On the Elusive Benefits of Protocol Offload

Position paper

Piyush Shivam
Department of Computer Science
Duke University
Durham, NC 27708 USA
shivam@cs.duke.edu

Jeffrey S. Chase[*]
Department of Computer Science
Duke University
Durham, NC 27708 USA
chase@cs.duke.edu

ABSTRACT

Periodic order-of-magnitude jumps in Ethernet bandwidth regularly reawaken interest in TCP/IP transport protocol offload. This time the jump to 10-Gigabit Ethernet coincides with the emergence of new network storage protocols (iSCSI and DAFS), and vendors are combining these with offload NICs to position IP as a competitor to FibreChannel and other SAN interconnects. But what benefits will offload show for application performance?

Several recent studies have presented conflicting data to argue that offload either does or does not benefit applications. But the evidence from empirical studies is often little better than anecdotal. The principles that determine the results are not widely understood, except for the first principle: Your Mileage May Vary.

This paper outlines fundamental performance properties of transport offload and other techniques for low-overhead I/O in terms of four key ratios that capture the CPU-intensity of the application and the relative speeds of the host, NIC device, and network path. The study also reflects the role of offload as an enabler for direct data placement, which eliminates some communication overheads rather than merely shifting them to the NIC. The analysis applies to Internet services, streaming data, and other scenarios in which end-to-end throughput is limited by network bandwidth or processing overhead rather than latency.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling Techniques

General Terms

Measurement, Performance, Design

Keywords

Protocol offload, TCP, IP, RDMA, network performance, high-speed networking, network storage

[*]This work is supported in part by the U.S. National Science Foundation (EIA-9972879 and ANI-126231) and by Cisco, HP Labs, Intel, IBM, and Network Appliance.

1. INTRODUCTION

Advancing network speeds and the rise of IP storage have renewed the debate over TCP/IP protocol offload. Storage systems and cluster services generate high data rates, motivating use of SAN networks such as FibreChannel to improve performance. As the bandwidth gap between Ethernet and non-IP SANs continues to decline, commodity Ethernet/IP networks offer a cost-effective and standards-based alternative for network storage and other demanding applications. A frequently cited drawback of IP networks is that data copying and TCP/IP protocol overheads can consume a significant share of host CPU cycles and memory system bandwidth, siphoning off end-system resources needed for application processing. This overhead can increase latency and limit application throughput relative to a non-IP SAN with equivalent wire speed. Protocol offload to the network device (NIC) is an attractive solution to reduce this communication overhead.

IP transport offload has been highly controversial for more than a decade [11, 2]. TCP/IP protocol processing is not itself the fundamental bottleneck that it is often perceived to be: several studies show that protocol costs are small relative to data-touching overheads and the cost to interface the TCP/IP stack to the NIC device and operating system [4, 9, 8, 3]. The popular textbook by Patterson and Hennessy [7] notes the "pitfall" that a long time-to-market for complex offload NICs leaves them at risk of being outstripped by advances in host CPU speeds: offload may do more harm than good if the NIC becomes a bottleneck [13]. A compensating factor is that transport offload is a structural change that exposes information useful for direct data placement from the NIC into host memory. Thus offload may make it possible to eliminate some portion of overhead entirely, rather than merely shifting it to the NIC. Examples of TCP/IP offload as a basis for direct data placement include iSCSI NICs and emerging IP standards for Remote Direct Memory Access (RDMA). This trend is a key reason why IP transport offload may be "a dumb idea whose time has come" [11].

Several recent empirical studies have presented conflicting data to argue that offload does or does not benefit applications [13, 12, 6, 10]. While these studies offer valuable real-world comparisons, their results depend on the specific technologies and applications studied. This paper identifies key system parameters for protocol offload and low-overhead communication, and uses basic algebra to analyze their effects independently and in combination. Our purpose is to illustrate interactions across the entire parameter space independent of any specific technology point or application. An improved understanding of the fundamental performance principles will help to guide the debate about offload and the interpretation of point studies.

Our study characterizes the potential benefits of transport offload

Table 1: Four LAWS ratios determine the potential benefit from protocol offload. Higher ratio values diminish the benefit.

