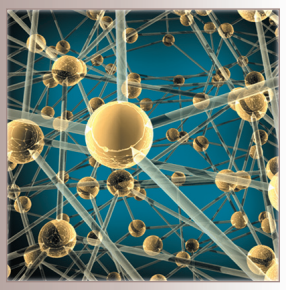

Burst Transmission for Energy-Efficient Ethernet

The proposed Energy-Efficient Ethernet (EEE) standard reduces energy consumption by defining two operation modes for transmitters and receivers: active and low power. Burst transmission can provide additional energy savings when EEE is used. Collecting data frames into large-sized data bursts for back-to-back transmission maximizes the time an EEE device spends in low power, thus making its consumption nearly proportional to its traffic load. An initial evaluation shows that the additional savings in the scenarios considered range from 5 to 70 percent for conventional users and approximately 50 percent for large data centers.

Pedro Reviriego and Juan Antonio Maestro
Universidad Antonio de Nebrija

José Alberto Hernández and David Larrabeiti
Universidad Carlos III de Madrid

Efficient energy use in communications is a growing concern for industry and governments worldwide. The massive number of communication devices we use today, together with their expected growth, have led researchers to conclude that we can save significant energy by applying energy-efficiency concepts in the design of communication systems.[1] Indeed, the Internet core is estimated to consume approximately 6 terawatt-hours (TWh) per year, a figure that we can significantly reduce if we deploy energy-aware protocols.

Ethernet is a good example of technology that could be more energy efficient; some people estimate that we could cut its energy use by more than 3 TWh.[2] To reduce waste, the IEEE P802.3az Energy-Efficient Ethernet (EEE) Task Force (see http://grouper.ieee.org/groups/802/3/az/public/) is introducing energy-efficiency enhancements to the existing Ethernet, a process expected to produce a new standard later this year.

Essentially, current Ethernet standards require both transmitters and receivers to operate continuously on a link, thus consuming energy all the time, regardless of the amount of data exchanged. The upcoming EEE standard aims to make energy consumption over a link a little more proportional to the amount of traffic exchanged[3] (for more information, see the "Energy-Efficient Ethernet" sidebar). Clearly, this change has profound implications

 1089-7801/10/$26.00 © 2010 IEEE

Energy-Efficient Ethernet

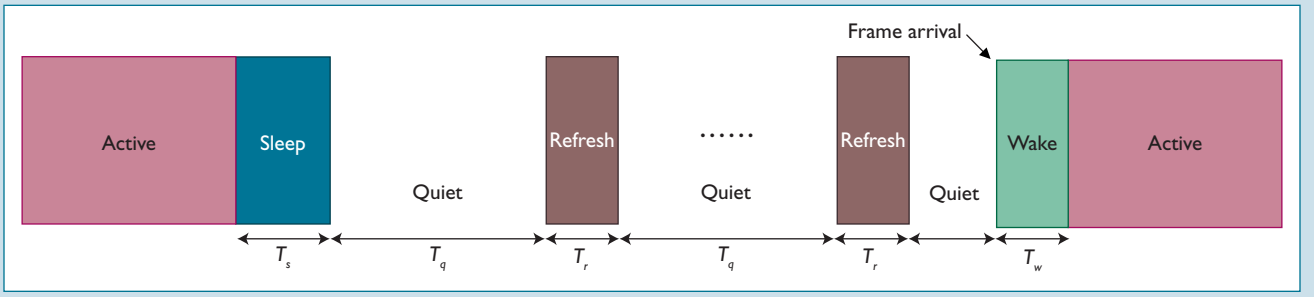

Figure A. Transitions between the active and sleep modes in Energy-Efficient Ethernet. $\mathbf{T_s}$ *is sleep time (the time needed to enter sleep mode);* $\mathbf{T_w}$ *is wake-up time (the time required to exit sleep mode). The transceiver spends* $\mathbf{T_q}$ *in the quiet (energy-saving) period but also has short periods of activity (*$\mathbf{T_r}$*) to refresh the receiver state.*

The main idea behind Energy-Efficient Ethernet (EEE) is to put the physical layer (PHY) into sleep (low-power) mode when no data is being transmitted. This potentially saves considerable energy because links are usually lightly loaded (see www.ieee802.org/3/eee_study/public/mar07/bennett_01_0307.pdf).

