Predictive power management method for network devices

P.C. Gunaratne and K.J. Christensen

Energy consumption can be reduced by powering down components within a router, switch, or other network device during inter-packet idle periods. A predictive power management method based on exponential smoothing and detection of very long idle periods is developed. Long idle periods are detected using quantile estimation. The method improves upon that of Hwang and Wu.

Introduction: Significant energy savings can be achieved if components in a router, switch, or other network device can power down to a low-power sleep state during the idle periods between packet arrivals [1]. What is needed is a power management method that can maximise the amount of sleep time and minimise the impact of wake-up time on system performance. A packet arrival to a network device triggers a forced wake-up (to power-up the sleeping components). This forced wake-up can result in added delay to packet processing since wake-up time is non-negligible. It is thus desirable to pre-wake-up the components before a packet arrival. A predictive method is needed to achieve pre-wake-up (and thus minimise the number of forced wake-ups) while also maximising sleep time.

Network traffic is well known to be bursty and self-similar [2]. Self-similarity can result in the majority of idle periods being small (and uniform), but the large part of the total idle time to occur in a few, very long idle periods. Being able to detect and exploit long idle periods is a difficult challenge. A method proposed by Hwang and Wu [3] addresses this challenge. In this method, exponentially weighted moving average (EWMA), or exponential smoothing, is used to predict the next idle period and if the predicted idle period is less than a previously calculated threshold (S_{th}), the device will power down (i.e. enter a low-power sleep state) until an event occurs that triggers a forced wake-up. If the predicted idle period is less than S_{th}, the device stays powered-on and a new prediction is made after every S_{th} time interval until either an idle period greater than S_{th} is predicted or a wake-up event occurs. A prediction is limited to a maximum of C times the previous prediction. If pre-wake-up is enabled, then if no event occurs by the end of the predicted idle period and the device is sleeping the device will wake-up at the end of the predicted idle period. Our method improves upon the method of Hwang and Wu.

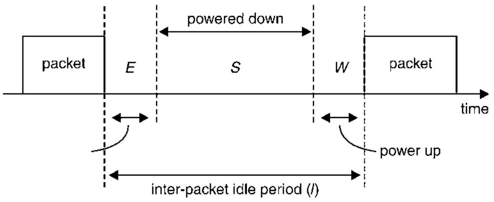

Fig. 1 *Inter-packet idle period*

The ExpQ method: Fig. 1 shows an inter-packet idle period (I), time for power-down (E), and time for power-up (W). The possible sleep time (S) is $S = I - (E + W)$. If $(E + W) < I$ then it is possible that the device can power-down and save energy. Fig. 2 shows the finite state machine (FSM) of the new EWMA plus quantile estimation method (we call this new method ExpQ). The four states of ExpQ are:

- Running – the device is fully powered-up and may be processing a packet.
- Running-wait – The device is counting down a timer T1. The device is fully powered-up and monitoring for packet arrivals.
- Sleep-wait – The device is counting down a timer T2. The device is in a sleep state and monitoring for packet arrivals.
- Sleep – The device is in a sleep state and monitoring for packet arrivals.

A prediction of the next idle period is $P_{i+1} = P_i \times (1 - \alpha) + I_i \times \alpha$ where P_{i+1} is the new prediction, P_i is the previous prediction, I_i is the time of previous idle period, and α is a constant ($0 \leq \alpha \leq 1$). We define a threshold value $T = E + W$. If $P < T$ the device will set the timer T1 to the estimated quantile time Q. The estimated quantile Q is calculated using the algorithm of Jain and Chlamtac [4]. If no packet arrival occurs before T1 expires, the device will power-down and sleep until a packet does arrive (and then triggers a forced wake-up). If $P > T$, the device sets timer T1 to the estimated quantile time Q and T2 to the predicted idle period time P and transits to the Sleep-wait state. If no packet arrival occurs before T2 expires, the device will power up and wait for T1 to expire in the Running-wait state. The ExpQ method contains four states compared to three states in the method of Hwang and Wu. The computational complexity is the same as Hwang and Wu. Long idle periods tend to distort predictions made by EWMA. Quantile estimation is used to identify and exclude these long idle periods.

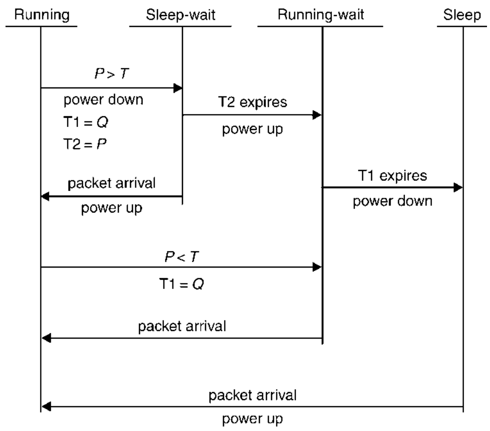

Fig. 2 *Finite state machine for ExpQ method*

Table 1 shows the sum of idle periods that exceed the given quantile as a percentage of the sum of all idle periods for the four Bellcore traces from [2]. This motivates the use of quantiles as a means of dynamically finding thresholds for identifying difficult-to-predict long idle periods. The Bellcore traces, each of which contains a million packet arrival events, are from an Ethernet network at the Bellcore Morris Research and Engineering Center and are known to exhibit a large degree of self-similarity and to be representative of real network traffic.

