# Stochastic Evaluation of Fair Scheduling with Applications to Quality-Of-Service in Broadband Wireless Access Networks

**Mohammed Hawa**

Ph.D. Defense
University of Kansas

# Presentation Outline

- Research Motivation.
- A scheduling architecture to support QoS in broadband wireless access networks.
  - Infrastructure MAC Protocol.
  - Support for UGS, rtPS, nrtPS and BE Traffic.
- Upper and lower bounds on mean waiting time of FQ algorithms under Poisson arrivals.
- Further study of three FQ policies (WFQ, SCFQ and SPFQ) using simulation.
- Discussion.

# Motivation

- A QoS architecture for packet switched networks (e.g., the Internet) introduces tools to treat packets differently, thus providing guaranteed services to end users.
- The Internet is expanding to the wireless realm, especially in recent years:
  - WLANs (IEEE 802.11 or Wi-Fi™), BWA Networks (IEEE 802.16 or WiMAX) and 3G Cellular.
- Need to expand QoS to the wireless side.

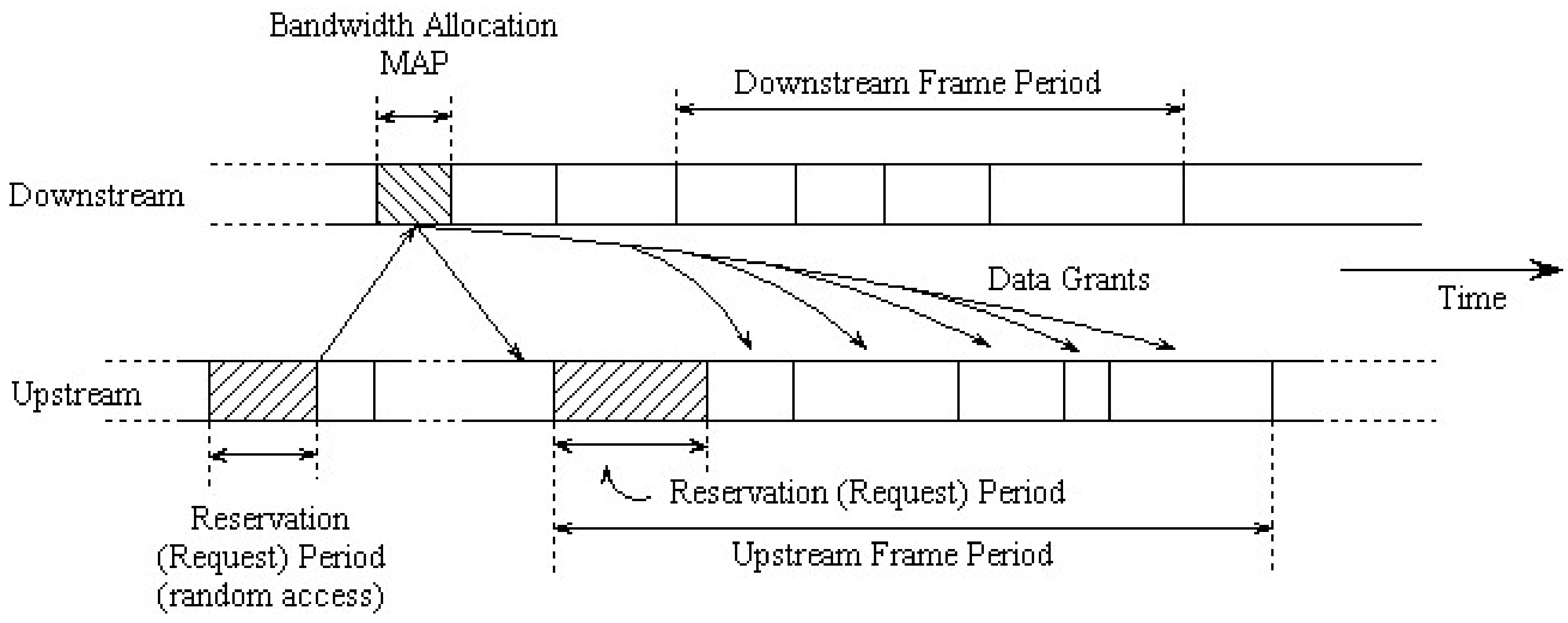
# Wireless QoS

- We introduce a new scheduling architecture to support QoS guarantees in BWA networks.
  - Should integrate with IETF QoS architectures for the wired Internet (IntServ and DiffServ).
  - Several new technical challenges: The wireless channel is a shared medium, bandwidth is a scarce resource, excessive amount of interference.
- QoS Management Functions:
  - Admission Control, Scheduling, Buffer Management and Congestion Control.
  - Concentrate on Scheduling.

# IEEE 802.16 MAC Protocol

- For wireless networks it is natural to integrate the QoS architecture with the MAC protocol.
  - IEEE 802.16 is the industry backed standard for broadband wireless access (BWA) systems.
- In IEEE 802.16, a Base Station (BS) controls many end user Subscriber Stations (SSs).
- Upstream and downstream channels are separated using FDD.
- A request/grant mechanism to coordinate transmission between multiple SSs.

# IEEE 802.16 Operation



# QoS in IEEE 802.16

- To support QoS, IEEE 802.16 defines the concept of a service flow.
- Upstream Service Flow Types in IEEE 802.16:
  - Unsolicited Grant Service (UGS)
  - Real-Time Polling Service (rtPS)
  - Non Real-Time Polling Service (nrtPS)
  - Best Effort (BE)

# Real-Time Service Flows

- **Unsolicited Grant Service (UGS)**:
  - Supports real-time traffic (Voice over IP).
  - Offers fixed size unsolicited data grants (transmission opportunities) on a periodic basis.
- **Real-Time Polling Service (rt-PS)**:
  - Supports real-time flows that generate variable size data packets on a periodic basis (MPEG).
  - Offers periodic unicast request opportunities. The SSs specify the size of the desired data grants.