Several methods exist for implementing sleep mode. The most obvious is to reduce link speed when little traffic is exchanged.[1] You can achieve this through autonegotiation,[2] which is already part of the IEEE 802.3 Ethernet standards. Autonegotiation is currently used during link setup to determine the highest link speed that both ends support. However, you can also use it to reduce energy consumption by selecting a lower speed; the lower the link speed is, the less power the devices consume. Nevertheless, speed autonegotiation requires from a few hundred milliseconds (ms) to a few seconds,[2] which is excessive for many applications.

To accelerate the speed change, researchers have proposed other alternatives, such as Rapid PHY Selection (RPS; see www.ieee802.org/3/eee_study/public/mar07/christensen_02_0307.pdf). RPS uses a frame exchange to renegotiate a speed change without restarting autonegotiation. So, the decision on the speed change can take much less time. Nevertheless, speed changes require adjusting many elements in the receivers — including equalizers, echo cancellers, and timing circuits — to the new speed. These changes require a nonnegligible amount of time during which the link is down. Additionally, although speed downgrades reduce energy consumption, they don't make it proportional to the actual traffic load. In short, speed changes mitigate the problem of energy consumption with minimal changes, but this can be further improved.

A better alternative is to put the device to sleep when no transmission is needed but wake it quickly upon data arrival without changing its speed. This is the option chosen by the IEEE 802.3az Task Force, which has analyzed the mechanisms to support the sleep mode for the different Ethernet speeds — say 100 Mbps, 1 Gbps, and 10 Gbps. This sleep mode freezes the elements in the receiver and wakes them in just a few microseconds (μs). Such sleep/active operation requires only minor changes to the hardware because the channel is quite stable.

Figure A shows a state transition example of a given link, following the IEEE 802.3az draft.[3] T_s sleep time (the time needed to enter sleep mode); T_w is wake-up time (the time required to exit sleep mode). The transceiver spends T_q in the quiet (energy-saving) period. Finally, the standard also considers the scheduling of short periods of activity T_r to refresh the receiver state to ensure that the receiver elements are always aligned with the channel conditions.

With 100Base-TX and 10GBase-T Ethernet, both the

continued on p. 52

on the design of the Ethernet's physical layer devices (PHYs) and might drive changes in other algorithms and upper-layer protocols.

Although frame scheduling is out of the scope of the standard, we could further reduce energy consumption using a burst-transmission algorithm. Burst transmission maximizes the time an EEE device spends in sleep mode, thus making its consumption nearly proportional to its traffic load. Therefore, collecting data frames into large-sized data bursts for back-to-back transmission can lead to even larger energy savings, in the range of 5 to 70 percent for conventional end users and approximately 50 percent for large data centers in the scenarios considered.[4]

Energy-Efficient Burst Transmission

The potential energy savings of burst-transmission EEE make it well worth considering. Something else to consider, however, is that burst-transmission EEE can intro-

Energy-Efficient Ethernet (cont.)

Table A. Proposed wake-up time (T_w), sleep time (T_s), and frame transmission times (T_{frame}) for different link speeds.

Speed	Minimum T_w (μsec)	Minimum T_s (μsec)	Frame size (bytes)	T_{frame} (μsec)	Single-frame efficiency (%)
100Base-TX	30.50	200.00	1,500	120.00	34.2
			150	12.00	4.9
1000Base-T	16.50	182.00	1,500	12.00	5.7
			150	1.20	0.6
10GBase-T	4.48	2.88	1,500	1.20	14.0
			150	0.12	1.6

transmitter and receiver can operate independently regarding active mode and sleep mode. In other words, the link can send data (in active mode) in one direction while it's idle (in sleep mode) in the opposite direction. However, this isn't permitted with 1000Base-T, in which the link enters or exits sleep mode in both directions at the same time. On the other hand, 100Base-TX and 1000Base-T permit a transition back to active mode during a transition to sleep mode without needing time to exit sleep mode, thus increasing efficiency.

