Fuzzy Congestion Control in Communications

Chrysostomos Chrysostomou, Andreas Pitsillides, Vasos Vassiliou

Department of Computer Science

University of Cyprus

04-07-2007

http://www.NetRL.cs.ucy.ac.cy

Talk outline

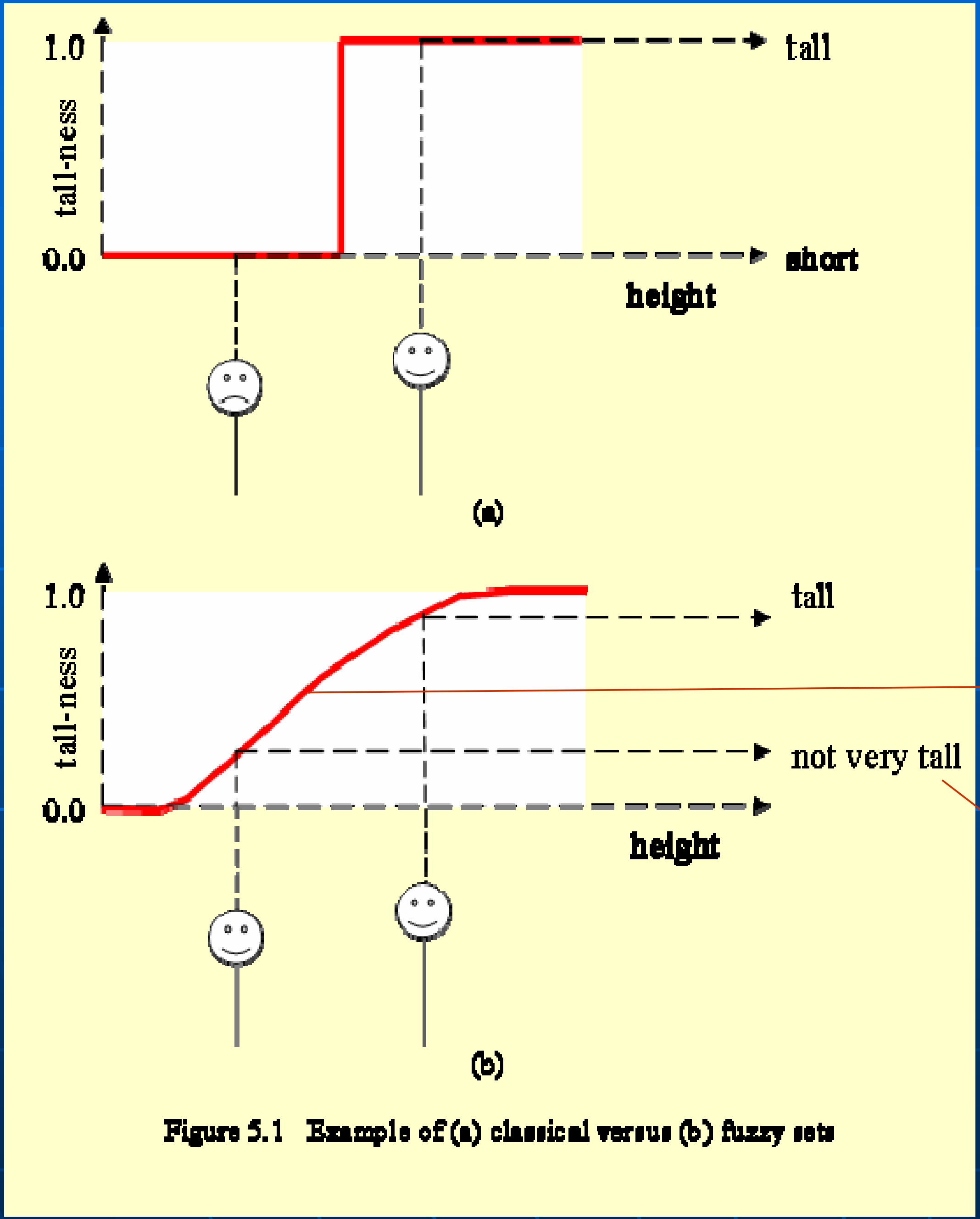
- Fuzzy Logic introduction
- Congestion Control in Communication Networks
- Two illustrative examples of our approach: FEM/FIO and ADIVIS
- Concluding remarks and future work
- Open discussion

Fuzzy Logic

- One of the tools of what is commonly known as Computational Intelligence.
- An extension and generalization of multivalued logic systems.
- It provides a set of mathematical methods for representing information in a way that resembles natural human communication, and for handling this information in a way that is similar to human reasoning.
- By using FL, a designer is able to blend qualitative linguistic expressions favored by human experts in the structure of control systems.

Fuzzy Sets

- Fuzzy set theory (Zadeh, 1965) and fuzzy inference are used to derive control laws.
- A fuzzy set is defined by a membership function that can be any real number in the interval [0, 1], expressing the grade of membership for which an element belongs to that fuzzy set.
  - *The truth of any statement becomes a matter of degree.*
- The concept of fuzzy sets enables the use of fuzzy inference, which in turn uses the knowledge of an expert in a field of application to construct a set of IF-THEN rules.



A fuzzy set can contain elements with only a partial degree of membership.

The degree of membership is changed gradually from *false (0)* to *true (1)*, rather than abruptly.

The characteristic function that defines the fuzzy set is known as a *membership function*.

Any value of height has a *partial membership* in the corresponding fuzzy set.

Fuzzy Membership Functions

- In fuzzy logic theory, the range of values for a given input or output space is often called the *universe of discourse*.
- A *membership function* is a curve that defines how each point in the *universe of discourse* is mapped to a membership value (or degree of membership) between 0 and 1.
- There are various types of membership functions (triangular, trapezoidal, Gaussian and bell-shaped curves, etc).

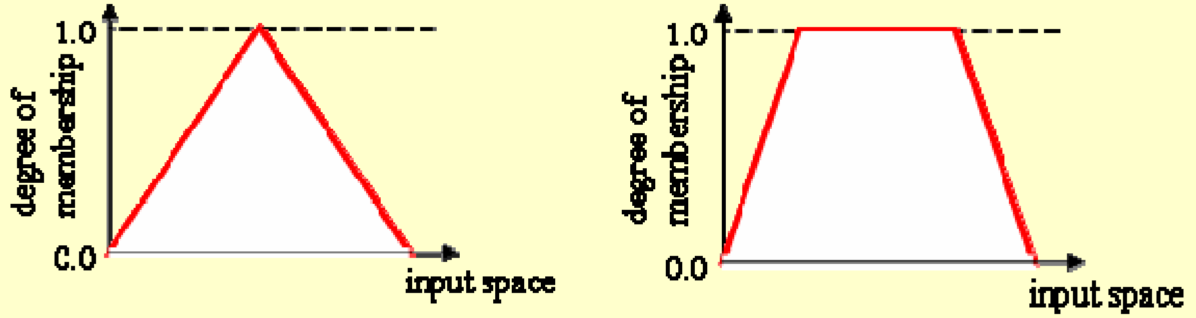


Figure 5.2 Triangular-type membership function of a fuzzy set

Figure 5.3 Trapezoidal-type membership function of a fuzzy set

Fuzzy Logical Operators

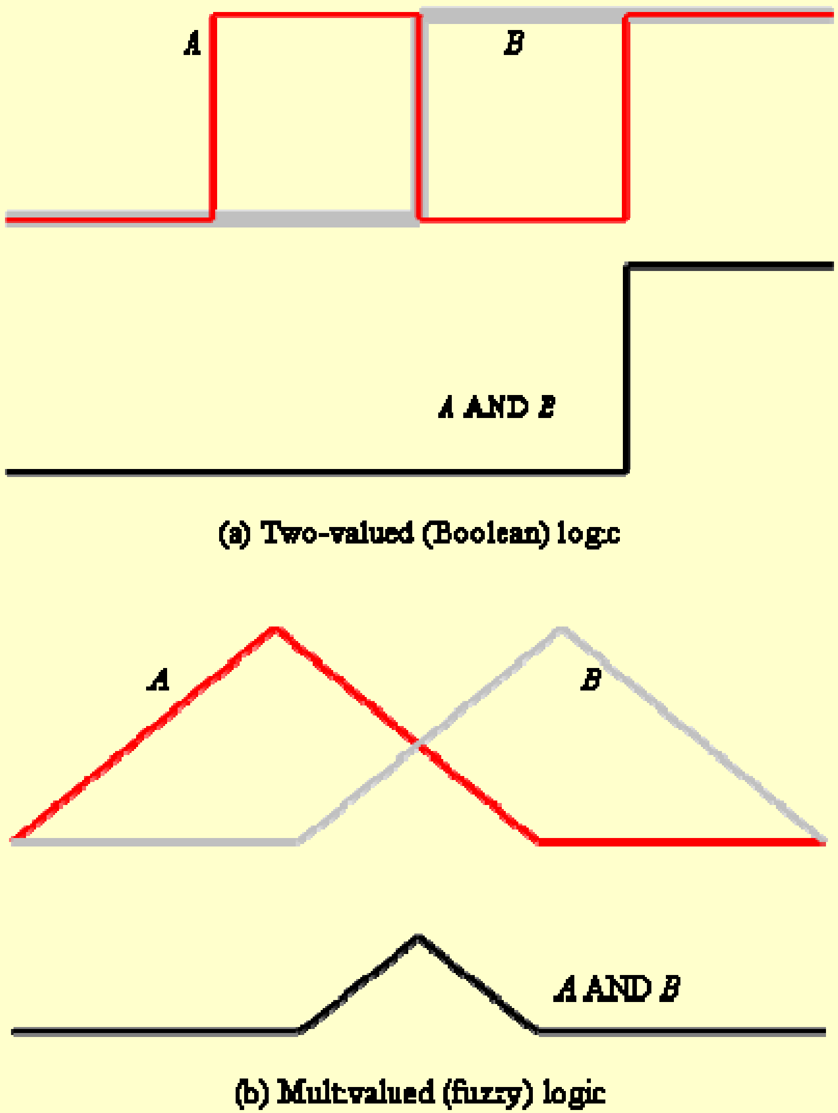


Figure 5.4 Intersection logical operation

- Three basic operations fulfil the needs of most typical fuzzy logic based systems.
  - Fuzzy Intersection (AND):
    - The *min* operator represents the intersection of two fuzzy sets A and B.
    - The elements of A and B are operated one-by-one and the minimum is taken as the output.

Fuzzy Logical Operators (cnt'd)

- Fuzzy Union (OR):
  - The *max* operator represents the union of two fuzzy sets A and B.
  - The elements of A and B are operated one-by-one and the maximum is taken as the output.
- Fuzzy Complement (NOT):
  - The complement of the membership function of a fuzzy set for each element is taken as the output.

- Other logical operations can also be defined.

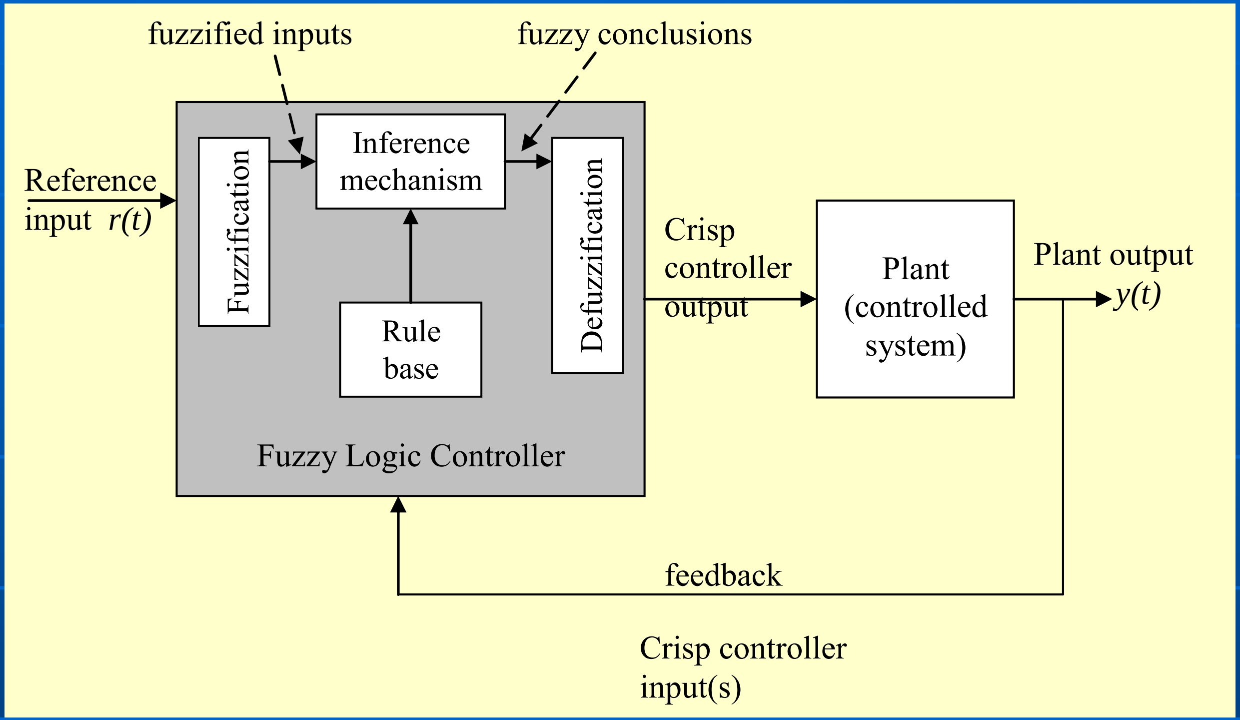
IF-THEN Rules

- Fuzzy sets and fuzzy logic operators are the so-called "subjects" and "verbs" of fuzzy logic, respectively.
- IF-THEN rule statements are used to formulate the conditional statements that comprise fuzzy logic.

IF x is A and y is B THEN z is C

- *A*, *B*, and *C* are *linguistic values* defined by fuzzy sets on the ranges *X*, *Y*, and *Z*, respectively, and represented by certain membership functions.
- These linguistic values are part of their corresponding *linguistic variables*, and define the partitions (operating regions) over the input/output space.
- A linguistic variable (describing the input/output, e.g. *x*, *y*, *z*) takes *words* as values.
- Ex.: linguistic variable called "height", would assume linguistically values like "short", "tall", "very tall", etc.
- The IF-parts of the rule statement are called the *antecedents* (or *premises*), while the THEN-part is called the *consequence.*
- The antecedent as well as the consequence of a rule can have multiple parts.

Fuzzy Logic Control System



- Fuzzy logic control system design essentially amounts to
  (1) choosing the fuzzy logic controller input(s) and output(s),
  (2) choosing the preprocessing that is needed for the controller input(s) and possibly postprocessing that is needed for the output(s) (i.e, normalisation of the input and output values),
  (3) designing each of the four components of the fuzzy logic controller

Rule Base

- It holds the knowledge of how best to control the system in the form of a set of IF-THEN rules.
  - Specifies the system behaviour against the input variables.
- A fuzzy system performs reasoning on every rule in this rule base toward a final inference.
- The operations performed on these rules are simple, which is advantageous regarding computational processing.
- Ex.:

  if queue is moderate and rate_of_change_of_queue is zero then drop_probability is low

  if queue is moderate and rate_of_change_of_queue is increasing then drop_probability is medium

  if queue is full and rate_of_change_of_queue is increasing then drop_probability is high

Inference Mechanism

- It emulates the expert's decision making in interpreting and applying knowledge about how best to control the plant.
- It basically evaluates which control rules are relevant at the current time, and then decides what the input to the controlled system should be.

Fuzzification of the Input Variables

- It converts the inputs into information that the inference mechanism can use to activate and apply rules.
- It determines the degree to which the inputs belong to each of the appropriate fuzzy sets via their membership functions.
- Each input is a crisp numerical value limited to the universe of discourse of the associated input linguistic variable, and the output is a fuzzy degree of membership in the qualifying linguistic set (always in the interval between 0 and 1).
    - that is, each input is "fuzzified" over all the qualifying membership functions required by the rules.
- Once the inputs have been fuzzified, we know the degree to which each part of the antecedent has been satisfied for each rule.

Fuzzification of the Input Variables (cnt'd)

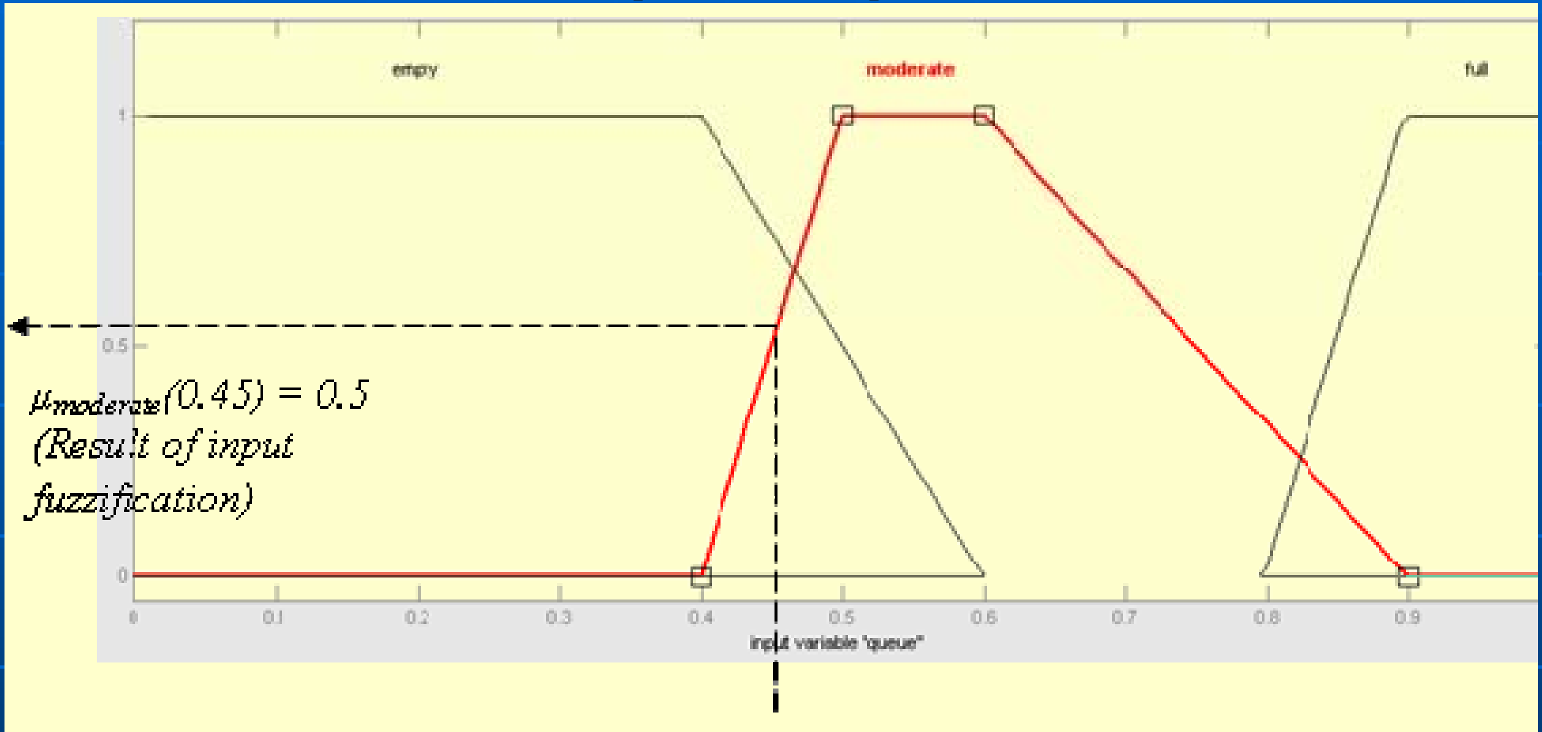


Figure 5.7 Fuzzification of the input variable

Given the graphical definition of "moderate queue", the "queue" being rated as 0.45, corresponds to $\mu_{moderate}(0.45) = 0.5$ for the "moderate queue" membership function. For such input value of queue, the corresponding degree of membership for the "empty queue" value is $\mu_{empty}(0.45) = 0.8$, and for the "full queue" is $\mu_{full}(0.45) = 0.0$