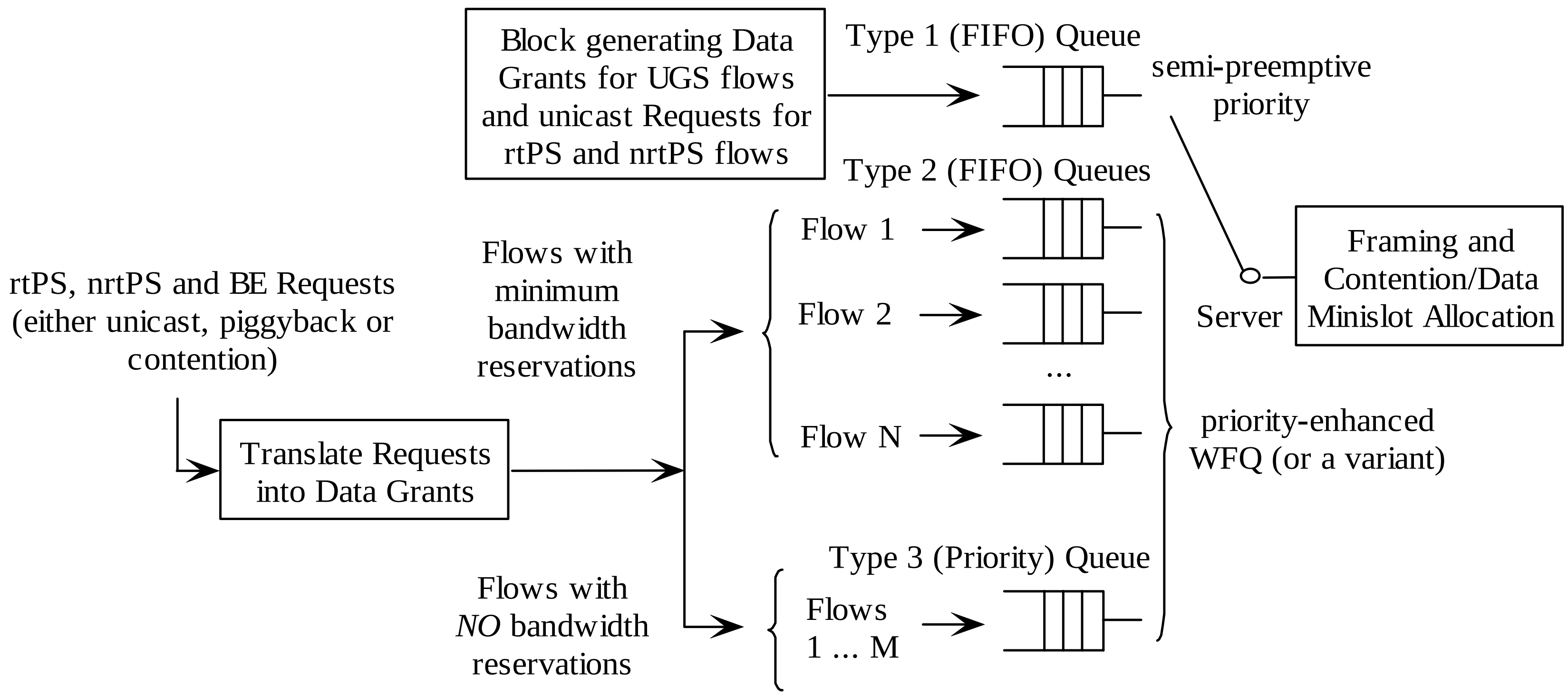
# Non Real-Time Service Flows

- **Non Real-Time Polling Service (nrt-PS)**:
  - Supports flows that require variable size data grants on a regular basis (high bandwidth FTP).
  - Offers infrequent unicast polls plus contention and piggybacking.
- **Best Effort (BE)**:
  - The SS uses contention and piggybacking only.
- Key service parameters for nrt-PS and BE:
  - Minimum Reserved Traffic Rate.
  - Traffic Priority.

# Scheduling in IEEE 802.16

- IEEE 802.16 defines the QoS service flow types, but does not define the scheduling algorithm to be used at the BS.
- IEEE 802.16 is very similar to DOCSIS, the *de facto* standard for HFC Cable Systems.
- No scheduling architectures have been proposed for IEEE 802.16 and those proposed for HFC networks do not specifically address the QoS requirements of DOCSIS (or IEEE 802.16).

# The New BS Scheduler



rtPS, nrtPS and BE Requests (either unicast, piggyback or contention)

Translate Requests into Data Grants

Flows with minimum bandwidth reservations

Flows with *NO* bandwidth reservations

Block generating Data Grants for UGS flows and unicast Requests for rtPS and nrtPS flows

Type 1 (FIFO) Queue

Type 2 (FIFO) Queues

Flow 1

Flow 2

...

Flow N

Type 3 (Priority) Queue

Flows 1 … M

semi-preemptive priority

priority-enhanced WFQ (or a variant)

Server

Framing and Contention/Data Minislot Allocation

# Scheduler Architecture

- Requests arrive at the BS.
  - Through contention, unicast requests and piggybacking.
- Requests are translated into upstream data grants.
- Data grants are scheduled on a frame-by-frame basis by building a corresponding allocation MAP:
  - The hardware block responsible for creating the MAP is represented by a server.
  - Each data grant (or unicast request opportunity) is treated as a packet. Actual transmission of the corresponding data packet takes place in the next frame.
- Data grants are queued in three types of buffers: Type 1, Type 2 and Type 3 buffers.

# Scheduler Advantages

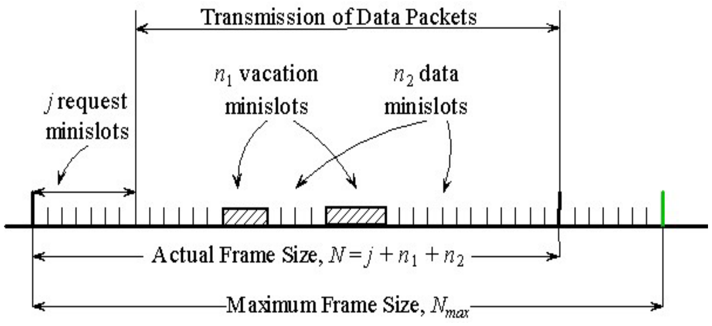
- Easy to implement in hardware thus gaining a performance advantage over software-based alternatives.
- Takes advantage of the Tolerated Jitter parameter for UGS to fit as many packets as possible in the upstream frame thus avoiding fragmentation and being more efficient.
- Lends itself to easier and straightforward performance analysis via classical queuing theory techniques.

# Novel Features

- **Semi-preemptive priority**: A grant undergoing service is sometimes allowed to complete service even if a grant of higher priority arrives in the meantime. This happens only when the newly arriving high priority grant can still be delivered within its deadline.
- **Priority-enhanced WFQ**: If two data grants (from two different queues) have identical WFQ virtual finish times, the first grant to be served is chosen based on its priority level.