Energy consumption is significant only during T_w, T_s, and T_r, with a small fraction occurring during T_q, with $T_q \gg T_r$. The IEEE 802.3az draft specifies the minimum and maximum values for T_w, T_s, T_q, and T_r for 100Base-TX, 1000Base-T, and 10GBase-T. Table A gives the minimum values for T_s and T_w, along with the subsequent frame transmission efficiencies for long and short frames.

Indeed, implementing sleep mode brings large power savings — close to 90 percent for 100Base-TX, 1000Base-T, and 10GBase-T with respect to the current standards, which operate at full power all the time. However, as Table A shows, the wake-up and sleep times are considerably high with respect to the frame transmission time T_{frame}, especially when the frame size is small in bytes. For example, assume that a given device is in sleep mode upon a frame arrival. At this point, the device must wake up (which takes T_w), transmit its frame (this takes T_{frame}), and go to sleep again (this takes T_s). In total, the transmission of a single frame takes $T_w + T_{frame} + T_s$, whereas only T_{frame} is for actual data transmission. This algorithm, on a 10-Gbps link, requires $T_w \geq 4.48$ μs, $T_s = 2.88$ μs, and $T_{frame} = 1.2$ μs for the transmission of a 1,500-byte data frame. This results in 14 percent efficiency because most of the time (and energy) is spent in waking the link and putting it back to sleep.

Such energy overhead is particularly high for small data frames and at high-speed rates. This is evident in Table A, in which the single-frame efficiency values refer to the energy spent on transmitting a single frame with respect to the total energy spent in the EEE process (waking the link, transmitting the frame, and putting the link to sleep). However, we can easily achieve high efficiency levels if the device is awake only when 100 data frames have arrived for transmission (about 94 percent in the previous example). This visually proves the benefits of collecting data frames and sending them as a single unit or burst over the link (burst transmission) with respect to single-frame transmission. (For more on burst transmission, see the main article.)

References

1. C. Gunaratne et al., "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Trans. Computers*, vol. 57, no. 4, 2008, pp. 448–461.
2. C.E. Spurgeon, *Ethernet: The Definitive Guide*, O´Reilly Media, 2000.
3. *IEEE P802.3az Energy-Efficient Ethernet Standard Draft D3.0*, IEEE P802.3az Energy-Efficient Ethernet Task Force, May 2010.

duce some burstiness to the traffic – which might degrade network performance – and also cause extra delay to the frames, because they must wait until a number of them have arrived. The balance of energy savings versus time delay is something that can be easily achieved (as we explain later), but it's important to remember this, and to design the energy-efficient data-transmission schedule to accomplish a clear goal: it should maximize the time the device spends in sleep mode, but it shouldn't delay data transmission excessively to affect the upper-layer protocols and the applications' performance.

Sergiu Nedevschi and his colleagues explored the use of burst transmission to improve energy efficiency for arbitrary values of wake-up and sleep timers, in the context of a generic network.[4] Here, we focus on the EEE standard draft, showing the energy savings of burst-transmission EEE for several representative scenarios.

Initial Experiments

Proportionality occurs when a linear relationship exists between the system's load and its energy consumption.[3] Unfortunately, this isn't the case for today's EEE standard, whose performance we recently studied.[5] Essentially, we discerned that the large values of the wake-up and sleep timer (see Table A in the sidebar), with respect to the frame transmission time, make EEE differ from proportionality. Following that study, we simulated the energy consumption of 100-Mbps, 1-Gbps, and 10-Gbps links at different traffic load values. For simplicity, we assumed that both link directions operate independently, although this isn't true for the 1000Base-T Ethernet standard. Links enter sleep mode only when no frames are pending for transmission. We simulated the wake-up and sleep timers in EEE following the current standard draft (summarized in Table A in the sidebar). Although the timers might change in forthcoming revisions, their actual values don't affect our reasoning here.

Additionally, our experiment considered 12,000-bit data frames arriving at the link following a Poisson process. This assumption can be a valid approximation for servers in large data centers dealing with many parallel independent connections. Finally, we assumed power consumption in sleep mode to be 10 percent of that in active mode for all Ethernet speeds, according to the estimates provided by different manufacturers during the EEE's standardization process.[6-8]

Figure 1 shows energy consumption versus traffic load for the current standards and after introducing the EEE's two power modes for the three link speeds (100 Mbps, 1 Gbps, and 10 Gbps). EEE assumes that the link becomes active upon a frame's arrival and is put into sleep mode as soon as no frames are ready for transmission.

