Power Management schemes in LAN Switches

Maruti Gupta, Satyajit Grover, Suresh Singh
Computer Science Department
Portland State University
{mgupta,satyajit,singh}@cs.pdx.edu

We examine the feasibility of introducing power management schemes in network devices in the LAN. Currently, such schemes exist to minimize the power consumption on devices such as desktops, notebooks and a number of other portable devices. However, no such dynamic power management schemes are available for network devices such as routers and switches. In this work, we take a look at the feasibility of introducing power management schemes in LAN switches and address questions arising from putting various components on switch interfaces to sleep.

We choose to begin with LAN devices and in particular LAN switches for several reasons. Firstly, LAN switches comprise the bulk of network devices in the LAN and they also consume the largest percentage of energy (see Table I in [1]) among them. Second, a majority of interfaces on LAN switches are directly connected to hosts, that intuitively would have longer periods of low traffic activity than on those interfaces that carry aggregated traffic from many hosts. Our approach to power management in switches here involves powering off or putting to sleep LAN switch components, interfaces, or entire switches. However, powering devices back into normal or higher power consuming modes causes a spike in energy consumed as well as takes a certain amount of transition time. Hence, the decision to put devices to sleep must be made such as to not result in an overall increase in power consumption or cause performance degradation due to loss of network connectivity and packet drops. Another side effect of putting ports or switches to sleep is that protocols running at layer 2 and above may be negatively affected.

In our first step, we collected and examined traffic data from our campus network LAN to see if there are enough periods of inactivity at various switch interfaces to justify sleeping. Our data shows that there are significant number of intervals during low as well as high activity periods when individual switch interfaces can be put to sleep to intervals ranging from more than 20 seconds to about 1 second. To address the question on when an interface should be put to sleep and for how long, we developed an abstract sleep model where we define three different sleep states (other than ON and OFF) for switch interfaces based on the *functionality* of the interface in each of these states.

- *Simple Sleep:* In Simple Sleep, the interface sets a sleep timer, and only wakes up when the timer expires. All packets arriving during the sleep period are lost.
- *HAS:* In (Hardware Assisted Sleep), an incoming[1] packet wakes up the interface but is lost since it is not buffered.
- *HABS:* In (Hardware Assisted Buffered Sleep), an incoming packet wakes up the interface and is buffered, thus requiring the input buffer to remain powered on to buffer incoming packets.

We then used our sleep model to develop algorithms for sleeping. The sleeping algorithm makes the decision on whether to sleep or not depending upon the traffic activity seen so far. In the case of Simple Sleep, we observed that Simple Sleep would be best used on interfaces connecting hosts to switches, especially if the ACPI implementation on these hosts were modified to inform the switch of their sleeping decisions. HABS is suited to use for all interfaces, especially those that cannot tolerate packet loss such as interfaces connecting switches to other switches and routers. To make a decision on when to sleep, we used a traffic estimation scheme where we used the exponentially weighted moving average filter (EWMA) to derive estimated inter-activity intervals as well as knowledge of the periodic behavior of the protocols. We used the estimated value obtained to make decisions to sleep based on whether sleeping would save energy or not. Using energy values and transition times obtained from wireless LAN cards for idle, active and low power modes, we implemented our sleeping algorithms on trace data obtained for several interfaces and observed significant energy savings. Our results show that sleeping is indeed feasible in the LAN and in case of the HABS model, with little or no impact on other protocols. However, we note that in order to maximize energy savings while minimizing sleep-related losses, we need hardware that supports sleeping.

1. REFERENCES

[1] M. Gupta and S. Singh, "Greening of the Internet", *ACM SIGCOMM'03*.

[1] A incoming packet includes those coming from the wire as well as the packets switched to a particular switch port from other ports