, storage, and software 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

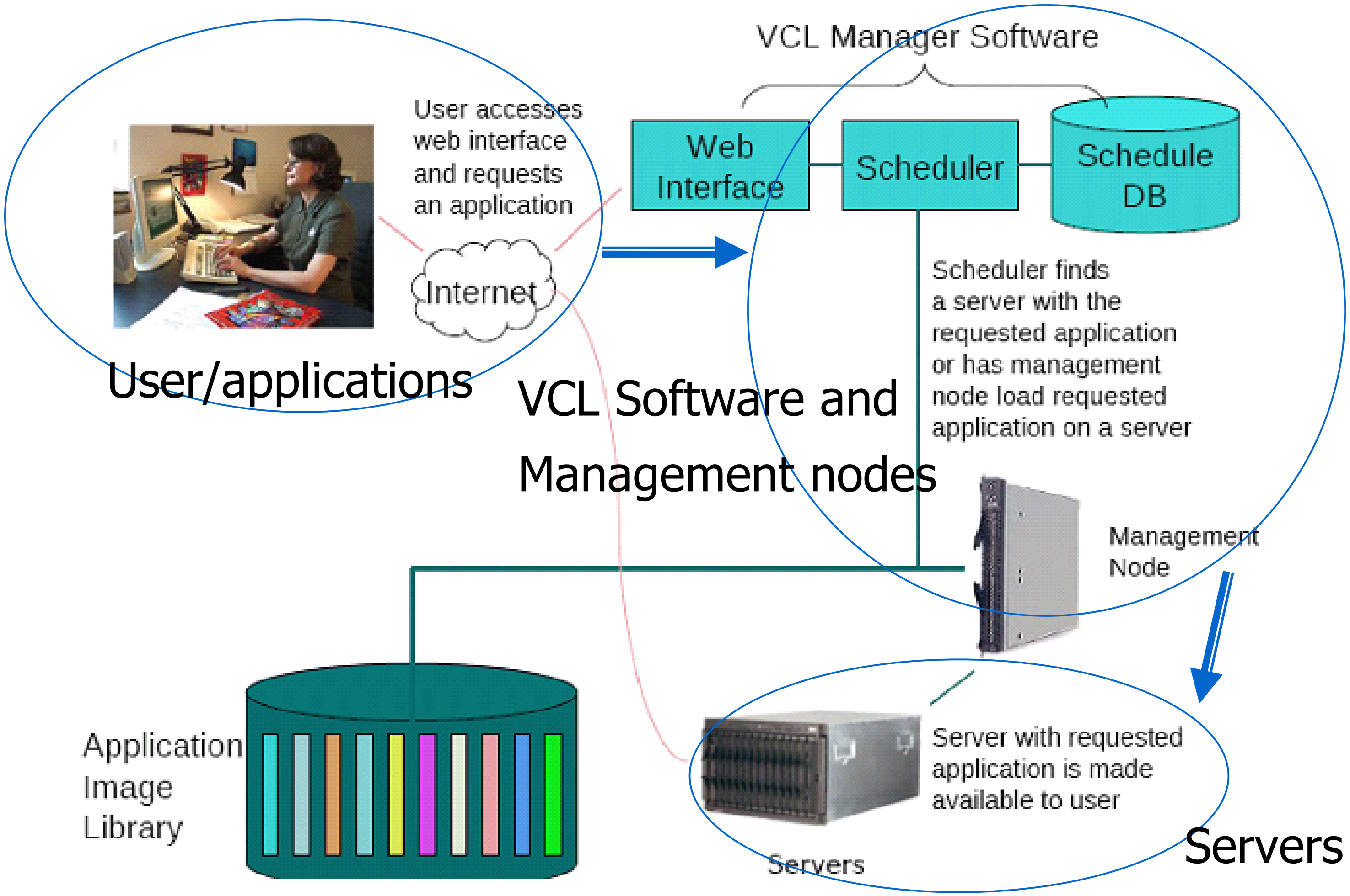
Dynamic cloud-based network model

Post Sec.