Application of Fuzzy Operators

- If the antecedent of a given rule has more than one part, the fuzzy operator used in the IF-part of the rule is applied to obtain one number that represents the result of the antecedent for that rule (in the interval between 0 and 1).
- The input to the fuzzy operator is the membership values derived from the fuzzification of the input variables.
- The output is a single truth value.
  - This is the *degree of support for the rule*.
- Choosing the *min-operation*, indicates that *we can be no more sure about the conjunction of the two statements in the IF-part, than we are about the individual terms that make them up.*

Application of Fuzzy Operators (cnt'd)

if queue is moderate and rate_of_change_of_queue is zero then drop_probability is low

$$\mu^{queue}_{moderate}(0.45) = 0.5\,,\ \mu^{rate\_queue}_{zero}(0.43) = 0.8$$

$$\mu_{antecedent} = \min\left(\mu^{queue}_{moderate}(0.45),\ \mu^{rate\_queue}_{zero}(0.43)\right) = 0.5 \qquad (5.6)$$

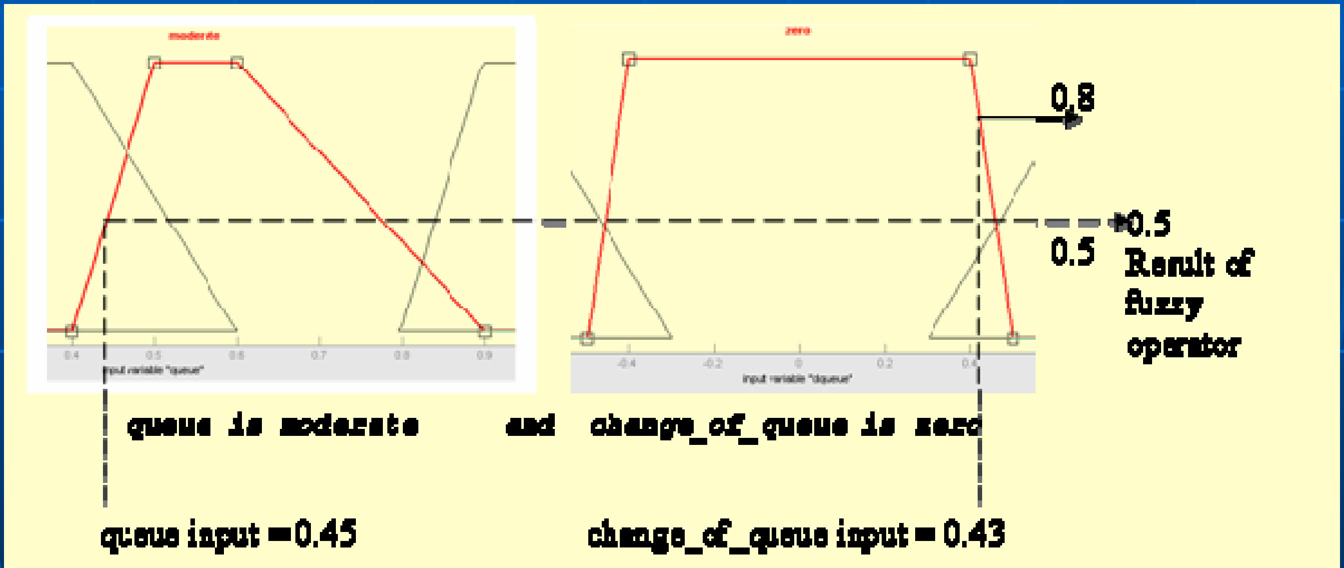


Figure 5.8 Application of the Fuzzy Operator AND (min)

we are 0.5 (or 50%) certain that this rule applies to the current situation.
The rule indicates that if its antecedent part has a degree of truth (certainty) then the action indicated by its consequent part should be taken

Implication

- The implication method shapes the consequent (the output fuzzy set) of a particular rule on the basis of the antecedent.
- Implication is implemented for each rule.
- The input of the implication method is a single number given by the antecedent, and the output is a fuzzy set.
- The most popular, well-known implication method, which is also used by the fuzzy logic AND operator, is the *min* (minimum) that *truncates* the output fuzzy set.
- The justification of using the *minimum* operator to represent the implication is that *we can be no more certain about our consequent than our antecedent.*
- All possible values in the universe of discourse of the specific output fuzzy set are compared with the generated antecedent single value, and the minimum is taken.

Implication (cnt'd)

$$\mu_{implied}(x_i) = \min\left(\mu_{antecedent}, \mu_{consequent}(x_i)\right)$$

a time-varying function that quantifies how certain the specific rule is that the output of the fuzzy logic system should take on certain values.

It has a certain degree of truth that the output of the fuzzy system should lie in a region around *low* values.

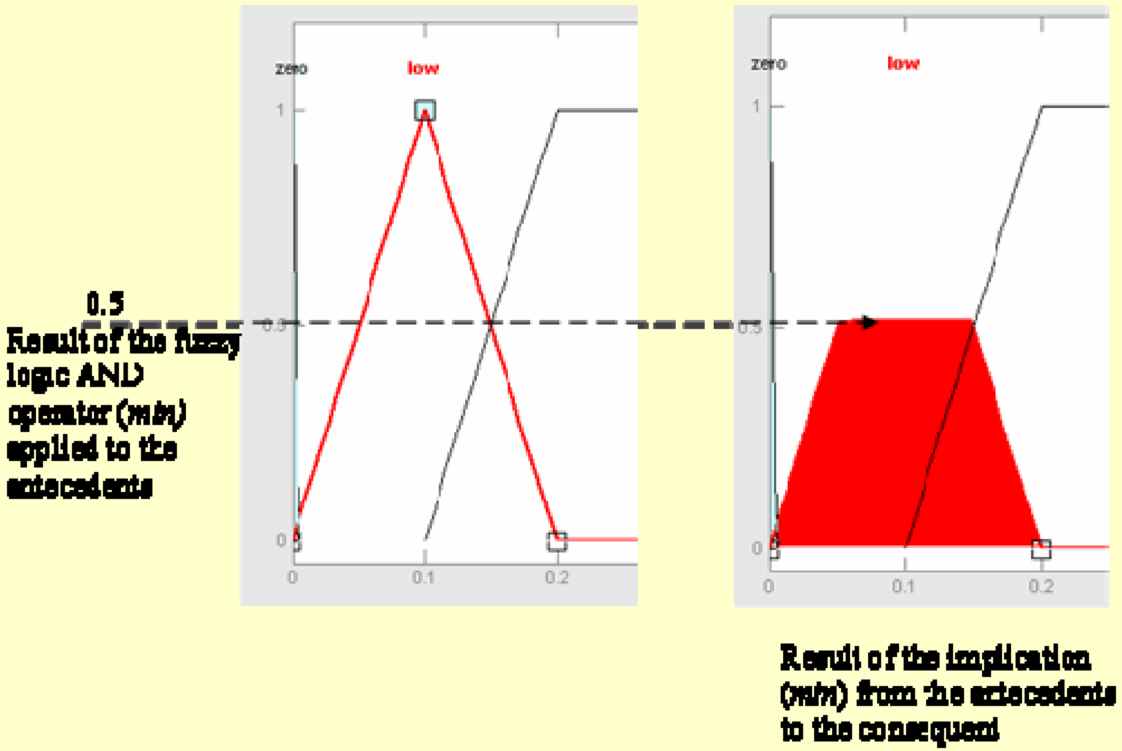


Figure 5.9 Application of the implication method (min)

Aggregation

- Since decisions are based on the testing of all of the rules in the *rule base* of a fuzzy system, the rules must be combined in some manner in order to make a decision.
- *Aggregation* is the method by which the fuzzy sets that represent the resulted (implied) outputs of each rule are combined into a *single fuzzy set*.
- The most popular and well-known aggregation method, which is also used by the fuzzy logic OR operator, is the *max*.
  - the resulting fuzzy set contains the maximum membership values among those generated by the implication process.

Aggregation (cnt'd)

$$\mu_{aggregated}(x_i) = \max\left(\mu^1_{implied}(x_i),\ \mu^2_{implied}(x_i),\ \ldots,\ \mu^N_{implied}(x_i)\right)$$

where $\mu_{aggregated}(x_i)$ is the *ith* membership value of the aggregated fuzzy output set, and $\mu^j_{implied}(x_i)$ is the membership value of the *ith* element belonging to the universe of discourse of the consequent – output fuzzy set, obtained in the implication process of the rule number *j* (*1* < *j* < *N*).

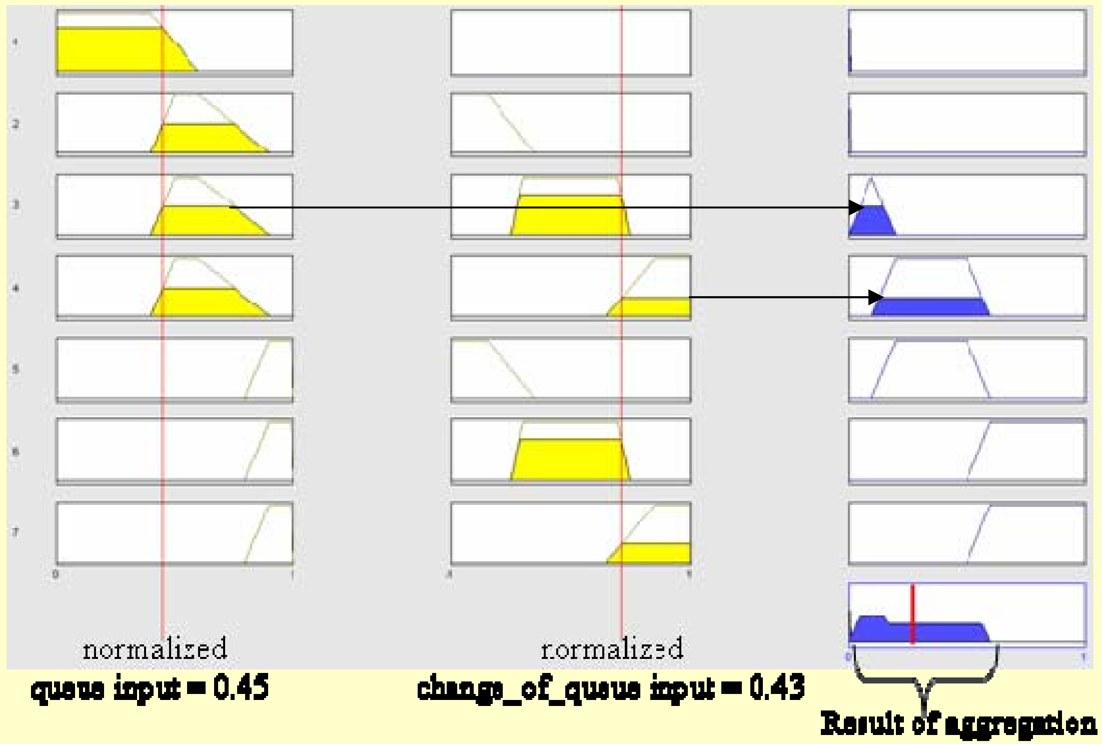


Figure 5.10 Application of aggregation method (max)

two rules contribute to the production of the aggregated output fuzzy set

Defuzzification

- The input for the defuzzification process is a fuzzy set (the aggregated output fuzzy set) and the output is a single number.
- The aggregated fuzzy set constitutes of a range of the maximum membership values of those implied fuzzy sets.
- As the aggregated fuzzy set encompasses a range of output values, a single output value must be resolved from the set.
- The final crisp output result is found at the last step of the fuzzy reasoning process: the process of defuzzification, which converts the fuzzy reasoning output, which is a fuzzy set, into a crisp value that represents the whole inference process outcome.

Defuzzification (cnt'd)

- There are various methods for defuzzification purposes.
- The most popular method is the *centroid* method, which returns the centre of area under the curve that represents the aggregated output fuzzy set.

$$p = \frac{\int_S y\mu_C(y)\,dy}{\int_S \mu_C(y)\,dy}$$

where $\mu_C(y)$ is the membership degree of *y* in the aggregated output fuzzy set *C*. The limits of integration correspond to the entire universe of discourse *S* of the output variable *p*.

- If discrete values are used, then $p = \frac{\sum_{i=1}^{k} y_i \mu_C(y_i)}{\sum_{i=1}^{k} \mu_C(y_i)}$

where the output universe of discourse *S* is discretized to *k* values.

Defuzzification (cnt'd)

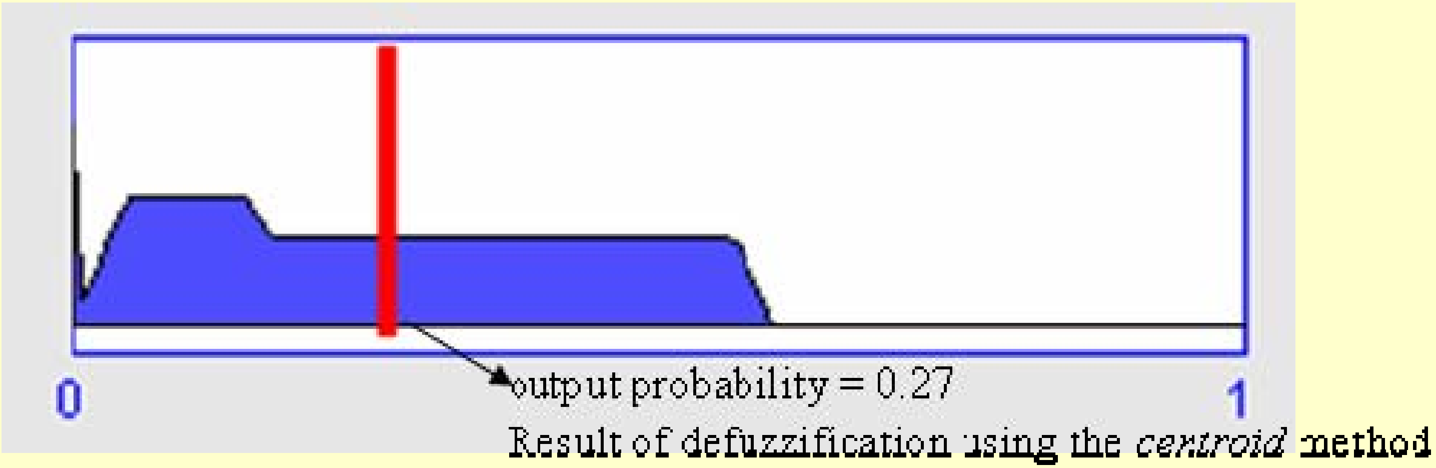


Figure 5.11 Application of defuzzification method (centroid)

Fuzzy Control Surface

- The nonlinear mapping from input to output implemented by the fuzzy logic controller is called the *control surface*.
- This mapping can be visualized by a nonlinear surface plot, where the controller's output is plotted against its inputs.
- The surface represents in a compact way all the information in the fuzzy logic controller.
- The rippled surface is created by the rules and the membership functions.

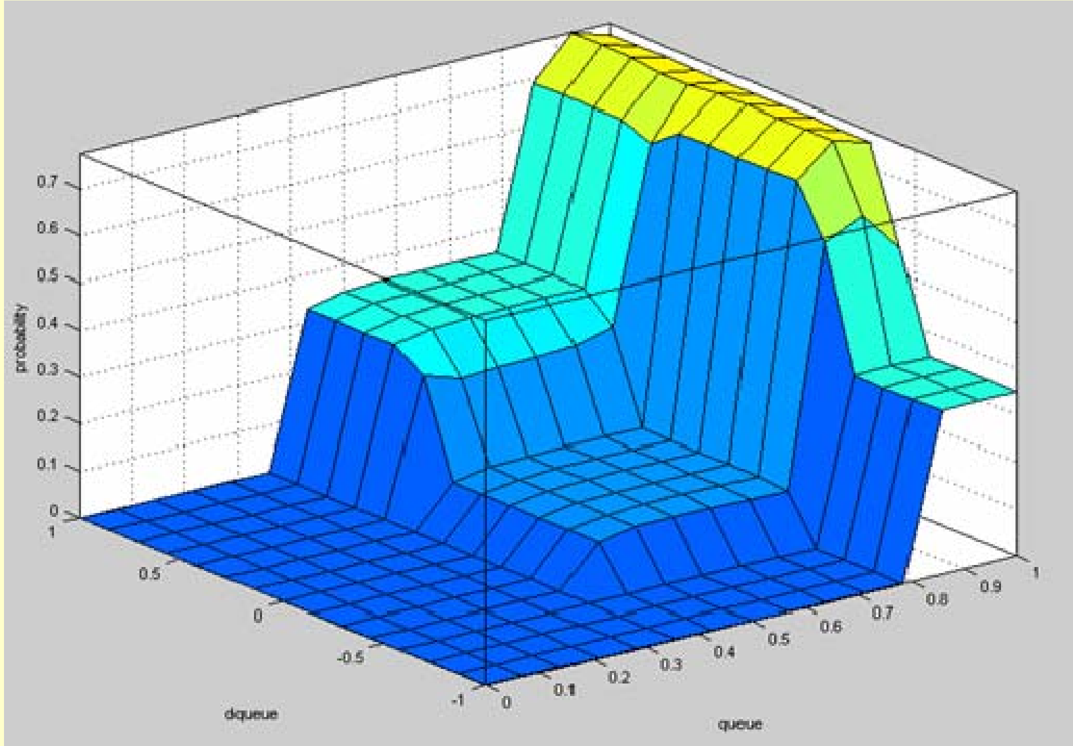


Figure 5.13 Nonlinear control surface of the fuzzy system used as an example

Fuzzy Control Inference Models

Three popular types

- ***(i) Mamdani Fuzzy Model:*** This method is associated with the use of Mamdani's 'min' operation rule as a fuzzy implication function. A crisp output is obtained through defuzzification.
- ***(ii) Larsen Fuzzy Model:*** This method is based on the use of Larsen's 'algebraic product' operation rule as a fuzzy implication function. A crisp control action can be deduced through defuzzification.
- ***(iii) TSK Fuzzy Model:*** This inferencing method (proposed by Takagi, Sugeno and Kang) produces crisp output as a result of the weighted average of each rule's output.

