Block-based Fair Queuing: An Efficient Network QoS Provisioning Algorithm for High-speed Data Transmission

Shu-Hsin Chang, Wei-Chih Ting, Chun-Yu Chuang and Shih-Yu Wang
Speaker: Shu-Hsin Chang
Date: 2012/05/15

Industrial Technology Research Institute

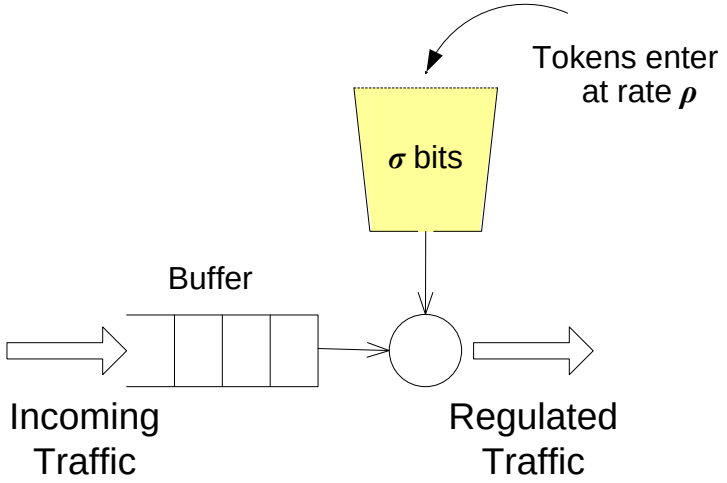
Outline

- Background
- Motivation
- Proposed method
- Simulation
- Summary

Industrial Technology Research Institute

Background

- **Traffic scheduling algorithm**
  - To allocate the limited bandwidth to all of the sessions sharing an outgoing link.
- **Traffic characteristics**
  - Maximum burst size
  - Average arriving rate
- **Performance requirements**
  - Maximum delay
  - Maximum latency
- **Traffic model (Token bucket)**
  - New tokens are continuously filling the bucket at a constant rate
  - The bucket has a maximum volume of token number
  - An arriving packet is released only when it can remove a number of tokens equal to its packet length



Industrial Technology Research Institute

Motivation

- **Premise**
  - There exists tradeoff between packet latency and computational complexity
    - packet latency ↓, computational complexity ↑

Characteristic / Service Discipline	Complexity	Start-up Latency
WFQ	$O(N)$	$p_{i,\max} / r_i + p_{\max} / C$
SCFQ	$O(\log N)$	$p_{i,\max} / r_i + \Sigma^{N}_{j=1} p_{j,\max} / C$
DRR	$O(1)$	$(3F - 2u_i) / C$

- **Observation**
  - All sessions suffer the same performance degradation in a simplified algorithm
- **Question**
  - How to reduce the computation time under the existing tradeoff?
- **Solution**
  - To increase the data length in each scheduling computation in WFQ algorithm
  - To save computation time through parameter setting, instead of applying simplified algorithm

Industrial Technology Research Institute

Previous Work

- **WFQ** (Weighted Fair Queuing)
  - **Concept**
    - Each session is reserved a positive real number as its service weight
    - Sessions are served at rates proportional to their service weights
  - **Mechanism**
    - Each arriving packet is stamped with a service tag
    - Packets are picked up for transmission in increasing order of their tag values.
  - **Service Tag**
    - The service tag for the k^{th} packet on session i is defined as