# Extra Features

- **Dynamic Contention Minislot Allocation**: An appropriate number of contention request minislots is allocated in each frame period to reduce collisions and to shorten contention resolution.
- A **Buffer Management** mechanism based on RED was also suggested.



Frame 0: Set $j_0 = j_{\min}$ (Initialization)

Frame $i$: Let, $j_i = \max\left\{\frac{3*(N_{\max} - j_{i-1} - n_{1,i-1})}{(k*l_d/l_c)}, j_{\min}\right\}$

If $Q \geq \alpha * (N_{\max} - j_{i-1} - n_{1,i-1})$, Set $j_i = j_{\min}$

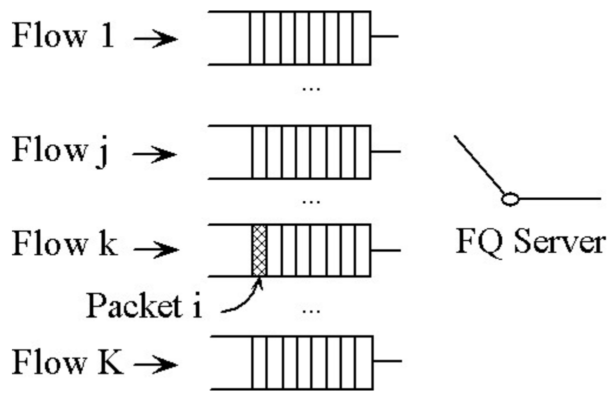
```
Start: $d = 0$
While $d < 7$ do{ (we stop at $d = 7 - 1$)
If $j^d > \beta r_d$, Set $j^d = j^d - \delta$
And set $j^e = j^e + \delta * a_e \Big/ \sum_{f=d+1}^{7} a_f$ for all $e = d+1, ..., 7$
}End While
```

# Analysis of FQ Algorithms

- Our Wireless QoS architecture is built mainly around Fair Queueing (FQ) algorithms.
  - Significant work on **deterministic** worst-case delay analysis of such FQ policies.
  - Little work on **stochastic** delay analysis.
- We introduce a new stochastic analysis method to evaluate upper and lower bounds on **mean** packet delay and **mean** buffer occupancy experienced by FQ policies.

# Background on FQ Policies

- Provide **fair** treatment for supported flows by splitting bandwidth based on pre-defined weights.
- Arriving packets are assigned **timestamps**.
- Packets are served in increasing order of their timestamps.
- Timestamps are either **virtual finish times** or **virtual start times**.



$$
\begin{aligned}
F_k^i &= \frac{L_k^i}{r_k} + S_k^i, \\
S_k^i &= \max(F_k^{i-1}, v(a_k^i)), \\
k &\in \mathbf{K}, \quad i = 1, 2, 3, \ldots
\end{aligned}
$$

# M/G/FQ Stochastic Analysis

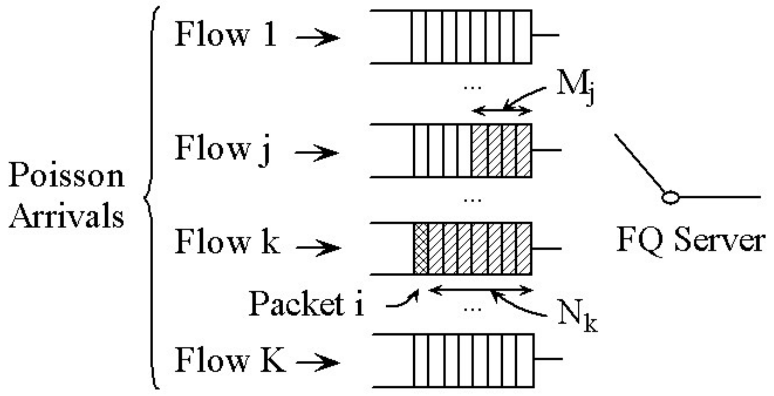
- Stochastic Analysis Assumptions:
  - Poisson arrivals, general packet length distribution and infinite buffer lengths.
- Finding Exact mean waiting time is not practical:

$$M_j = \sum_{n_k=0}^{\infty} E\left[M_j \,|q_k = n_k\right] \cdot P\left[q_k = n_k\right]$$

$$\begin{aligned}
&E\left[M_j \,|q_k = n_k\right] = \\
&\sum_{l_j=0}^{m'_j-1} \left( \sum_{m_j=l_j}^{m'_j-1} m_j \cdot P_{m_j-l_j}\left(\lambda_j \cdot R + \rho_j \cdot (n_k + m_j)\right) \right) \cdot P\left[q_j = l_j\right] \\
&+m'_j \cdot \left( 1 - \sum_{l_j=0}^{m'_j-1} \left( \sum_{m_j=l_j}^{m'_j-1} P_{m_j-l_j}\left(\lambda_j \cdot R + \rho_j \cdot \left(n_k + m'_j\right)\right) \right) \cdot P\left[q_j = l_j\right] \right)
\end{aligned}$$

# Upper and Lower Bounds

- An arrival at the kth queue must wait for:
  - R: Residual Time
  - $N_k$: Packets in queue k
  - $M_j$: Packets not in queue k but with smaller timestamps.
- Upper and lower bounds on $M_j$ can be found in terms of $N_k$ by using the FQ bounded fairness criterion.



$$W_k = R + \frac{1}{\mu}\left(N_k + \sum_{j \in \mathbf{J}} M_j\right)$$

$$W_k = \frac{1}{2}\overline{X^2}\lambda + \rho_k W_k + \frac{1}{\mu}\sum_{j \in \mathbf{J}} M_j$$

# Using the Bounded Fairness

- Fairness can be mathematically quantified by a **bound** $\Psi$ on the difference in normalized services received by different flows in a FQ system.
- Many scheduling policies exhibit this **bounded fairness** criterion including WFQ, SCFQ, SPFQ and SFQ.

$$M_j \leq \min\left( (N_k + 1)\frac{r_j}{r_k} + \overline{\psi_j}\frac{r_j}{\overline{X}C},\ N_j + \lambda_j W_k \right)$$

$$M_j \geq \min\left( \max\left( (N_k + 1)\frac{r_j}{r_k} + \overline{\psi_j}\frac{r_j}{\overline{X}C} - 1,\ 0 \right),\ N_j + \lambda_j W_k \right)$$