Lag ratio	α	Ratio of host processing speed to NIC processing speed.
Application ratio	γ	Ratio of normalized application processing to communication processing; CPU-intensity of the application.
Wire ratio	σ	Ratio of host saturation bandwidth to raw network bandwidth; portion of network bandwidth the host can deliver without offload.
Structural ratio	β	Ratio of the normalized processing overhead for communication with offload to the overhead without offload; portion of normalized overhead that remains in the system (either NIC or host) after offload.

for application throughput as a function of four key ratios. The ratios—Lag (α), Application (γ), Wire (σ), and Structural (β) or LAWS—capture speed differences between the host and the network, the CPU-intensity of the application, and structural factors that may eliminate work in the offload case. We hold constant any other factors that may affect communication processing work. Table 1 summarizes the LAWS ratios and Section 2 describes them in detail.

The acronym LAWS suggests that these ratios expose fundamental properties of transport offload and related techniques to reduce I/O overhead. Although other factors may further limit the benefits in practice, these properties are "laws" in the same sense as Amdahl's Law, rather than approximations from a model that one may choose to accept or reject based on empirical validation. The chief limitation of this study is that it applies only when applications are throughput-limited, i.e., they are fully pipelined. Since overhead affects latency, latency-sensitive applications may benefit from offload in ways that are not captured here. Thus our study complements LogP [5] and related analyses that focus on latency.

Our metric of merit is *marginal improvement in peak application throughput*. The key conclusions of our analysis are as follows:

- Offload is worthwhile primarily for applications with modest processing load (low γ); the benefit for more CPU-intensive applications is bounded by $1/\gamma$ and diminishes to zero.
- The benefit of offload may be arbitrarily large for a host that is not fast enough to communicate at wire speed ($\sigma << 1$)—next year's network connected to last year's computer—but in this case the large benefits apply to a vanishingly small range of low-γ applications. At the other extreme, slow networks ($\sigma >> 1$) show little or no benefit from offload because the host handles the processing easily without assistance. In all cases the benefit is bounded by $1/\sigma$.
- For "realistic" cases with $\sigma \geq 1$ the benefits are bounded by a factor of two. Offload is most beneficial when the host is evenly balanced between communication overhead and application processing before offload ($\gamma = \sigma = 1$).
- If the NIC technology lags behind the host ($\alpha > 1$), then the benefit is bounded by $1/\alpha$ and offload may damage performance if (but only if) the NIC saturates before the network path ($\alpha\beta > \sigma$). However, structural improvements ($\beta < 1$) expand the range of systems that benefit from a given NIC technology. This quantifies how direct data placement can improve the engineering economics for TCP/IP offload by easing time-to-market pressure.

2. LAWS PARAMETERS AND MEASURES

Table 2 summarizes the input parameters for our analysis. Consider an application executing on a reference host with a single CPU attached to a network of bandwidth B. Application processing generates CPU work a per unit of bandwidth on average over some interval; communication processing generates CPU work o per unit of bandwidth. The parameter o encapsulates all factors that may affect communication processing work (e.g., cycles per byte), such as packet size, the specific transport protocol used, packet loss rates, implementation tricks to streamline protocol processing, and the ratio of transmits to receives.

Table 2: Input parameters for the LAWS ratios.

o	CPU occupancy for communication overhead per unit of bandwidth, normalized to a reference host
a	CPU occupancy for application processing per unit of bandwidth, normalized to a reference host
X	Occupancy scale factor for host processing
Y	Occupancy scale factor for NIC processing
p	Portion of communication overhead o offloaded to NIC
B	Bandwidth of the network path

Since the units of bandwidth and time are arbitrary, we can express a and o in units of CPU occupancy (utilization) normalized for the reference host. We introduce a scaling parameter X on occupancy to capture variations in processing power relative to the reference host. If the application moves data at network speed, then the CPU spends a portion aXB of its time on application processing and a portion oXB on communication overhead. If these processing costs saturate the CPU ($aXB + oXB > 1$), then the system is host-limited and the maximum throughput is $1/(aX + oX)$. Thus peak throughput doubles ($1/X$) on a host that is twice as fast as the reference host ($X = 0.5$). Choosing $X = 1/P$ represents the maximum potential benefit from a multiprocessor with P reference processors under the assumption that any unit of work may execute on any CPU with no additional overhead.

Suppose now that the system offloads a portion p of the communication overhead to the NIC. Then raw communication at network speed B consumes only $(1-p)oXB$ of host processing power and the host can reach higher speeds before it saturates. LAWS holds o fixed whether the work runs on the host or the NIC, but we introduce an occupancy scaling parameter Y to represent the NIC processing power relative to the reference host; thus communication at network speed consumes a share $poYB$ of NIC processing capacity, and the NIC can move data at a rate $1/poY$ before saturating.