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

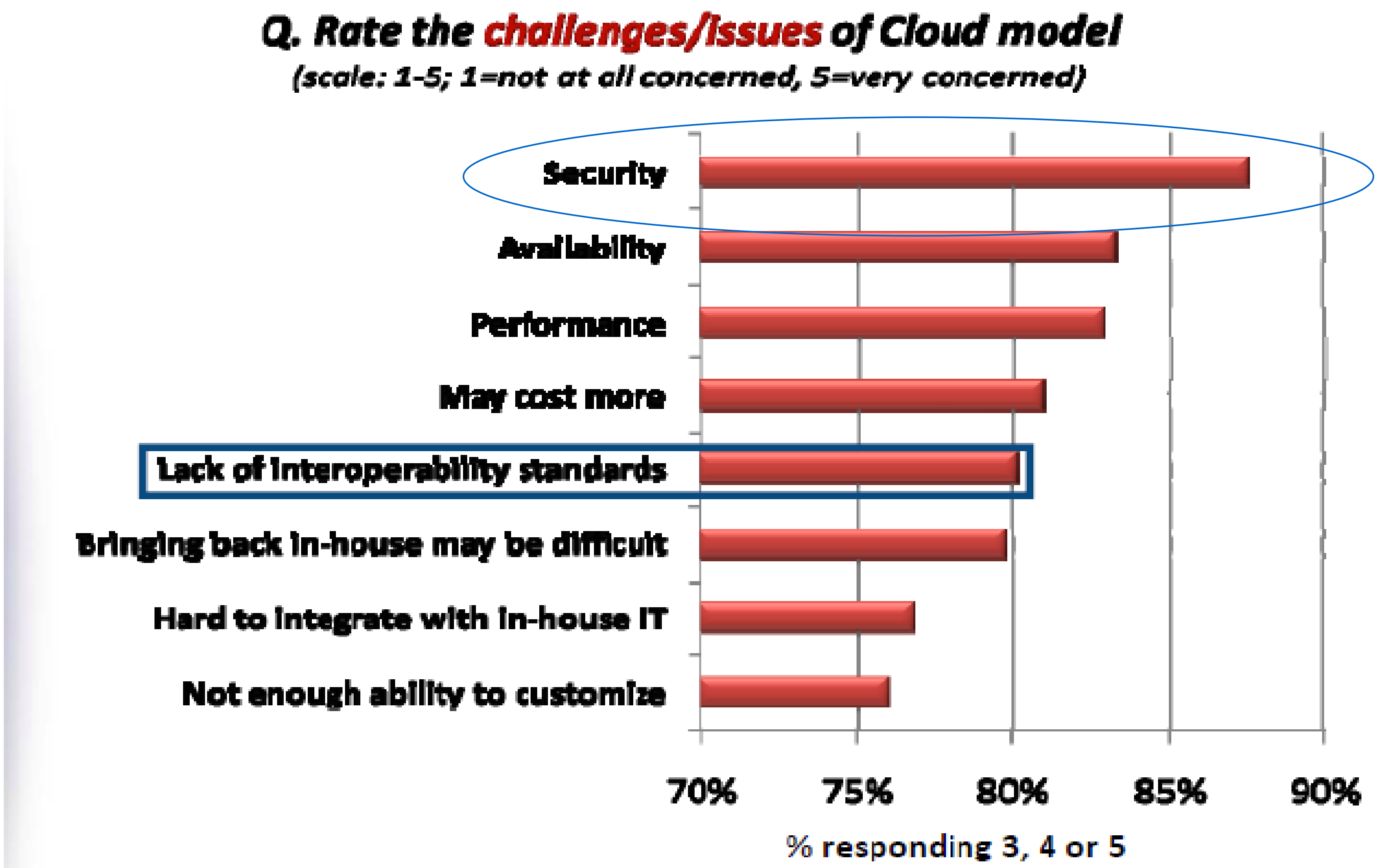
Dynamic cloud-based network model



North Carolina State University VCL model

http://vcl.ncsu.edu/

What's Worrisome about Cloud?