$$F_{i,0} = 0$$
$$F_{i,k} = \max\{V(a_{i,k}), F_{i,k-1}\} + p_{i,k} / w_i$$

    - **System virtual time**

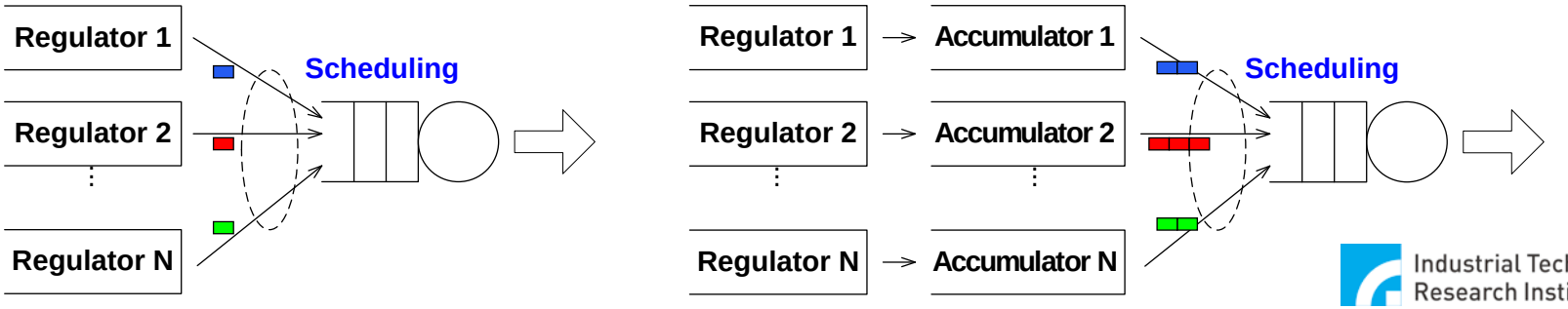
$$V(t_0) = 0$$

$$\frac{dV(t)}{dt} = \frac{1}{\sum_{j \in B(t)} w_j}$$

Industrial Technology Research Institute

Proposed Method

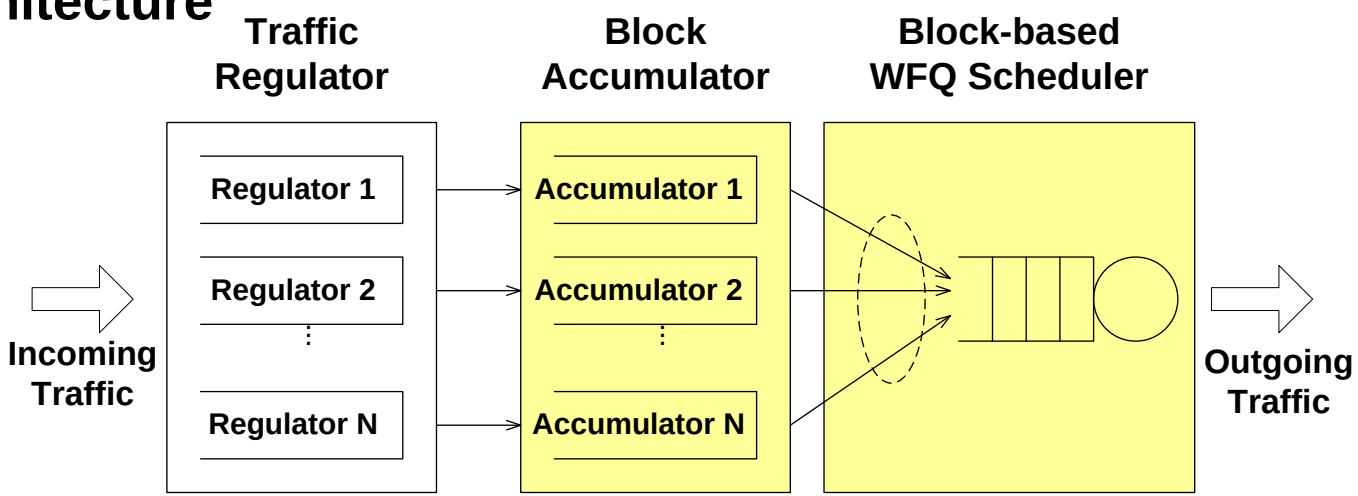
- **Block-based Weighted Fair Queuing (BWFQ)**
  - An extension of WFQ algorithm
  - Two parameters
    - **weight (w_i) :** determine the ratio of service rate
    - **granule (g_i) :** determine the data length for scheduling
- **Concept**
  - Packets from each session are aggregated to blocks in advance
  - The order of data transmission is arranged in unit of block
- **Performance**
  - By assigning great granules to delay-insensitive sessions
    → computation time is saved
  - By assigning small granules to delay-sensitive sessions
    → QoS is guaranteed



