Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_535-1

Single and Multiple Buffer Processing

Sergey I. Nikolenko*[a] and Kirill Kogan[b]
[a]Laboratory of Mathematical Logic, Steklov Institute of Mathematics, St. Petersburg, Russia
[b]IMDEA Networks, Madrid, Spain

Keywords Buffer management policies • Admission control • Online algorithms

Years and Authors of Summarized Original Work

2004; Kesselman, Mansour
2012; Keslassy, Kogan, Scalosub, Segal
2012; Kesselman, Kogan, Segal
2012, 2013; Kogan, López-Ortiz, Nikolenko, Sirotkin
2014; Eugster, Kogan, Nikolenko, Sirotkin

Problem Definition

Buffer management policies are online algorithms that control a limited buffer of packets with homogeneous or heterogeneous characteristics, deciding whether to accept new packets when they arrive, which packets to process and transmit, and possibly whether to push out packets already residing in the buffer. Although settings differ, the problem is always to achieve the best possible competitive ratio, i.e., find a policy with good worst-case guarantees in comparison with an optimal offline clairvoyant algorithm. The policies themselves are often simple, simplicity being an important advantage for implementation in switches; the hard problem is to find proofs of lower and especially upper bounds for their competitive ratios. Thus, this problem is more theoretical in nature, although the resulting throughput guarantees are important tools in the design of network elements. Comprehensive surveys of this field have been given in the past by Goldwasser [9] and Epstein and van Stee [7].

General Model Description

We assume discrete slotted time. A packet is *fully processed* if the processing unit has scheduled the packet for processing for at least its required number of cycles. Each packet may have the following characteristics: (i) *required processing*, i.e., how many processing cycles the packet has to go through before it can be transmitted; (ii) *value*, i.e., how much the packet contributes to the objective function when it is transmitted; (iii) *output port*, i.e., where the packet is headed (in settings with multiple output ports, it is usually assumed that processing occurs independently at each port, so it becomes advantageous to have more busy output ports at a time); and (iv) *size*,

The work of Sergey Nikolenko was partially supported by the Government of the Russian Federation (grant 14.Z50.31.0030).

*E-mail: snikolenko@gmail.com

i.e., how many slots (bytes) a packet occupies in the buffer. The objective of a buffer management policy is to maximize the total value of transmitted packets. Different settings may assume that some characteristics are uniform.

Competitive Analysis

Competitive analysis provides a uniform throughput guarantee for online algorithms across all traffic patterns. An online algorithm ALG is said to be α*-competitive* with respect to some objective function f (for some $\alpha \geq 1$ which is called the *competitive ratio*) if for any arrival sequence σ the objective function value on the result of ALG is at least $1/\alpha$ times the objective function value on the solution obtained by an offline clairvoyant algorithm, denoted OPT.

Problem 1 (Competitive Ratio). *For a given switch architecture, packet characteristics, and an online algorithm* ALG *in a given setting, prove lower and upper bounds on its competitive ratio with respect to weighted throughput (total value of packets transmitted by an algorithm).*

Key Results

Policies and lower and upper bounds on their competitive ratios are outlined according to problem settings; the latter differ in which packet characteristics they assume to be uniform and which are allowed to vary, and additional restrictions may be imposed on admission, processing and/or transmission order, and admissible packet characteristics.

Uniform Processing, Uniform Value, Shared Memory Switch

Since all packets are identical, the problem for a single queue with one output port is trivial. We consider an $M \times N$ shared memory switch that can hold B packets, with a separate processor on each output port. All packets require a single processing cycle and have equal value; the goal is to maximize the number of transmitted packets. Each packet is labeled with an output port where it has to be processed and transmitted.

Non-Push-Out Policies

Kesselman and Mansour [14] show an adversarial logarithmic lower bound: no non-push-out policy can achieve competitive ratio better than $d/2$ for $d = \log_d N$. On the positive side, they present the Harmonic policy that allocates approximately $1/i$ of the buffer to the ith largest queue and, for its variant, the Parametric Harmonic policy, show an upper bound of $c \log_c N + 1$.