As Figure 1 shows, current Ethernet standards operate at maximum power all the time, thus consuming 100 percent of their energy regardless of the traffic load. By introducing the two power modes, EEE achieves energy consumption ratios more proportional to the traffic load, showing important energy savings, especially at low loads. However, for 1000Base-T and 10GBase-T, the relationship between load and energy consumption is still far from proportionality, which would appear as a straight line from the plot's bottom-left to its top-right corner. Basically, such poor results result from the large values of the sleep and wake-up timers compared to a single frame's actual transmission time. This is because most of the energy

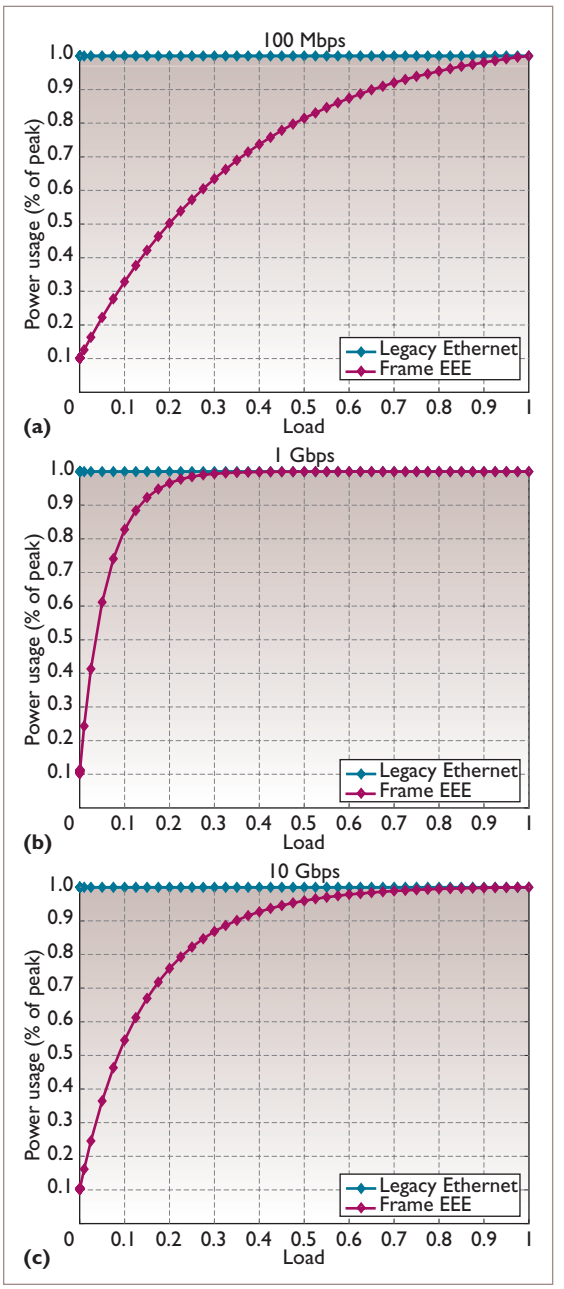

Figure 1. Energy consumption versus traffic load for the different Ethernet speeds: (a) 100 Mbps, (b) 1 Gbps, and (c) 10 Gbps. Current standards (the legacy Ethernet, plotted in blue) operate at maximum power all the time, consuming full energy regardless of the traffic load. Energy-Efficient Ethernet (called here Frame EEE and plotted in red) allows energy consumption to be more proportional to the traffic load, which should save much energy.