Industrial Technology Research Institute

Proposed Method

- **Components**
  - **Traffic Regulator :**
    - To regulate traffic from each session to conform to a token bucket model
  - **Block Accumulator :**
    - To combine packets from each session to form data blocks
  - **WFQ Scheduler :**
    - To sort the blocks from all sessions for transmission according to WFQ algorithm
- **Architecture**



Industrial Technology Research Institute

Block-based WFQ

- **Comparison of complexity**
  - BWFQ has the same complexity of WFQ algorithm

Characteristic / Algorithm	Complexity	Start-up Latency
WFQ	$O(N)$	$p_{i,\max} / r_i + p_{max} / C$
BWFQ	$O(N)$	$(g_i + p_{i,\max}) / r_i + p_{max} / C$

- **Comparison of computation time**
  - BWFQ saves computation time by scheduling the data in unit of block

Component	Traffic Regulator	Block Accumulator	WFQ Scheduler		
			Service tag	Virtual time	Sorting
Formula	$\max\{0,(p_{i,k}-\sigma_i(t))\} / \rho_i$	$\max\{0,(g_i-q_i(t))\} / r_i$	$\max\{V(a_{i,k}), F_{i,k-1}\} + p_{i,k} / C \cdot w_i$	$dV(t)/dt = t / (\Sigma_{j \in B(t)} w_j)$	
Complexity	$O(1)$	$O(1)$	$O(1)$	$O(N)$	$O(\log N)$
WFQ	1/packet	0	1/packet	1/packet	1/packet
BWFQ	1/packet	1/packet	1/block	1/block	1/block