Push-Out Policies

The best known policy is Longest Queue Drop (LQD): accept packets greedily if the buffer is not full; if it is, accept the new packet and then drop a packet from the longest queue (destined to the output port with the most packets assigned to it). Aiello et al. [1, 10] show that the competitive

DOI 10.1007/978-3-642-27848-8_535-1

ratio of LQD is between $\sqrt{2}$ and 2; they also provide nonconstant lower bounds for other popular policies and a general adversarial lower bound of $\frac{4}{3}$ on the competitive ratio of any online algorithm.

Uniform Processing, Uniform Value, Multiple Separated Queues

In an $N \times 1$ switch where each of N input queues has a separate independent buffer of size B, a policy must select which input queue to take a packet from and set admission policies for input queues. For uniform values, the problem was closed by Azar and Litichevskey [3] with a deterministic policy with competitive ratio converging to $\frac{e}{e-1} \approx 1.582$ for arbitrary B; a matching lower bound was shown by Azar and Richter [4].

Uniform Processing, Variable Values, Single Queue

Here, there is only one output port (a single queue), and each packet is fully processed in one cycle; however, packets have different values, making it desirable to drop packets with smaller value and process packets of larger value. It is easy to show that the Priority Queue (PQ) policy that sorts packets with respect to values and pushes out smaller values for larger ones is optimal. Research has concentrated on models with additional constraints: non-push-out policies that are not allowed to push admitted packets out and the FIFO model where packets have to be transmitted in order of arrival. Another important special case considers two possible values: 1 and $V > 1$.

Non-Push-Out Policies

Aiello et al. [2] consider five online policies for the two-valued case, considering the specific cases of $V = 1$, $V = 2$, and $V = \infty$. Andelman, Mansour, and Zhu provide a deterministic policy (Ratio Partition) that achieves optimal $\left(2 - \frac{1}{V}\right)$-competitiveness [26]. In the case of arbitrary values between 1 and $V > 1$, they show that the optimal competitive ratio is $\ln V$, proving tightly matching bounds of $1 + \ln V$ and $2 + \ln V + O(\ln^2 V/B)$ [2, 26].

Push-Out Policies

In the FIFO model, there has been a line of adversarial lower bounds culminating in the lower bound of 1.419 shown by Kesselman, Mansour, and van Stee [18] that applies to all algorithms, with a stronger bound of 1.434 for $B = 2$ [2, 26]. As for upper bounds, in this simple model the FIFO greedy push-out policy (accept every packet to end of queue, then push out the packet with smallest value if buffer has overflown) has been shown by Kesselman et al. to be 2-competitive [17]; in the two-valued case, they provide an adversarial lower bound of 1.282, and a long line of improvements for the upper bound has led to the optimal Account Strategy policy of Englert and Westermann [6]. They show an adversarial lower bound of $r = \frac{1}{2}(\sqrt{13} - 1) \approx 1.303$ for any $B \geq 2$ and $r_\infty = \sqrt{2} - \frac{1}{2}(\sqrt{5 + 4\sqrt{2}} - 3) \approx 1.282$ for $B \to \infty$ and show that Account Strategy achieves competitive ratio r for arbitrary B and r_∞ for $B \to \infty$. Thus, in the push-out two-valued case, the gap between lower and upper bounds has been closed completely.

Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_535-1

Uniform Processing, Variable Values, Multiple Separated Queues

Kawahara et al. [11] consider an $N \times 1$ switch with N separated queues, each of which has a distinct buffer of size B and has a value α_j associated with it, $1 = \alpha_1 \leq \ldots \leq \alpha_N = \alpha$. A policy selects one of N queues, maximizing total transmitted value; [11] provides matching lower and upper bounds for the PQ policy as $1 + \frac{\sum_{j=1}^{n'} \alpha_j}{\sum_{j=1}^{n'+1} \alpha_j}$, where $n' = \arg\max_n \frac{\sum_{j=1}^{n} \alpha_j}{\sum_{j=1}^{n+1} \alpha_j}$, and an adversarial lower bound $1 + \frac{\alpha^3+\alpha^2+\alpha}{\alpha^4+4\alpha^3+3\alpha^2+4\alpha+1}$ for any online algorithm. Azar and Richter [4] show that any r-competitive policy for a FIFO queue with variable values yields a $2r$-competitive policy for multiple queues. Kobayashi et al. [21] show that an r-competitive policy for unit values and multiple queues yields a $\min\left\{Vr, \frac{Vr(2-r)+r^2-2r+2}{V(2-r)+r-1}\right\}$-competitive policy for the two-valued case.