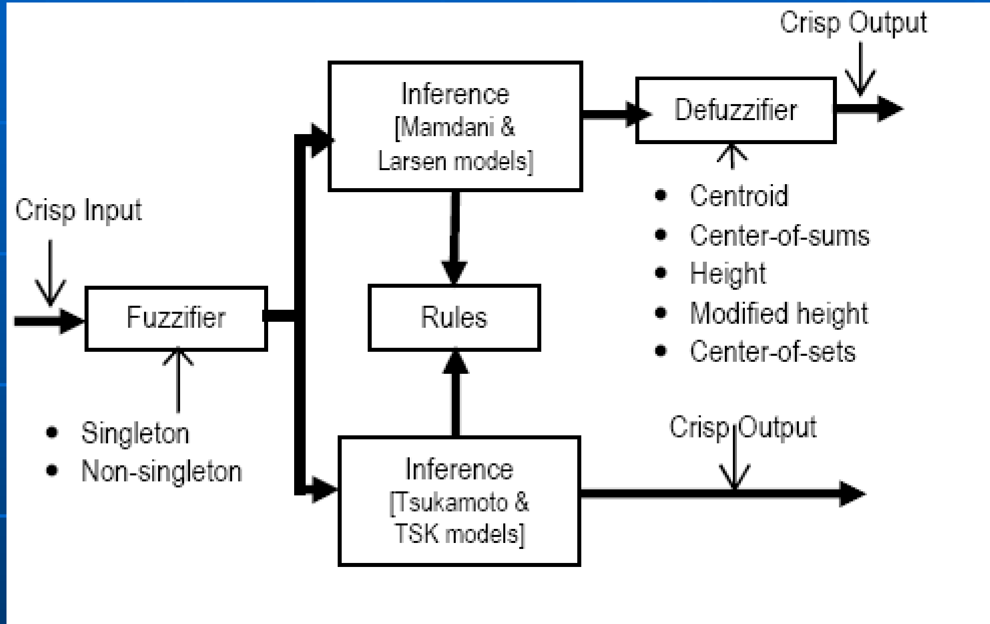


Figure 1.15: Basic elements of fuzzy logic system (FLS) and representative implementations.

Why Use Fuzzy Logic Control

- Fuzzy Logic Control
  - particularly appealing in nonlinear complex systems where
    - satisfactory analytic models are impractical
    - their behavior is well understood and can be captured by linguistic models
- Has solid theoretical foundation (at times controversial)
  - achieves 'inherent' robustness and reduces design complexity
- Part of what is termed Intelligent Control, or Computational Intelligence Control
- Remarkable success demonstrated in research literature and commercial products in many diverse disciplines

Why Use Fuzzy Logic Control (cnt'd)

- An appealing alternative to conventional control methods when systems follow some general operating characteristics that can be linguistically described, and a detailed process understanding is unknown or traditional systems models become overly complex.
- Fuzzy Logic Control concentrates on attaining an intuitive understanding of the way to control the process, incorporating human reasoning in the control algorithm.
- It is independent of mathematical models of the controlled system, thus achieving inherent robustness and reducing design complexity.

Why Use Fuzzy Logic Control (cnt'd)

- The main idea is that if the fuzzy logic control is designed with a good (intuitive) understanding of the system to be controlled, the limitations due to the complexity system's parameters introduce on a mathematical model can be avoided.
  - A common approach in classical control theory is to either ignore such complex parameters in the mathematical model, or to simplify the model to such an extent (in order to obtain some stability results), which render the designed controllers and their derived stability bounds overly conservative.
- The control algorithm is encapsulated as a set of linguistic rules, leading to algorithms describing what action should be taken based on system behaviour observations.

Congestion Control in Communications: An Overview

A recent remark

‘Networks are very complex. Do not kid yourselves otherwise.’

Debasis Mitra, Senior VP Research, Bell Labs
Panel discussion at Infocom 2001

Modelling of the Shrew (beast): Quest for a ‘Model’ Network Model (Organiser: Ariel Orda)

Rapid changes in Networking technologies

Networking technology and techniques are in constant state of change

Fixed Networks evolution	
Network Type	**Transfer media type**
Telegraph – mid 19th cent.	Morse code
Telephone system – late 19th cent.	Voice
N-ISDN – early 80's	Voice + data
LAN – early 80's	Data + lately some multimedia
B-ISDN – early 90's (includes ATM technology)	Integrated services (multimedia)
Internet – early 90's (commercialisation)	Data + lately some multimedia
'New Network(s)' – ? Internet + or ? Grid ?	access from everywhere, anytime for all communication: voice, video, data, TV, ...

bits/sec

Gbits/sec

Rapid changes in Networking technologies

Wireless Networks evolution	
Network Type	**Transfer media type**
Analog Mobile – early 70s	Voice
GSM – early 80s	Voice
GSM + GPRS – early 90s	Voice + data
UMTS – early 2000s	Integrated services
4th Generation – mid 2000s	Integrated services over heterogeneous networks
WLAN – mid 90's	Data + some multimedia
WiMAX – mid 2000s	Data + multimedia
Ad-hoc, MANETS – mid 90's	Data + multimedia
Sensor – early 2000's	Data + multimedia?
'new' information transfer techniques??	??

Congestion Control (CC) for Packet-based Communication Networks

- Active research area for over 30 years
- Still remains one of most important research challenges
  - demands on Internet, e.g. QoS, multiservice, multimedia (existing TCP only Best Effort/Elastic traffic)
  - high speed internet (existing TCP does not scale well)
  - wireless networks (existing TCP does not perform well)
  - new networking techniques / technologies: diff-serv, peer-to-peer, ad-hoc and sensor networks, etc…
- A multi-faceted problem with many different approaches, not only on how we affect control, but also on the control structure, as well as the theoretical development
  - even a definition cannot be universally accepted by research community
    - Internet Congestion Control Research Group, ICCRG, debate in 2006
  - an Internet Draft on metrics only now issued
    - Transport Modeling Research Group, TRMG, March 07: submitted to IRTF to be considered as Informational
  - research challenges are still debated
    - ICCRG February 2007

Causes and effects of congestion

To control congestion we must understand:

- what is that we are trying to control and how we can model and quantify it (what metrics)
- how and where we can control it

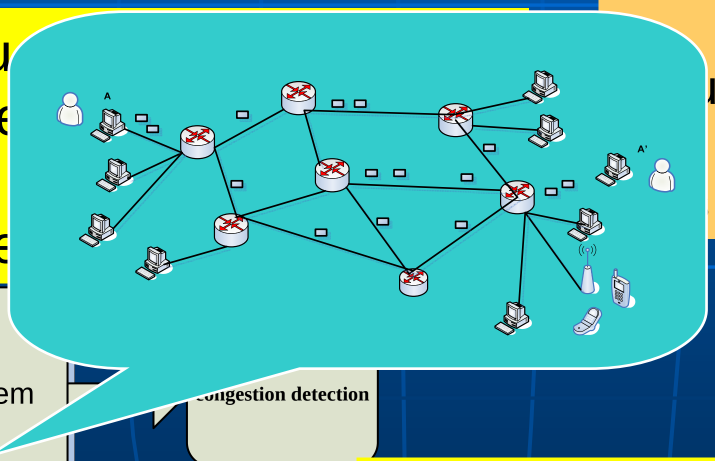
If we are going to u

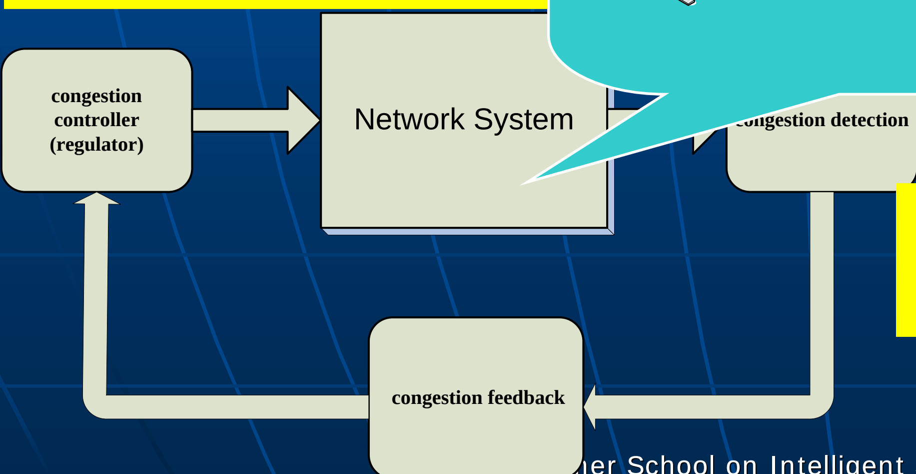
- how and where
  and control it?
- what type of fe

Quantifying and modelling congestion not easy

- congestion is caused by constrained resources
- felt by users as
  - delay
  - loss
  - throughput

ut these metrics are
bjective and conflicting,
seen by applications





FB control system elements

Inherent control difficulty: control elements are also within network system

Different approaches toward CC

- diverse range of control approaches adopted
  - open loop, closed loop, proactive, reactive, adaptive
  - flow control, admission control, bandwidth control, routing
  - control structure
    - different feedback schemes and combinations and placement of CC elements
      - end-to-end, network assisted (AQM)
        - implicit, explicit, single-bit (e.g. ECN), multi-valued (ER)
      - at sender, at sender+receiver, at sender+receiver+routers
- diverse range of mathematical tools adopted
  - From fuzzy logic to game theory, from feedback control theory to utility functions and linear and nonlinear programming
- focus only on feedback based control approaches, regulating flow from sources

Our view: Networks behave as *complex dynamical systems*

- Internet is largest and most complex artificially deployed system
- possesses similar structural properties to other complex systems pervading science
  - heterogeneous subsystems (sources and routers) performing complex functions
  - interconnected by heterogeneous links (wired, wireless), often incorporating complex dynamics themselves
- many other factors contribute to immense complexity
  - large-scale and size
  - fragmented nature of underlying infrastructure
  - semi-hierarchical organization
  - extreme heterogeneity - diverse network technologies and communication services
  - distributed management of resources
  - complex structures which arise in implementation of the layered protocols

Our view: Networks behave as *complex dynamical systems*

- *identified* ***control difficulties*** *in networks include*
  - *non-linear time varying* dynamics
  - non-linear interacting subsystems
  - *large delays* in relation to system time scales
  - *tight control demands*
  - *largeness* of network system
  - *lack of appropriate measurements*
  - *placement of controls and lack of adequate local controls*
  - *uncertain feedback path*
  - *uncertain responsiveness of the system*

in-build control structures were never engineered in the network system

Our view: Networks behave as *complex dynamical systems*

- introduced ***advanced control theoretic techniques***, exploiting dynamic nature of network to come up with effective solutions
  - solid theoretical foundation in effectively dealing with complex dynamical systems
  - remarkable success in many diverse systems

- demonstrated
  - that despite reported control difficulties, control theory provides effective control with tight QoS provision and good properties
  - proposed control theoretic techniques are effective in delivering efficient solutions in highly complex network systems, such as ATM and Internet, outperforming existing approaches

CC design objective

- goal of designed CC protocols is to guide network to a ‘***stable’ equilibrium***
  - *‘tight’* QoS control featuring
    - high utilization
    - tolerable queue sizes
    - tolerable packet drops
  - good *transient and steady state* response
    - smooth responses with no or minimal oscillations
    - fast convergence
    - settles to fair allocations (max-min sense)
  - *robust* and *scalable* with respect to changing network dynamics
    - bandwidths and delays
    - number of users
    - network size
  - simple to implement, effective design

Resource Allocation

- Effective and fair resource allocation is necessary in any constrained system
- In our study of resource allocation we provided solutions to
  - Wireless Networks
    - Streaming video

Fuzzy Logic based AQM Congestion Control in TCP/IP Networks

"Quote 1"

'So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.'

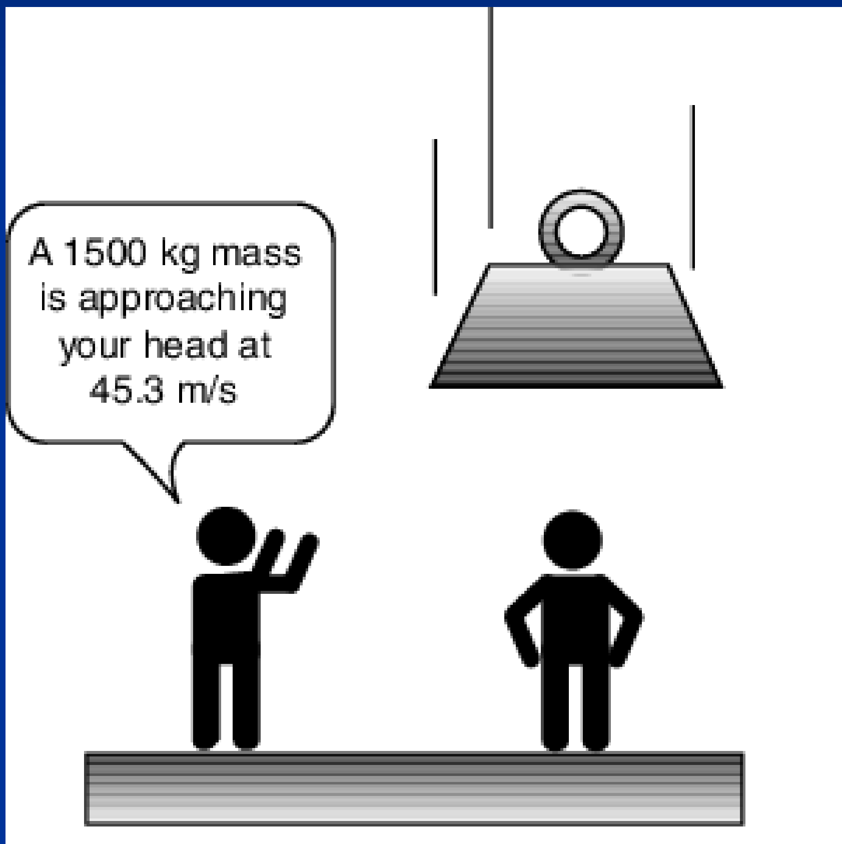
Albert Einstein

“Quote 2”

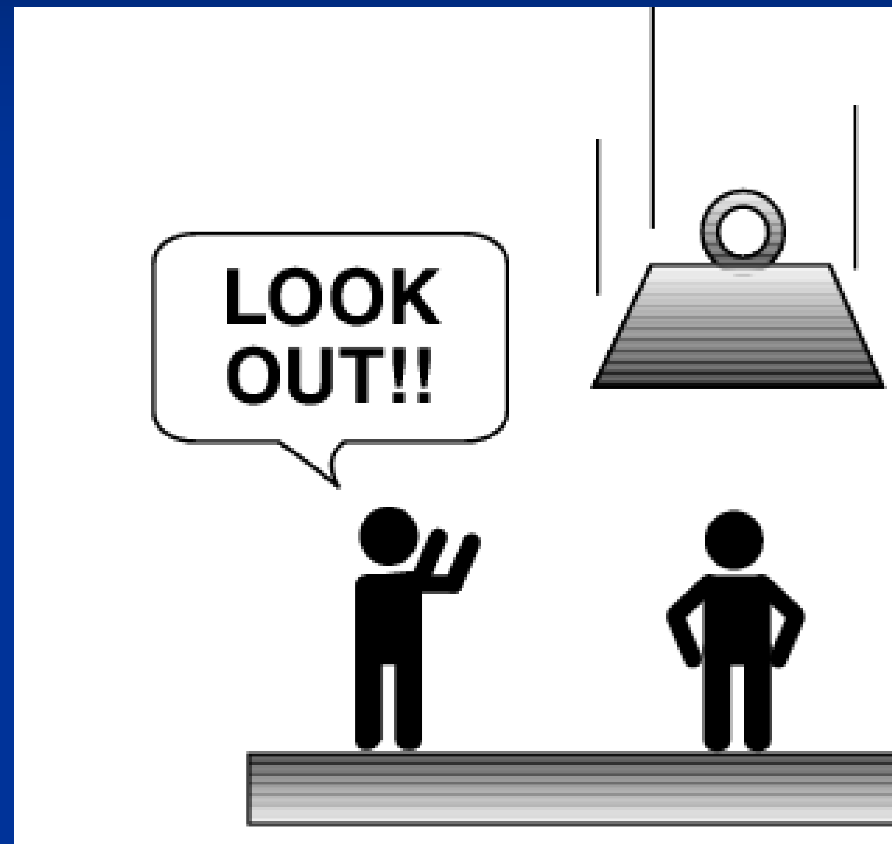
‘As complexity rises, precise statements lose meaning and meaningful statements lose precision.’

Lotfi Zadeh (“father” of Fuzzy Logic)

Precision and significance in real world



Precision



Significance

Fuzzy CC (ATM, Internet, DiffServ)

- Encouraged by success of *Fuzzy Logic Control (FLC)* in *robustly* controlling nonlinear complex systems, especially where a *dynamic mathematical model* is *impractical* to obtain
- Successfully solved a range of problems in both ATM and TCP networks (Best Effort and Diff-Serv), thus also demonstrating universality of FLC approach
- Evaluated performance in OPNET and ns-2 based discrete event simulators
  - robust and effective control in large set of scenarios with widely differing traffic characteristics and topological differences, thus verifying robustness
  - good steady state and transient behaviour
  - in all cases FLC approach outperformed well known schemes reported in literature
- Since 1993, addressed many network problems using FLC

Our FLC-based Research Study

- investigated the complex, but challenging, concepts of
  - ATM QoS aware flow control
  - TCP/AQM (Active Queue Management) CC
    - BE (best-effort)
    - Diff-Serv (differentiated services) environments

- FOCUS OF PRESENTATION IS TCP/AQM CC

Problem Statement

- **Main Aim**: provide effective control for high link utilization with low loss and queuing delay
  - focus on AQM mechanisms with ECN support, thus keeping TCP’s CC mechanisms unchanged
    - Internet standards track protocol RFC3168
    - most routers support ECN
  - furthermore, address Diff-Serv congestion control at core for aggregated QoS support

Motivation

- Current CC solutions based on AQM/ECN are *ineffective to meet diverse needs of today's Internet*
  - they have serious limitations and drawbacks, as identified in literature

- Extremely difficult for traditional modeling techniques to capture the network's essential dynamics
  - even if they do resulting model is overly complex

- Common approach in classical control theory is to
  - ignore such complex parameters in mathematical model
  - simplify model, often making overly conservative with restrictive stability bounds

- Given need for effective control methodology
  - to capture system behaviour under widely differing operating conditions
    - investigate usefulness of **fuzzy logic control** to meet such objectives

Contributions

- Offer significant improvements
  - achieve high link utilization and regulated queues
  - fast system response and robustness to varying system dynamics (differing topologies and traffic conditions)
  - in Dif-Serv, adequate and effective differentiation among different priority classes
- Demonstrate that Fuzzy Logic based AQM control methodology better handles nonlinearities and dynamics, in contrast with existing, well-known, conventional counterparts