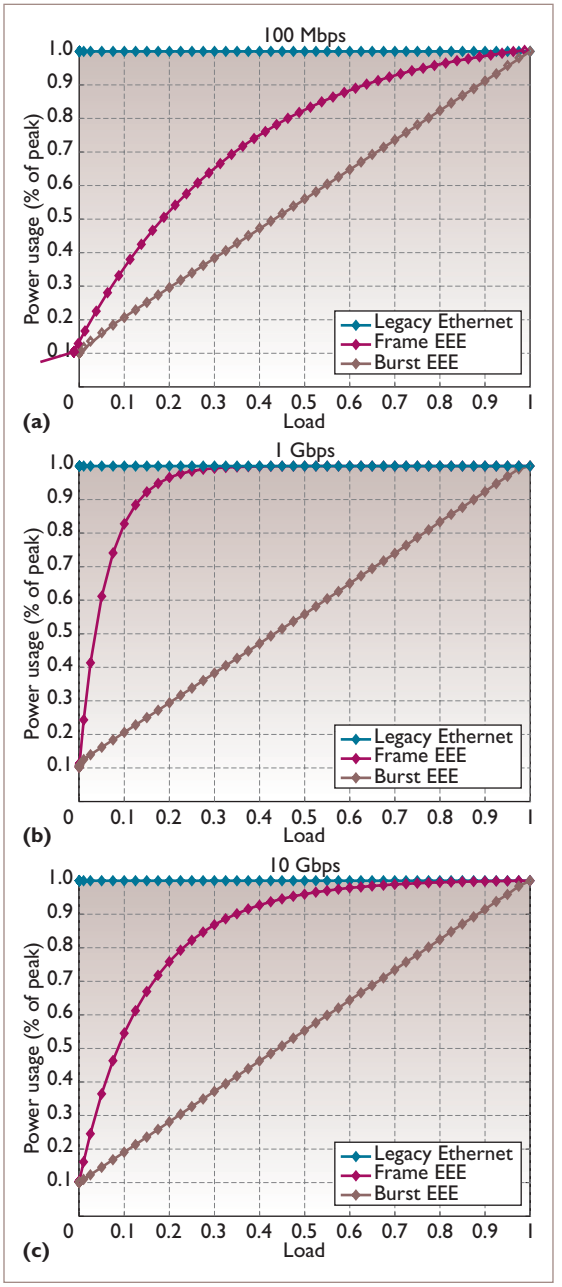

Figure 2. Energy consumption vs. traffic load when using burst transmission at (a) 100 Mbps, (b) 1 Gbps, and (c) 10 Gbps. The energy plots now approach the proportional relationship between energy and load.

is spent on waking the link and putting it to sleep, rather than on data transmission. This is particularly harmful with high-speed links. We can achieve much better efficiency, then, by making the link active for only the transmission of a large number of data frames (burst transmission), rather than for the transmission of a single frame.

As we previously stated, gathering frames into bursts adds delay to the frames until a burst unit is complete. We can bound this delay by using a timer-driven assembler that collects frames only during some T_{as} units of time and wakes the link for transmission after this timer expires. So, the timer T_{as} acts as an upper-delay bound because every frame waits no longer than this amount. It should be carefully designed on the basis of the delay requirements of both users and applications. Most residential users tolerate a few tens of milliseconds of delay. Also, the assembly timer might produce some buffer overflows if too many packets arrive within T_{as}. To avoid this situation, we can use the timer in conjunction with a data-size threshold so that when sufficient frames arrive at the network card, the burst releases without waiting for the timer to expire. The following experiments consider a maximum buffer size of 1,000 data frames, so that if this buffer fills up, it sends the data straightaway.

To illustrate the benefits of burst transmission in EEE, Figure 2 extends the previous simulation results with a 10-ms timer burst-transmission scheduler. As the figure shows, the energy plots now approach the proportional relationship between energy and load.

Figure 3 shows the ratio between using frames and bursts in terms of energy consumption for different timer values. Using bursts always saves substantial energy, especially at medium load levels for 100Base-TX and at low load levels for 1000Base-T and 10GBase-T. Also, Figure 3 shows greater energy savings for large values of T_{as}. For instance, $T_{as} = 1$ ms doesn't introduce excessive delay and still can provide large potential savings compared to single-frame EEE transmission. By far, the largest potential energy savings are for 1000Base-T and 10GBase-T links, which are also those that consume more power — approximately 1 and 5 watts, respectively. In data centers in which we deployed 1000Base-T and 10GBase-T, burst transmission provides a clear benefit not only by saving energy but also by cooling down the equipment.

Finally, it's well known that Ethernet LAN traffic shows self-similarity[9] and that data frames' interarrival times aren't exponentially distributed.[10] Nevertheless, the previous simulation results provide a first approach to the expected energy savings.