Uniform Processing, Variable Values, Shared Memory Switch

Several output queues, each with a processor, share a buffer of size B, and each unit-sized packet is labeled with an output port and an intrinsic value from 1 to V. Eugster, Kogan, Nikolenko, and Sirotkin [8] show a $(\sqrt[3]{V} - o(\sqrt[3]{V}))$ lower bound for the LQD (Longest Queue Drop) policy, an $\frac{1}{2}(\min\{V, B\} - 1)$ lower bound for the MVD (Minimal Value Drop) policy, and a $\frac{4}{3}$ lower bound for the MRD (Maximal Ratio Drop) policy.

Uniform Processing, CIOQ Switches

In CIOQ (Combined Input–Output Queued) switches, one maintains at each input a separate queue for each output (also called Virtual Output Queuing, VOQ). To get delay guarantees of an input queuing (IQ) switch closer to those of an output queuing switch (OQ), one usually assumes increased *speedup* S: the switching fabric runs S times faster than each of the input or the output ports. Hence, an OQ switch has a speedup of N (where N is the number of input/output ports), whereas an IQ switch has a speedup of 1; for $1 < S < N$, packets need to be buffered at the inputs before switching as well as at the outputs after switching. This architecture is called a CIOQ switch.

Uniform Values

Consider an $N \times N$ CIOQ switch with speedup S. Packets of equal size arrive at input ports, each labeled with the output port where it has to leave the switch. Each packet is placed in the input queue corresponding to its output port; when it crosses the switch fabric, it is placed in the output queue and resides there until it is sent on the output link. For unit-valued packets, Kesselman and Rosén [15] proposed a non-push-out policy which is 3-competitive for any S and 2-competitive for $S = 1$. Kesselman, Kogan, and Segal [13] show an upper bound of 4 on the competitiveness of a simple greedy policy.

Variable Values

For up to m packet values in $[1, V]$, Kesselman and Rosén [15] show two push-out policies to be $4S$- and $8\min\{m, 2\log V\}$-competitive. Azar and Richter [5] propose a push-out policy β-PG

Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_535-1

with parameter β; Kesselman et al. [20] show that the competitive ratio of β-PG is at most 7.5 for $\beta = 3$ and at most 7.47 for $\beta = 2.8$. Kesselman and Rosén [16] consider CIOQ switches with PQ buffers (transmit the highest value packet) and show that this policy is 6-competitive for any S.

Uniform Processing, Crossbar Switches

In the buffered crossbar switch architecture, a small buffer is placed on each crosspoint in addition to input and output queues, which greatly simplifies the scheduling process. For packets with unit length and value, Kesselman et al. [20] introduce a greedy switch policy with competitive ratio between $\frac{3}{2}$ and 4 and show a general lower bound of $\frac{3}{2}$ for unit-size buffers. For variable values and PQ buffers, they propose a push-out greedy switch policy with preemption factor β with competitive ratio between $(2\beta - 1)/(\beta - 1)$ (3.87 for $\beta = 1.53$) and $(\beta + 2)^2 + 2/(\beta - 1)$ (16.24 for $\beta = 1.53$). For variable values and FIFO buffers, they propose a β-push-out greedy switching policy with competitive ratio $6 + 4\beta + \beta^2 + 3/(\beta - 1)$ (19.95 for $\beta = 1.67$) [19].

Uniform Values, Variable Processing, Single Queue

In this setting, each packet contributes one unit to the objective function, but different packets have different processing requirements, i.e., they spend a different number of time slots at the processor. We denote maximal possible required processing by k.

Non-Push-Out Policies

For a single queue and packets with heterogeneous processing, non-push-out policies have not been considered in any detail. Kogan, López-Ortiz, Nikolenko, and Sirotkin [23] have shown that any greedy non-push-out policy is at least $\frac{1}{2}(k + 1)$-competitive. It remains an open problem to find non-push-out policies with sublinear competitive ratios or show that none exists.