We can now express the key ratios from Table 1 as:

- **Lag** ratio ($\alpha = Y/X$) is the ratio of host CPU speed to NIC processing speed. We may view the lag in terms of NIC time to market: assuming that both hosts and NICs advance at equivalent rates following Moore's Law—a doubling of processing power every 18 months—then $\alpha = 2$ corresponds to a NIC lagging host technology by 18 months. Higher α means the NIC lags further behind the host.

Table 3: Some useful values derived from the parameters.

$(1-p)oX$	Host occupancy for network processing
$1/(1-p)oX$	Raw host throughput ($\gamma = 0$)
$1/((1-p)oX + aX)$	Peak throughput when host-limited
$poY\beta$	NIC occupancy per unit bandwidth
$1/poY\beta$	Peak throughput when NIC-limited
B	Peak throughput when network-limited

- **Application** ratio ($\gamma = a/o$) captures the CPU-intensity or compute/communication ratio of an application: higher γ means the application is more CPU-intensive. For example, $\gamma = 2$ means that the application processing per unit of bandwidth is double the fundamental communication overhead, i.e., overhead accounts for 1/3 of the processing work for the application. We may view γ in terms of Amdahl's Law, which tells us that the benefit of offload or any other technique for low-overhead I/O declines as γ increases and the application itself becomes the primary bottleneck.

- **Wire** ratio ($\sigma = 1/oXB$) is the ratio of host saturation bandwidth to peak network bandwidth. Much of our analysis focuses on the case where the host can process raw communication at network speed but saturates the host CPU in doing so ($\sigma = 1$). This is the best case for transport offload that is still "realistic" in the sense that the host is powerful enough to drive its network fully without offload. If $\sigma < 1$, then σ gives the peak percentage of wire speed that the host can deliver without offload assist.

- **Structural** ratio (β) captures the effect of offload as an enabler for direct data placement and other techniques to eliminate overhead rather than merely shifting it to the NIC. β represents the portion of overhead that is unavoidable. For example, $\beta = 0.6$ could indicate that an offload NIC can demultiplex received data earlier and place it correctly in host memory (e.g., using RDMA), saving 40% of the overhead by eliminating a copy. For simplicity, we consider β only for the full offload case, i.e., we take $\beta = 1$ whenever $p < 1$. Communication consumes only a portion $oY\beta$ of NIC processing capacity per unit of bandwidth, rather than oY.

Table 3 summarizes some measures derived from the parameters. We can see that peak throughput in all scenarios is given by $min(B, 1/poY\beta, 1/((1-p)oX + aX))$. Higher throughput numbers are better, so the *marginal improvement in peak throughput* or "benefit" is defined as $(after - before)/before$. We may multiply by 100 to express the marginal improvement value as a percentage. The base (*before*) case uses no offload ($p = 0$), so its peak throughput is given by $min(B, 1/(oX + aX))$.

Several unaccounted factors may lead to throughputs lower than these ideal values. Contention within the system may cause a or o to increase with load, particularly in multiprocessors. If the system is bursty or suffers from limited concurrency then it may fail to utilize its resources fully, so latency may affect throughput. In these cases the expressions in Table 3 give *upper bounds* on throughput. It is possible that unaccounted factors (particularly latency) could impact throughput for non-offload cases more than for the offload cases; if so, the results could understate the benefit from offload. However, the results are reliable for fully pipelined systems when the parameters in Table 2 can be given accurately.

LAWS is similar to the LogP [5] model with some differences and extensions reflecting their complementary goals. LogP provides a framework to analyze scalable parallel programs, which tend to be latency-sensitive. Thus LogP focuses on latency as a primary parameter, and its o and g (NIC overhead) parameters reflect a fixed division of communication-related overheads between the host and the NIC. LAWS, on the other hand, is a framework to analyze the effect of shifting and varying these overheads according to the structure of the communication system, and to quantify their impact on peak application throughput. This impact is independent of the number of processors in the system, in contrast to the per-processor communication bandwidth in LogP.