Source: IDC Enterprise Panel 3Q09 N=263

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

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

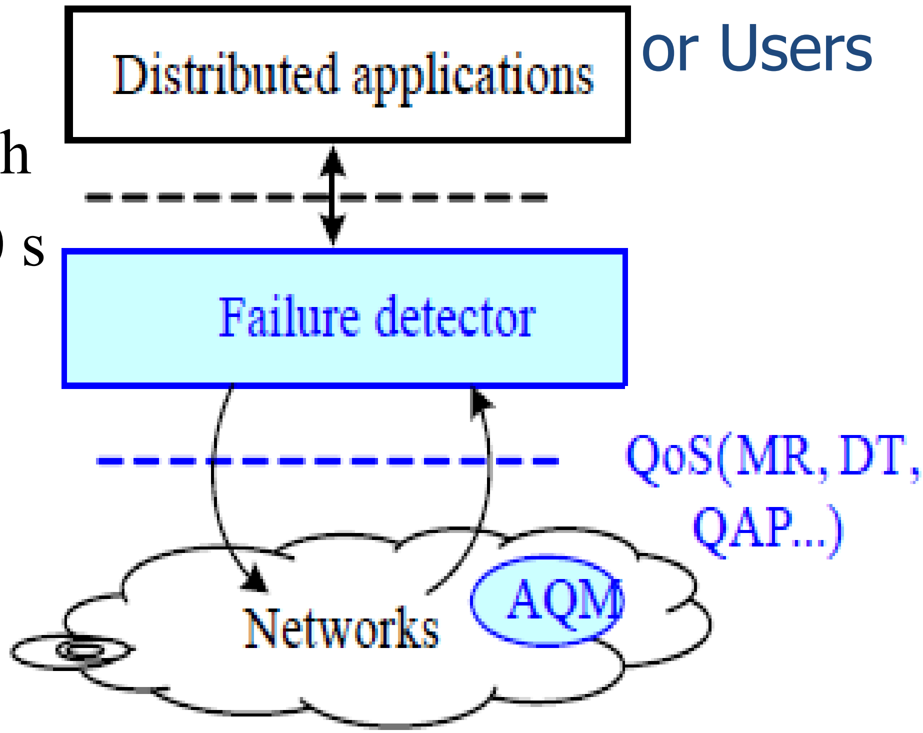
Algorithm (parameters):

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.

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

Failure Detectors (FDs): Outline

- **Problems, Model, QoS of Failure Detectors**
- **Existing Failure Detectors**
- **Self-tuning FD (S FD): IPDPS12, ToN**
  - Self-tunes its parameters

1. Outline of failure detectors

- **Introduction**
- **Existing Failure Detectors**
- **Self-tuning FD (S FD)**

Failure Detectors (FDs)

- *Importance of* FD :

  Fundamental issue for supporting dependability

  Bottleneck in providing service in node failure

- *Necessity*:

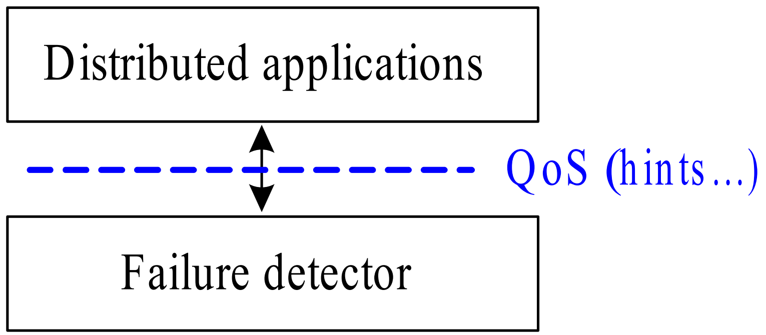
  To find an acceptable and optimized FD

