ENHANCED NON-CONGESTIVE QUEUE: A METHOD FOR DIFFERENTIATING SERVICE IN WIRELESS SENSOR NETWORK INTERNETWORKING

Seyed Enayatallah Alavi[1], Marjan Naderan-Tahan[2], Mohamad Aminian[3]

[1] Shahid Chamran University of Ahvaz, Ahvaz, Iran; se_alavi@yahoo.co.uk
[2] Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.naderan@scu.ac.ir
[3] Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.aminiyan@gmail.com

Abstract. We propose Enhanced Non-Congestive Queuing as a scheduling paradigm that allows for efficient interoperation of sensor networks with the Internet. This method promotes conditionally small data packets, which require minor service times, with respect of the maximum delay they can handle that preventing expiration for packets, as long as their total service times cause insignificant delays to other packets in the queue.

Introduction. Scheduling paradigms of packet networks do not match well the requirements of sensor applications. Such applications do not really cause significant delays, raising naturally the issue of whether they deserve a prioritized service or not.

Our primary assumption is that sensor applications generate packets in form of non-congestive traffic. However, other applications may fall into this category as well, if we judge exclusively on the basis of packet length. This observation calls for a new metric for application fairness as well, which relies mainly on the delay rather than throughput. In order to avoid the cost of packet preparation for differentiated services, we take advantage of two distinctive properties of typical sensor data:

1) The small size of sensor packets.
2) The small data volume of sensor-generated data flows.

Our work is based on Non-Congestive Queuing (NCQ) approach. The key idea of this approach is the service discipline called: “Less Impact Better Service” (LIBS).

II Related works. A lot has been done in the networking community aiming at controlling traffic based on its characteristics. Controlling is implemented either through scheduling or through dropping policies mainly aiming at penalizing high - bandwidth - demanding flows rather than favoring low - bandwidth - demanding flows. Floyd and Fall introduced mechanisms based on the identification of high-bandwidth flows from the drop-history of RED. The RED-PD algorithm (RED with Preferential Dropping) uses per-flow preferential dropping mechanisms. Two other approaches that use per-flow preferential dropping with FIFO scheduling are Core-Stateless Fair Queuing (CSFQ) and Flow Random Early Detection (FRED). CSFQ marks packets with an estimate of their current sending rate. The router uses this information in conjunction with the flow’s fair share estimation in order to decide whether a packets needs to be dropped. FRED does maintain a state although only for the flows which have packets in the queue.

III. Enhanced non-congestive queuing. First, we assume different classes of packets. We use three priority queues. The queue with lower number has higher priority. When packet received by router first based on delay flag we decide to put it in first two high prior queues or the second two queues.

Second step is making decision based on size of packet. So we assume PL (packet length) as critical parameter in our work. Based on what we said the packet with smaller size and lowest flow should receive better services, so we considering two threshold PL1=130 bytes and PL2 as average of length of receiving packets. And we define function P_{PL} as first parameter to decision (if PL2 > PL1 we use (1) but if PL2<PL1 we use (2)):

$$P_{PL} = \begin{cases} 1 & PL \leq PL1 \\ \dfrac{(PL - PL1)}{(PL2 - PL1)} & PL1 < PL < PL2 \\ 0 & PL \geq PL2 \end{cases} \quad (1)$$

Where PL is the packet length if any received pocket.

$$P_{PL} = \begin{cases} 1 & PL \leq PL1 \\ 0 & PL \geq PL1 \end{cases} \quad (2)$$

The second parameter is the priority probability. This probability will be calculated separately for packet that classified in first step and it is called HPP (high priority probability) and LPP (low priority probability):

$$HPP = \frac{number\ pockets\ in\ Queue\ 0}{number\ of\ all\ received\ delay\ sensitive\ pockets}$$
$$LPP = \frac{number\ of\ all\ received\ delay\ sensitive\ pockets}{number\ of\ all\ received\ delay\ sensitive\ pockets} \quad (3)$$

To continue the first step classification, we integrate two mentioned parameters P_{PL} and HPP/LPP and introduce Final Probability (FP) as an average of the two other probabilities. We use average with same weight. So we define FP as follow :

$$FP = \begin{cases} \dfrac{(P_{PL} + HPP)}{2} & for\ delay\ sensitive\ pockets \\ \dfrac{(P_{PL} + LPP)}{2} & for\ non-delay\ sensitive\ pockets \end{cases} \quad (4)$$

At the end based on FP, we finalize classification of the received packet. After classification of packets, we need a scheduling method. We choose simple priority scheduling. This method service queues based of their priority.

IV Comparison. Our ultimate goal of this work is introduction some enhancement on NCQ algorithm. In this part we prepare a table which is compare ENCQ with NCQ.

Table(1) : comparison between ENCQ and NCQ

	ENCQ	NCQ
Complexity	Simple	Simple
Flexibility	Adaptive thresholds	Solid thresholds
Considering QoS	Yes	No
Scheduling paradigm	Simple	Simple
Fairness aspects	Bandwidth and delay	Bandwidth

ENCQ is more complex than NCQ but in return has adaptive parameters that help network to react more efficient in different situations. QoS features especially delay is considered in ENCQ which is make the method more realistic and more appropriate in wireless sensor networks.

V Conclusion. In this work we demonstrate that ENCQ can be adjusted to promote service for sensor applications without damaging traditional internet applications. In simple terms, ENCQ increases the amount of satisfied users within a system. Also, our approach solve some of deficiencies of NCQ algorithm and illustrate some way to make the main algorithm NCQ more appropriate for distinct applications.