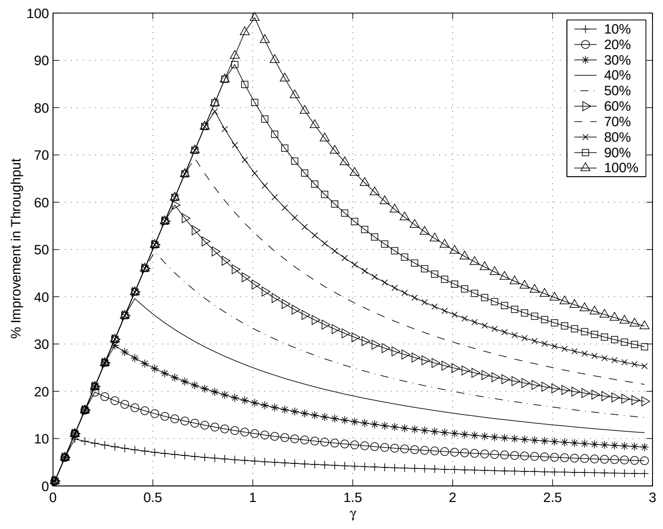


Figure 1: Benefit of offload as a function of application CPU-intensity (γ) for varying degrees of offload (p), holding other ratios at 1 (see Section 3.1). The benefit is bounded by a factor of two in this ideal realistic case ($\sigma = 1$), and is bounded by $1/\gamma$ as applications become more CPU-intensive. Amdahl's Law dictates that the benefit approaches zero as γ increases.

3. ANALYSIS

This section explores the parameter space outlined in the previous section. The entire analysis assumes that the system is host-limited before offload, since offload always yields zero benefit otherwise. Sections 3.1 and 3.2 vary the application (γ) and lag (α) ratios to illustrate effects on the marginal improvement in peak throughput from varying degrees (p) of protocol offload. Next, Section 3.3 varies the structural (β) ratio for the full offload ($p = 1$) case, to show the effect of direct data placement or RDMA on the range of applications (characterized by γ) that can benefit from offload for a given lag ratio. All of these results assume that $\sigma = 1$, i.e., the host is just powerful enough to process raw communication at wire speed—the "best realistic case" for offload. Section 3.4 shows the effect of varying σ values based on a general formulation of the LAWS analysis.

3.1 Effect of Application Ratio (γ)

Figure 1 shows the peak marginal benefit of offload for a range of applications characterized by their CPU-intensity (the γ ratio), holding the other three ratios equal to 1. It assumes that the NIC does not lag the host ($\alpha = 1$), so the offload NIC can never limit throughput.

We make the following observations from Figure 1:

- Offload can improve application throughput by at most a factor of two (100%). The peak benefit is obtained at $\gamma = 1$,

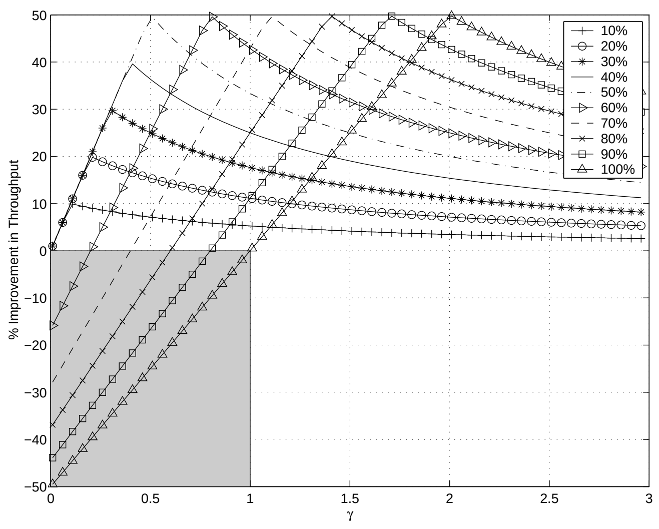


Figure 2: Effect of offload as a function of application CPU-intensity (γ) for varying degrees of offload (p). This graph is similar to Figure 1 except that the lag ratio is $\alpha = 2$: the NIC technology lags the host by 18 months. The slower NIC diminishes the benefit of offload when it saturates for communication-intensive (low-γ) applications; the maximum benefit is reduced to $1/\alpha$, and offload does more harm than good in the shaded region (see Section 3.2).

when the host CPU is exactly balanced between application processing and communication overhead without offload.