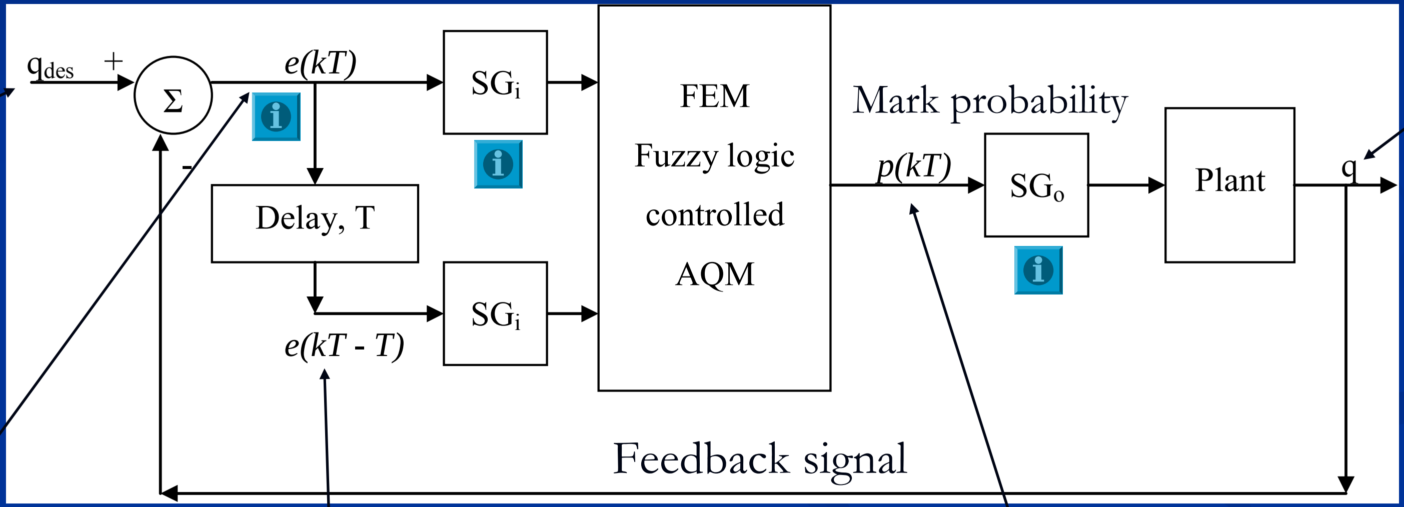
Proposed Mechanisms: BE

- Proposed AQM scheme for BE (Best-Effort) environments
  - **Fuzzy Explicit Marking (FEM)**
    - regulates queues of IP routers at predefined levels
      - by achieving a specified target queue length (TQL)
    - in same spirit as RED
    - Fuzzy inference engine (FIE) operates on router buffer queues

Proposed Mechanisms: BE (cnt'd)

- **Feedback system model of FEM**

desired queue length



queue length

based on two network state inputs

• **error on instantaneous queue length** for *two consecutive sampling intervals*

like RED, FEM dynamically calculates ***mark probability*** p(kT)

Proposed Mechanisms: BE (cnt'd)

- **System <u>model of FEM</u>**

surface structure

Table 6.1 FEM Linguistic rules – Rule base

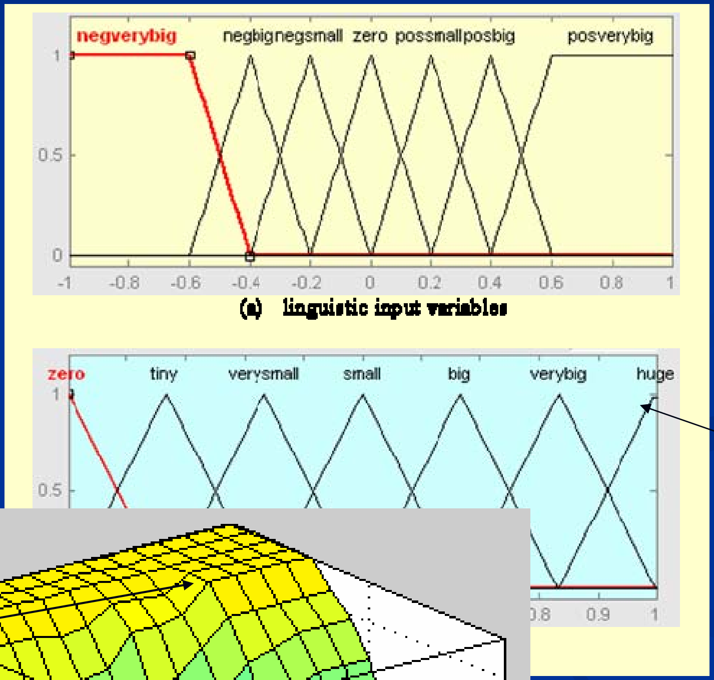
p(kT)		Q (kT - T)						
		NVB	NB	NS	Z	PS	PB	PVB
	NVB	H	H	H	H	H	H	H
	NB	B	B	B	VB	VB	H	H
	NS	T	VS	S	S	B	VB	VB
Q (kT)	Z	Z	Z	Z	T	VS	S	B
	PS	Z	Z	Z	Z	T	T	VS
	PB	Z	Z	Z	Z	Z	Z	T
	PVB	Z	Z	Z	Z	Z	Z	Z

Knowledge-base (linguistic rules) generated from IF-THEN control r
e.g.:

IF *e(kT)* is *NVB* AND *e(kT – T)* is
THEN *p(kT)* is *H*

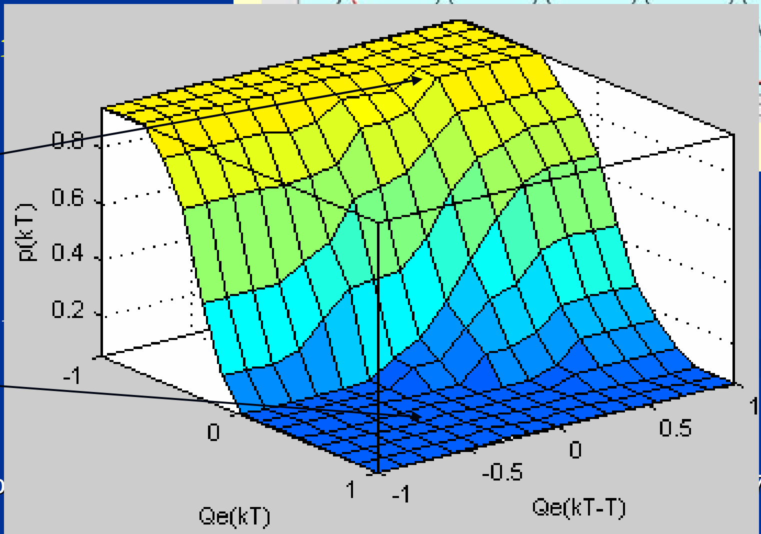
IF *e(kT)* is *PVB* AND *e(kT – T)* is
THEN *p(kT)* is *Z*

deep structure



Membership functions

linguistic values



mark probability responsive due to human reasoning and inbuilt nonlinearity

Proposed Mechanisms: Diff-Serv

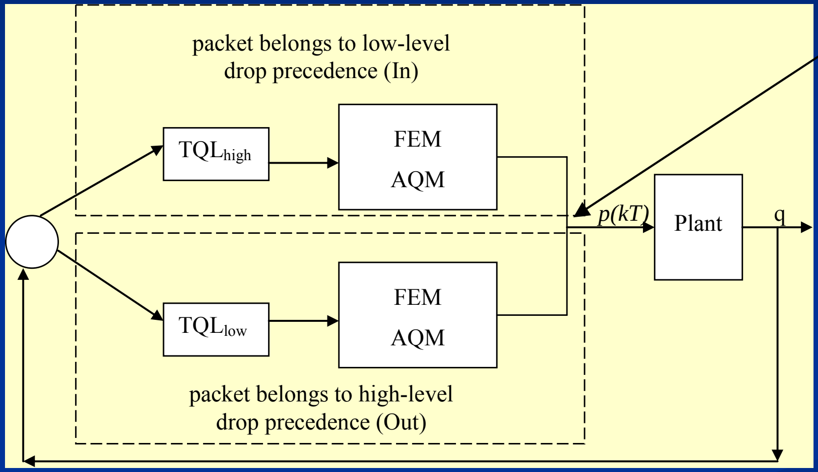
- **Diff-Serv Fuzzy Logic Control Design (FIO)**
  - **goal** to achieve **same performance as BE**
    - provide effective congestion for Diff-Serv, plus
    - **differentiated treatment of traffic aggregates**
  - built on fuzzy controller designed for BE environments
  - **two identical FEM controllers** used
    - one for each differentiated traffic aggregate (FIO – FEM In-Out)
      - high-priority (low drop precedence / In packets)
      - low-priority (high drop precedence / Out packets)

      hence offering (differentiated) QoS in traffic aggregates

Proposed Mechanisms: Diff-Serv (cnt'd)

- Two different TQLs, one for each FEM controller

TQL for low-priority < TQL for high-priority



high- and low-priority traffic share a FIO queue

- objective: regulate queue at TQL_{low}, where *mark* probability for high-priority traffic is closer to zero
  - **but**, if high-priority traffic >> low-priority traffic, at least regulate queue at TQL_{high}
    - not enough low-priority traffic to maintain TQL_{low}
    - in this case *mark* probability for low-priority traffic is closer to 1

- accomplish both differentiation and bounded delay, by regulating queue between two TQLs, depending on network traffic

Simulative Performance Evaluation

- Use extensive simulative evaluation to demonstrate effectiveness and robustness, in **both BE and Diff-Serv environments**
- Comparison made with other published results with well-known, AQM schemes
  - A-RED, PI, REM, and AVQ for BE networks
  - RIO and TL-PI for Diff-Serv networks
- Performance of AQM schemes evaluated using most widely used simulator, NS-2, in different topologies and wide varying network conditions
- In all cases, our approach outperforms all others in all scenarios and network conditions

additional results FEM, FIO

FEM Evaluation

Effect of traffic load (increase flows from 100- 500) provides some time-varying dynamics and scalability. Single-bottleneck link (TQL = 200 packets ~ 100 msec)

- FEM outperforms other AQM schemes
  - high link utilization, low delay and delay variation
  - exhibits more stable, robust behavior with bounded delay
- lowest drops, over large traffic load

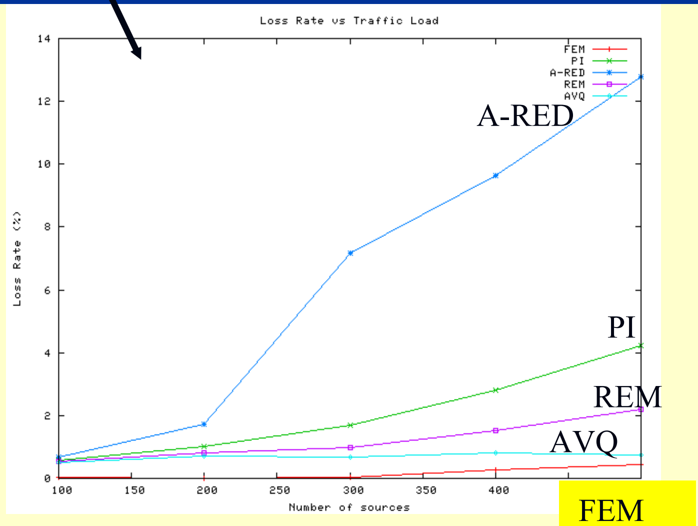


Figure 7.10 Scenario I-5: Loss Rate vs Traffic Load (for 100-500 flows)

loss

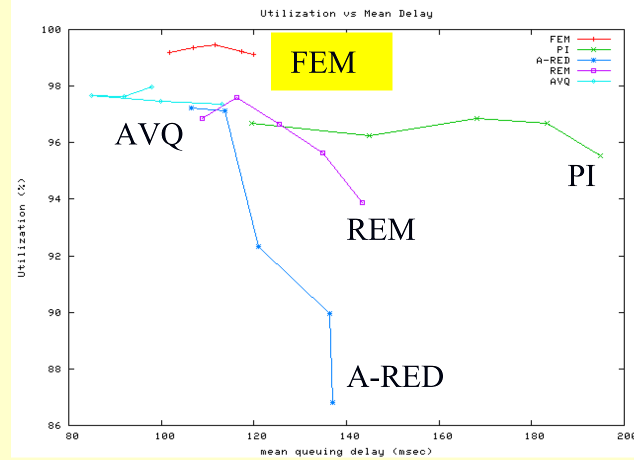


Figure 7.11 Scenario I-5: Utilization vs Mean Delay (for 100-500 flows)

delay

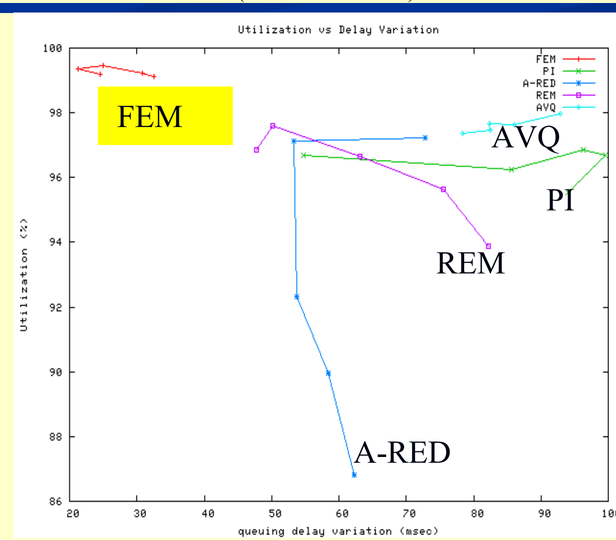
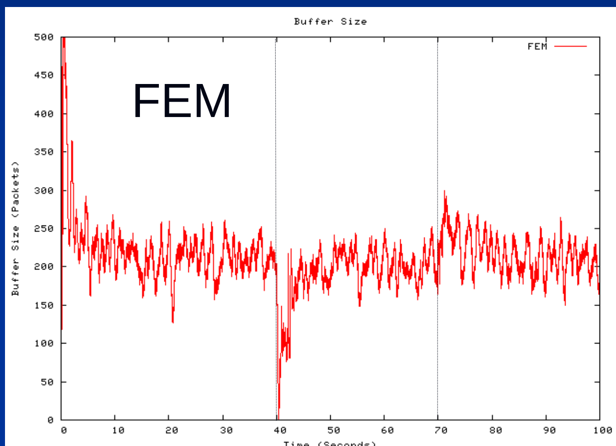


Figure 7.12 Scenario I-5: Utilization vs Delay Variation (for 100-500 flows)

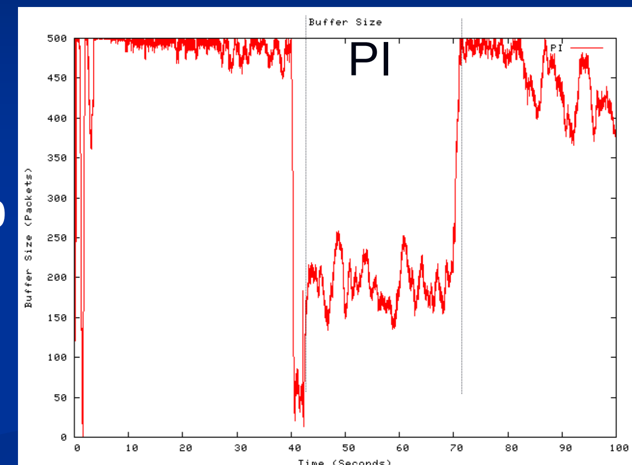
delay variation

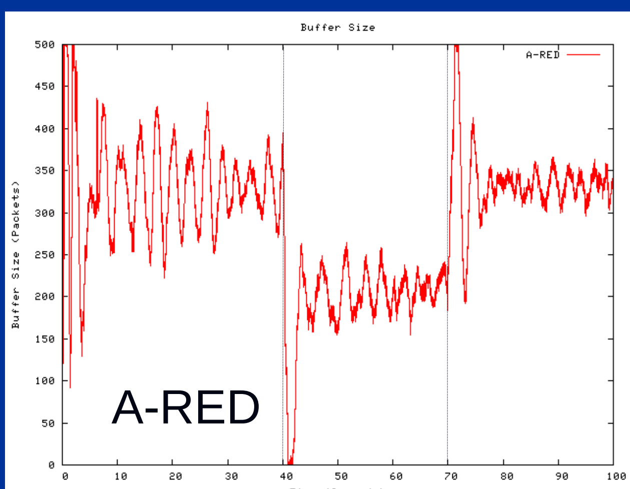
FEM Evaluation (cnt’d)

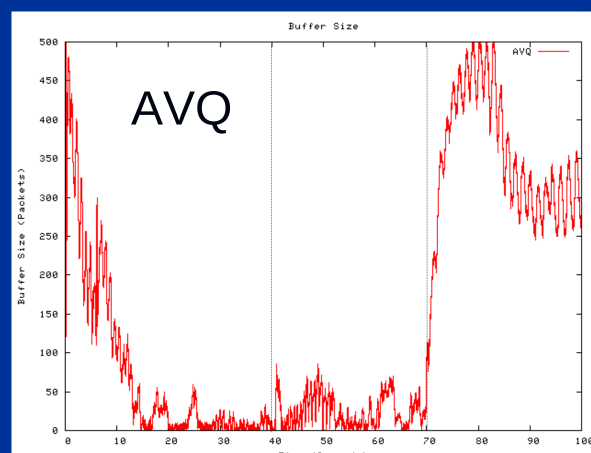
queue length evolution: sudden change in traffic conditions

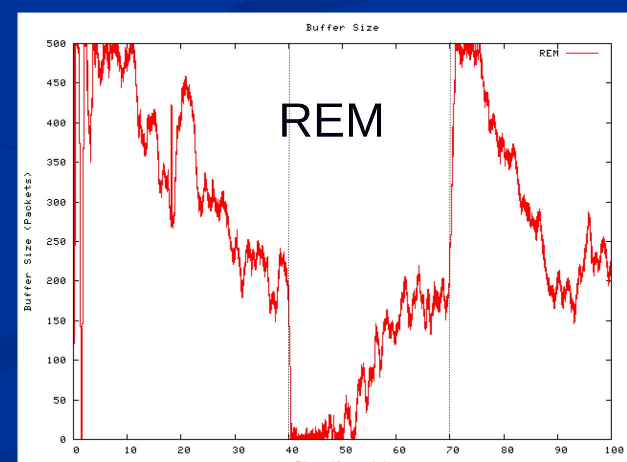


600 users at t=0
- **300 leave network t=40**
- **300 re-enter network t=70**









multiple bottleneck links

Diff-Serv FIO Evaluation

- Single-bottleneck link
  - Effect of increasing high-priority traffic
    - increases from 2%, 10%, and 90% of total traffic
  - FIO outperforms other schemes with **better differentiation** provided in favor of the high priority traffic

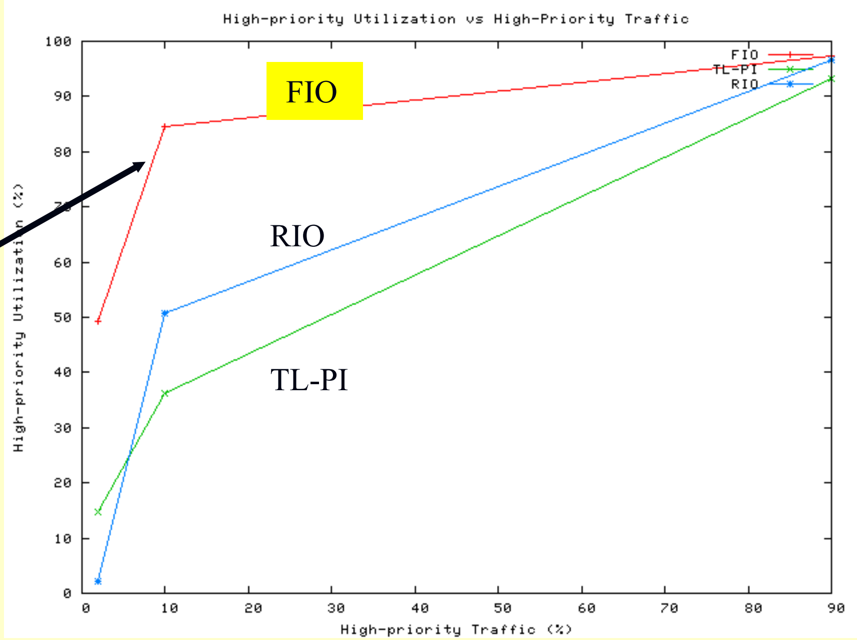


Figure 9.4 Scenarios I-1-3: Utilization of high-priority traffic vs percentage of high-priority traffic
(high-priority traffic increases from 2%, 10, and 90% of the total traffic)