Failure Detectors

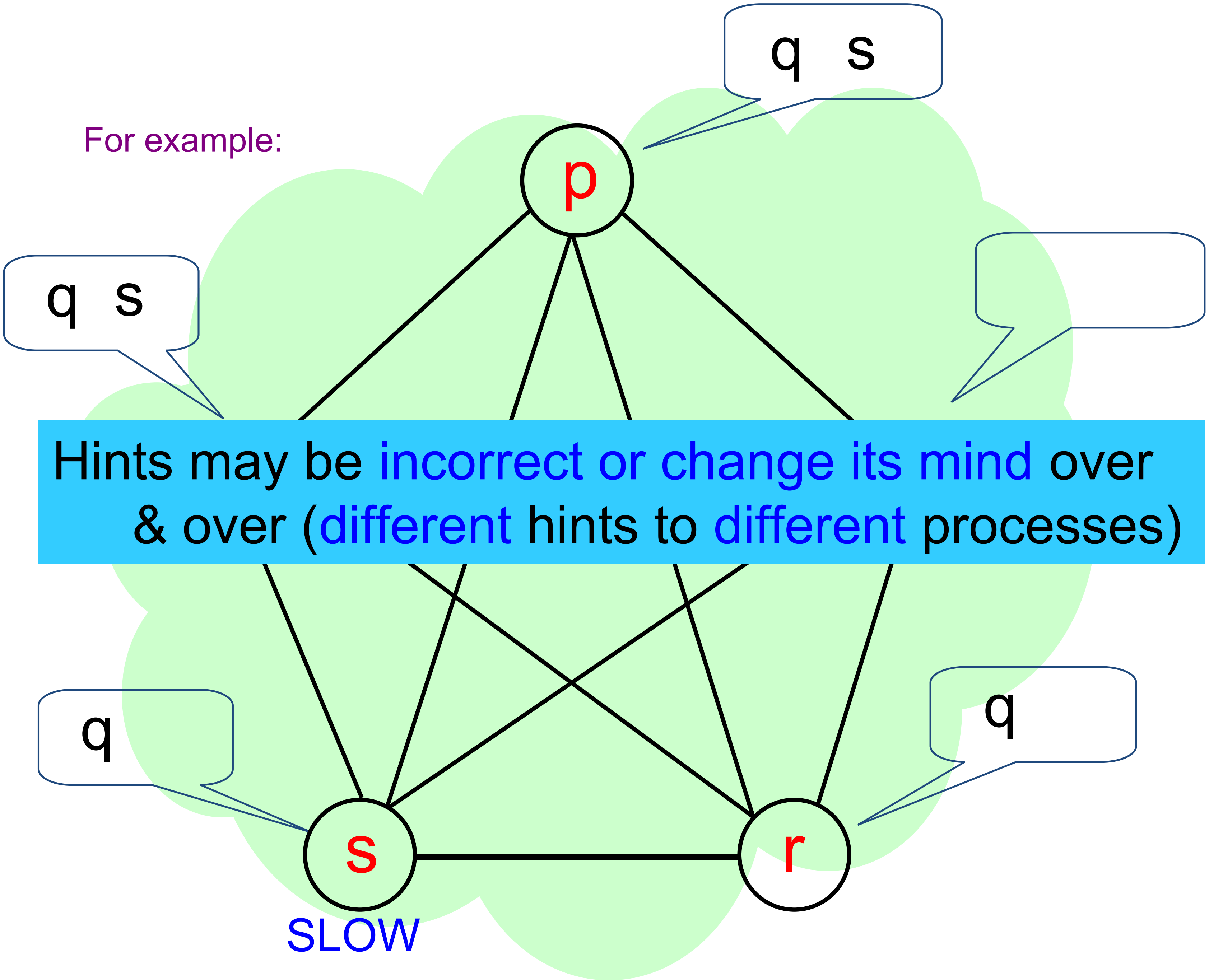
An FD is a distributed oracle that provides hints about the operational status of processes (Chandra-Toueg).