Push-Out Policies

Keslassy et al. [12] showed that again, for a single queue, PQ (Priority Queue) that sorts packets with respect to required processing (smallest first) is optimal; research has concentrated on the FIFO case, where packets have to be transmitted in order of arrival. Kogan et al. [24] introduced *lazy* policies that process packets down to a single cycle but then delay their transmission until the entire queue consists of such packets; then all packets are transmitted out in as many time slots as there are packets in the queue. In [24], LPO (Lazy Push-Out) was proven to be at most $(\max\{1, \ln k\} + 2 + o(1))$-competitive; [24] also provides a lower bound of $\lfloor \log_B k \rfloor + 1 - O(1/B)$ for both PO (push-out FIFO) and LPO; for large k this bound matches the upper bound up to a factor of $\log B$. Proving a matching upper bound for the PO policy remains an important open problem. In the two-valued case, when packets may have required processing only 1 or k, LPO has a lower bound of $2 - \frac{1}{k}$ and a matching upper bound of $2 + \frac{1}{B}$ [24]. Kogan, López-Ortiz, Nikolenko, and Sirotkin [23] introduce *semi-FIFO* policies, separating processing order from transmission order so that transmission can conform to FIFO constraints while processing order remains arbitrary. Lazy policies thus become a special case of semi-FIFO policies. The authors

Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_535-1

show a general upper bound of $\frac{1}{B}\log_{\frac{B}{B-1}} k + 3$ on the competitive ratio of any lazy policy and a matching lower bound of $\frac{1}{B}\log_{\frac{B}{B-1}} k + 1$ for several processing orders. In the two-valued case, when processing is only 1 or k, this upper bound improves to $2+\frac{1}{B}$, so any lazy policy has constant competitiveness. LPQ (Lazy Priority Queue) also falls in the semi-FIFO class; its competitiveness is between $\left(2-\frac{1}{B}\left\lceil\frac{B}{k}\right\rceil\right)$ and 2 even for arbitrary processing requirements. Kogan et al. [22] consider a generalization with packets of varying size, considering several natural policies and showing an upper bound of $4L$ for one of PO policies, where L is the maximal packet size.

Copying Cost

An important generalization of the heterogeneous processing model was introduced by Keslassy et al. [12]. They attach a penalty α called copying cost to admitting a packet in the queue; thus, the objective function is now $T-\alpha A$, where T is the number of transmitted packets and A is the number of accepted ones, and it becomes less advantageous to push packets out. To deal with copying cost, the authors propose to use β-push-out policies that push a packet out only if its required processing is at least $\beta > 1$ times less than the required processing of a packet which is being pushed out. Keslassy et al. [12] consider the PQ_β policy (Priority Queue with β-push-out) and show that it is at most $\frac{1}{1-\alpha\log_\beta k}\left(1+\log_{\frac{\beta}{\beta-1}}\frac{k}{2}+2\log_\beta k\right)(1-\alpha)$-competitive. Kogan, López-Ortiz, Nikolenko, and Sirotkin [23] show that for any processing order, a β-push-out lazy policy LA_β has competitive ratio at most $\left(3+\frac{1}{B}\log_{\frac{\beta B}{\beta B-1}} k\right)\frac{1-\alpha}{1-\alpha\log_\beta k}$. They show a lower bound $\frac{1-\alpha}{1-\alpha\log_\beta k}$ on the competitive ratio of any β-push-out policy, which matches the additional factor in the upper bound. In the two-valued case, the upper bound becomes $\left(2+\frac{1}{B}\right)\frac{1-\alpha}{1-2\alpha}$, and the authors also show a matching lower bound of $\frac{(2B-2)(1-\alpha)}{(B-1)(1-2\alpha)+(1-\alpha)}$.

Uniform Values, Variable Processing, Multiple Separated Queues

Consider k separate queues of size B each; packets with required processing i fall into the ith queue, and the processor chooses which queue to process on a given time slot. Push-out is irrelevant since queues are independent and packets in a queue are identical. Kogan, López-Ortiz, Nikolenko, and Sirotkin [25] show linear lower bounds for several seemingly attractive policies: $\frac{1}{2}\min\{k,B\}$ for LQF (Longest Queue First), k for SQF (Shortest Queue First), $\frac{3k(k+2)}{4k+16}$ for PRR (Packet Round Robin), and an almost linear lower bound of $\frac{k}{H(k)}$, where $H(k)=\sum_{i=1}^{k}\frac{1}{i}\approx \ln k+\gamma$, for CRR (Cycle Round Robin). They introduce a policy called MQF (Minimal Queue First) that processes packets from a nonempty queue with minimal processing requirement. They show that MQF is at least $\left(1+\frac{k-1}{2k}\right)$-competitive and prove a constant upper bound of 2. For the two-valued case with two queues, 1 and k, Kogan et al. [25] show exactly matching lower and upper bounds for MQF of $1+\left(1+\left\lfloor\frac{aB-1}{b}\right\rfloor\right)/\left(B+\left\lceil\frac{1}{a}\left(b\left\lfloor\frac{aB-1}{b}\right\rfloor+1\right)\right\rceil\right)$.