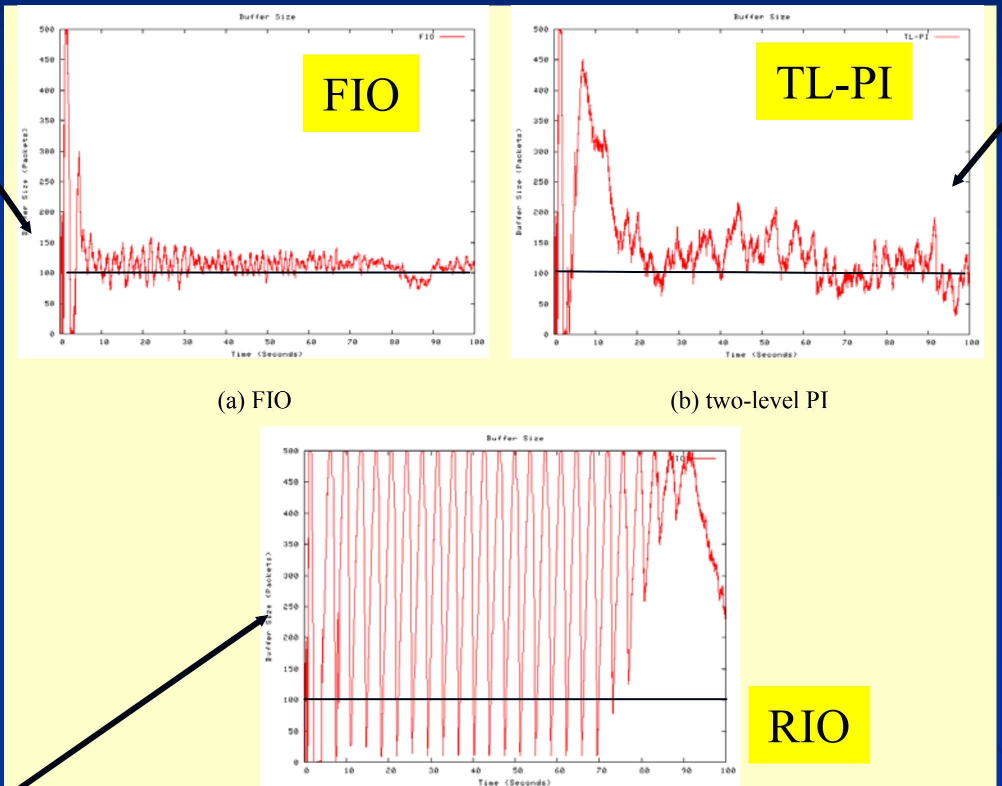
Diff-Serv FIO Evaluation (cnt'd)

- Single-bottleneck link
  - Effect of increase of high-priority traffic – (2% of total traffic) on queue behaviour

FIO regulates its queue to TQL_{low}

TL-PI tries to regulate the queue at TQL_{low} however slow and large delay variation

TQL_{low} = 100
TQL_{high} = 200



(a) FIO

(b) two-level PI

(c) RIO

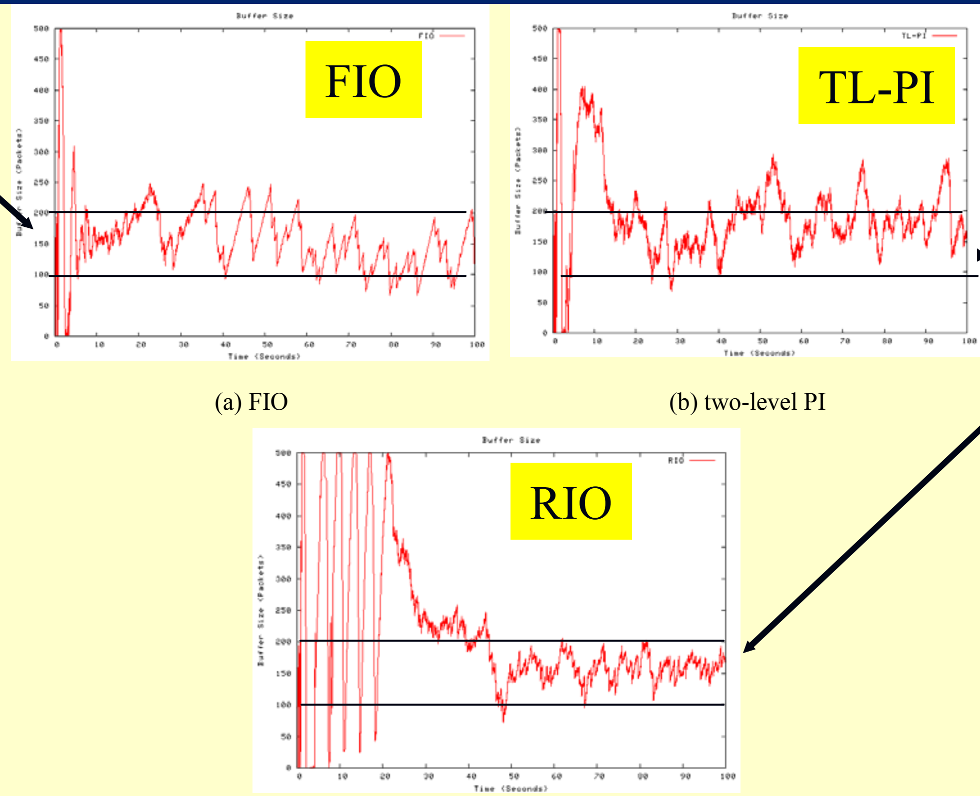
Figure 9.1 Scenario I-1: Queue lengths

RIO cannot regulate at all the queue and exhibits very large queue fluctuations

Diff-Serv FIO Evaluation (cnt'd)

- Single-bottleneck link
  - Effect of increase of high-priority traffic – cnt'd (10% of total traffic)

FIO accomplishes a bounded delay, between the two TQLs



(a) FIO

(b) two-level PI

(c) RIO

Figure 9.2 Scenario I-2: Queue lengths

TL-PI and RIO slowly regulate their queue, after a significant transient period

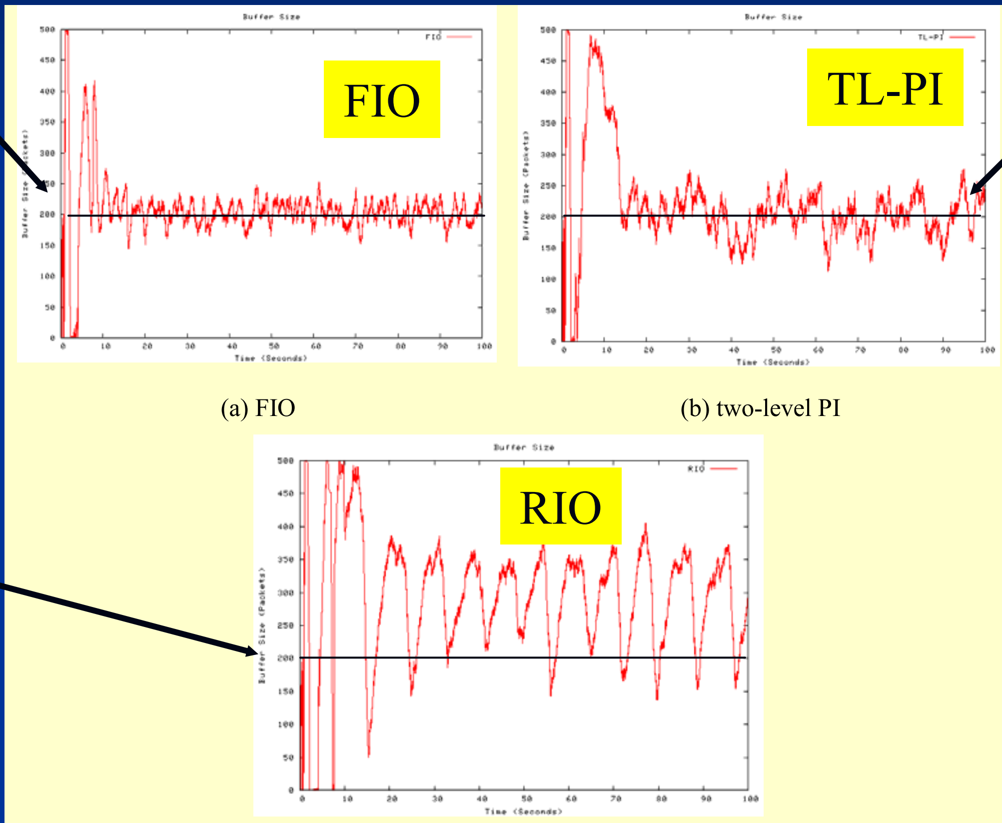
TQL_{low} = 100
TQL_{high} = 200

Diff-Serv FIO Evaluation (cnt'd)

- Single-bottleneck link
  - Effect of increase of high-priority traffic – cnt'd (90% of total traffic)

FIO regulates its queue at **TQL_{high}**

RIO exhibits large queue fluctuations



(a) FIO

(b) two-level PI

(c) RIO

Figure 9.3 Scenario I-3: Queue lengths

TL-PI shows a sluggish response and less tight control than FIO

TQL_{low} = 100
TQL_{high} = 200

Diff-Serv FIO Evaluation (cnt'd)

- Single-bottleneck link, 100 flows (10% of total traffic is high-priority)
  - Effect of time-varying dynamics (between t=40sec – 70sec, only low-priority traffic is active, i.e 90 flows)

FIO is very robust against the dynamic traffic changes, successfully manages to regulate queue length at TQL for low-priority, between t=40-70sec

RIO fails to regulate queue

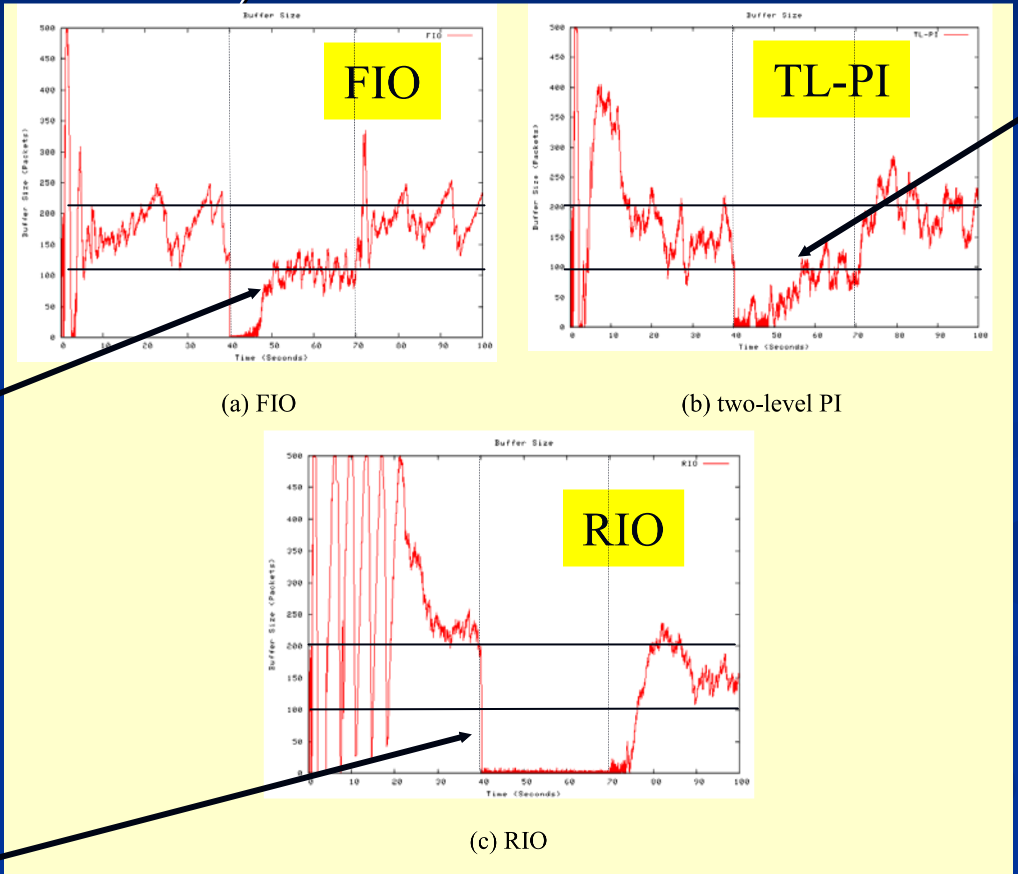


Figure 9.5 Scenario I-4: Queue lengths

TL-PI slow response to regulate queue

$TQL_{low} = 100$
$TQL_{high} = 200$

All high priority traffic leaves the network at 40 seconds and reenters at 70 sec

Concluding Remarks

- Due to complexity of dynamic network parameters, FLC, a robust **intelligent control methodology,** is adopted to effectively control the network system under widely varying operating conditions and dynamic changes
  - provide effective CC and QoS support within BE, and furthermore effective differentiation for Diff-Serv environments
    - FIO uses same controller as BE (FEM) for both priority classes
    - no retuning needed

Future Work

- supplement simulation with analytical proof of behaviour
  - inherent robustness of fuzzy systems
- adaptive tuning scaling factor of probability function using formal adaptive control theory
- large scale behaviour
- 'real' system tests
  - LINUX based pilot Diff-Serv network in UCY NetRL and Monash Networks Lab
- Extend to other network environments
  - mobile/wireless networks
  - high speed internet and rate-based multimedia transport framework
  - Ad-hoc and sensor networks

Resource Allocation in Communication Networks

ADIVIS: An Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Control

Adaptive Video Streaming

- **Motivation**

  - Compressed video streams exhibit large variations in their data rates

  - Unpredictable network environment (bandwidth, delay, loss, etc.)

  - Heterogeneity of video-enabled end devices (different capabilities, requirements)

  - Wireless environment → erroneous and time-varying conditions

- **Objectives**

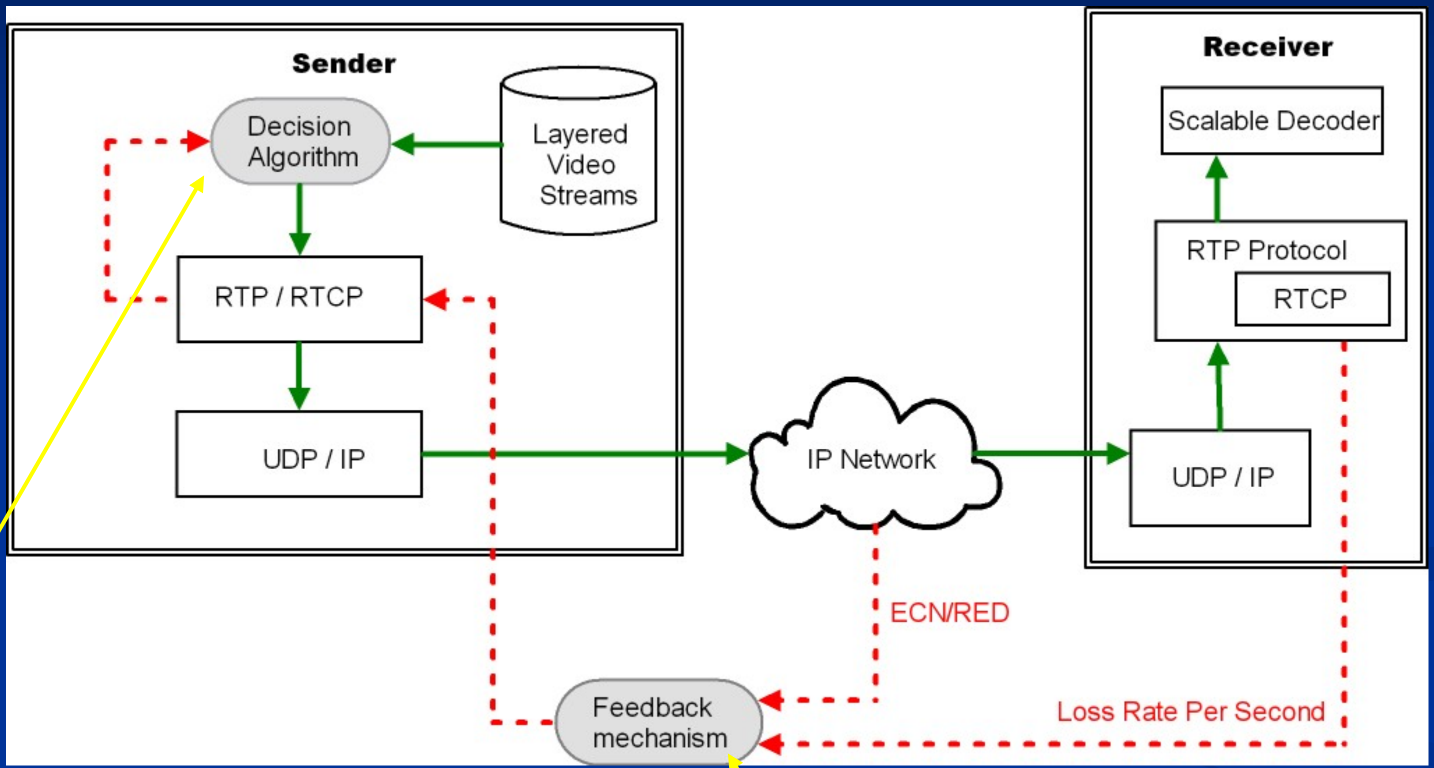
  - Responsiveness to dynamic changes & different (user/network) demands.

  - Scalability, adaptability against network complexity & heterogeneity: scalable content encoding, adaptive transmission rates.