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:



Hints may be incorrect or change its mind over & over (different hints to different processes)

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):

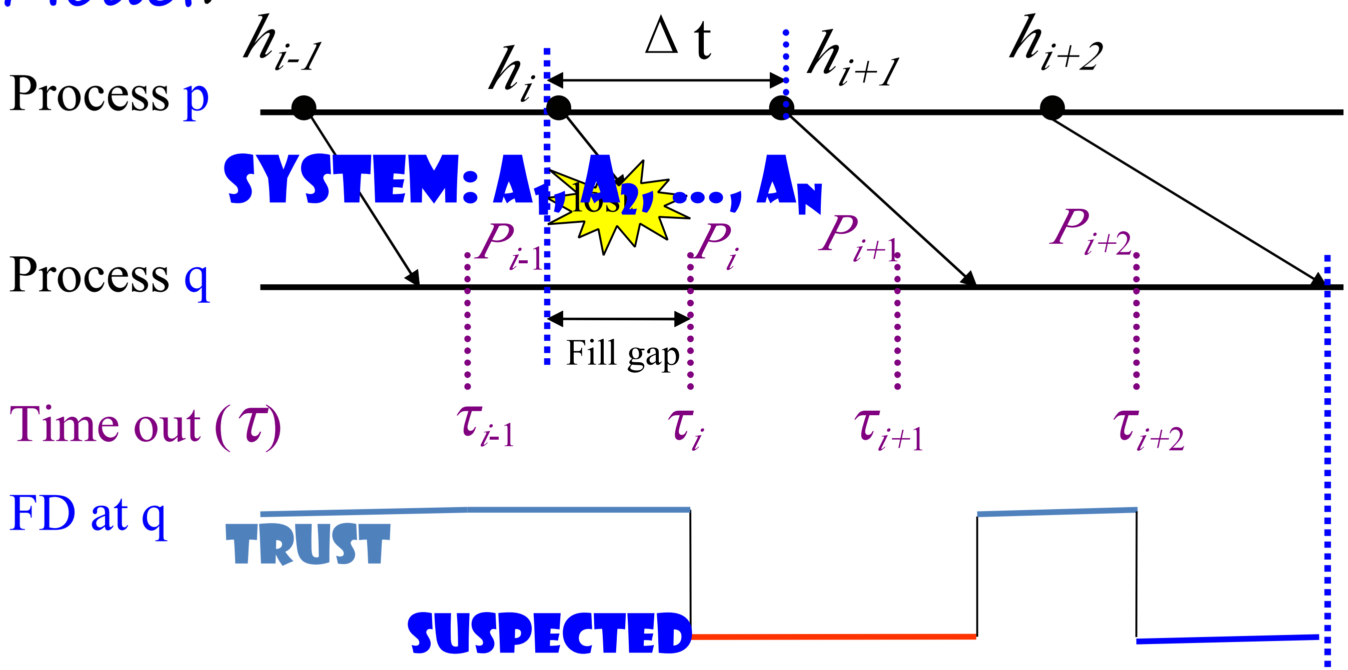
    Period from *p* starts crashing to *q* starts suspecting *p*
  - Mistake rate (MR):

    Number of false suspicions in a unit time
  - Query Accuracy Probability (QAP):

    Correct probability that process *p* is up

FD Problems, Model

- *Model*:



- *Problems*:
  - High probability of message loss, change topology
  - Difficult caused by unpredictability of network

2. Outline of failure detectors

- **Introduction**
- **Existing Failure Detectors**
- **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

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

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;	

[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.

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

2 Existing FDs: Kappa FD [3]

- Basic Kappa-FD scheme:

[3] N.Hayashibara. Accrual Failure Detectors. Doctoral Thesis, Japan Advanced Institute of Science and Technology, June, 2004.

- $T^i_{st} = EA_k + (i - k - 1) \cdot \Delta_i$
- $c^i(t) = c(t - T^i_{st})$
- $$c(t) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \int\limits_{-\infty}^{t} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases}$$
- $$sl_{qp}(t) = \kappa(t) = \sum_{i=k+1}^{\infty} c(t - T^i_{st})$$

Variables:

EA_k: expected time to arrive;

$T_{:}$ starting time;

Δ (t): sending interval;

c: the contribution;

Approximated from Norm. D.

Problem: How about the performance evaluation?