Industrial Technology Research Institute

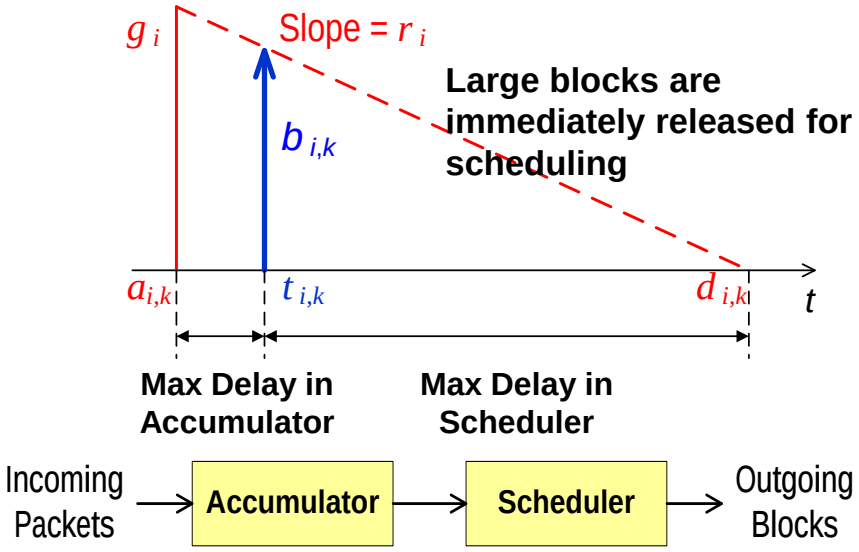
Block-based WFQ

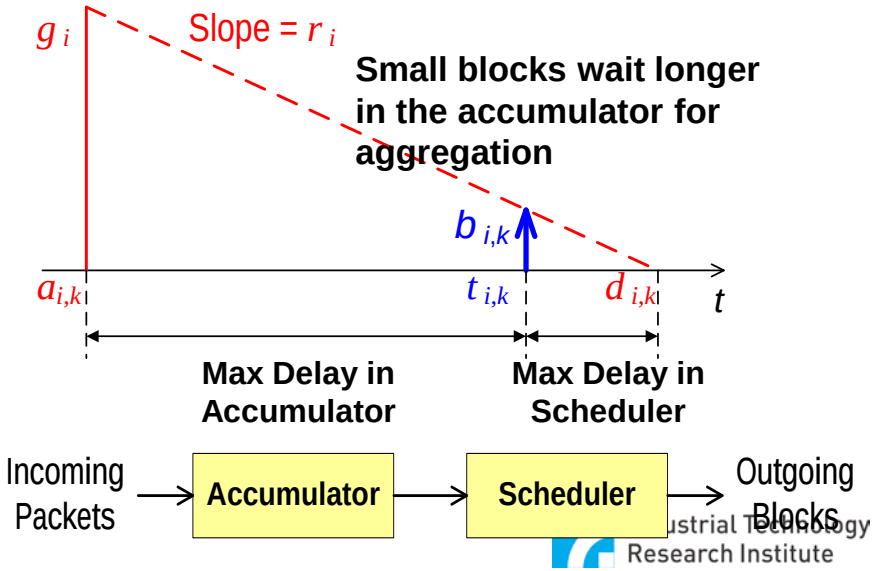
- **Expectation of maximum delay**
  - Mechanism of block aggregation introduces an extra delay no more than g_i/r_i

$$D_i^{*,BWFQ} \leqq D_i^{*,WFQ} + g_i/r_i$$

$D_i^{*,X}$: max delay of X algorithm for session i
g_i : granule of session i
r_i : min service rate for session i
$b_{i,k}$: the k^{th} block size on session i

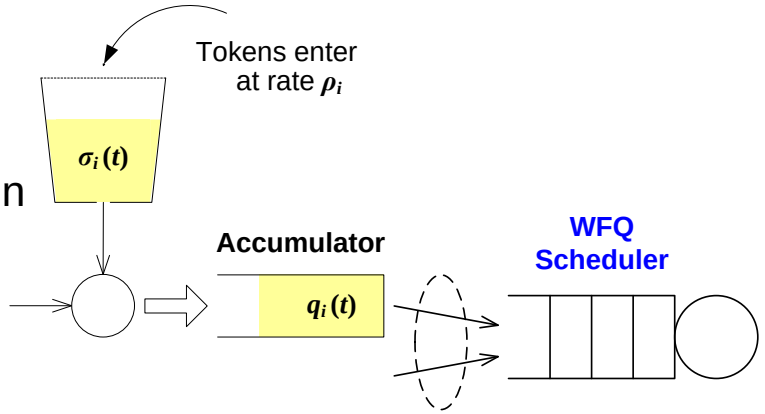
- **Maximum accumulating time**
  - The max service delay in the Scheduler is $b_{i,k}/r_i$
    → The max waiting time in the Accumulator is limited by $(g_i - b_{i,k})/r_k$





Block-based WFQ

- **Potential burst size**
  - For each session i, the data released to Scheduler has a potential burst size of $\sigma_i(t)+q_i(t)$
    - $\sigma_i(t)$ **:** The token number in the Bucket at time t
    - $q_i(t)$ **:** The length of all packets waiting in the Accumulator at time t



- **Influence**
  - The delay upper bound guaranteed by the Scheduler may be broken.
- **Additional constraint**
  - the maximum block size B_i should compensate for the potential increment in burst size.

$$B_i = g_i - \max\{0, \sigma_i(t_{i,1}) + b_{i,1} - \sigma_i\}$$

B_i **:** max block size of session i
g_i **:** granule of session i
$b_{i,1}$ **:** the size of the first block in a busy period
$t_{i,1}$ **:** release time of the first block from Accumulator
σ_i **:** max token number