Table 1: Percentage of inter-packet idle period times that exceed a given quantile

Quantile	Oct89Ext	Oct89Ext4	pAug89	pOct89
60%	95.49%	97.77%	83.59%	94.88%
70	91.89	96.09	73.87	90.88
80	85.27	92.11	61.29	81.17
90	73.23	79.93	46.61	60.77
95	62.34	66.30	36.45	44.23

Performance evaluation: Trace-driven simulation was used to compare the ExpQ method to the Hwang and Wu method. Traffic input consisted of the Bellcore traces [2] and synthetic traces. The shortest mean idle period among the four Bellcore traces is 1.249 milliseconds (this trace also gives the worst performance compared to the other methods based on the K metric (defined below) when using the ExpQ method) and the synthetic traces approximate this value. The four synthetic traces are composed of Poisson arrivals (exponential idle periods) with the rate of arrivals equal to 801 packets per second (labelled as Exp in Table 3), interrupted Poisson process (IPP) arrivals with rates of arrival set to 10,000 packets per second and with the on-to-off rate set to 12.5 per second and off-to-on rate set to 1 per second, and two traces of unbounded Pareto distributed arrivals with $\alpha = 1.2$, $\beta = 0.0002083$ (Par12) and $\alpha = 1.5$, $\beta = 0.0004163$ (Par15), respectively. The performance measures were sleep time as a percentage of the total idle time and forced wake-ups as a percentage of the total number of packet arrivals in the trace. A new metric $K = L \times (1 - F)$ where L is sleep time as a fraction of the total idle time and F is the number of forced wake-ups minus forced wake-ups that occur at the end of the longest 5% of idle periods as a fraction of the number of arrivals in the trace is defined. The possible values for K range from 0 to 1, and a larger value signifies better performance. The energy saved by sleeping during a long idle period mitigates the cost of the forced wake-up, thus our metric ignores forced wake-ups that occur at the end of the longest 5% of idle periods. This metric was also calculated without ignoring forced wake-ups that occur at the end of long idle periods and by ignoring forced wake-ups that occur at the end of the longest 5, 10, 15, and 20% of idle periods. For all cases, when taking the average value for all traces the ExpQ method outperformed the HW and HWP methods. As a greater percentage of idle periods are ignored, the differences between the K values for the three methods narrow.

Table 2: Performance evaluation with Bellcore traces

	Oct89Ext	Oct89Ext4	pAug89	pOct89
Sleep time as percentage of all idle time				
HW	99.99	99.78	85.91	64.96
HWP	21.21	25.12	50.98	36.97
ExpQ	53.93	55.62	59.66	54.99
Forced wake-ups as percentage of all arrivals				
HW	99.52	93.19	64.06	28.20
HWP	54.12	53.23	38.81	19.28
ExpQ	46.50	47.78	30.37	21.48
K				
HW	0.055	0.118	0.352	0.497
HWP	0.097	0.118	0.313	0.301
ExpQ	0.310	0.320	0.450	0.450

The method proposed by Hwang and Wu was evaluated with both pre-wake-up disabled (HW) and pre-wake-up enabled (HWP). For the HW and HWP methods, $S_{th} = 1.5$ ms and $C = 2$. For the ExpQ method, $E + W = 1.5$ ms and $Q = 0.90$ quantile. For all methods, the constant $\alpha = 0.5$. Tables 2 and 3 show the performance of the ExpQ method compared with the Hwang and Wu HW and HWP methods for the traffic inputs described above. On average, the ExpQ method spends 20% less time than HW sleeping and has 35% less forced wake-ups. ExpQ spends 133% more time sleeping than the HWP method and has 3% more forced wake-ups. Using the K metric, the ExpQ method outperforms both the HW and HWP methods.

Table 3: Performance evaluation with synthetic traces

	Exp	IPP	Par12	Par15
Sleep time as percentage of all idle time				
HW	39.84	92.14	51.01	38.73
HWP	32.31	0.22	9.56	17.31
ExpQ	32.87	93.53	59.04	41.76
Forced wake-ups as percentage of all arrivals				
HW	39.83	0.39	13.73	22.17
HWP	32.27	0.26	11.29	18.63
ExpQ	32.84	10.69	19.01	23.65
K				
HW	0.259	0.919	0.460	0.321
HWP	0.226	0.002	0.086	0.145
ExpQ	0.240	0.880	0.510	0.340

Conclusion: We have designed and evaluated a new predictive power management method for devices with components capable of transiting to a low-power sleep state. The ExpQ method was compared against the well-known method proposed by Hwang and Wu for the case of self-similar network traffic as an input. The results show that quantile estimates can be used as thresholds to detect long idle periods and that the performance of the method is superior to the methods it was compared against.

 29 March 2005
Electronics Letters online no: 20051149
doi: 10.1049/el:20051149

P.C. Gunaratne and K.J. Christensen (*Department of Computer Science and Engineering, University of South Florida, 4202 East Fowler Avenue, ENB 118, Tampa, Florida 33620, USA*)

E-mail: christen@cse.usf.edu

References

1 Gupta, M., and Singh, S.: 'Greening of the Internet'. Proc. ACM SIGCOMM, August 2003, pp. 19–26
2 Leland, W., Taqqu, M., Willinger, W., and Wilson, D.: 'On the self-similar nature of Ethernet traffic (extended version)', *IEEE/ACM Trans. Netw.*, 1994, **2**, (1), pp. 1–15
3 Hwang, C.-H., and Wu, C.-H.W.: 'A predictive system shutdown method for energy saving of event-driven computation', *ACM Trans. Des. Autom. Electron. Syst.*, 2000, **5**, (2), pp. 226–241
4 Jain, R., and Chlamtac, I.: 'The P^2 algorithm for dynamic calculation of quantiles and histograms without storing observations', *Commun. ACM*, 1985, **28**, (10), pp. 1076–1085