3. Outline of failure detectors

- Introduction
- Existing Failure Detectors
- **Self-tuning FD (S FD):**

  Self-tunes its parameters

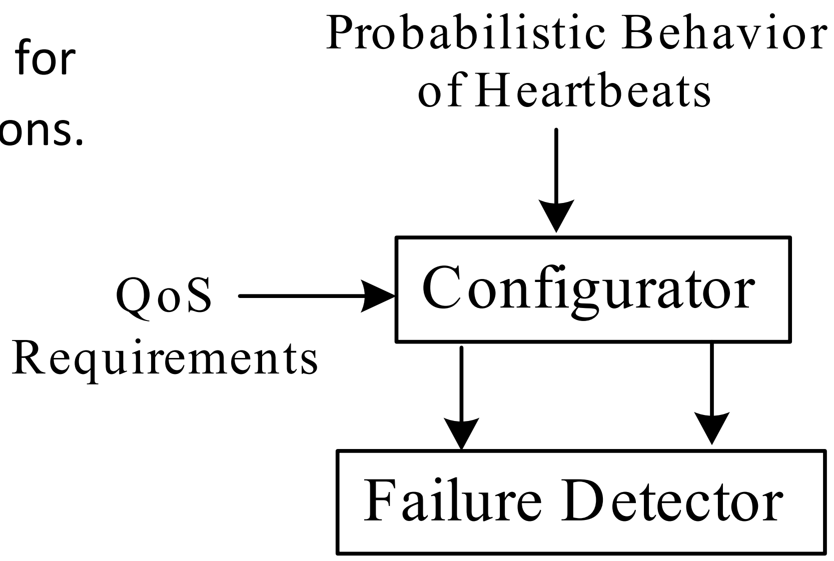
3. Self-tuning FD

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

Chen FD [30]

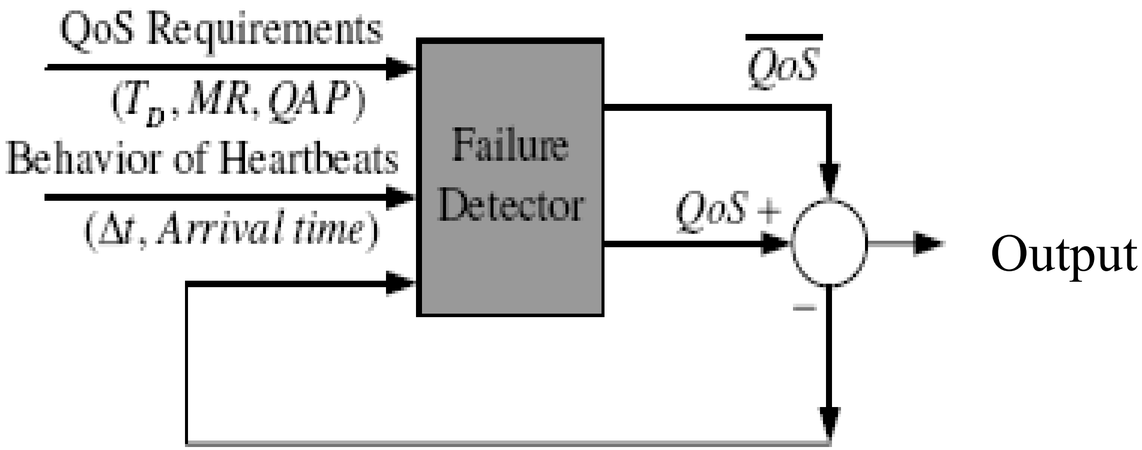
- Gives a list QoS services for users -- different parameters
- For certain QoS service -- match the QoS requirement
- Choose the corresponding parameters -- by hand.

Problem: it is not applicable for actual engineering applications.



3. 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:



How to design Self-tuning schemes to match it?