Solution: System Adaptation

- Network Adaptation Techniques (NATs)
  - adaptation to network parameters (avail. bandwidth, loss, delay, jitter, etc.)
  - The basic requirements of NATs are (1) to provide accurate information on the network load, (2) to distinguish between core congestion and wireless link errors, (3) to recognize a change in the possible bandwidth due to changes in the wireless link conditions, and (4) to adapt accordingly the transmission rate at the source.
- Content Adaptation Techniques (CATs)
  - scalable (layered) video content
  - The techniques for reducing the transmitted information are based on dropping layers

ADIVIS-based System Diagram



Fuzzy Decision Algorithm

Evaluates the available network bandwidth and decides in a fuzzy manner the optimal number of layers that should be sent by adding or dropping layers

Adaptive Feedback Mechanism

Collects (a) receiver's critical info (packets loss rate per second), and (b) measurements obtained by the core network (packets marked – ECN/RED)

Fuzzy Decision Algorithm

- 2 linguistic input variables: $D_{LRPS}(kT)$, $DN_{ECN}(kT)$
- 1 linguistic output value: a(kT)

  (T: decision period)

- Linguistic Rule Base:

a(kT)		DN_ECN(kT)						
		NVB	NB	NS	Z	PS	PB	PVB
D_LRPS(kT)	NVB	H	H	VB	B	S	VS	T
	NB	H	VB	VB	B	S	VS	T
	NS	VB	B	B	B	S	VS	T
	Z	B	B	B	B	S	VS	T
	PS	S	S	S	S	VS	VS	T
	PB	VS	VS	VS	VS	VS	VS	T
	PVB	T	T	T	T	T	T	T

Fuzzy Decision Algorithm (2)

- $D_{LRPS}(kT) = LRPS(kT) – LRPS(kT-T)$, $\epsilon[-1,1]$

  - increasing/decreasing trend of LRPS
- $N_{ECNsc}(kT) = N_{ECN}(kT) / N_{ps}(kT)$

  $DN_{ECNsc}(kT) = N_{ECNsc}(kT) - N_{ECNsc}(kT)$, $\epsilon[-1,1]$

  - increasing/decreasing trend of marked packets percentage
- $avail\_bw(kT) = a(kT)*avail\_bw(kT-T)$, $\epsilon[-0.5,1.5]$

  - gradual increase/quick reduction

Fuzzy Decision Algorithm (3)

- Defuzziffied crisp values of $\alpha(kT)$ are used for the evaluation of the available bandwidth:

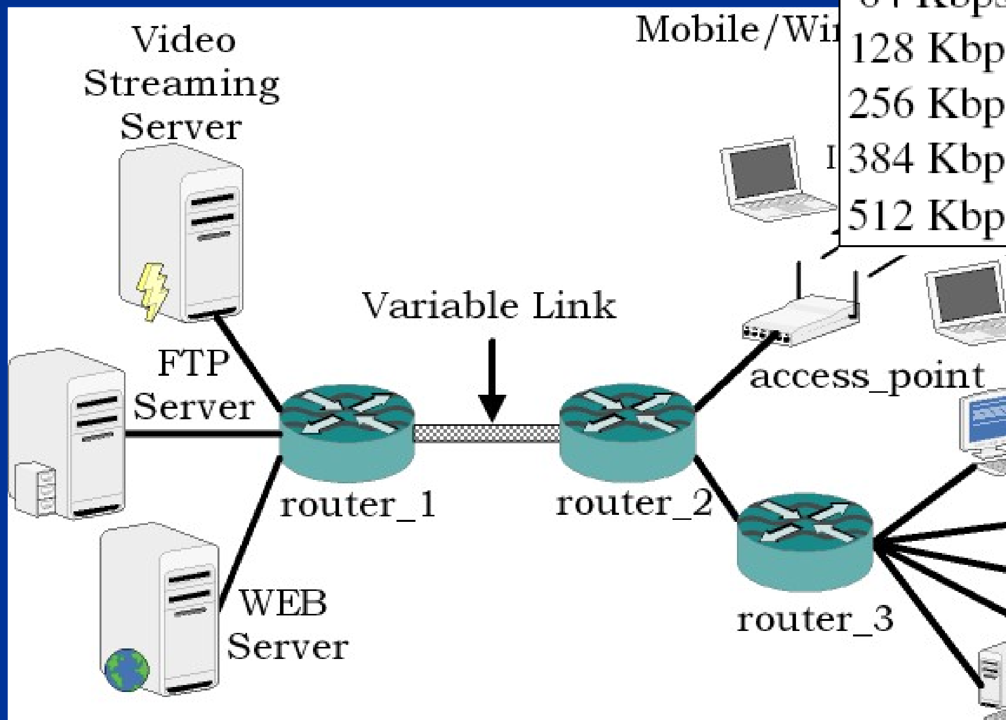
  $$avail\_BW(kT) = \alpha(kT) * avail\_BW(kT-T)$$

- Defuzziffied output value ranges from 0,5 to 1,5.
- Decision algorithm tries to "guess" the available bandwidth. Thus a "gradual" increase is allowed when there is available bandwidth and reduced congestion, whereas quick action is taken to reduce the rate to half in case of severe congestion.
- Time hysteresis is introduced in order to avoid frequent transitions from one layer to another which may cause instability (non aggressive layer selection approach).

Evaluation Setup & Scenarios

Variable link parameters:

Link Bandwidth		*Propagation Delay*
64 Kbps	768Kbps	10 ms
128 Kbps	1 Mbps	100 ms
256 Kbps	2 Mbps	200 ms
384 Kbps	4 Mbps	400 ms
512 Kbps		800 ms



Test sequence:
Foreman: 30fps, 176x144

Video Stream Bit Rate	
64 Kbps	384 Kbps
96 Kbps	512 Kbps
128 Kbps	768 Kbps
192 Kbps	
256 Kbps	

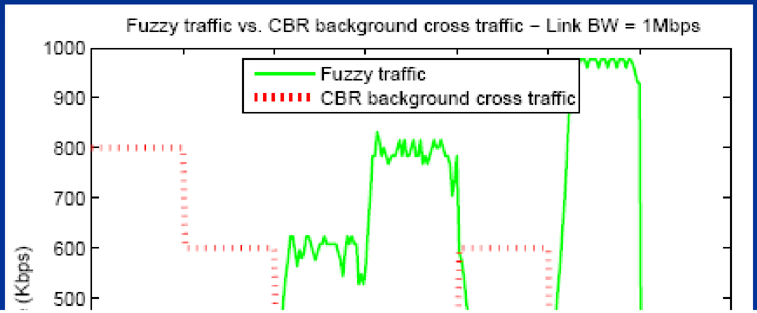
Decision Period: T = 0.5 secs

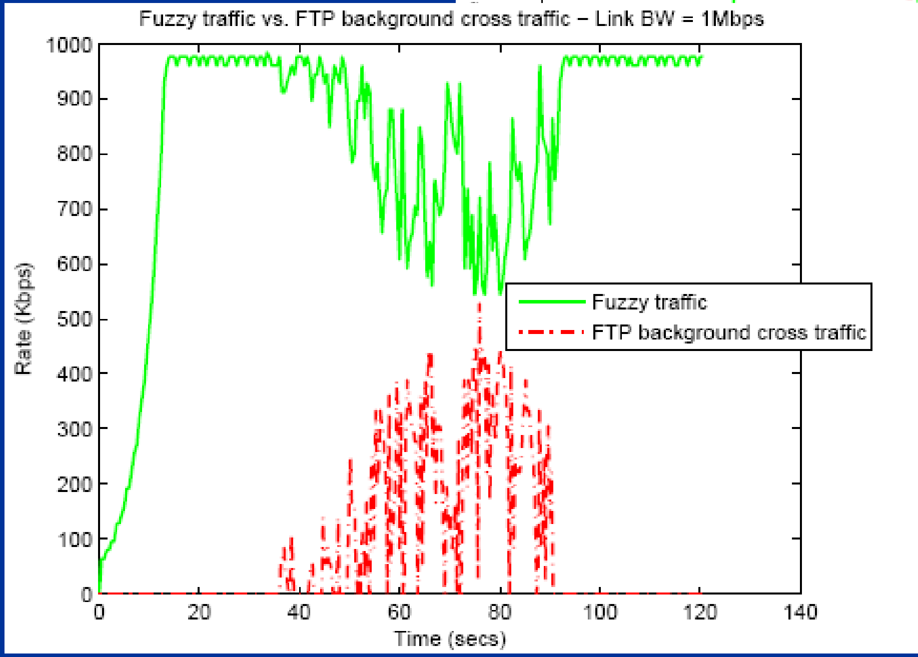
Responsiveness

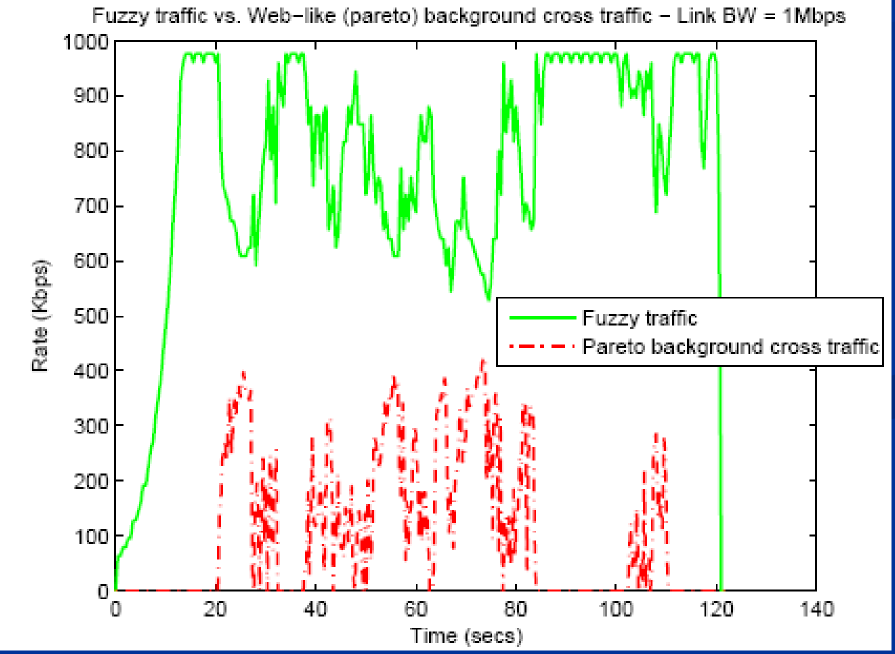
FTP cross-traffic (bursty)

CBR cross-traffic

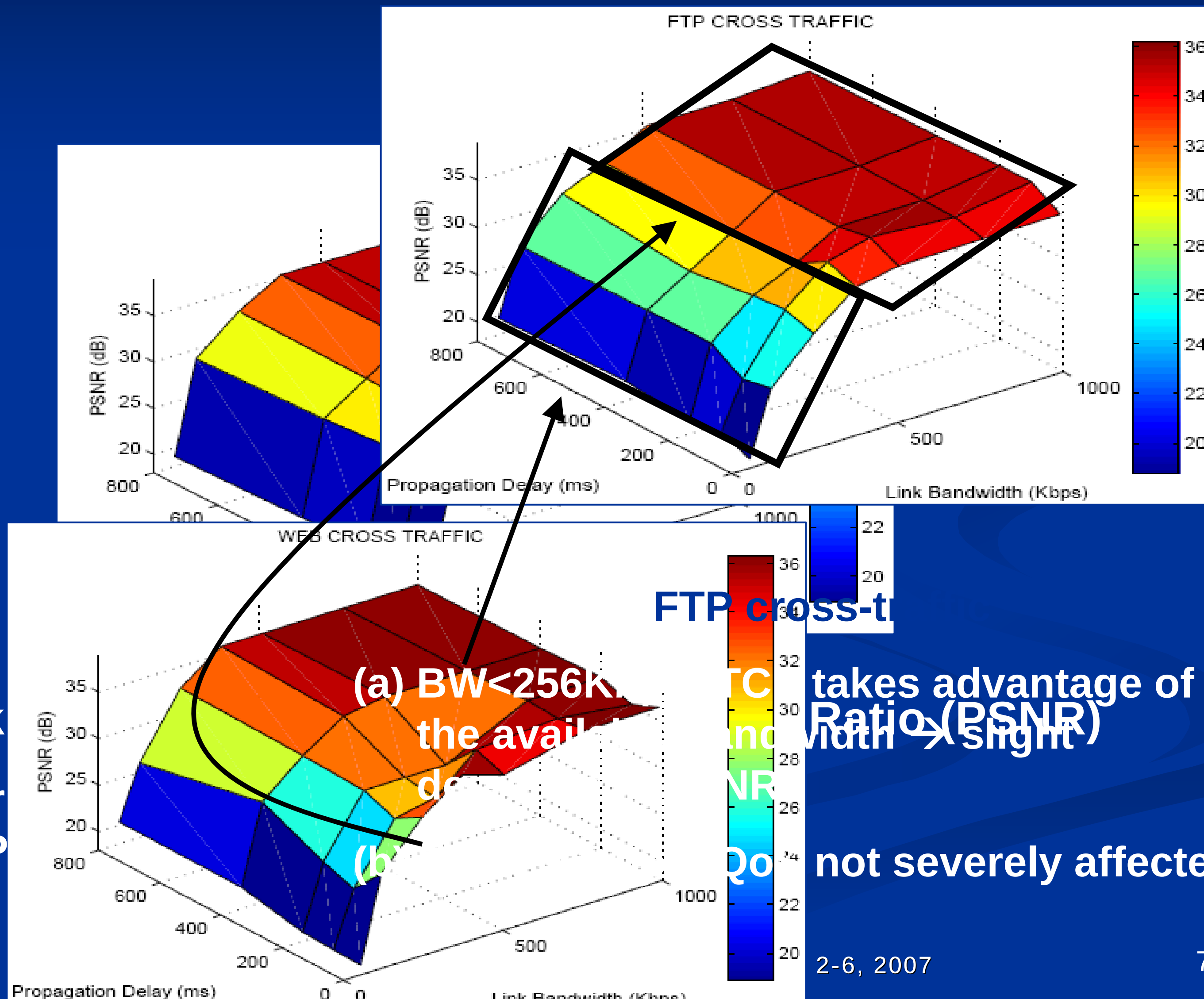
Web cross-traffic (bursty)







QoS: Link BW Perspective



FTP cross-t

Higher link

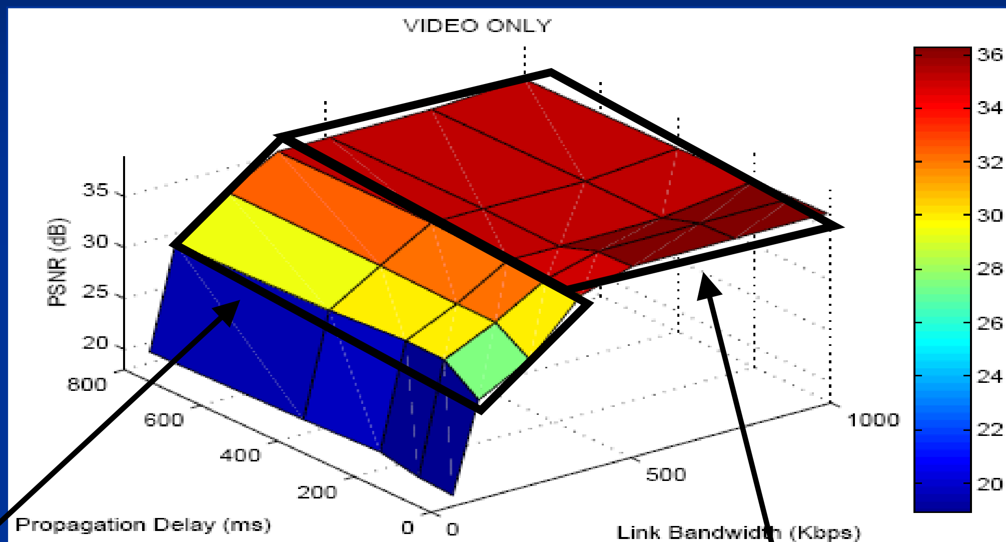
Web cross-traffic
More aggressive than FTP → lower PSNR than in FTP scenarios

(a) BW<256K... TC... takes advantage of the avail... bandwidth → slight de... PSNR

Ratio (PSNR)

(b) ... Qo... not severely affected

QoS: Prop. Delay Perspective

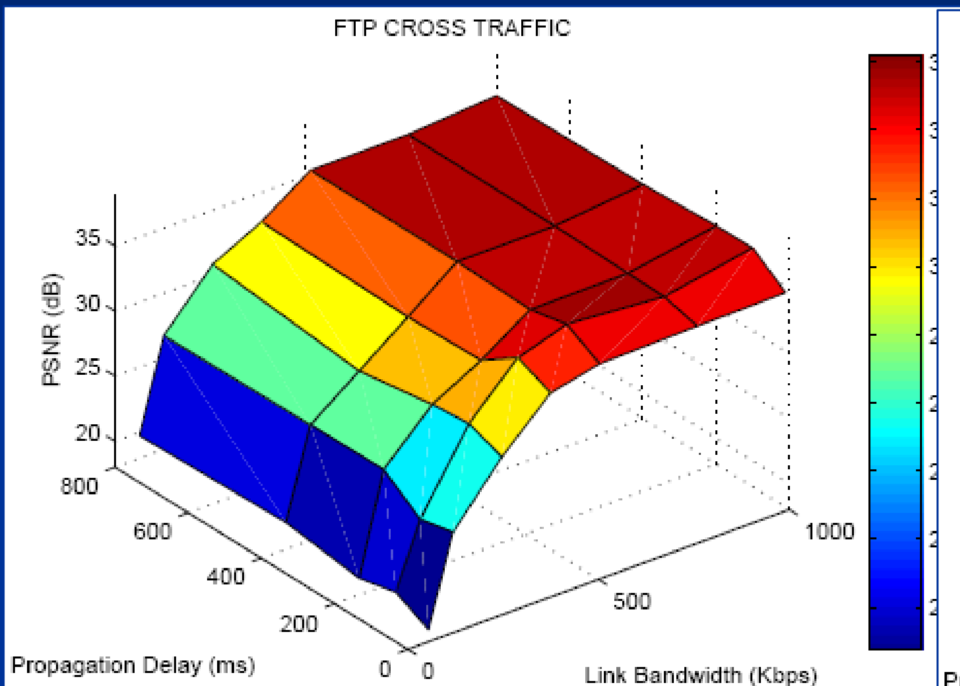


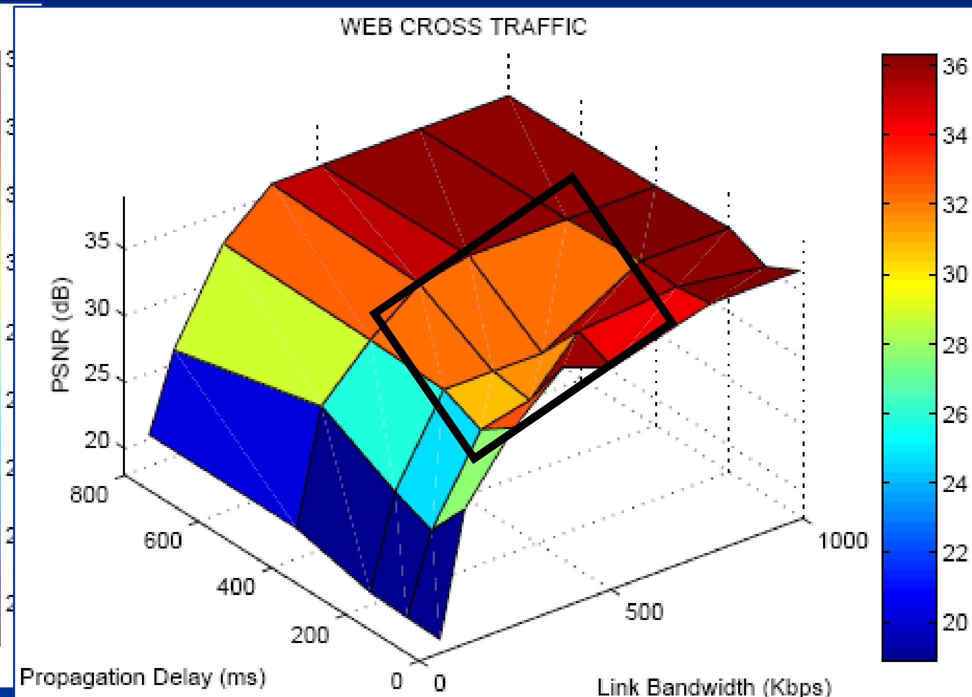
128Kbps<BW<256Kbps: Shorter Prop. Delay → lower PSNR

BW>512Kbps: Shorter Prop. Delay → higher PSNR

- Long propagation delay → delayed decision-making → slow pace of adaptation + High BW (>512Kbps) → smaller PSNR
- Short propagation delay → quick decision-making → fast pace of adaptation + Low BW (<256Kbps) → fast transitions (higher packet loss) → smaller PSNR