Industrial Technology Research Institute

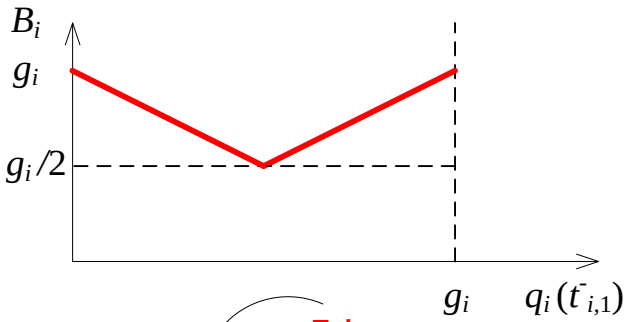
Block-based WFQ

- **The lower bound of maximum block size**
  - ***Theorem*:** For a session i that conforms to a token bucket (σ_i, ρ_i) , where $\rho_i < r_i$

$$B_i = g_i - \max\{0, \sigma_i(t_{i,1}) + b_{i,1} - \sigma_i\}$$

$$> \max\{q_i(t_{i,1}), g_i - q_i(t_{i,1})\} > g_i / 2$$

B_i : max block size of session i in a busy period
g_i : granule of session i
$t_{i,1}$: release time of the first block from Accumulator
$q_i(t)$: the length of all packets of session i waiting in the Accumulator at time t