# The Results

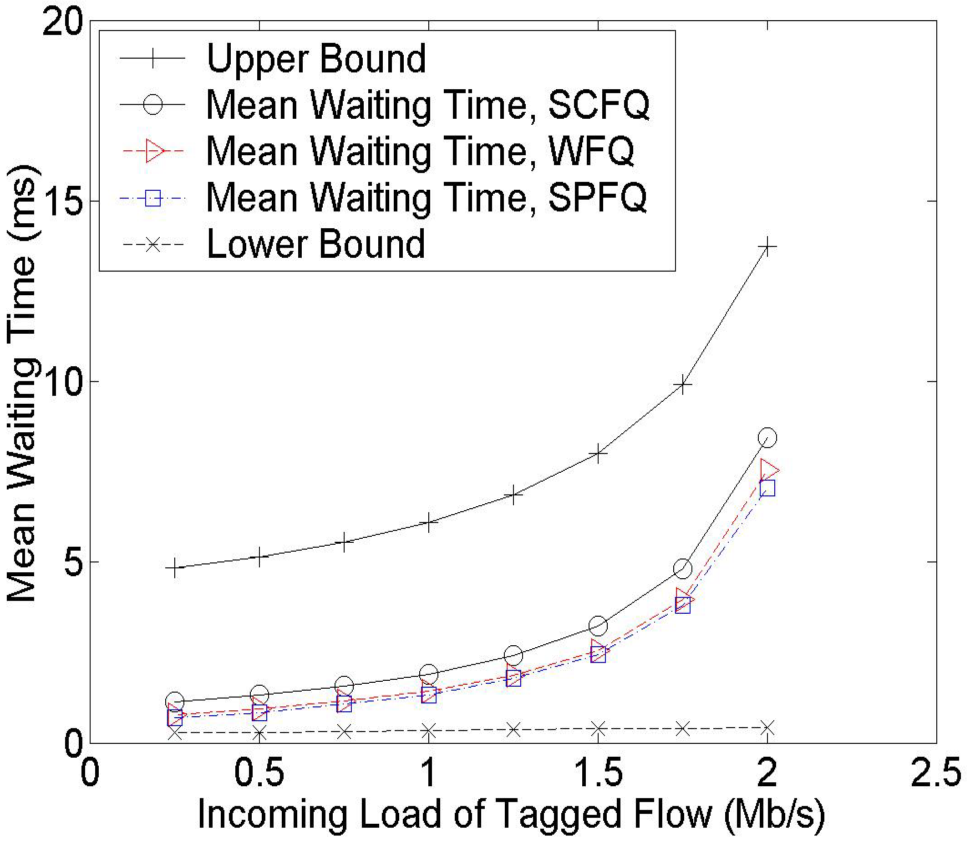
- The Upper and Lower bounds on Mean Packet Waiting Time:

$$\frac{\frac{1}{2}\overline{X^2}\lambda}{1-\rho_k} \leq W_k \leq \frac{\frac{1}{2}\overline{X^2}\lambda + \sum\limits_{j\in\mathbf{J}}\left[\frac{1}{\mu}\frac{r_j}{r_k} + \overline{\psi_j}\frac{r_j}{C}\right]}{1-\rho_k\frac{\sum\limits_{j\in\mathbf{K}} r_j}{r_k}}$$

- The Upper and Lower bounds on Mean Buffer Occupancy can be found using Little's Law $N_k = \lambda_k W_k$.

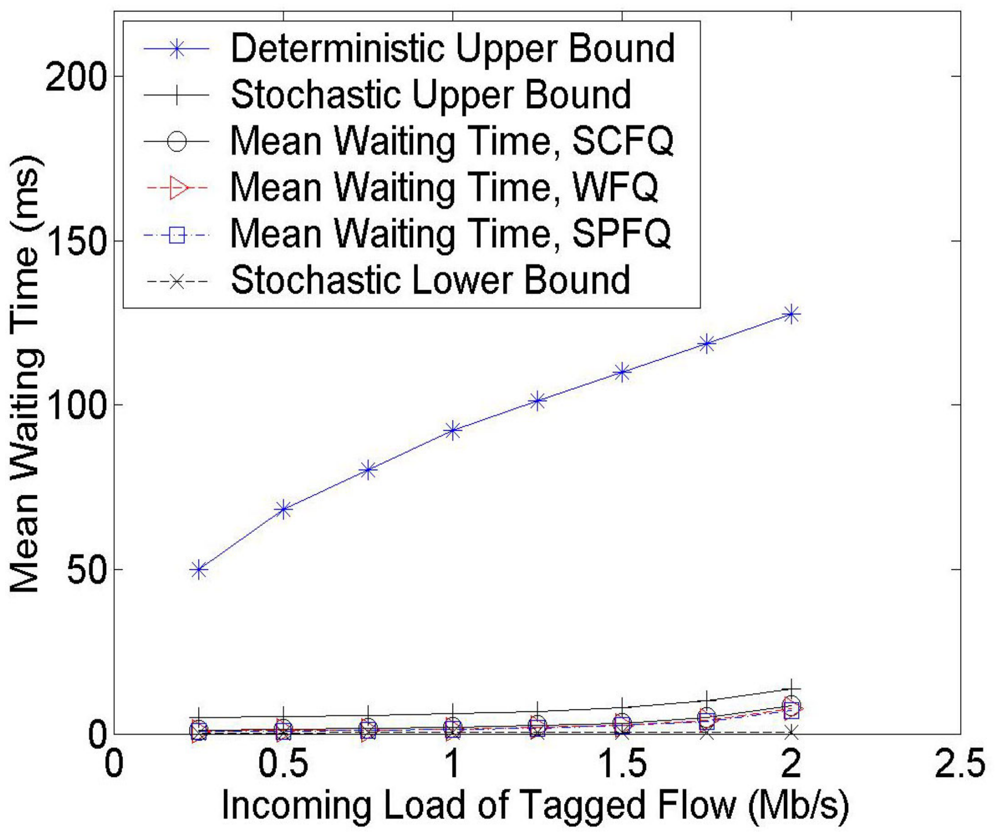
# Experimental Results

- Link capacity: 10 Mb/s.
- Four Flows with each making a 2.5 Mb/s reservation.
- Tagged flow sends at 0.25 – 2.0 Mb/s.
- Other Flows send at 2.5, 2.75, 2.75 Mb/s.
- Packet Length is Uniform between 500 and 1500 bytes.
- Tested Algorithms: WFQ, SCFQ and SPFQ.