QoS: Prop. Delay Perspective





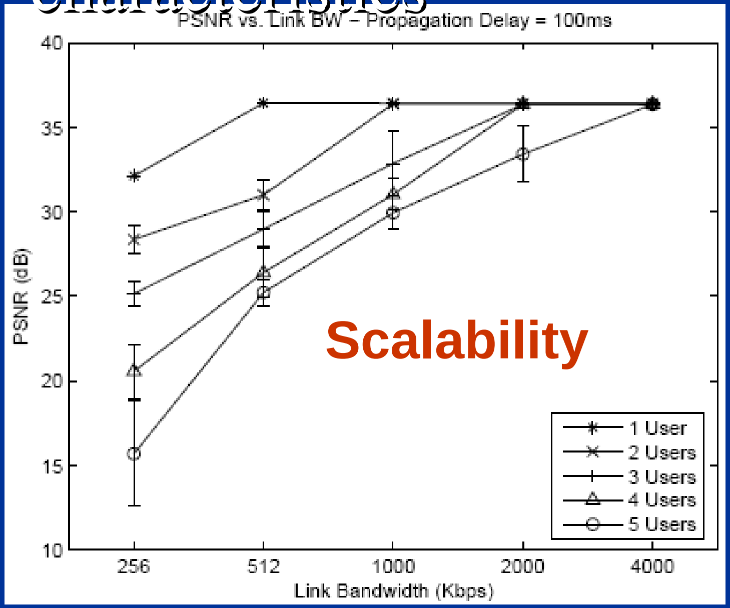
Shorter Prop. Delay → Lower PSNR:

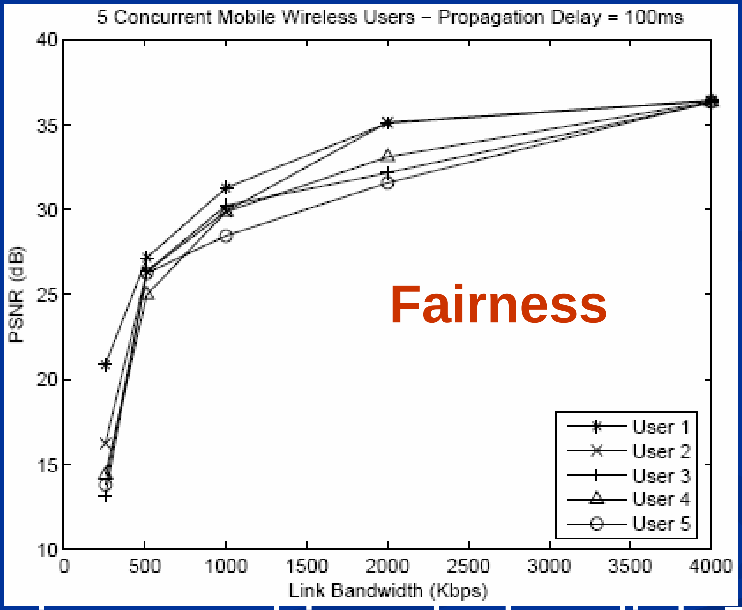
FTP evolves at fast and aggressive pace (TCP-based behavior)

The same behavior as in FTP traffic but here the impact of prop. delay is more severe.

Scalability and Fairness

- Fuzzy decision algorithm operates individually for each user
- Consider multiple identical users with the same connection characteristics



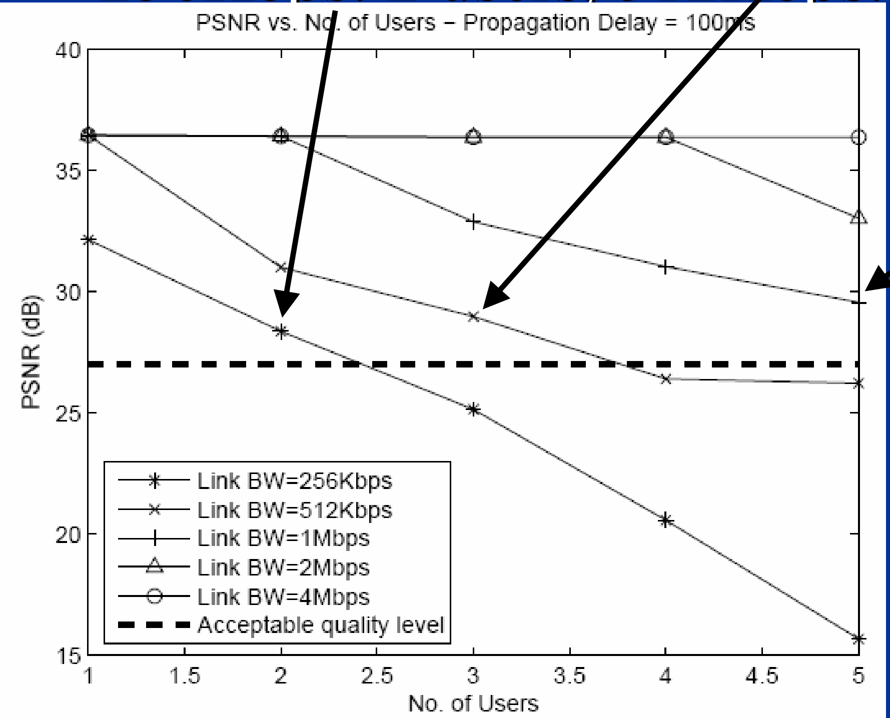


- Bandwidth is shared among all active users ➔ graceful degradation

Fairness achieved when link BW is inadequate of handling aggregated traffic

System Capacity

- If BW high enough to sustain aggregated video transmission rate → all users supported at high quality levels (>27dB *)
- 256Kbps: 2 users, 512Kbps: 3 users, 1Mbps: 5 users



Lowest limit (27dB) for acceptable OQ based on Mean Opinion Score (MOS) categories: GOOD & EXCELLENT (*)

** V. Vassiliou, P. Antoniou, I. Giannakou, and A. Pitsillides "Requirements for the Transmission of Streaming Video in Mobile Wireless Networks," International Conference on Artificial Neural Networks (ICANN), Athens, Greece, September 10-14, 2006.*

Conclusions

- Combination of NATs and CATs to achieve acceptable QoS in unpredictable mobile/wireless environments
- Fuzzy rate controller captures the available BW and finely adapts the video transmission rate
- Responsiveness is maintained
- High Objective Quality (PSNR) in the presence of CBR, FTP and Web cross traffic
- System scales up, offering graceful performance degradation
- Available BW is fairly shared among active users
- Capacity planning

Resources

http://www.NetRL.cs.ucy.ac.cy

NetRL brochure

- Funded projects
  - Almost 5 million Euro
- Research Lab
  - Simulation tools
    - OPNET (60 licences)
    - Ns-2
    - UMTS simulators based on Ns-2 and OPNET
      - developed as part of the EC funded projects SEACORN, B-BONE and C-MOBILE
  - Testbeds (CISCO and LINUX based)
    - Testbeds – Pilot networks
      - CISCO routers, switches, firewall
      - ‘home’ build LINUX based routers and gateways
      - sensor network

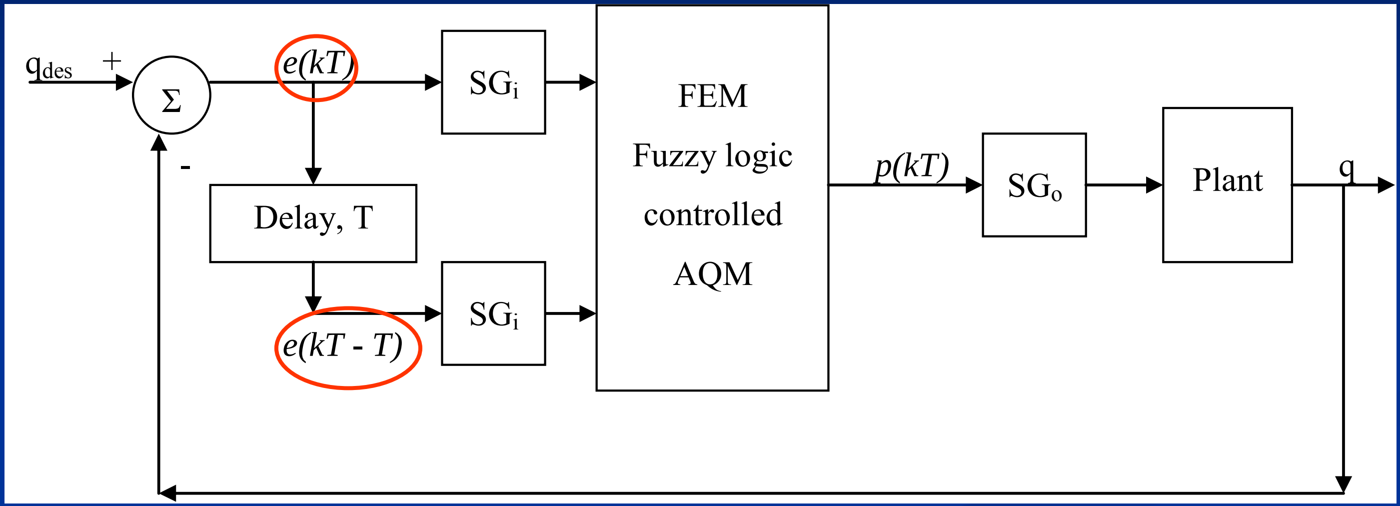
Discussion

Supplementary Slides for Fuzzy CC

Proposed Mechanisms: BE

- **System model of FEM**
  - Adopted most widely used, simplest MISO FLC (Mamdani-based)
    - avoid exponential increase of rule base and increasing controller complexity
  - Design of rule-base is done based on
    - *Completeness:* all kinds of situations of system behaviour are taken into consideration
    - *Consistency:* rule-base does not contain any contradiction
  - Philosophy behind knowledge-base of FEM controller is
    - being aggressive when queue length 'largely' deviates from TQL
    - smoothly respond when queue length is around TQL
    - All other rules represent intermediate situations, thus providing the control mechanism with a highly dynamic action

System model of FEM (cnt'd)

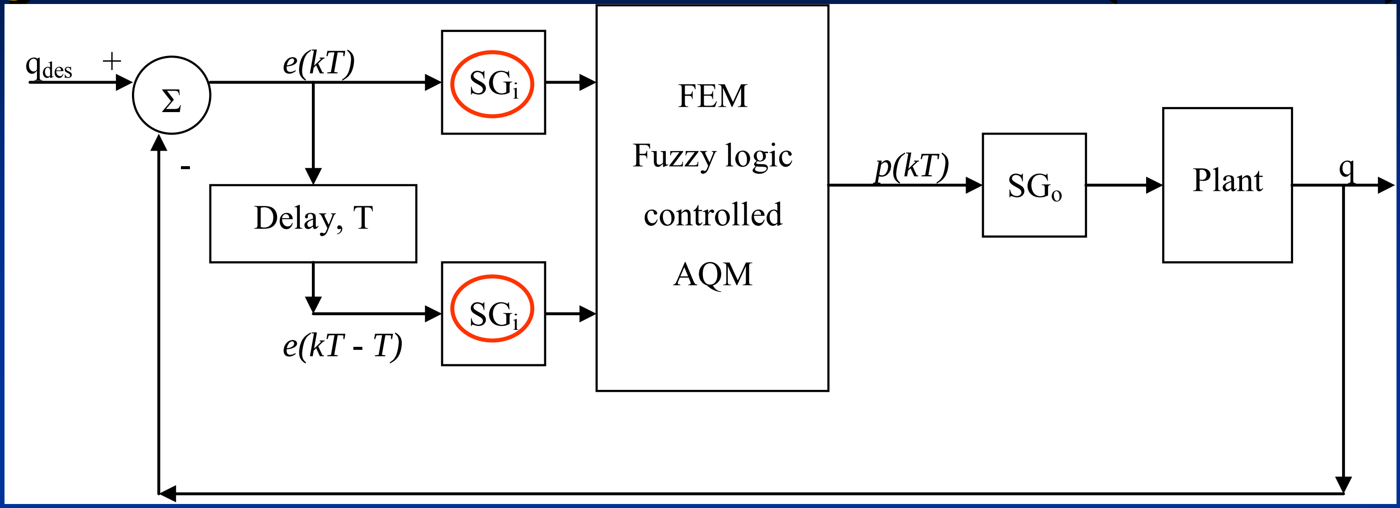


- *e(kT)* is the error on the controlled variable queue length, *q(kT)*, at each sampling interval *kT*

$$e(kT) = q_{des} - q(kT)$$

- *e(kT-T)* is the error on queue length with a delay *T* (at the previous sampling interval)

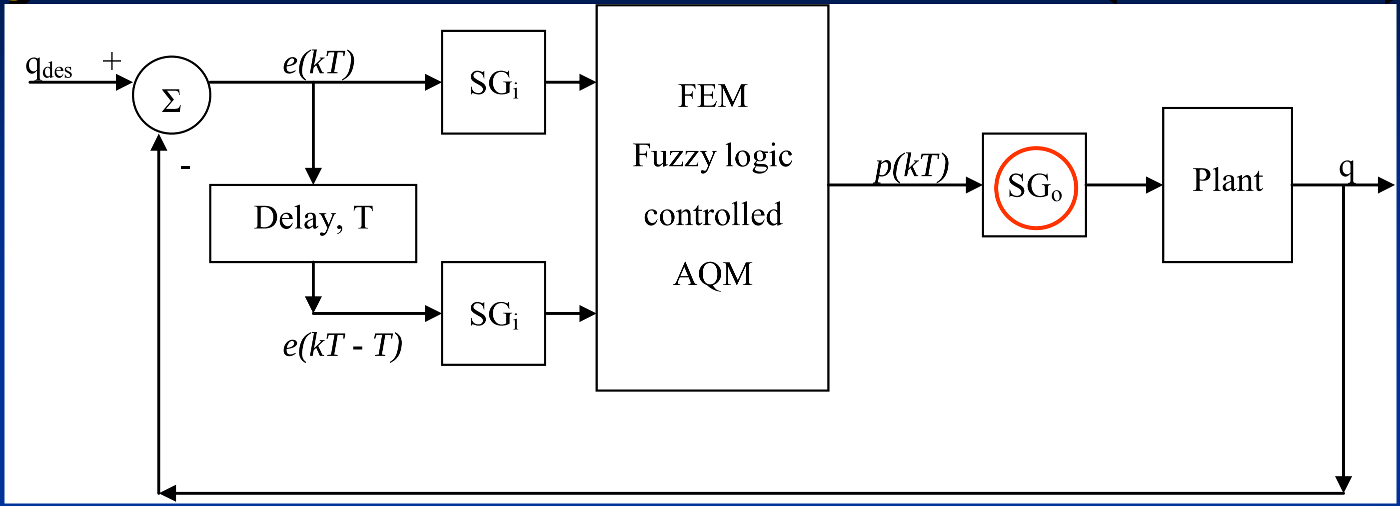
System model of FEM (cnt'd)



- *SGi* are the input scaling gains, determining the range of values (universe of discourse) for a given controller input
- For greater flexibility, and generality, the universe of discourse for each input is "normalized" to the interval [-1, +1], by means of constant scaling factors

$$SG_i = \begin{cases} -\dfrac{1}{q_{des} - BufferSize}, & q_{inst} > q_{des} \\ \dfrac{1}{q_{des}}, & otherwise \end{cases}$$

System model of FEM (cnt'd)



- *SGo*, the output scaling gain, is determined so that the range of outputs that is possible is the maximum, but also ensuring that the input to the plant will not saturate around the maximum.
- Following the approach of Floyd, Gummadi, and Shenker (2001) *SGo* is dynamically set to a value indicating the *maximum mark probability* (initially keep *SGo* to 10%)

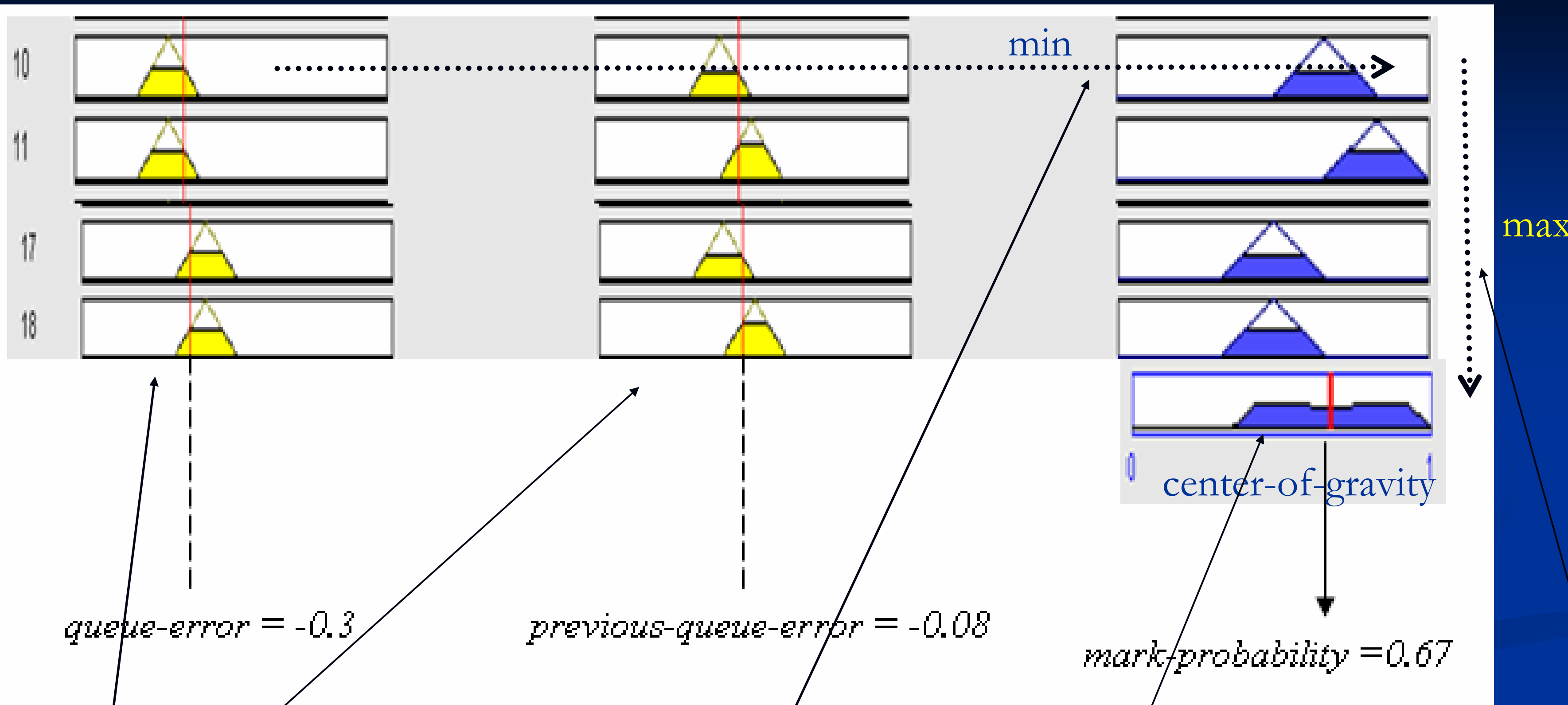
  *IF q(kT) > 1,1*TQL THEN increase SGo by 0,01*
  *IF q(kT) < 0,9*TQL THEN decrease SGo by 0,9*
  } AIMD
- The dynamic selection of *SGo* based on formal adaptive control theory is a subject of future research

