

# D R A F T

## Whack-on-LAN: Inexpensive Remote PC Reset

Grant Ayers   Kirk Webb   Mike Hibler   Jay Lepreau

*School of Computing, University of Utah*  
www.emulab.net   www.flux.utah.edu

*University of Utah Flux Group Technical Note FTN-2005-05 (DRAFT)*  
*December 2005*

### Abstract

Hardware and software lockups pose serious problems for the deployment, use, and maintenance of remotely deployed PCs. They require on-site human intervention, which can yield excessive downtime and unavailability. The best way to cope with this issue is to introduce a mechanism that enables a remote hardware reset, irrespective of a remote machine's hardware or software state.

We have developed a lightweight, cost-effective device capable of forcing a hardware reset over Ethernet. We describe its design and implementation, discuss relevant security implications, and review potential extensions to the technology.

## 1 Introduction

Any computer that is operated through a network instead of its own keyboard or other human input devices can be considered "remotely deployed." Typically this means servers, but it can also include a person's home or work machine, depending upon his or her current location. Most modern operating systems allow a remote user the same functionality as a console user, but with certain obvious restrictions. For example, a remote user cannot exchange CDs or pull the power plug. In other words, a remote user can only have complete control of the *software* and its capabilities. Because of this, a system crash or accidental network misconfiguration can effectively bar a remote user from even communicating with the machine, thus requiring physical on-site intervention. This is especially problematic in large-scale environments such as business and educational facilities, where the latency of physical intervention can be high and downtime is very expensive.

A reliable solution for restoring a computer to a responsive state must be inherently software independent, in case of a system crash or complete lockup. A user with physical access simply presses the reset button or interrupts power to the machine for a moment. This should also be the goal for a remote reset system.

Several remote solutions exist, but each has undesirable infrastructure requirements, is not always practical, or incurs too much cost. Hardware watchdogs rely on operating system specific drivers and can be mistakenly misconfigured. Stand-alone power control or reset units are expensive, costing several hundred dollars, and require additional infrastructure such as an available phone line or another ethernet connection.

For these reasons, stand-alone reset units are unsuitable for small or scattered deployment (such as machines spread across a building).

We have developed a modification to the Wake-on-LAN specification in commodity ethernet cards that diverts a wakeup signal to a hardware reset. We refer to this capability as Whack-on-LAN. It requires no additional infrastructure, is especially well suited for scattered environments, and can cost as little as five dollars per machine.

## 2 Design

Wake-on-LAN (WOL) was developed to provide a means to remotely power on a system over a local area network, and is supported by several commodity ethernet adapters. A WOL-enabled network adapter draws standby power from the ATX power supply when the machine is off, and monitors the network for specific data. AMD created the “Magic Packet” specification [1] to define this data as a short preamble followed by 16 repetitions of the adapter’s MAC address anywhere inside the frame. Many cards will also recognize multicast, unicast, ARP, or link activity as a valid wakeup event. When a wakeup event occurs, the network adapter signals the computer through a special WOL cable that is connected to the motherboard, which then switches on the ATX power supply.

Because it is designed to function when the rest of the computer is off, Wake-on-LAN is completely driver and software independent. Its only requirement is the standby power provided by the ATX power supply when the machine is off. This means that outside network activity can be guaranteed to trigger a machine-state independent event while the network adapter’s WOL circuitry is enabled. We use this unique feature for the hardware reset paradigm.

Normally, when a wakeup event occurs, the network adapter sends a signal on the LAN WAKE-UP line of the WOL cable, which is connected to the motherboard. If WOL was enabled in the motherboard BIOS, the ATX power supply is switched on. However, we wish to use this signal to force a hardware reset, which is equivalent to pressing the reset switch on a PC. The reset switch on a PC is connected to a two-pin header on the motherboard. When pressed, the switch connects the two pins, which closes the circuit that resets the hardware. By diverting the LAN WAKE-UP line to the base of an appropriate transistor, and connecting the source and drain of the transistor to the reset pins on the motherboard, the wakeup signal can effectively become an electronic reset button. The physical reset switch can remain connected to the same reset pins to allow both methods of resetting the computer. This new configuration also bypasses the WOL circuitry on the motherboard, making motherboard WOL support and BIOS settings irrelevant. Additionally, the standby power from the ATX power supply is no longer required because the machine is already on. By eliminating these requirements, Whack-on-LAN support is dependent only upon a WOL-compliant Ethernet card and a motherboard with reset functionality. It should be noted, however, that while reset functionality is almost universal, a minority of proprietary vendors have opted to exclude the reset header pins on some of their motherboards. In this case, a header or extending wires must be soldered on manually.