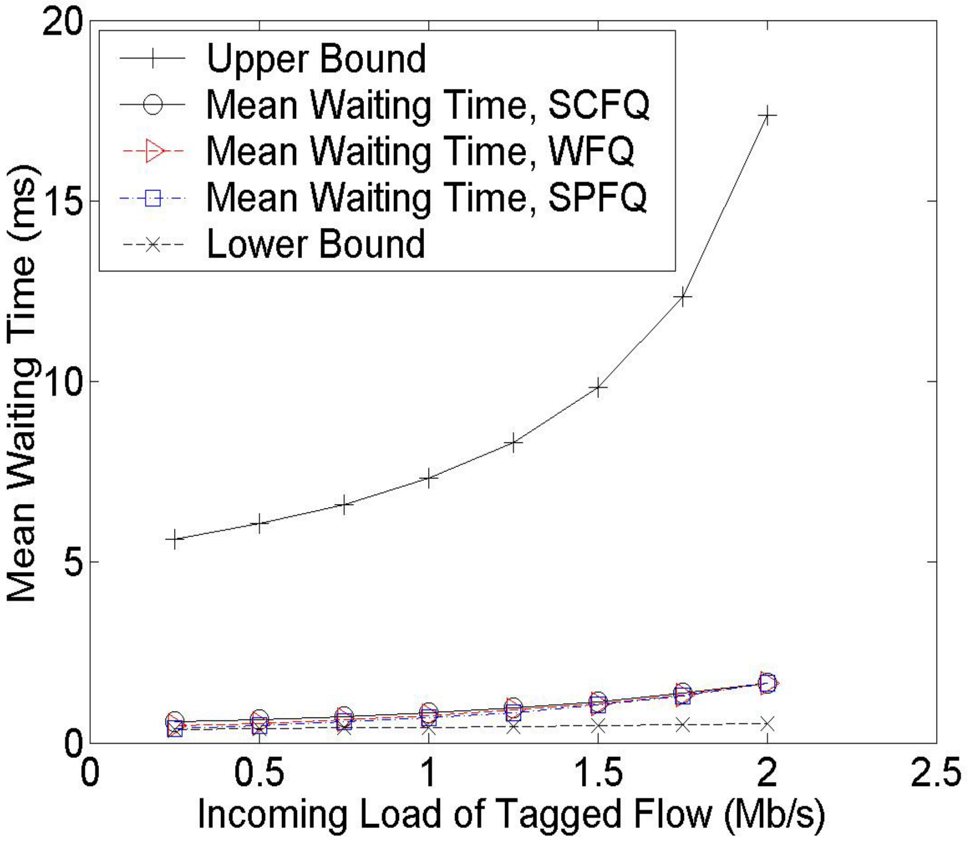
# Deterministic Bound!

- The same above experiment except: We show the corresponding **deterministic** upper bound.
- The stochastic bounds are much better than the deterministic bound.
- **Caution**: the comparison is not mathematically rigorous.



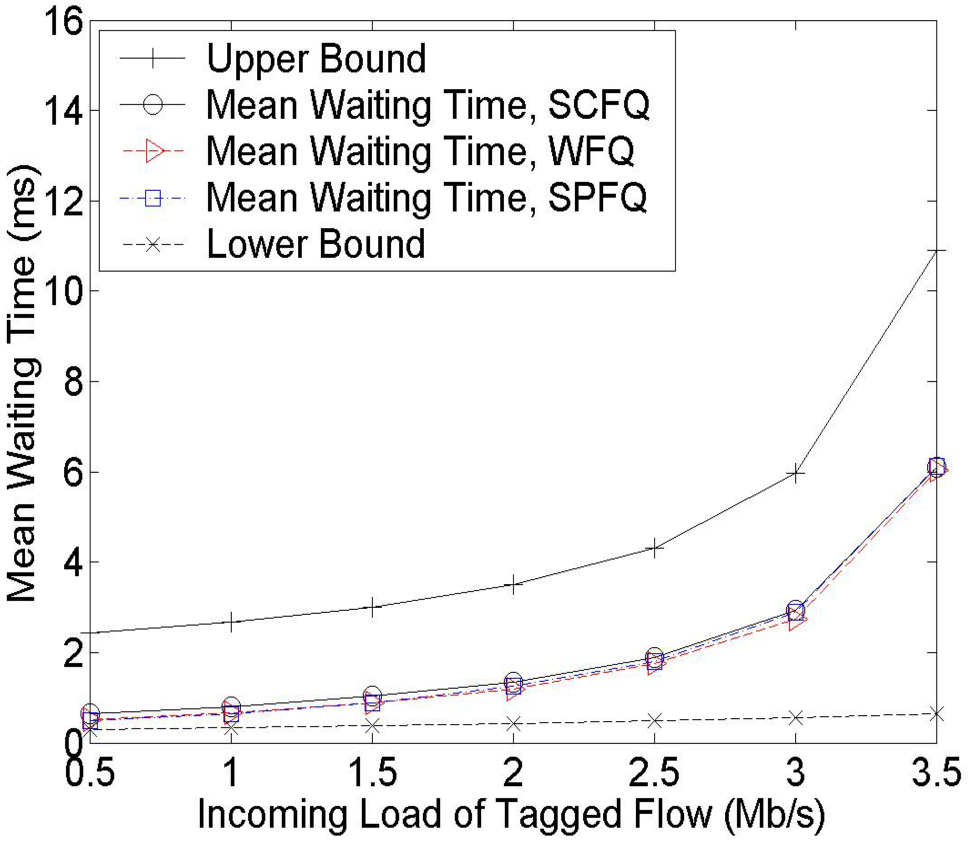
# Reducing the System Load

- The same above experiment except: All non-tagged Flows send at **2.0 Mb/s**.
- The upper bound is weaker, because the loss of $N_j + \lambda_j W_k$ information is more obvious when the load on the non-tagged flows is reduced.
- Lower bound gives a good quick estimate of mean waiting time.