  - ***Proof*:**
    - Case1) $q_i(t_{i,1}) < g_i / 2$

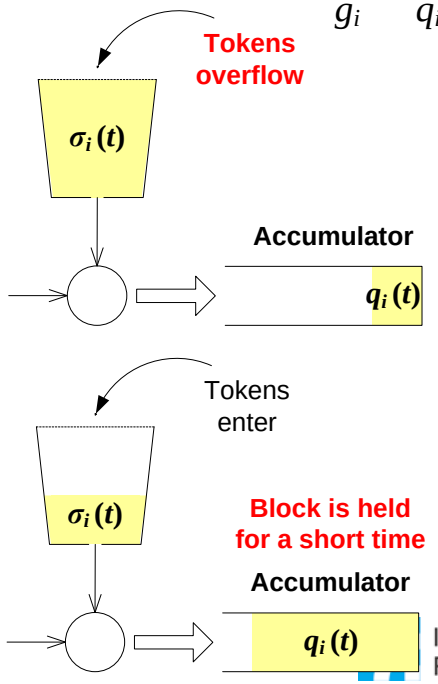
      We can prove that

      $$B_i = g_i - \max\{0, \sigma_i(t_{i,1}) + b_{i,1} - \sigma_i\} > q_i(t_{i,1})$$

    - Case2) $q_i(t_{i,1}) \geqq g_i / 2$

      We can prove that

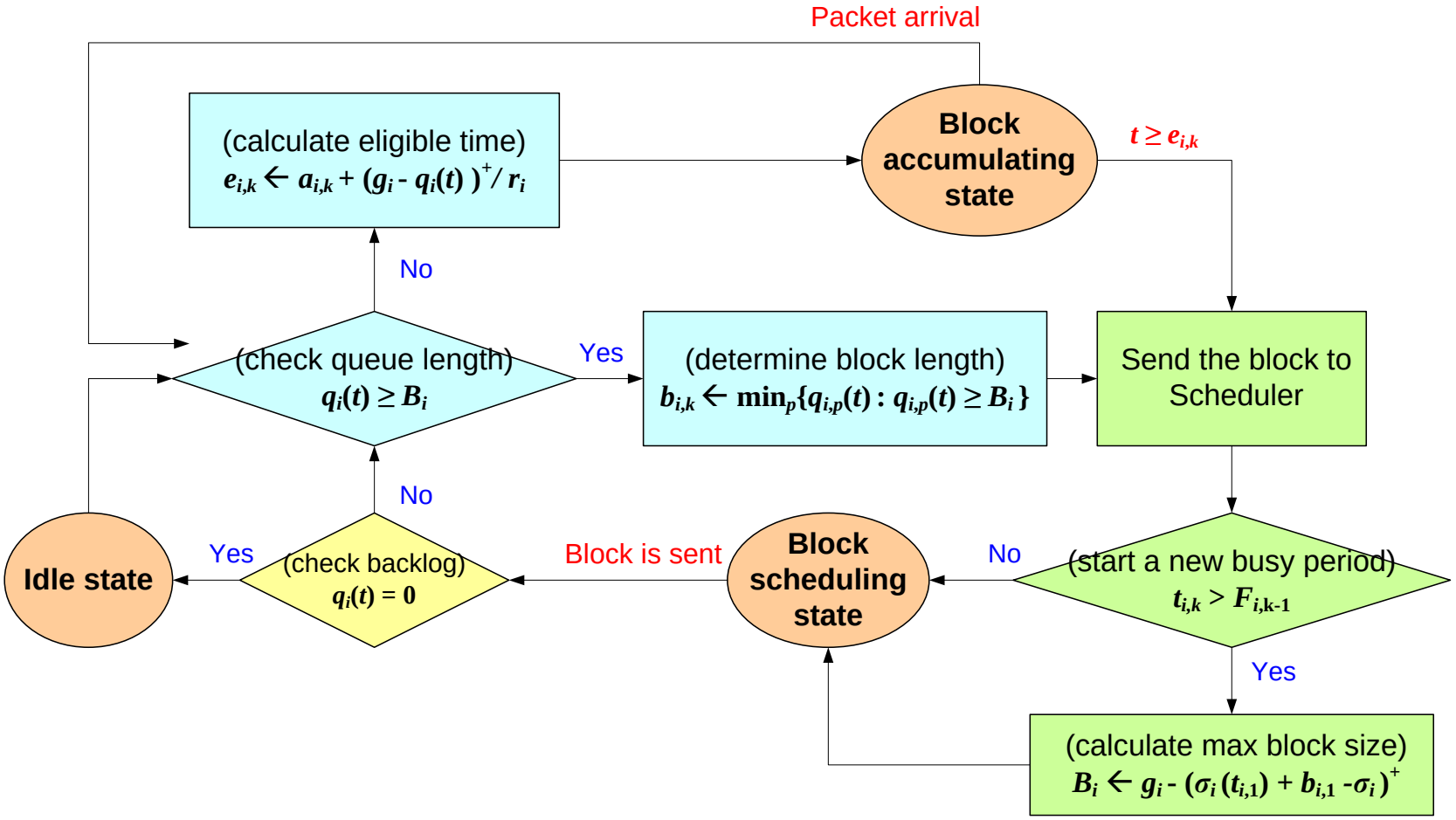
      $$B_i = g_i - \max\{0, \sigma_i(t_{i,1}) + b_{i,1} - \sigma_i\} > g_i - q_i(t_{i,1})$$



Industrial Technology Research Institute

Block-based WFQ

- **Flow chart**



Industrial Technology Research Institute

Block-based WFQ

- **The upper bound of packet delay**
  - ***Theorem*:** Given a session ***i*** that conforms to a token bucket (σ_i, ρ_i), and $\rho_i < r_i$, the BWFQ server guarantees a delay bounds as

$$D_i^{*,BWFQ} \leqq D_i^{*,WFQ} + g_i / r_i$$

$D_i^{*,X}$: max delay of X algorithm for session i
g_i : granule of session i
r_i : min service rate for session i

  - ***Proof***:
    - Packet departure time

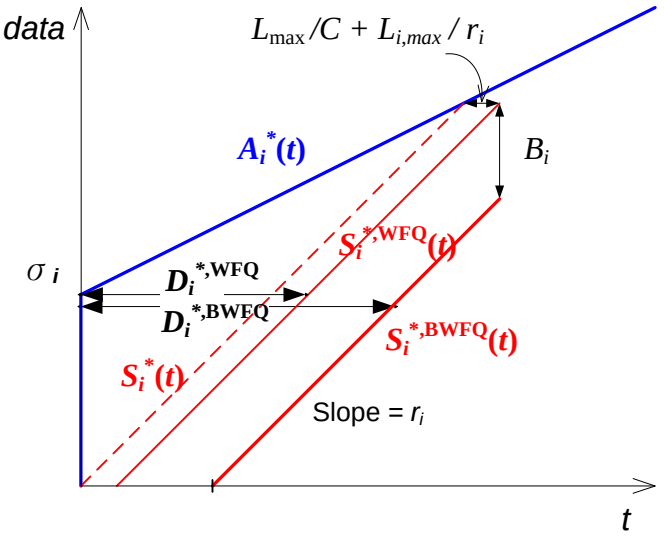
$$d_{i,k} \le \min\left\{t : S_i(t_{i,1}, t) = \sum_{u=1}^{k} b_{i,u}\right\} + \frac{p_{\max}}{C}$$