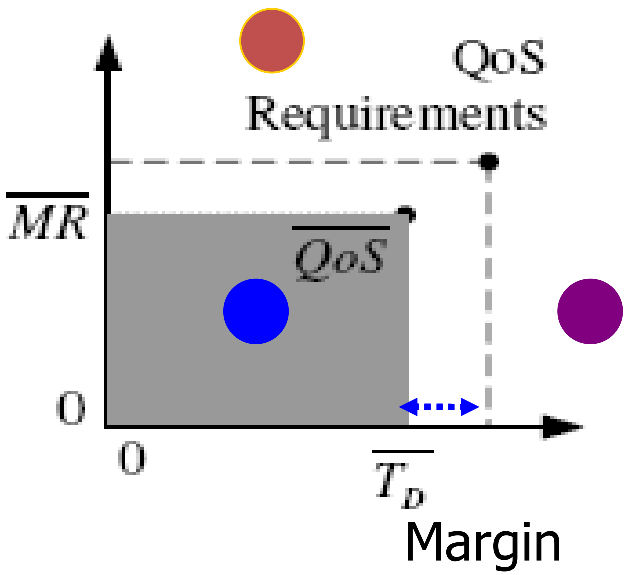
3. Self-tuning FD

- Basic scheme:

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

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

0;
>0;
<0;



Variables.:

EA_{k+1}: theoretical arrival;

SM: safety margin;

τ_{k+1}: timeout delay;

α.: a constant;

Experimental Environment

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

- S FD, Phi FD [18-19], Chen FD [30], and Bertier FD [16-17]
- Cluster, WiFi, LAN, WAN (USA-Japan, Germany-USA, Japan-Germany, Hongkong-USA, Hongkong-Germany)

Experiment Settings:

- For an arbitrarily long period (p-q)
- Without network breaking down
- Heartbeats UDP/IP
- CPU below the full capacity
- Logged heartbeat time
- Replayed the receiving time

… …

Exp. WAN (example)

- *WAN exp. Settings (USA-Japan):*

  - USA: planet1.scs.stanford.edu (p);
    Japan: planetlab-03.naist.ac.jp (q)

  - HB sampling (over one week)
    - Sending 6,737,054 samples;
    - Loss rate 0 %;
    - Ave. sending interval: 12.825 ms;
    - Ave. RTT: 193.909 ms;

    …

S FD: Settings

- Conducted on PlanetLab http://www.planet-lab.org/
- Nodes in USA, Europe (Germany), Japan, HongKong

- *WAN: Locations and hostnames*

Sender		Receiver	
country	hostname	country	hostname
USA	planet1.scs.stanford.edu	Japan	planetlab-03.naist.ac.jp
Germany	planetlab-2.fokus.fraunhofer.de	USA	planet1.scs.stanford.edu
Japan	planetlab-03.naist.ac.jp	Germany	planetlab-2.fokus.fraunhofer.de
China	planetlab2.ie.cuhk.edu.hk	USA	planet1.scs.stanford.edu
China	planetlab2.ie.cuhk.edu.hk	Germany	planetlab-2.fokus.fraunhofer.de

S FD: Statistics

- Statistics of Cluster, LAN, WiFi, WAN:

	Heartbeats		Heartbeat period			RTT
	total (♯msg)	loss rate	send (mean)	recv (mean)	recv (stddev)	(avg.)
WAN-1	6,737,054	0%	12.825 ms	12.83 ms	14.892 ms	193.909 ms
WAN-5	7,008,170	4%	12.367 ms	12.94 ms	16.557 ms	362.423 ms