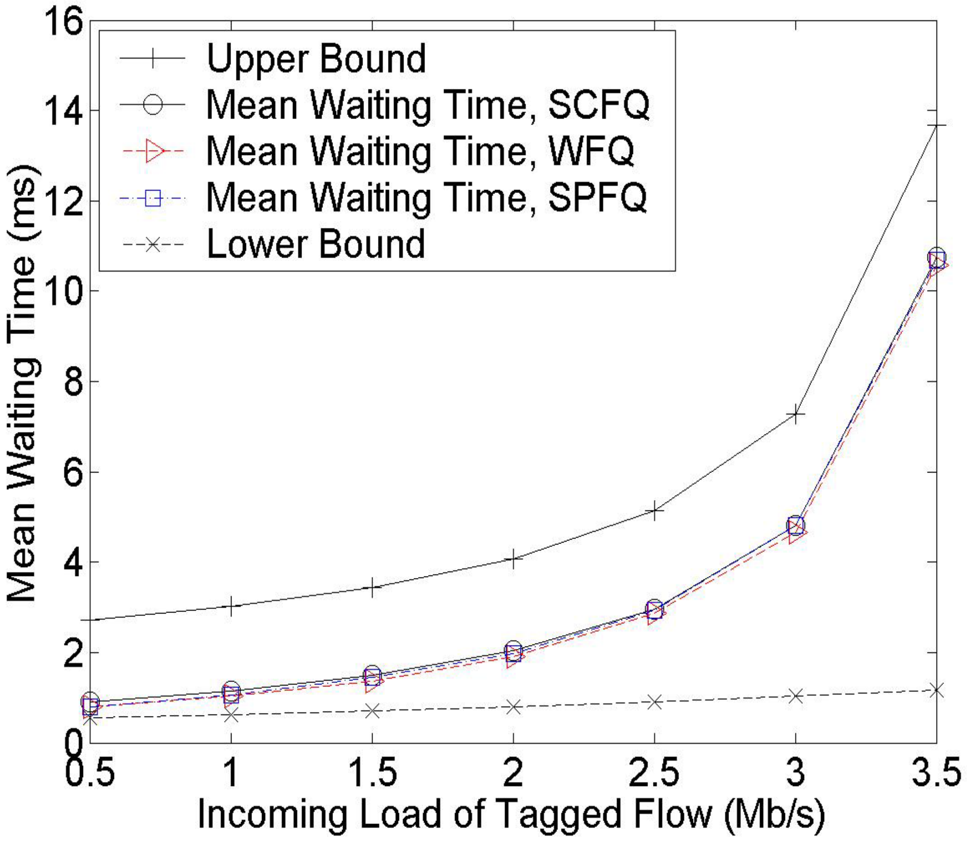
# Diverse Flow Reservations

- Link capacity: 10 Mb/s.
- 1st Flow has 4 Mb/s and 2nd Flow has 6 Mb/s reservation.
- Tagged flow sends at 0.5 – 3.5 Mb/s.
- Other flow sends at a fixed 6.0 Mb/s.
- Packet Length is Uniform between 500 and 1500 bytes.
- Tested Algorithms: WFQ, SCFQ and SPFQ.



# Different Packet Distribution

- The same above experiment except: Packet Length is **Exponential** with mean of 1000 bytes.
- Mean waiting time bounds work correctly for various flow setups and packet length distributions.

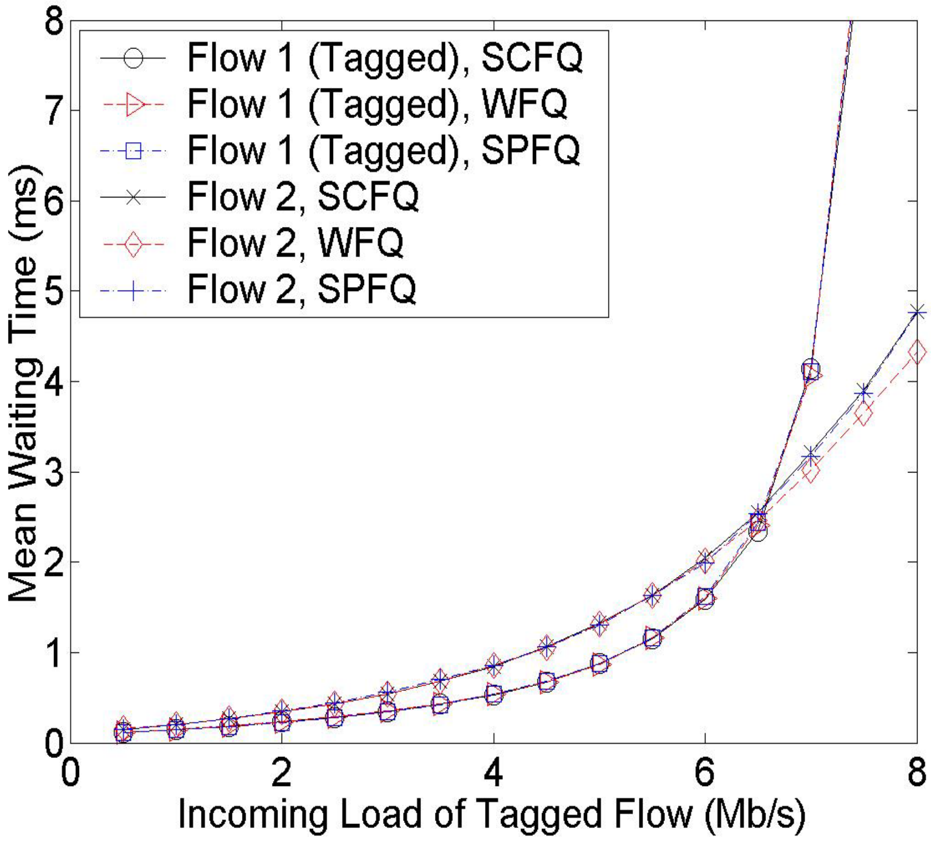


# Simulation Study of FQ

- Further study of FQ policies under random arrivals using simulation.
- As long as **none** of the incoming flows is misbehaving, then WFQ, SCFQ and SPFQ algorithms exhibit virtually the same **mean waiting time**.
- Other FQ policies do not necessarily perform the same (e.g., SFQ).
- The performance of WFQ, SCFQ and SPFQ is different than the reference GPS policy.