Using only a transistor, the total cost for remote reset capability can be less than one dollar. It is usually beneficial, however, to create a more complex device that takes into account the voltage and current that pass through the transistor, and limit them with resistors if necessary. This device can then be attached to the network adapter or any other component inside the computer. It can also be mass produced.

## 3 Implementation in Emulab

Emulab [6, 5] is a time- and space-shared network testbed with hundreds of publicly accessible nodes. Among them are over twenty fixed wireless nodes, which are spread throughout a large building. Due to

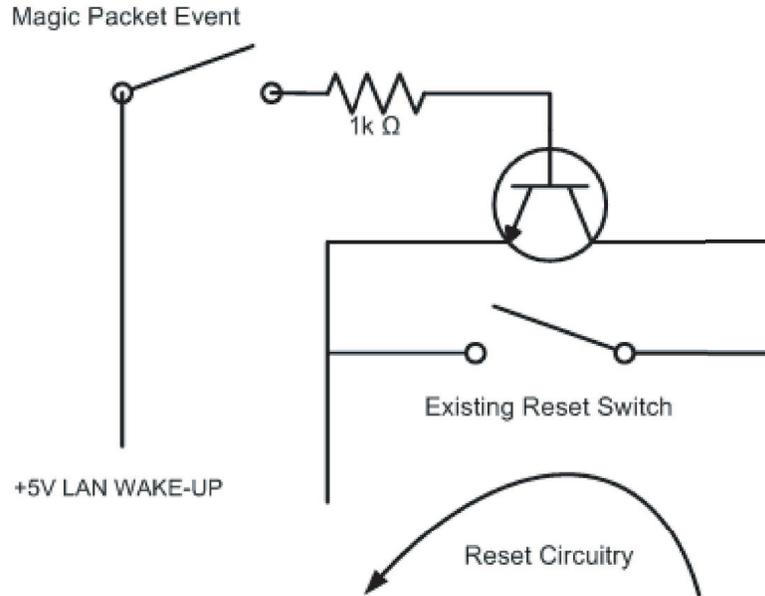


Figure 1: Whack-on-LAN circuit diagram

their diverse and decentralized locations, network and other connections are very limited. Additionally, physical access is restricted or not always available for many of these locations.

Emulab users are granted root access to the machines they use. They can load arbitrary software on them, including complete disk images and operating systems. This great power and flexibility has often created problems when experimental software or misconfigurations have locked the machines or prevented them from booting. In consequence of this and the frequently high latencies of physical intervention, over forty percent of Emulab’s wireless nodes were locked during any given time. Small-scale reset units were prohibitively expensive for large numbers of single nodes, and an additional ethernet connection was usually not available. Furthermore, watchdogs were inappropriate due to the variety of software environments in use.

We developed Whack-on-LAN modules for the Intel PRO/100 S series network adapters [3] in each of these nodes. Each module contains a generic NPN transistor, a 1k-Ohm resistor, a two-pin header for connecting a reset cable from the module to the motherboard, and an LED for diagnostic purposes. The resistor and LED are connected between the LAN WAKE-UP line and the base of the transistor, and the module itself is mounted within the case of each PC. The total parts cost for thirty modules was approximately twenty dollars. After deploying Whack-on-LAN on each of the wireless nodes, downtime was dramatically decreased from forty to three percent. The remaining downtime is due almost exclusively to miscellaneous node component failures. We have also had to take into account the Ethernet duplex mismatch problem [4], in which Ethernet frames may occasionally be lost due to auto-negotiation failure. We make attempts to correctly configure the network interface card and the delivering switch, and retransmit the Magic Packet if necessary.