- The benefit diminishes rapidly for more CPU-intensive applications. It is easy to derive that the benefit is bounded by $1/\gamma$; more specifically it is bounded by $p/(\gamma - p + 1)$. Thus offload can improve throughput by at most 33% for a system that spends only 25% of its CPU time in communication-related overhead ($\gamma = 3$).

- When $\gamma < p$ the benefit is equal to γ independent of the degree of offload p. This occurs because the system is network-limited after offload, so increasing the degree of offload provides no further benefit. The benefit reaches zero when $\gamma = 0$; at $\sigma = 1$ the the host can handle raw communication ($\gamma = 0$) at wire speed without NIC assistance, so offload provides zero benefit.

3.2 Effect of Lag Ratio (α)

Figure 2 shows how the potential benefits of offload decline for a NIC that is slower than the host ($\alpha > 1$). This is a well-known pitfall discussed in the classic Patterson and Hennessy [7] text. In this example the NIC handles communication at only half of the wire speed B; raw communication at bandwidth B just saturates the host ($\sigma = 1$), but the NIC processes communication only half as fast as the host ($\alpha = 2$). This corresponds to a NIC lagging host technology by 18 months attached to a network that is well-matched to the host. We make the following observations generalizing the results from Section 3.1.

- The lagging NIC reduces the maximum potential benefit from offload. It is easy to derive that the maximum benefit is $1/\alpha$, and that it occurs at $\gamma = \alpha$ in the full offload ($p = 1$) case; more specifically, it occurs at $\gamma = p\alpha + p - 1$.

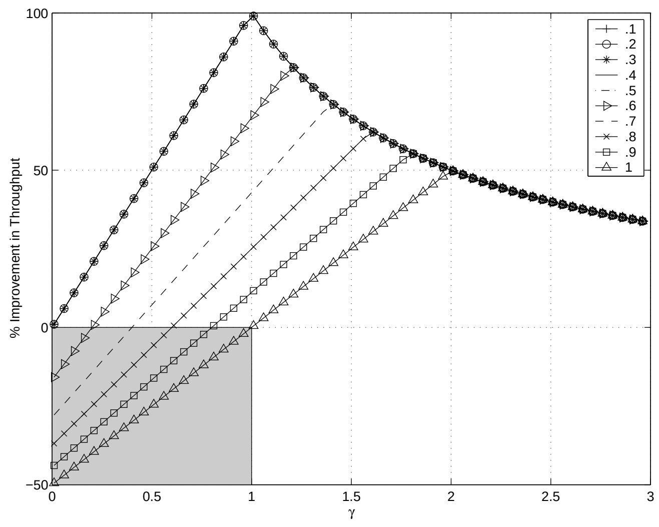


Figure 3: Potential benefit of full offload as a function of CPU-intensity (γ) for varying β, with $\alpha = 2$ and $\sigma = 1$ (see Section 3.3). β improves the NIC-limited cases by a slope factor of $1/\beta$, offsetting the slower NIC. This shows how enhancements such as direct data placement can improve the economics of offload.

- Offload hurts performance for low-γ applications that saturate the NIC before the host CPU. This effect is evident in the shaded region of Figure 2. Shifting overhead away from the host does not improve throughput because the host is not saturated, and the saturated NIC limits the data rate below the level achieved without offload. In this example the marginal improvement or decline is $(\gamma + 1)/p\alpha - 1$ when the system is NIC-limited after offload; with full offload the line is $(\gamma - 1)/2$.

- The benefits of offload are independent of α when the NIC does not limit throughput. The effects are unchanged from the results in Section 3.1: in the CPU-limited (high γ) cases the system delivers improvement $1/\gamma$, and in the network-limited cases the benefit is γ and peaks at $\gamma = p$.

3.3 Effect of Structural Ratio (β)

The previous section characterizes the sensitivity of offload solutions to NIC technology and time-to-market. This effect may be offset if transport offload serves as an enabler for structural improvements to eliminate overhead. In particular, extensions to offload NICs may process the data stream to place incoming data directly in designated host memory buffers to avoid copying within the host. For example, many iSCSI NICs place data directly for network storage over TCP/IP, and emerging IP standards for Remote Direct Memory Access (RDMA) can generalize direct data placement to other network applications. The long history of research on copy avoidance has generated many partial solutions that do not depend on offload, but NIC-based solutions offer crucial advantages from the standpoint of generality, ease of deployment, and interoperability [2].