# Case: Two Flows

- Link capacity: 10 Mb/s.
- 1st Flow has 7 Mb/s and 2nd Flow has 3 Mb/s reservation.
- Tagged flow sends at 0.5 – 8.0 Mb/s.
- Other flow sends at a fixed 2.0 Mb/s.
- Packet Length is Uniform between 500 and 1500 bytes.
- Tested Algorithms: WFQ, SCFQ and SPFQ.

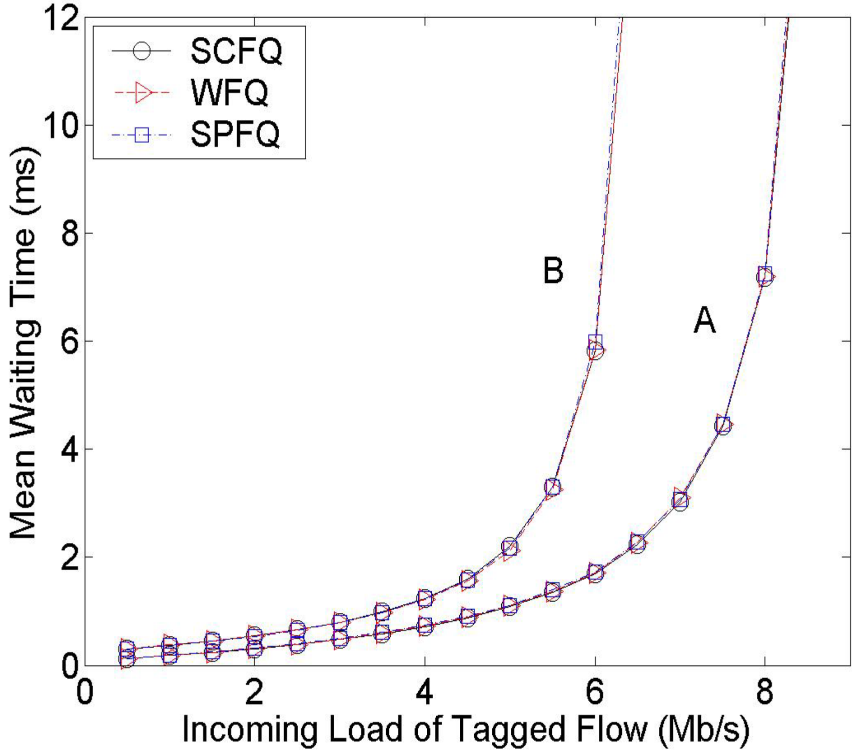


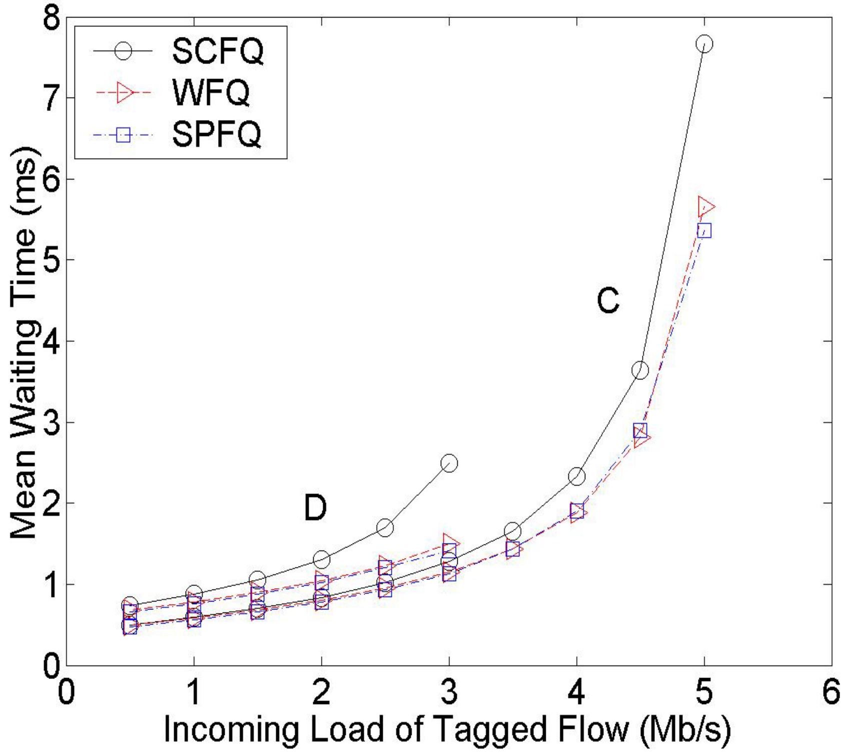
# The Result

- WFQ, SCFQ and SPFQ perform practically the same when the tagged flow is transmitting below 7 Mb/s.
- Once the tagged flow starts misbehaving, the mean delay of the different FQ algorithms starts to differ.
- This result applies for different number of incoming flows, reservation scenarios, arrival rates and packet length distributions.
- Example: 4 flows, exponential packet length distribution (see table).

| Case | $\lambda_1\overline{L}$ | $\lambda_2\overline{L}$ | $\lambda_3\overline{L}$ | $\lambda_4\overline{L}$ |
|---|---|---|---|---|
| Reservations | 6 | 1 | 2 | 1 |
| A | 0.5 – 9 | 0.25 | 0.5 | 0.25 |
| B | 0.5 – 7 | 0.75 | 1.5 | 0.75 |
| C | 0.5 – 5 | 1.25 | 2.5 | 1.25 |
| D | 0.5 – 3 | 1.75 | 3.5 | 1.75 |