Using Burst Transmission in Real Scenarios

To further validate burst transmission and estimate the potential savings more accurately, we performed an analysis based on traffic measurements collected from four real scenarios.

Scenario 1 involved a residential user downloading video content from the Internet. This user connected via 100-Mbps Ethernet to his or her Asymmetric Digital Subscriber Line (ADSL) router. As Table 1 shows, a 10-ms burst-assembly timer produces a downstream energy savings of 9.25 percent using single-frame EEE transmission. The upstream savings is only 5.91 percent because the transmitted data is mostly TCP acknowledgments.

Scenario 2 involved two users exchanging a file over the same 100-Mbps LAN as in the previous scenario. As Table 1 shows, EEE can potentially achieve large energy savings — especially with upstream burst-transmission scheduling (73.49 percent). That's because the link is lightly loaded and the average frame length is much smaller, thus reducing the overhead of waking the link and putting it to sleep. However, this example highlights single-frame EEE's primary shortcoming: its limited efficiency when transmitting small frames.

Scenario 3 involved a 1000Base-T university access link with highly multiplexed Internet traffic. When computing the results, we considered both directions to operate independently; however, for 1000Base-T, both directions must enter active or sleep mode at the same time. Nevertheless, this issue doesn't affect the results significantly because in this case the link was in active mode 90 percent of the time in both directions. As Table 1 shows, we can reduce energy by at least 70 percent if we employ burst-transmission scheduling on the university access link, given its low load.

Scenario 4 involved a few server traces from Google's data centers, where energy consumption is a major concern. The traces belonged to three typical server types: a file server that's also involved in search queries (scenario 4a), a server devoted to search queries (scenario 4b), and a server acting as both the file and application server (scenario 4c). In each case, we concluded that we can achieve important energy savings for servers that operate at low loads with small frames on average. Particularly, the energy savings are symmetrical for the search server (scenario 4b) and asymmetrical in the other two cases, with more efficiency shown in the input direction. Concerning link load, the energy savings with single-frame EEE increase by more than 50 percent on low-loaded servers (as with input in 4a and both input and output in 4b and 4c). In the very low load cases with small frame sizes (such as input in 4a and 4c), burst-transmission EEE can achieve important energy savings — up to 90 percent with legacy