Uniform Values, Variable Processing, Shared Memory Switch

In this setting, multiple queues with shared memory are implemented in the same way as for uniform processing and heterogeneous values: there are N output ports, each output port manages

DOI 10.1007/978-3-642-27848-8_535-1

a single output queue Q_i, and each output queue collects packets with the same processing requirement (so packets in a given queue are identical).

Non-Push-Out Policies

Eugster, Kogan, Nikolenko, and Sirotkin [8] consider non-push-out policies and show that NHST (Non-Push-Out Harmonic with Static Threshold: $|Q_i|$ is bounded by $\frac{B}{r_i Z}$) is $(kZ + o(kZ))$-competitive, NEST (Non-Push-Out with Equal Static Threshold: $|Q_i|$ is bounded by B/n) is $(N + o(N))$-competitive, NHDT (Non-Push-Out with Harmonic Dynamic Threshold: accept into Q_i if $\sum_{s=1}^{m} |Q_{j_s}| < \frac{B}{H_k}\left(1 + \frac{1}{2} + \ldots + \frac{1}{m}\right)$, where $j_1 \ldots j_m = i$ are queues for which $|Q_j| \geq |Q_i|$) is $(\frac{1}{2}\sqrt{k \ln k} - o(\sqrt{k \ln k}))$-competitive; finding better non-push-out policies is an open problem.

Push-Out Policies

The work [8] also shows lower bounds on the competitive ratio of well-known policies: $(\sqrt{k} - o(\sqrt{k}))$ for LQD (Longest Queue Drop), $(\ln k + \gamma)$ for BQD (Biggest Packet Drop), and $\left(\frac{4}{3} - \frac{6}{B}\right)$ for LWD (Largest Work Drop). The main result of [8] is that LWD is at most 2-competitive.

Open Problems

1. Close the gap between competitive ratios $\frac{4}{3}$ (lower bound for any policy) and 2 (upper bound for LQD) in the uniform processing, uniform value case.
2. Do there exist policies with constant competitive ratio in the uniform processing, variable values, shared memory multiple output queues setting?
3. Do there exist non-push-out policies with sublinear competitive ratio in the case of a single queue with packets with variable processing and uniform values?
4. Prove an upper bound on the competitiveness of PO (push-out) policy in the single-queue FIFO model with heterogeneous required processing and uniform values.
5. Do there exist non-push-out policies with logarithmic competitive ratio in the case of multiple output ports with shared memory that contain packets with variable processing and uniform values?
6. Design efficient policies for CIOQ and crossbar switches with packets with heterogeneous processing and uniform values; prove bounds on their competitive ratios.
7. Design efficient policies and prove bounds on their competitive ratios for the case of packets with both variable values and heterogeneous processing requirements in all of the above settings.

Cross-References

▸ Packet Switching in Multi-Queue Switches General
▸ Packet Switching in Single Buffer