Such structural optimizations eliminate a portion $1 - \beta$ of the overhead rather than merely shifting it to the NIC. This improves the economics of transport offload because it eases time-to-market pressure to keep NICs current with host technology (low α). In essence, β acts as an adjustment to α: the bandwidth point at which communication overhead saturates the NIC is $1/\alpha\beta$ times the host saturation point rather than $1/\alpha$. Figure 3 shows the effect for different values of β, holding $\alpha = 2$ as in Section 3.2. We make the following observations:

- Decreasing β compensates for a slow NIC by increasing the slope of the improvement line by a $1/\beta$ factor when the system is NIC-limited after offload. In the NIC-limited case, offload improves or diminishes throughput by $(\gamma+1)/\alpha\beta - 1$. When $\alpha\beta \leq 1$ the system can never be NIC-limited after offload, so offload can no longer hurt performance. For example, if offload enables structural changes that eliminate 50% of the overhead ($\beta = 0.5$) then the NIC technology may lag the host by 18 months on the Moore's Law curve ($\alpha = 2$) without risk of the NIC becoming a bottleneck.
- The maximum possible improvement from offload is bounded by $1/\alpha\beta$, and it occurs when $\gamma = \alpha\beta$.
- Like α, β influences performance only when the system is NIC-limited after offload. When $\gamma \geq \alpha\beta$ the system is host-limited after offload; it returns to the $1/\gamma$ improvement curve independent of α and β. When $\alpha\beta \leq 1$ the system becomes net-limited when $\gamma < 1$, and it returns to the γ improvement line.

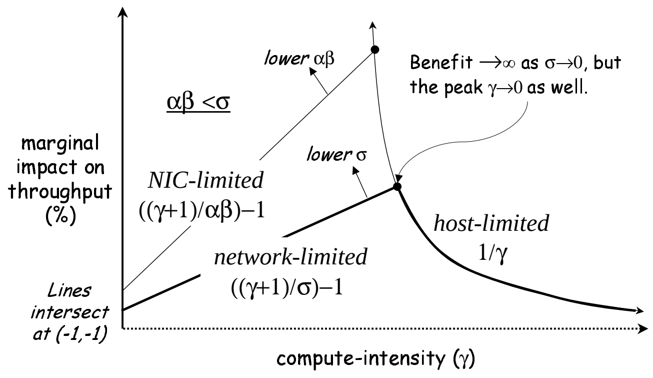
3.4 Effect of Wire Ratio (σ)

Until this point the analysis has assumed the "best realistic case" in which the host speed just matches the network bandwidth ($\sigma = 1$). CPU speeds improve along a smooth curve, while network advances follow a step function. This means that network technologies in the market at any given time span two orders of magnitude. Thus it is important to explore the impact of protocol offload with relatively faster networks ($\sigma < 1$) or slower networks ($\sigma > 1$). Some have argued that network bandwidth will continue to advance at twice the rate of Moore's Law [1], resulting in a long-term shift toward lower σ values. Indeed, much of the current interest in offload is a response to 10-Gigabit Ethernet and other high-speed network technologies, which push σ lower at least temporarily.

If we hold fundamental overhead constant, then the value of σ varies with the inverse of the network speed B. To explore the effect of σ it is therefore sufficient to consider just the case in which the system becomes network-limited after offload, since this is the only case in which B and σ can impact the result in ways not covered by our previous analysis. As previously stated, our analysis assumes that the system is host-limited rather than network-limited before offload, since the benefit of offload is always zero otherwise. Thus B has no effect if the system is also host-limited or NIC-limited after offload.

It is easy to derive that the benefit of offload is $(\gamma + 1)/\sigma - 1$ when the system is network-limited after offload. This is analogous to the NIC-limited case discussed in the previous two sections, in which the peak benefit for full offload is given by $(\gamma+1)/\alpha\beta - 1$. The marginal benefit of offload is determined by the points at which these lines intersect the familiar $1/\gamma$ improvement curve, which applies to the case in which the system is host-limited after offload. Figure 4 illustrates the combination of these factors, generalizing the earlier figures. Note that the graph does not specify the point at which these lines intersect the y-axis; indeed, the impact of offload may be negative, as in Figures 2 and 3. This is why the x-axis in Figure 4 is a dotted line.