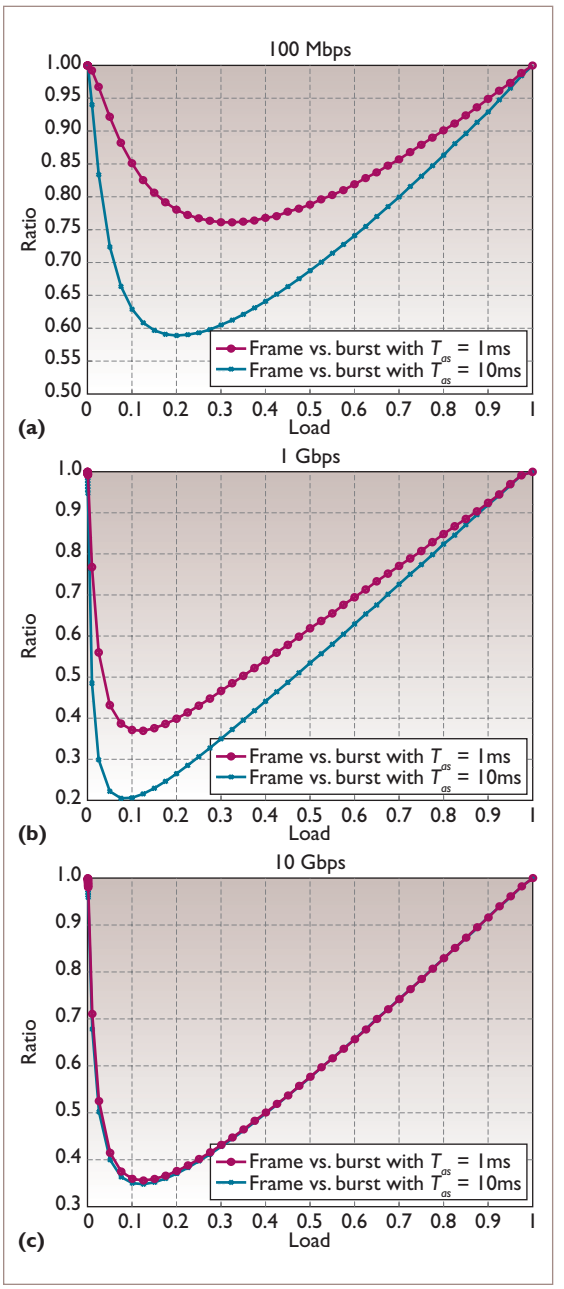

Figure 3. Energy consumption ratio of EEE burst and frame transmission at (a) 100 Mbps, (b) 1 Gbps, and (c) 10 Gbps. Larger assembly timers (T_{as}) save more energy.

Table 1. Energy consumption estimates for different measurement-based scenarios.

Scenario	Direction	Speed	$Energy_{frame}$ (% of peak)*	$Energy_{burst}$ (% of peak)*	Link load (%)	Average frame size (bytes)	Energy savings (%)
1. Residential user video download	Download	100 Mbps	12.75	11.57	1.43	1,444	9.25
	Upload	100 Mbps	10.99	10.34	0.04	90	5.91
2. Residential user file transfer	File	100 Mbps	78.68	74.25	71.13	1,499	5.63
	Acknowledgments	100 Mbps	44.92	11.91	1.39	77	73.49
3. University Internet access link	Download	1 Gbps	92.80	23.47	10.94	679	74.71
	Upload	1 Gbps	96.20	27.24	17.66	919	71.68
4a. Data center: file and search server	Input	1 Gbps	65.90	12.60	1.22	87	80.88
	Output	1 Gbps	72.92	57.73	52.21	1,497	20.83
4b. Data center: search server	Input	1 Gbps	45.28	18.85	8.51	945	58.37
	Output	1 Gbps	42.30	17.73	7.23	934	58.09
4c. Data center: file and application server	Input	1 Gbps	61.37	11.77	0.65	130	80.82
	Output	1 Gbps	57.10	14.72	4.02	749	74.22

*$Energy_{frame}$ is the energy used for single-frame transmission; $Energy_{burst}$ is the energy used for burst transmission.

Ethernet and beyond 80 percent with single-frame EEE. Basically, single-frame EEE provides important energy savings in high-speed scenarios at low loads, but we can greatly improve these with burst-transmission EEE if the average frame is small.

Burst transmission brings many open issues; here we look at four.

First, a frame traversing multiple links might experience excessive delay. We can avoid this by implementing burst transmission only on the last-hop links. This will ensure that at most two assembly timers contribute to the delay. Because most links are connected to end stations, most of the energy savings would still occur. Those links are also normally lightly loaded, so burst transmission has great potential to improve their energy efficiency. We could also use burst transmission on the links between switches but use small values for the assembly timers. This is possible because such links are usually highly loaded and operate at high speed, and small assembly timers can achieve significant energy improvements. For example, to assemble 100 frames of 10,000 bits on a 100-Mbps link with a 1-percent load, we need 100 ms, whereas we need only 1 ms for a 1-Gbps link with a 10-percent load.

Second, large data bursts might cause buffer overflow on the switches, especially on the uplink ports. Fortunately, these are typically overdimensioned (10×) and can deal with multiple simultaneous burst arrivals. However, the impact of the increased burstiness on network performance needs careful study.

Third, increasing the round-trip time (RTT) translates to decreased TCP throughput. However, we can use small values of the assembly timer to mitigate burst-transmission EEE's negative effects on TCP throughput. For instance, EEE assembly timers in the order of a few milliseconds should have a limited impact on the TCP throughput for connections with RTT values of tens of milliseconds, while saving substantial energy. Additionally, for the same throughput, an RTT increase also requires an increment in the TCP window, which causes TCP throughput to reach its steady state a bit later. Finally, correlated losses due to buffer overflow might trigger TCP's congestion-avoidance mechanisms, with subsequent performance degradation. Deployment of burst-transmission EEE must carefully consider all these cause-and-effect issues.

