Optimization of TCP segment size for file transfer

by R. M. Bournas

In this paper, we study the problem of optimal Transmission Control Protocol (TCP) segment size for file transfer from hosts to clients. The criterion of optimality is the minimization of the amount of TCP and IP (Internet Protocol) processing by the sender. The parameters that govern the host-processing cost include the cost for processing both the outgoing TCP segments and incoming TCP acknowledgments, the TCP window size, the maximum transferable unit (MTU) size of the network, and the network reliability factor. We study the variations of the sender processing cost as a function of the TCP segment size and the network reliability factor. We show that there exists a network reliability factor γ_0 such that 1) for all network reliability factors $\gamma \leq \gamma_0$, the optimal TCP segment size equals the MTU size less the sizes of the TCP and IP headers (the sender processing cost increases with the TCP segment size in this case); and 2) for each $\gamma > \gamma_0$, there exists an optimal TCP segment size that is greater than the MTU size. Moreover, the optimal TCP segment size is an increasing function of the network reliability factor. We also derive a sufficient condition for the optimal TCP segment size to be greater than the MTU size. In this case, a lower bound for the optimal TCP segment size can be obtained as a simple function of the network reliability factor.

Introduction

In this paper, we consider the problem of optimal TCP (Transmission Control Protocol) segment size for file transfer from hosts to clients (outbound). The criterion of optimality is the minimization of the amount of TCP/IP (Internet Protocol) processing by the sender. The TCP segment size is usually chosen to maximize file-transfer throughput (see [1–5] and the references therein). However, on some hosts, such as the IBM S/390* and AS/400* large mainframe servers, the processing cost is expensive, and there is thus a need to minimize the TCP/IP processing. In addition, these systems are required to support high transaction rates for database and file server applications, so there is a need to minimize the consumption of CPU cycles. To the best of the author's knowledge, this type of optimization has not been studied prior to this work.

The system we describe[1] consists of a file server connected to clients through a TCP/IP communications network. In most TCP implementations known to the author, the size of a TCP segment equals the maximum transferable unit (MTU) size, a function of the network, less the sizes of TCP, IP, and network-medium headers. The processing of small TCP segments and TCP acknowledgments can be costly in systems where input/output processing is expensive—particularly the processing of TCP acknowledgments, as these are very small in size yet require processing all the way from the network layer to the TCP layer. One then may be inclined

[1] U.S. Patent Office application RA9-96-003, August 8, 1996.

0018-8646/97/$5.00 © 1997 IBM

to ask how to reduce the processing of acknowledgments. One way is to increase the size of TCP segments. By making the size of TCP segments large, we reduce the amount of outbound and inbound (acknowledgments) processing because, for a fixed file size, the number of TCP segments and acknowledgments to be processed decreases as the TCP segment size increases. However, using large TCP segments may cause other problems, as we explain below.

As described in detail in the following section, when the IP layer receives a TCP segment larger than the MTU size of the network medium, it fragments it into smaller pieces equal to the MTU size, for network transmission. IP fragmentation is often thought of as bad for performance because when such a fragment is lost or erroneous, the entire TCP segment that contains it must be retransmitted. Such TCP segments must then be reprocessed at the IP layer and retransmitted, which causes the host processing cost to increase.

We then have the following situation. On one hand, the cost of network-communications processing decreases as the TCP segment size increases because the number of TCP segments and acknowledgments to be processed decreases. On the other hand, the network-communications processing cost increases as the TCP segment size increases because of the retransmissions of TCP segments that occur when one of their fragments is lost or erroneous. This suggests that there is an optimal TCP segment size—one that leads to the smallest number of host CPU cycles consumed. This optimum size is a function of the various host processing costs and the network reliability factor.

It is not our goal in this paper to develop a methodology to calculate the network reliability factor, γ (the long-term average of the ratio of the number of successfully received network packets to the total number of transmitted packets). This depends on several parameters, such as the number of hops in the network-communications path, the error rate of each hop, the buffer size of the switching nodes, and the traffic intensity. A detailed and precise study of this subject exceeds the scope of this paper and should be the topic for further investigation. However, for some particular cases, such as one- or two-hop networks, one could make a rough estimate of γ by experimentation.