In general, the impact of offload for each value of γ is given by the minimum of the three functions evaluated at γ, depicted by the bold tracking line in Figure 4. If $\alpha\beta < \sigma$ (as in Figure 4)



αβ <σ
lower αβ
Benefit →∞ as σ→0, but the peak γ→0 as well.
lower σ
marginal impact on throughput (%)
NIC-limited ((γ+1)/αβ)−1
network-limited ((γ+1)/σ)−1
host-limited 1/γ
Lines intersect at (-1,-1)
compute-intensity (γ)

Figure 4: The interaction of NIC-limited, network-limited, and host-limited scenarios. The marginal improvement is given by the network-limited (if $\alpha\beta \leq \sigma$) or NIC-limited (if $\sigma \leq \alpha\beta$) scenario until the peak benefit is reached at the intersection with the host-limited ($1/\gamma$) improvement curve.

then the system can never become NIC-limited after offload because the network saturates before the NIC. Similarly, if $\alpha\beta > \sigma$ then the system can never become network-limited after offload because the NIC saturates before the network. Taking the constant $c = max(\alpha\beta, \sigma)$, we can see that the benefit (or impact) of offload is given by the interaction of the $1/x$ curve and the $((x+1)/c) - 1$ line, as depicted in Figure 5. It is easy to see that the line always intersects the curve at $x = c$, and that this is the point of maximum benefit $1/c = 1/x$. This is the fundamental basis for many of our previous results.

With respect to the impact of σ, note the following:

- The benefit of offload is bounded by $1/\sigma$ as well as by $1/\gamma$. The peak benefit occurs when the system saturates both the network and the host after offload ($\gamma = \sigma$ and $\alpha\beta < \sigma$).
- The maximum benefit of offload is unbounded as σ approaches zero—a fast network and a slow host. However, these large benefits apply to a vanishingly small range of gamma values representing little or no application processing of the data; in fact, the peak benefit occurs as γ also approaches zero. For applications that spend at least as much CPU time on application processing as communication overhead ($\gamma \geq 1$), the benefit of offload is always bounded by $1/\gamma$: offload can never improve throughput by more than a factor of two for these applications even on fast networks.
- Offload has little value on relatively slower networks ($\sigma >> 1$). For example, a host that is powerful enough to communicate at 10 times the network bandwidth before saturating ($\sigma = 10$) can never obtain more than a 10% benefit ($1/\sigma$) from offload. Moreover, this diminished benefit applies to an increasingly narrow range of γ values: as σ increases, the benefits accrue only for increasingly compute-intensive applications ($\gamma = \sigma$) and are correspondingly smaller in accordance with Amdahl's Law.
- More efficient protocols and implementations increase both γ and σ, undermining the benefits of offload.

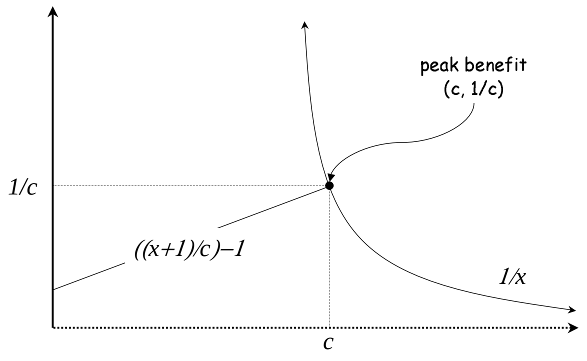


Figure 5: An abstraction of the functions governing the benefit of offload, where $c = max(\alpha\beta, \sigma)$. The peak benefit $1/c$ occurs when $x = c$, i.e., $\gamma = max(\alpha\beta, \sigma)$.

4. CONCLUSION

The contribution of this paper is to capture many of the factors governing the effectiveness of TCP/IP offload in terms of simple relationships among the four LAWS ratios. Point studies may be misleading if they evaluate systems and applications that lie within narrow regions of the parameter space. The LAWS analysis provides a comprehensive basis for interpreting experimental results and framing the debate over offload technologies. The results apply to throughput-limited applications such as Internet servers; we plan to address latency-sensitive applications in future work.

It would be wrong to conclude from our analysis that TCP/IP offload is not worthwhile. Indeed, our study supports the conclusion that offload is more worthwhile now than it has ever been, as a result of faster networks and direct data placement pushing σ and β downwards. In practice it is reasonable to expect benefits up to 50% or perhaps as much as a factor of two for ideal communication-intensive applications on high-speed networks. In these scenarios offload NICs offer a relatively inexpensive opportunity to gain a few months or even a year or more on the Moore's Law curve.