    - Packet arrival time

$$a_{i,k} \ge \max\left\{\tau : A_i(a_{i,1}, \tau) = \sum_{u=1}^{k-1} b_{i,u}\right\} \ge$$

    - Maximum delay time

$$D_i^{*,BWFQ} = \max_{k \ge 1}\left\{d_{i,k} - a_{i,k}\right\}$$
$$= \max_{\tau > 0}\left\{t - \tau : S_i^*(t) = A_i^*(\tau) + g_i + p_{i,\max}\right\} + p_{\max}/C$$
$$\le \cdots = D_i^{*,WFQ} + g_i/r_i$$



Simulation

- **Simulation Model**

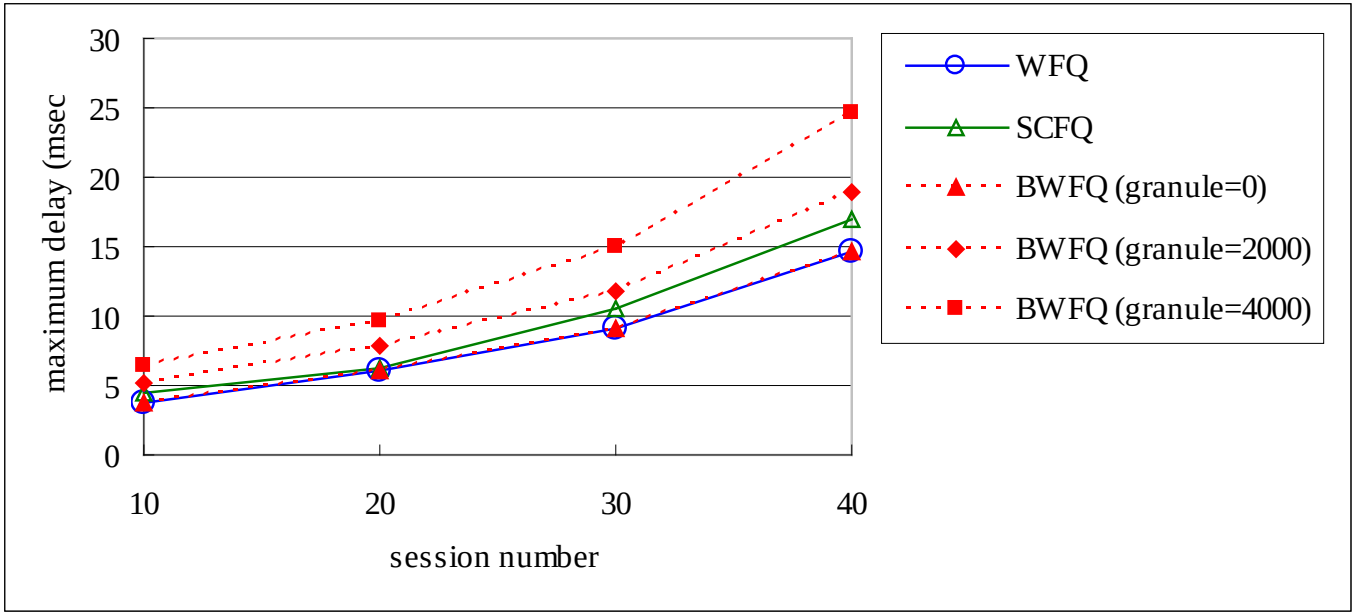
Parameter	Value
Total bandwidth (C)	**100** Mbps
Number of sessions (n)	**10, 20, 30,** and **40**
Granule (g)	**0, 1000, 2000, 3000, 4000** bytes
Source	ON-OFF traffic model
Packet size (l)	Uniformly distributed between **(100,1500)** bytes
Max burst size (σ)	**5000** bytes
Average rate (ρ)	**99 / n** Mbps
Service weight (w)	**1 / n**
Simulation length	**600** sec

- **Algorithm**
  - **WFQ, SCFQ, BWFQ**
- **Metrics**
  - **Maximum delay** : The max packet delay time in the BWFQ server.
  - **Average block size** : The average size of transmission data in each scheduling computation.