We show that there exists a network reliability factor γ_0 such that 1) for all network reliability factors $\gamma \leq \gamma_0$, the optimal TCP segment size equals the MTU size less the sizes of the TCP and IP headers (the sender processing cost increases with the TCP segment size in this case); and 2) for each $\gamma > \gamma_0$, there exists an optimal TCP segment size that is greater than the MTU size less the header sizes. Moreover, the optimal TCP segment size is an increasing function of the network reliability factor. We also derive a sufficient condition for the optimal TCP segment size to be greater than the MTU size. In this case, a lower bound for the optimal TCP segment size can be obtained as a simple function of the network reliability factor.

The paper is organized as follows. In the following section, we formulate the problem, and in the next section, we analyze it, determining for which network reliability factors the optimal TCP segment size is larger than the MTU size. For such network reliability factors, we prove the existence of a unique optimal TCP segment size, at which the host processing cost reaches a global minimum. In the section on bounds, we derive a sufficient condition on the network reliability factor for the optimal TCP segment size to be greater than the MTU size. We also obtain a lower bound for the optimal TCP segment size as a simple function of the network reliability factor. In the following section, we present some examples of optimal TCP segment size computation. We draw conclusions in the final section.

Problem formulation

In this section, we formulate the problem to be solved. First, we briefly review the File Transfer Protocol (FTP) algorithm. The data flow for outbound TCP/IP data transmission is depicted in **Figure 1**. Data are read from disks or other storage media of the host by the FTP layer and transferred in blocks of equal size to the TCP layer for processing. The TCP layer then encapsulates the data into segments of equal size, calculates the checksum of the data (parity check for data integrity), prepares a TCP header, and passes control to the IP layer. The latter breaks each TCP segment (if necessary) into IP datagrams of equal size, prepares a header for each IP datagram, and invokes the network layer, which appends a network-medium header and transmits the packet(s). We ignore the anomalies associated with the final blocks, segments, and datagrams. For a detailed overview of TCP/IP, the reader is referred to [1]. The amount of data in a network packet equals the MTU size less the sizes of headers for the network medium, TCP, and IP. Upon successful reception of the data, the receiver of the TCP layer sends an acknowledgment to the sender. The TCP architecture suggests that an acknowledgment be sent for every two TCP segments received; however, different implementations may use different acknowledgment algorithms. For the sake of simplicity, we do not take into account the size of TCP, IP, or network-medium headers in the problem formulation. These header sizes are typically twenty to forty bytes each.

We now define the variables used in the problem formulation:

- S_b (bytes): Size of data block transferred between FTP and TCP.
- p (bytes): Maximum network packet size (MTU).
- n: Number of network packets of size p in a TCP segment. (TCP segment size equals np.) We assume that n is an integer.
- C_m (instructions per byte): Data-moving cost. This is the cost per byte to move data from the FTP buffer to the TCP buffer.
- C_k (instructions per byte): Cost per byte to calculate the checksum in a TCP segment.
- C_b (instructions per data block): Fixed overhead cost to transfer a data block from the FTP layer to the TCP layer.
- $C(n)$ (instructions): Total host cost to process a data block, from FTP through IP levels.

Our model consists of moving two TCP segments from the IP layer to the network layer. The reason for two segments is that once an acknowledgment is received from the recipient, there is room for two more segments in the TCP window (maximum number of unacknowledged data bytes transmitted by the sender). We assume here that the data flow is in equilibrium. If we define

- C_{IPb} (instructions): Cost to move the contents of the IP buffer, containing one or more IP datagrams, from the IP layer to the network layer (or from the network layer to the IP layer),
- $C_{2TCPs}(n)$ (instructions): Cost to move two TCP segments of size np each from the IP layer to the network layer, and
- S_{IPb} (bytes): size of IP buffer (assumed to be fixed),

then

$$C_{2TCPs}(n) = \text{ceil}\left(\frac{2np}{S_{IPb}}\right) C_{IPb} \leq nC_{2TCPs}(1), \quad (1)$$

where ceil(x) denotes the smallest integer greater than or equal to x. The inequality follows from ceil$(nx) \leq n\,\text{ceil}(x)$. To complete the description of variables, we define