Finally, next-generation Ethernet standards (40/100G) might benefit from burst-transmission EEE if the approach to making them energy efficient is based on the same active/sleep transitions as in 100Base-TX, 1000Base-T, and 10GBase-T. Essentially, given the increase of link speed by one order of magnitude with respect to 10GBase-T, the waking and sleeping values (T_w and T_s) must be rescaled by one order

of magnitude, too. Otherwise, the power overheads of an eventual deployment of EEE with large values of T_w and T_s would be even higher than those of Table A in the sidebar, with much time (and energy) spent on waking and putting to sleep the link for transmitting a single frame. In such a case, it might be worth collecting several frames before waking a link, given the high energy cost of activating a link.

Acknowledgments

We thank Kevin Fu, Greg Chesson, Luis André Barroso, and Urs Hölzle from Google for the traces collected in their data centers and used in the experiments. We carried out the research described in this article partly with the support of the Building the Future Optical Network in Europe project, a Network of Excellence funded by the European Commission through the Seventh Information and Communication Technologies Framework Program and the Spanish project TIN2008-06739-C04-01/TSI.

References

1. M. Gupta and S. Singh, "Greening of the Internet," *Proc. 2003 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications*, ACM Press, 2003, pp. 19–26.
2. C. Gunaratne et al., "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Trans. Computers*, vol. 57, no. 4, 2008, pp. 448–461.
3. L.A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *Computer*, vol. 40, no. 12, 2007, pp. 33–37.
4. S. Nedevschi et al., "Reducing Network Energy Consumption via Sleeping and Rate-Adaptation," *Proc. Usenix Symp. Networked System Design and Implementation*, Usenix, 2008, pp. 323–336.
5. P. Reviriego et al., "Performance Evaluation of Energy Efficient Ethernet," *IEEE Communications Letters*, vol. 13, no. 9, 2009, pp. 697–699.
6. J. Chou et al., "Proposal of Low-Power Idle: 100Base-TX," presentation at the IEEE 802.3 Jan. 2008 meeting; www.ieee802.org/3/az/public/jan08/chou_01_0108.pdf.
7. M. Grimwood et al., "Energy Efficient Ethernet 1000BASE-T LPI Timing Parameters," presentation at the IEEE P802.3 July 2008 meeting; www.ieee802.org/3/az/public/jul08/grimwood_02_0708.pdf.
8. D. Taich et al., "10GBASE-T Low-Power Idle Proposal," presentation at the IEEE P802.3 May 2008 meeting; www.ieee802.org/3/az/public/may08/taich_02_0508.pdf.
9. W.E. Leland et al., "On the Self-Similar Nature of Ethernet Traffic," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, 1994, pp. 1–15.
10. V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modelling," *IEEE/ACM Trans. Networking*, vol. 3, no. 3, 1995, pp. 226–244.

Pedro Reviriego is an associate professor in computer science at Universidad Antonio de Nebrija. His research interests are fault-tolerant systems, evaluating communication network performance, and the design of physical-layer communication devices. Reviriego has a PhD in telecommunications engineering from Universidad Politécnica de Madrid. He's participated in the IEEE 802.3 standardization for 10GBase-T. Contact him at previrie@nebrija.es.

Juan Antonio Maestro manages the Computer Architecture and Technology Group at Universidad Antonio de Nebrija. His research interests include signal processing and real-time systems, fault tolerance, and reliability. Maestro has a PhD in computer science from Universidad Complutense de Madrid. Contact him at jmaestro@nebrija.es.

José Alberto Hernández is an associate professor of computer networking at Universidad Carlos III de Madrid. His research interests include the areas at which mathematical modeling and computer networks overlap. Hernández has a PhD in computer science from Loughborough University. Contact him at jahgutie@it.uc3m.es.

David Larrabeiti is a full professor of computer networking at Universidad Carlos III de Madrid. His research interests include the design of future Internet infrastructures, ultrabroadband multimedia transport, and traffic engineering of Internet Protocol-Generalized Multiprotocol Label Switching (IP-G/MPLS) networks. Larrabeiti has a PhD in telecommunication engineering from Universidad Politécnica de Madrid. Contact him at dlarra@it.uc3m.es.

cn *Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.*