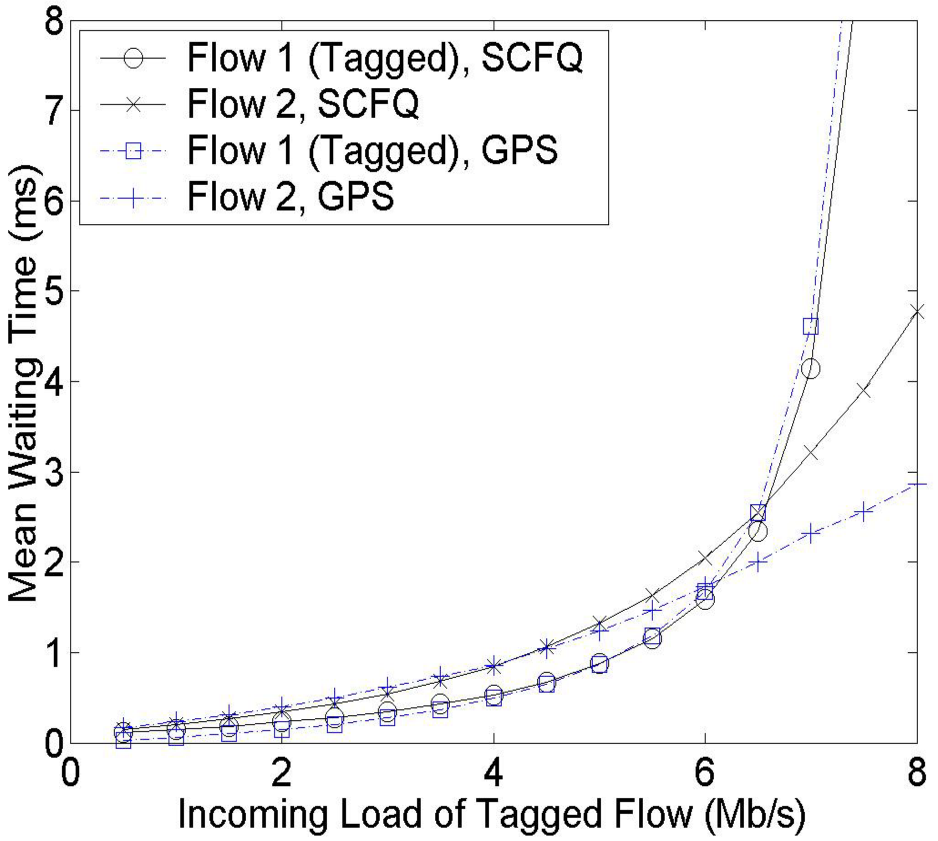
# Case: Four Flows





# Comparison with GPS

- The same earlier experiment with two flows except: Tested algorithms are SCFQ and GPS.
- Conclusion: Although WFQ, SCFQ and SPFQ are attempting to emulate GPS, they fail to emulate its exact behavior under random arrivals.



# Virtual Capacity

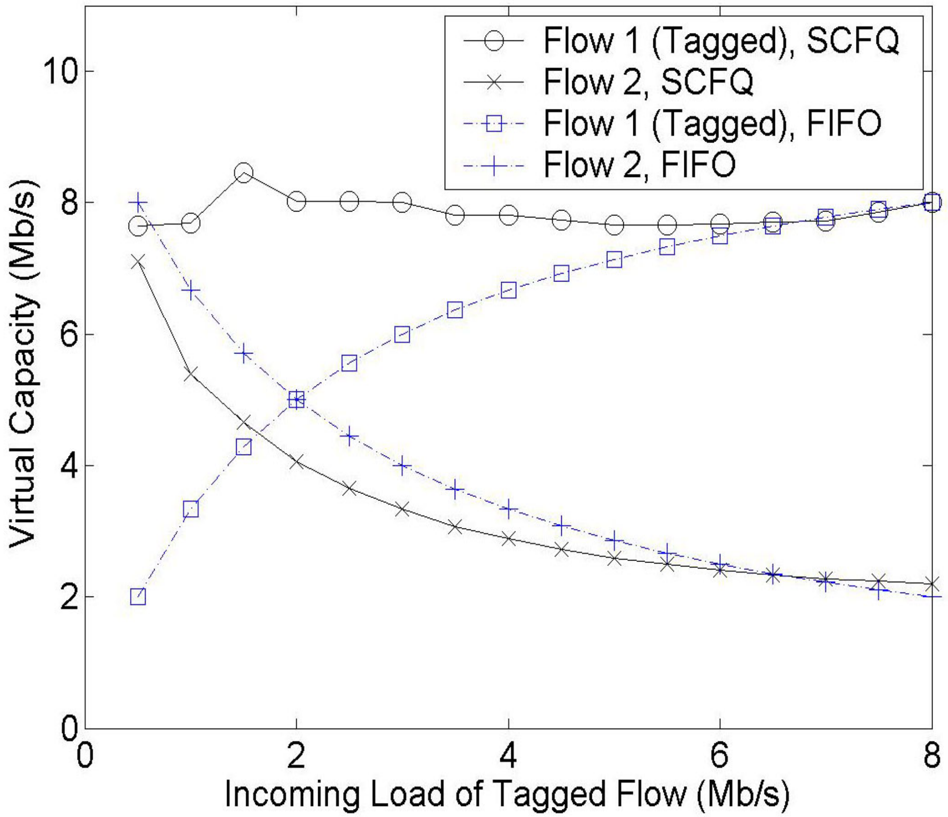
- A new concept to simplify comparing FQ algorithms to other queueing systems.
- Define the mean waiting time as:

$$W_k = \frac{R}{1 - \frac{\lambda_k}{\mu'_k}} = \frac{R}{1 - \frac{\lambda_k \overline{L_k}}{c_k}}$$

- Converts the result for waiting time in M/G/FQ to an equivalent M/G/1 system.
- Useful in visualizing the operations of FQ systems (Show **bias** in flow treatment).

# Comparison with FIFO

- The same earlier experiment with two flows except: Tested algorithms are SCFQ and FIFO.
- Lower reservation flow is actually mistreated (compared to FIFO) so that the flow with higher reservation receives a lower delay.
- Flow mistreatment grows smaller as $r_2$ approaches $r_1$.



# Contributions

- A new QoS scheduling architecture for both IEEE 802.16 and DOCSIS.
  - Performance evaluation using OPNET was slowed down by the DOCSIS module bugs.
- Simple and reasonably tight upper and lower bounds on mean waiting time of FQ algorithms under Poisson arrivals.
- Study of stochastic performance of packet-based FQ policies: WFQ, SCFQ and SPFQ.
- A new Queueing System, called H-F$^2$Q.

# Publications and Submissions

- Mohammed Hawa and David W. Petr, “Stochastic Evaluation of Fair Queueing Systems,” Submitted to IEEE Journal on Selected Areas in Communications, April 2003.
- Mohammed Hawa and David W. Petr, “M/G/FQ: Stochastic Analysis of Fair Queueing Systems,” IEEE 2nd International Conference on Networking, pp. 368-381, 2002.
- Mohammed Hawa and David W. Petr, “Quality of Service Scheduling in Cable and Broadband Wireless Access Systems,” Tenth International Workshop on QoS, pp. 247-255, 2002.

# Discussion

- Thank you!