System model of FEM

- A Mamdani-based model (most commonly seen in fuzzy literature – adopt due to its simplicity and investigate its suitability through extensive simulations):
  - *Fuzzification*: interprets a crisp/numeric value for *error on queue length (e(kT)* or *e(kT-T)*) into a linguistic value (such as *Negative Very Big)* with a grade of membership.
  - *Fuzzy AND operator used in the IF-part of each rule:* We choose the *min-operation*, indicating that *we can be no more sure about the conjunction of the two statements in the IF-part, than we are about the individual terms that make them up.*
  - *Implication: min-operation* selected for the implication of each rule, as *we can be no more sure about our consequent (THEN-part) than our antecedent (IF-part).*
  - *Aggregation:* The *max-operation* is used to all the resulted output fuzzy sets
  - *Defuzzification:* gives a crisp/numeric value
    - It takes aggregated fuzzy output of rules and results in control input to the plant.
    - We use -most common method- *center of gravity (centroid of area)* of aggregated fuzzy output set *C:*
      - To reduce computations, discretize output universe of discourse *Y* into *m* values.
      - The use of symmetric triangular membership functions makes computation easy.

$$p_k = \frac{\sum_{j=1}^{m} y_j \times \mu_C(y_j)}{\sum_{j=1}^{m} \mu_C(y_j)}$$

Illustrative Example of FEM



For each input each rule visited

minimum membership values of inputs to corresponding linguistic values found

for each corresponding output (for each rule)

linguistic value truncated (using the *min* operator)

using center-of-gravity defuzzification method, compute a numerical value for controller's output

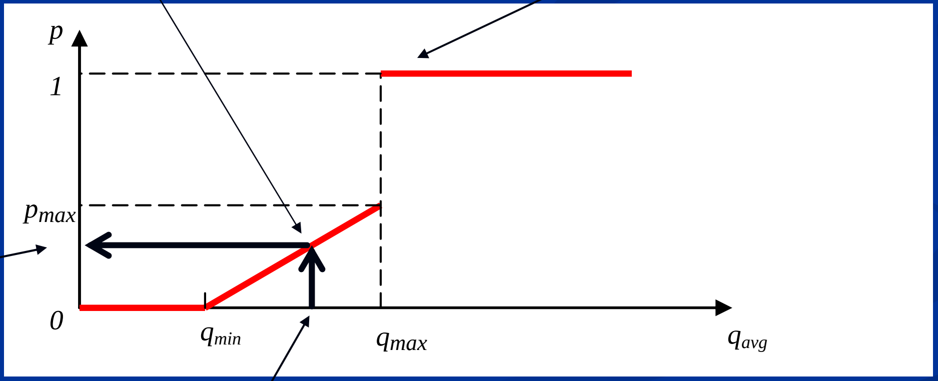
take aggregated of all four implied fuzzy sets using the max operator

Related Work for Best Effort

- Random early detection (RED)
  - first popular AQM mechanism proposed
  - a heuristic-based AQM technique
    - sets some min and max *marking* thresholds in router queues

If average queue between min and max RED randomly ***marks*** packets based on a probability depending on average queue

If average queue exceeds max threshold, every packet is dropped



current probability of packet mark

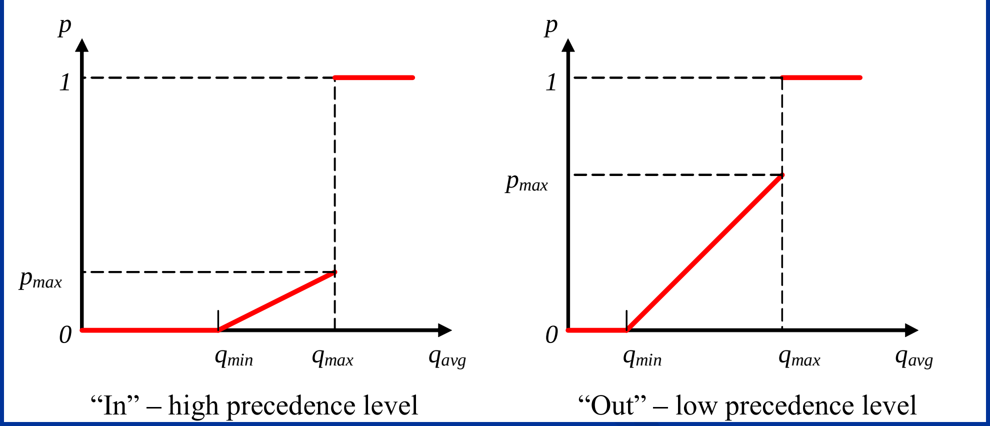
current average queue

Related Work for Best Effort

- Adaptive-RED (**A-RED**)
  - enhancement of RED
    - Dynamically adjusts maximum *mark* probability to keep average queue length half way between min and max thresholds
- Proportional-Integral (**PI**) controller
  - A linear control theory based AQM technique
    - uses classical control theory techniques to stabilize router queue length around a target value
- Random exponential marking (**REM**)
  - An exponentially increasing based probability law
    - uses instantaneous queue size and its difference from a target value to calculate *mark* probability based on an exponential law
- Adaptive Virtual Queue (**AVQ**)
  - Virtual queue-based dropping scheme
    - uses a virtual queue to regulate link utilization, rather than queue length

Related Work for Diff-Serv

- Aim to preferentially drop/mark low-priority packets against high-priority packets, during congestion
  - RED In/Out (RIO)
    - most popular implementation based on RED

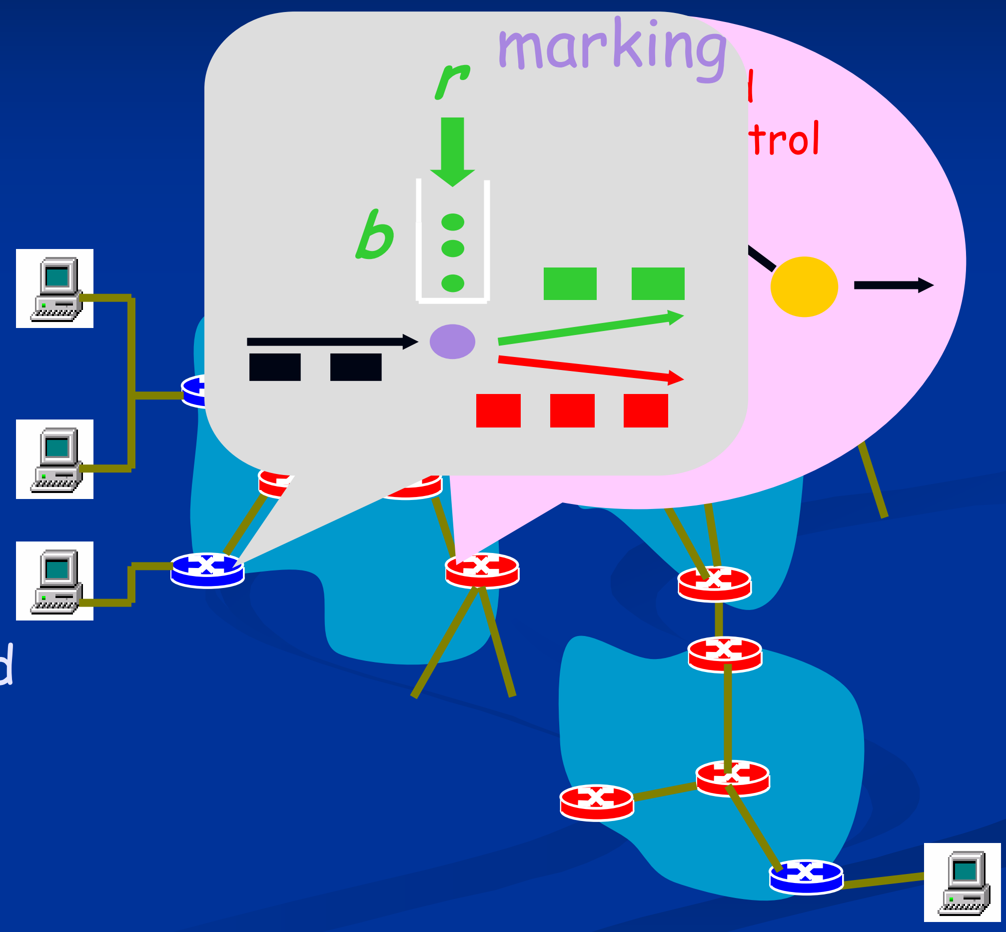


    - For "In" packet, RIO uses average queue length of "In" packets
    - For "Out" packet, RIO uses total average queue length
  - Two-level PI controller (TL-PI)
    - introduces two target queue lengths, which correspond to two levels of drop precedence

Diff-serv Architecture

Edge router:

- per-flow traffic management
- marks packets as in-profile and out-profile



Core router:

- per class traffic management
- buffering and scheduling based on marking at edge
- preference given to in-profile packets

Simulative Evaluation (cnt’d)

- examine effects on AQM schemes, of **wide varying network conditions**:
  - dynamic traffic changes – time-varying dynamics
  - traffic load factor
  - heterogeneous propagation delays and different propagation delays at bottleneck links
  - different link capacities
  - introduction of noise-disturbance (background traffic) to the network (e.g. short-lived TCP connections)
  - introduction of reverse-path traffic
  - different types of competing data streams, like TCP/FTP and TCP/Web traffic, as well as unresponsive traffic (UDP-like)

Simulative Evaluation (cnt'd)

- We use both single- and multiple-congested links networks, as well as topologies with congestion at peripheral links.

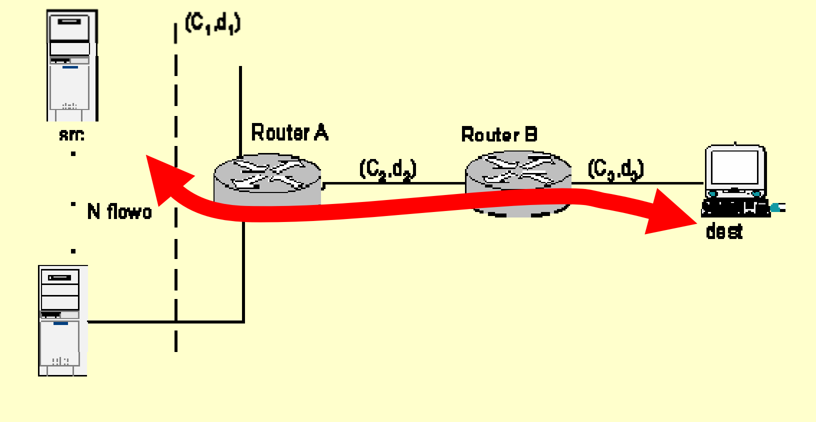


Figure 7.1 Single-bottleneck network topology I

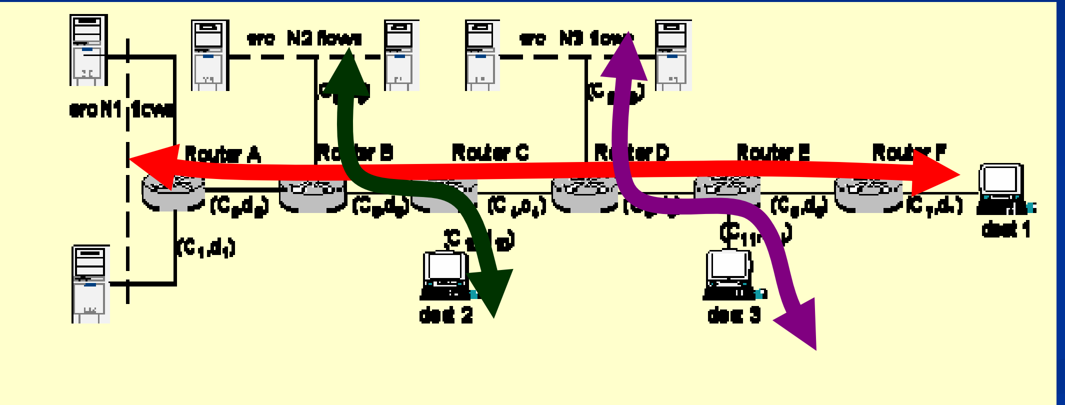


Figure 7.24 Multiple-bottleneck network topology

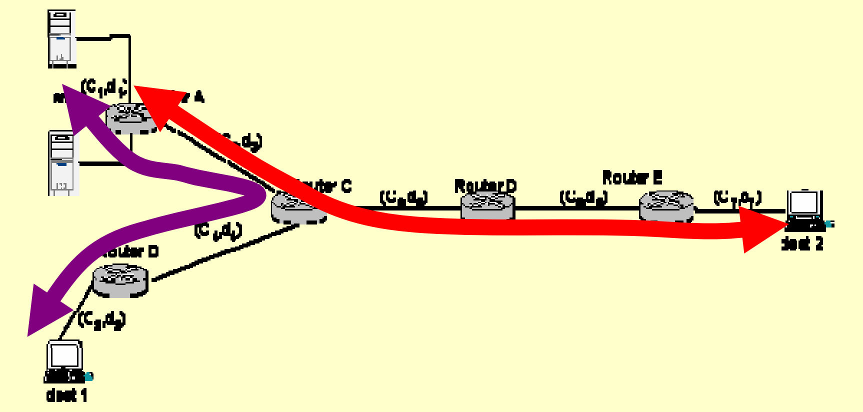


Figure 7.21 A network topology with congestion at peripheral links

General Supplementary slides

Congestion definition

- Simplistic definitions?
  - Congestion is the overloading of communication entities (e.g. switches/routers; sensed by a queue build-up and or packet losses) with arriving packets
  - Congestion occurs when resource demands exceed the capacity (can be viewed as a resource allocation problem)
    - large number of resources such as buffers, link bandwidths, etc. If total traffic wanting to enter link is more than its bandwidth, link is said to be congested
    - Congestion collapse occurs when some resources are consumed by traffic that will be later discarded, and perhaps retransmitted
- A different definition?
  - Network congestion is a state of degraded performance from the perspective of a particular user. A network is said to be congested from the perspective of a user if that user's utility has decreased due to an increase in network load
    - Most of us understand the effect of congestion (e.g. waiting for ever to download a movie trailer when normally it only takes a few minutes). But how do we quantify?

INTERNET-DRAFT TMRG METRICS March 2007

INTERNET-DRAFT TMRG, METRICS March 2007

Table of Contents

no consensus in IETF or research communities about metrics that congestion control should be designed to optimize

document makes this explicit

ICCRG meeting in February 07

Open areas for research in CC

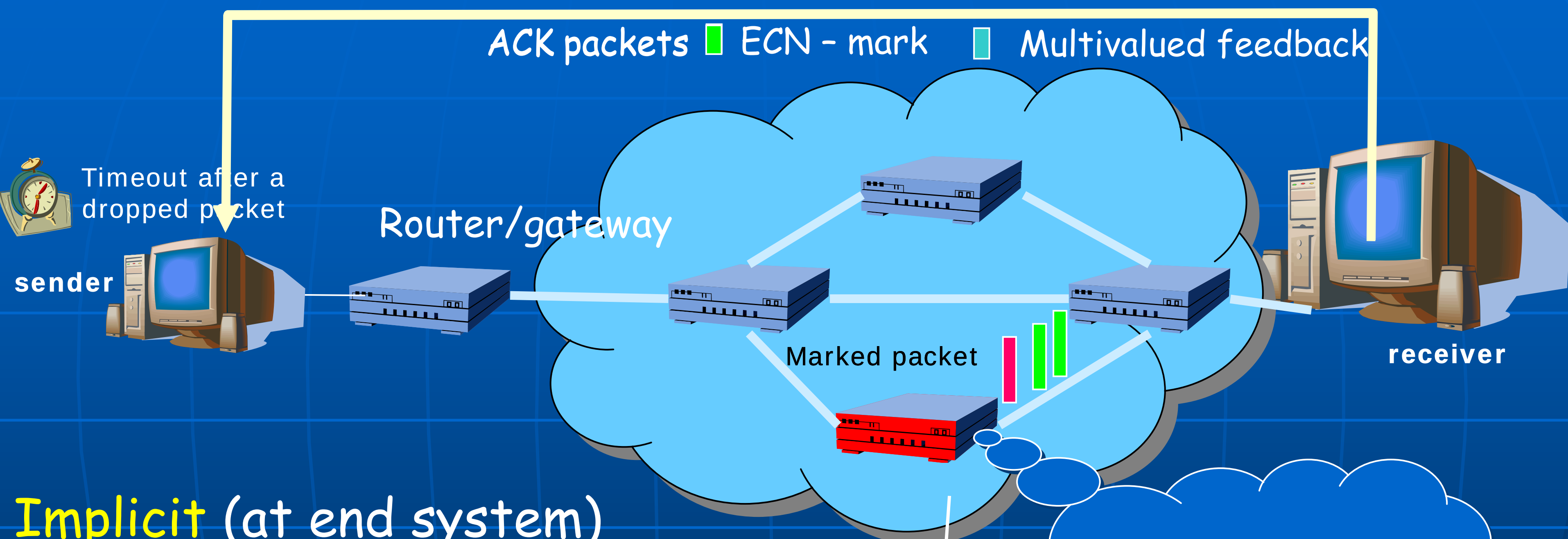
1) high-rate schemes
2) detection and proper treatment of corruption loss
3) media-flows, including small packet flows
4) router support (implicit / explicit feedback, etc.)
5) pseudowires
6) pre-congestion notification
7) precedence for elastic traffic
8) misbehaving senders and receivers

– Real-time media applications

– Impact of VoIP and IPTV

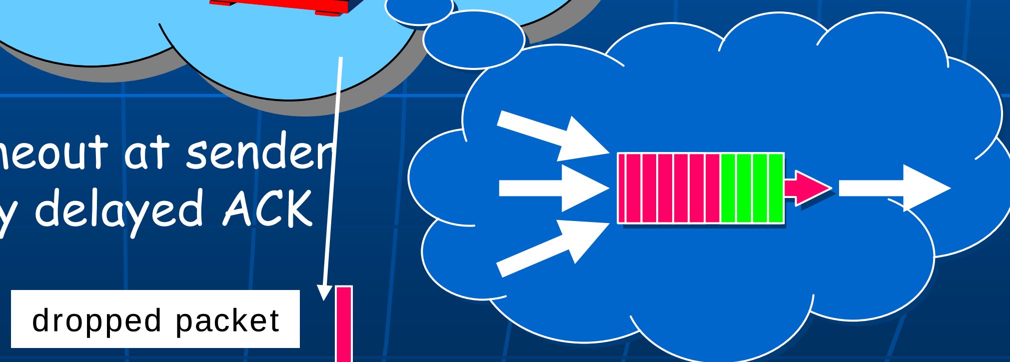
– Interactions with
    - QoS
    - Traffic Engineering
    - Lower-layer technologies

Congestion control: feedback



Implicit (at end system)

- packet loss sensed by timeout at sender
- delay sensed at sender by delayed ACK



Explicit (network assisted)

- binary/single bit (ECN)
  - packet marked at router and sent back to sender by receiver in ACK
- multivalued / ER
  - computed by router, sent back to sender by receiver in ACK

Complex systems

- no universally accepted definition of a “complex system”
  - it is multifaceted, and its definition cannot be compressed into a simple statement
- most researchers agree on many characteristics that would make a system complex
  - presence of nonlinear dynamics in the plant (or process) to be controlled
- other characteristics of complex systems include
  - uncertainties or time-variations in system behaviour
  - operation of the system far from equilibrium

    even if a stable equilibrium exists, the system may be prevented from approaching it by external disturbances or input signals in general