DOI 10.1007/978-3-642-27848-8_535-1

Recommended Reading

1. Aiello W, Kesselman A, Mansour Y (2008) Competitive buffer management for shared-memory switches. ACM Trans Algorithms 5(1):1–16
2. Andelman N, Mansour Y, Zhu A (2003) Competitive queueing policies for QoS switches. In: Proceedings of the 4th annual ACM-SIAM symposium on discrete algorithms, Baltimore, pp 761–770
3. Azar Y, Litichevskey A (2006) Maximizing throughput in multi-queue switches. Algorithmica 45(1):69–90
4. Azar Y, Richter Y (2005) Management of multi-queue switches in QoS networks. Algorithmica 43(1-2):81–96
5. Azar Y, Richter Y (2006) An improved algorithm for CIOQ switches. ACM Trans Algorithms 2(2):282–295
6. Englert M, Westermann M (2009) Lower and upper bounds on FIFO buffer management in QoS switches. Algorithmica 53(4):523–548
7. Epstein L, van Stee R (2004) Buffer management problems. SIGACT News 35(3):58–66
8. Eugster P, Kogan K, Nikolenko SI, Sirotkin AV (2014) Shared-memory buffer management for heterogeneous packet processing. In: Proceedings of the 34th international conference on distributed computing systems, Madrid
9. Goldwasser M (2010) A survey of buffer management policies for packet switches. SIGACT News 41(1):100–128
10. Hahne EL, Kesselman A, Mansour Y (2001) Competitive buffer management for shared-memory switches. In: 13th ACM symposium on parallel algorithms and architectures, Crete Island, pp 53–58
11. Kawahara J, Kobayashi K, Maeda T (2012) Tight analysis of priority queuing policy for egress traffic. CoRR abs/1207.5959
12. Keslassy I, Kogan K, Scalosub G, Segal M (2012) Providing performance guarantees in multipass network processors. IEEE/ACM Trans Netw 20(6):1895–1909
13. Kesselman A, Kogan K, Segal M (2008) Best effort and priority queuing policies for buffered crossbar switches. In: Structural information and communication complexity, 15th international colloquium (SIROCCO 2008), Villars-sur-Ollon, 170–184 http://dx.doi.org/10.1007/978-3-540-69355-0_15
14. Kesselman A, Mansour Y (2004) Harmonic buffer management policy for shared memory switches. Theor Comput Sci 324(2-3):161–182
15. Kesselman A, Rosén A (2006) Scheduling policies for CIOQ switches. J Algorithms 60(1):60–83
16. Kesselman A, Rosén A (2008) Controlling CIOQ switches with priority queuing and in multistage interconnection networks. J Interconnect Netw 9(1/2):53–72
17. Kesselman A, Lotker Z, Mansour Y, Patt-Shamir B, Schieber B, Sviridenko M (2004) Buffer overflow management in QoS switches. SIAM J Comput 33(3):563–583
18. Kesselman A, Mansour Y, van Stee R (2005) Improved competitive guarantees for QoS buffering. Algorithmica 43(1-2):63–80
19. Kesselman A, Kogan K, Segal M (2010) Packet mode and QoS algorithms for buffered crossbar switches with FIFO queuing. Distrib Comput 23(3):163–175
20. Kesselman A, Kogan K, Segal M (2012) Improved competitive performance bounds for CIOQ switches. Algorithmica 63(1–2):411–424

Encyclopedia of Algorithms
DOI 10.1007/978-3-642-27848-8_535-1

21. Kobayashi K, Miyazaki S, Okabe Y (2009) Competitive buffer management for multi-queue switches in QoS networks using packet buffering algorithms. In: Proceedings of the 21st ACM symposium on parallelism in algorithms and architectures (SPAA), Portland, OR, USA, pp 328–336
22. Kogan K, López-Ortiz A, Nikolenko S, Scalosub G, Segal M (2014) Balancing Work and Size with Bounded Buffers. In: Proceedings of the 6th international conference on communication systems and networks (COMSNETS 2014), Bangalore, pp 1–8
23. Kogan K, López-Ortiz A, Nikolenko SI, Sirotkin AV (2012) A taxonomy of semi-FIFO policies. In: Proceedings of the 31st IEEE international performance computing and communications conference (IPCCC2012), Austin, pp 295–304
24. Kogan K, López-Ortiz A, Nikolenko SI, Sirotkin AV, Tugaryov D (2012) FIFO queueing policies for packets with heterogeneous processing. In: Proceedings of the 1st Mediterranean conference on algorithms (MedAlg 2012), Ein Gedi. Lecture notes in computer science, vol 7659, pp 248–260
25. Kogan K, López-Ortiz A, Nikolenko SI, Sirotkin A (2013) Multi-queued network processors for packets with heterogeneous processing requirements. In: Proceedings of the 5th international conference on communication systems and networks (COMSNETS 2013), Bangalore, pp 1–10
26. Zhu A (2004) Analysis of queueing policies in QoS switches. J Algorithms 53(2):137–168