Industrial Technology Research Institute

Simulation Result

- **Maximum delay**

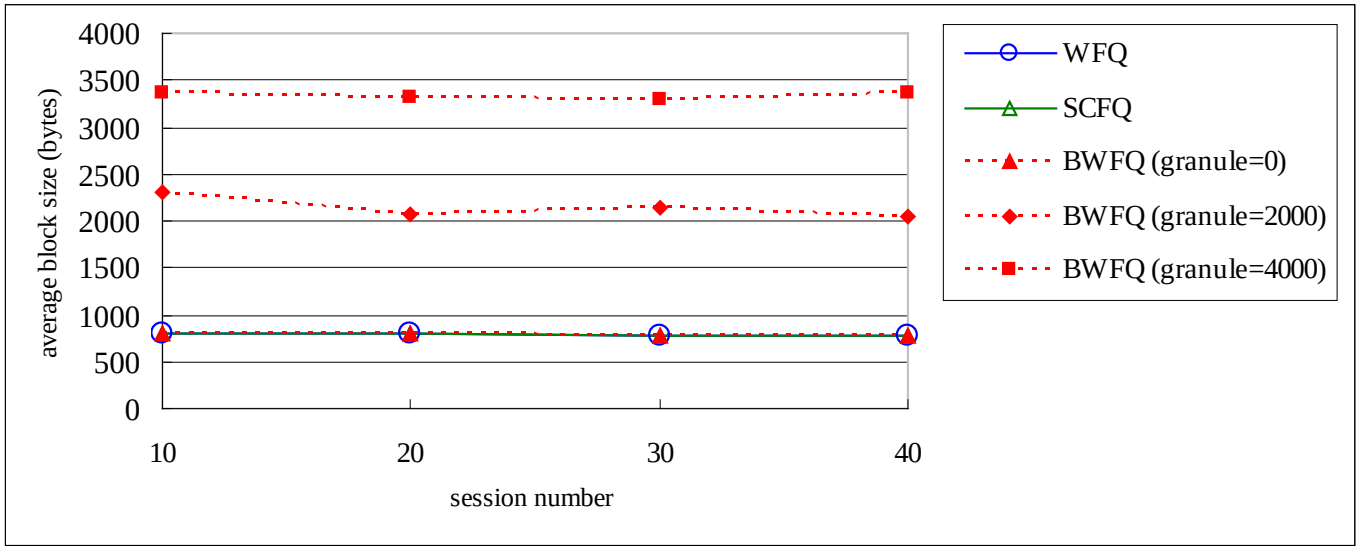


- **Comparison**
  - Compared to SCFQ, BWFQ provides lower delay for sessions with granule = 0
  - Compare to WFQ, the extra delay in BWFQ is bounded by g_i / r_i

Industrial Technology Research Institute

Simulation Result

- **Average block size**



- **Comparison**
  - Compared to WFQ, BWFQ reduces the computation time by increasing the data length in each scheduling computation
  - The average block size increases with granule, and is independent of the session number

Industrial Technology Research Institute

Summary

- We proposed the Block-based WFQ algorithm
  - Concept
    - To allocate a suitable amount of computation resource to each session through parameter setting
  - Method
    - Use two parameters to control the minimum bandwidth and delay upper bound
    - Dynamically aggregate packet in advance
    - Arrange the data transmission order in unit of block
  - Performance
    - Save the computation time by setting great granules for delay-insensitive sessions
    - Guarantee the delay upper bound by setting small granules for delay-sensitive sessions
- Limitation
  - BWFQ has the same complexity as WFQ

Industrial Technology Research Institute

THANK YOU

Shu-Hsin Chang

sh_chang@itri.org.tw

Industrial Technology Research Institute