However, the LAWS analysis exposes fundamental limits on the benefits of offload, and the fragile—even "elusive"—nature of those benefits. The benefits of offload accrue primarily for low-γ applications such as transaction servers and network storage servers with large numbers of disk arms, visualization engines, and streaming data processing. Faster networks and memory system limitations may increase the benefits of offload appreciably by pushing σ down, but the larger benefits may not extend beyond a handful of synthetic (low-γ) benchmarks in the lab. Indeed, the trends for industrial-strength Internet services may favor higher γ values as these services are increasingly based on powerful but slow scripting languages such as Perl, Python, and PHP. We have measured γ ratios as high as 40 for dynamic-content Web service benchmarks, limiting the benefit of offload in those settings to a few percentage points.

The LAWS analysis is sufficiently abstract to apply to any technique to reduce I/O overhead or shift it to another component of the system. For example, many in the industry view TCP/IP offload as essential to enable IP-based networking to compete with SANs such as FibreChannel. This may be so, but the performance advantage of today's SANs relative to IP networks is similarly limited according to the LAWS results.

Acknowledgments

We thank the anonymous reviewers and our shepherd Vijay Pai for their help in improving the paper. This work also benefited from discussions with Omer Asad, Kostas Magoutis, Alvin Lebeck, Jeff Mogul, Jaidev Patwardhan, Prasenjit Sarkar, and Dan Sorin.

5. REFERENCES

[1] Offload White Paper. http://www.national.com/appinfo/networks/files/whitepaper1.pdf, 2003.

[2] Jeffrey S. Chase. TCP implementation. In High-Performance TCP/IP Networking, Mahbub Hassan and Raj Jain, editors. Prentice Hall, December 2003. ISBN 0-13-064634-2. *http://issg.cs.duke.edu/publications/tcp.pdf*

[3] Jeffrey S. Chase, Andrew J. Gallatin, and Kenneth G. Yocum. End system optimizations for high-speed TCP. *IEEE Communications, Special Issue on High-Speed TCP*, 39(4):68–74, April 2001.

[4] David D. Clark, Van Jacobson, John Romkey, and Howard Salwen. An analysis of TCP processing overhead. *IEEE Communications Magazine*, 27(6):23–29, June 1989.

[5] David E. Culler, Richard M. Karp, David A. Patterson, Abhijit Sahay, Klaus E. Schauser, Eunice Santos, Ramesh Subramonian, and Thorsten von Eicken. LogP: Towards a realistic model of parallel computation. In *Proceedings 4th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 1993.

[6] Matt DeBergalis, Peter Corbett, Steve Kleiman, Arthur Lent, Dave Noveck, Tom Talpey, and Mark Wittle. The Direct Access File System. In *USENIX Symposium on File and Storage Technologies (FAST)*. USENIX, March 2003.

[7] John L Hennessy and David Patterson. Computer Architecture: A Quantitative Approach, 2nd Edition. Morgan Kauffman Publishers, 1995.

[8] Van Jacobson. 4BSD header prediction. *ACM Computer Communication Review*, 20(2):13–15, April 1990.

[9] Jonathan Kay and Joseph Pasquale. The importance of non-data touching processing overheads in TCP/IP. In *Proceedings of the SIGCOMM Symposium on Communications Architectures and Protocols*, pages 259–268. ACM, September 1993.

[10] Kostas Magoutis, Salimah Addetia, Alexandra Fedorova, and Margo I. Seltzer. Making the most out of direct-access network attached storage. In *2nd USENIX Symposium on File and Storage Technologies (FAST)*, March 2003.

[11] Jeff Mogul. TCP offload is a dumb idea whose time has come. In *Workshop on Hot Topics in Operating Systems (HotOS)*. May 2003.

[12] M.Rangarajan, A. Bohra, K. Banerjee, E. Carrera, R. Bianchini, L. Iftode, and W. Zwaenepoel. TCP Servers: Offloading TCP Processing in Internet Servers—Design, Implementation and Performance. Technical Report, DCS-TR-481.

[13] Prasenjit Sarkar, Sandeep Uttamchandani, and Kaladhar Voruganti. Storage over IP: When does hardware support help? In *2nd USENIX Symposium on File and Storage Technologies (FAST)*, March 2003.