## 4 Security

WOL is implemented in different ways by different vendors. Almost all adapters support AMD’s “Magic Packet,” and several support multicast, unicast, ARP, or link activity wake-up events. The latter four are inherently insecure for most networking environments, as anyone who can reach them can almost certainly

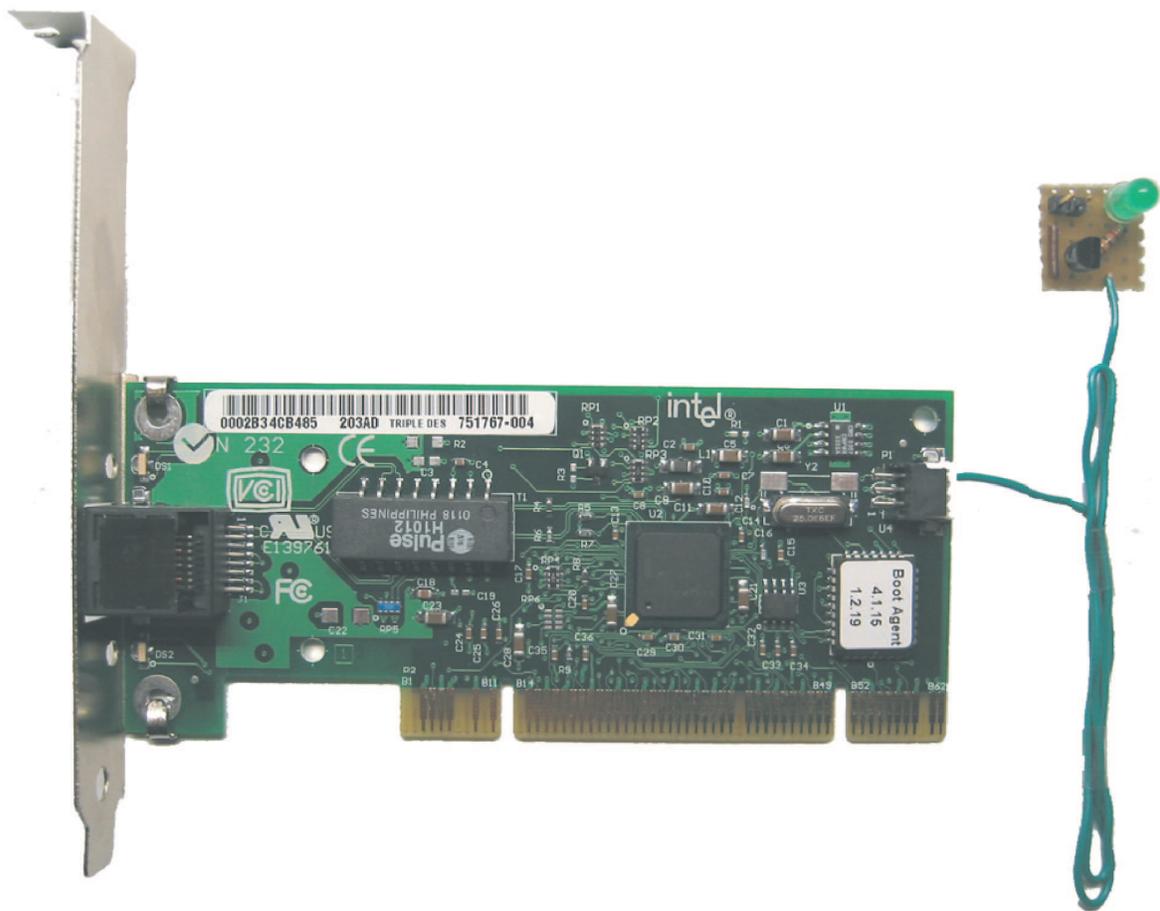


Figure 2: Intel PRO/100 S with Whack-on-LAN circuit

generate a wake-up event, as can standard network traffic. In the context of rebooting, this is especially important and must be considered carefully. In some networks, for example, it may be possible to filter outside multicast traffic heading for an internal unicast-only network, and use multicast triggers for Whack-on-LAN. Of course, this would depend on a trusted internal network. In any case, all of the trigger events supported by an adapter may be enabled or disabled on an individual basis through software such as *ethtool* [2].

AMD's Magic Packet is the safest option for Whack-on-LAN because it requires knowledge of the target machine's MAC address. Furthermore the MAC address must be repeated 16 times immediately after six bytes of hex 'FF', which virtually eliminates accidental wake-up events from random or everyday traffic. However, the Magic Packet specification allows this data to be anywhere inside an Ethernet frame. This means that the Magic Packet sequence can be embedded into any network protocol on any OSI layer that can be successfully routed to the intended host. This is hard for most firewalls to detect, since it requires the examination of every incoming packet that could reach the host. Without proper security, it is feasible for a malicious user to perform a Denial-of-Service (DoS) attack if the attacker is aware of the Whack-on-LAN functionality, if the MAC address is known, and if a packet or frame can be successfully routed to the target host.