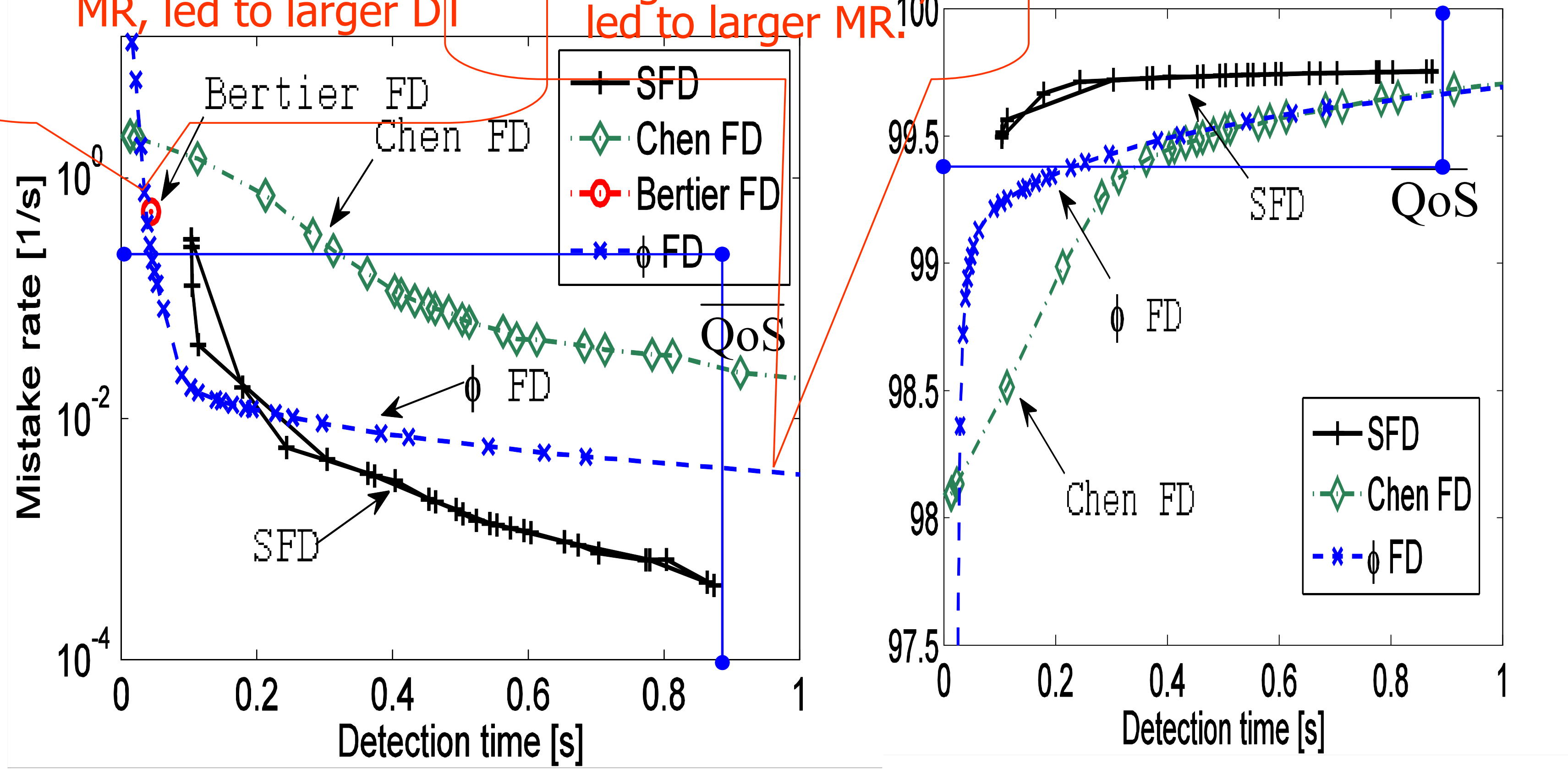
WAN-1: USA-Japan; WAN-2: Germany-USA;
WAN-3: Japan-Ger.; WAN-4: HK-USA; WAN-5: HK-Germany

WAN-1: USA-Japan; WAN-5: HK-Germany

3 Self-tuning FD

SFD adjusts next freshness point to get shorter MR, led to larger DT

SFD adjusts next freshness point to get shorter TD, led to larger MR.



MR and QAP comparison of FDs (logarithmic).

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 $\tau(k+1)$ t shorter TD gradually --- it led to a little larger MR.

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

Future Work for FD

- Self-tuning FD;
- Indirection FD;
- New schemes: different Probability Distribution;
- New schemes: different architectures;
- FD-Network: dependable network software in cloud;

Thank you

Neal Naixue Xiong

AP, in CS&IST, Georgia State University

Georgia State University | IS& T INFORMATION SYSTEMS AND TECHNOLOGY