There are several existing approaches to preventing DoS attacks through Whack-on-LAN. "SecureOn" is an extension to AMD's Magic Packet specification that incorporates a configurable six-byte passcode into the sequence the adapter must recognize. A brute-force DoS attempt with a 48-bit key would be impractical in nearly any network environment and could be easily detected. The key could also be changed frequently to thwart brute-force attempts and to prevent replay attacks after a legitimate reboot. Unfortunately, SecureOn is not supported by many network adapters, most notably the Intel Pro/100 S adapters, which have proven independent WOL circuitry. In Emulab, we use trusted networks to ensure security with Whack-on-LAN. Each time a node is rebooted, a VLAN is created between a control switch and an experimental network interface that is inaccessible to the outside world. The Magic Packet is sent through the VLAN and then the VLAN is terminated.

## 5 Considerations and Extensions

As mentioned earlier, many vendors have implemented WOL functionality in differing ways on their network adapter chipsets. Due to this lack of conformity, special attention must be paid to ensure that specific network adapters can be utilized for Whack-on-LAN.

The most significant difference between Wake-on-LAN and Whack-on-LAN to a network adapter is that Whack-on-LAN is utilized when the machine is fully on, rather than in a sleep state. This is acceptable for network adapters such as the Intel PRO/100 S, which enable the WOL logic regardless of power states as long as WOL functionality has not been disabled on the card. Other adapters may not behave the same way, enabling their WOL circuitry only when the SLEEP# pin on the PCI bus is asserted, or when normal network activity has been disabled. Another important consideration is the existence of a WOL connector on the network adapter. The PCI 2.2 revision introduced the ability to send Power Management Events across the bus, thus eliminating the need for a separate WOL cable to connect the network adapter to the motherboard. As a result, many newer network adapters that support WOL may not have the WOL connector. This is also the case for onboard Ethernet controllers. Implementing Whack-on-LAN on these devices is therefore harder because the LAN WAKE-UP signal must be acquired through an intermediate solder or other custom modification.

The natural solution to these vendor-specific WOL discrepancies is unified vendor support for an updated standard. The original WOL specification with AMD's Magic Packet provided only one type of wake-up trigger and no security provisioning. It was only much later that vendors began supporting such extensions as additional wake-up triggers and SecureON, none of which is required or necessarily defined by any

standard. If chipset vendors were to incorporate all of the features described by this paper, namely power state-independent WOL circuitry, security options, and backwards compatibility with a WOL connector, both Wake-on-LAN and Whack-on-LAN could see far more widespread use and demand in the remote network world.

## 6 Conclusion

We have presented Whack-on-LAN, a lightweight and extremely inexpensive remote reset solution over Ethernet. We discussed its potential for use in varying applications, and characterized its strengths in scattered or small deployment, cost, and infrastructure requirements. We evaluated its usefulness and success in Emulab, noting a drastic improvement in overall uptime. Finally, we discussed important security issues pertinent to several types of networking environments, and related current and future considerations for the technology.

## Acknowledgements

This work was supported in part by the National Science Foundation, through the Networking Research Testbeds (NRT) program under Award number ANI-0335296, and by NSF grant CNS-0435485.

## References

- [1] AMD Corp. Magic Packet Technology. White Paper 20213, Rev A, AMD, Nov. 1995. [http://www.amd.com/us-en/assets/content\\_type/white\\_papers\\_and\\_tech\\_docs/20213.pdf](http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/20213.pdf).
- [2] ethtool. <http://sourceforge.net/projects/gkernel/>.
- [3] Intel Corp. Intel PRO/100 S Network Adapter. [http://www.intel.com/network/connectivity/products/pro100s\\_adapter.htm](http://www.intel.com/network/connectivity/products/pro100s_adapter.htm).
- [4] S. Shalunov and R. Carlson. Detecting Duplex Mismatch on Ethernet. In *Passive and Active Measurement Workshop*, Boston, MA, Mar. 2005. <http://www.pam2005.org/PDF/34310138.pdf>.
- [5] B. White, J. Lepreau, and S. Guruprasad. Lowering the Barrier to Wireless and Mobile Experimentation. In *Proc. HotNets-I*, Princeton, NJ, Oct. 2002.
- [6] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002.