- a (instructions per segment): TCP processing cost to prepare a header (independent of segment size). This is also the cost to process an acknowledgment.
- d (instructions per network packet): IP processing cost to prepare a header (independent of packet size).
- $b \equiv d + C_{2TCPs}(1)/2$ (instructions per network packet): IP and network processing cost.
- $c \equiv (a + d + C_{IPb})/2$ (instructions per TCP acknowledgment): TCP, IP, and network processing cost.
- γ: Probability of a successful network packet transmission.

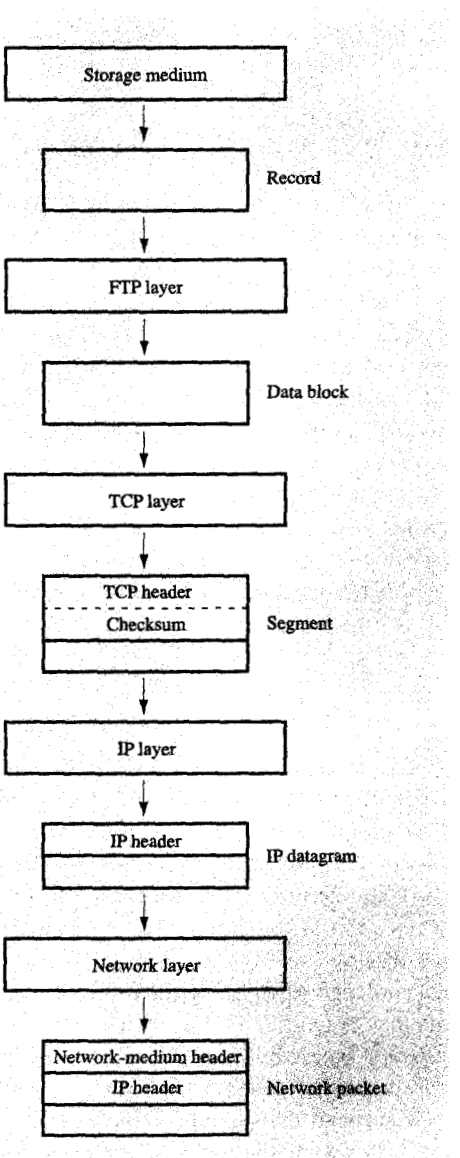

Figure 1

Data flow (processing layers and data entities) for outbound (host-to-client) data transmission in TCP/IP networks.

- $r \equiv C_{IPb}/2a$: Ratio of network processing cost to the sum of the TCP and IP processing costs (when $a = d$).
- $m \equiv \text{ceil}(2p/S_{IPb})$: Number of times to invoke the network layer to read or write two packets from or to the network, respectively.

It is assumed throughout this paper that each network packet carries p bytes, the maximum amount of data allowed by the network medium. When the TCP segment

size is equal to the MTU size (less the sizes of the TCP, IP, and network-medium headers), the total host cost to process a block of data from FTP to IP levels is given by

$$C(1) = C_b + (C_m + C_k)S_b + \frac{S_b}{p}\left(a + \frac{b+c}{\gamma}\right). \quad (2)$$

In Equation (2), the cost due to TCP, IP, and network processing (the third term on the right-hand side) is calculated as follows. For each TCP segment, there is a TCP processing cost of a, there are IP and network processing costs totaling b, and there is an acknowledgment-processing cost of c. Note that when a TCP segment is retransmitted, it is reprocessed by the IP and I/O layers only. The average number of times a packet is transmitted until it is received successfully, when the probability of successful network transmission of a packet is γ, is given by $1/\gamma$.

When the TCP segment size is an integral multiple n of the MTU size p, the total host processing cost is bounded by

$$C(n) \le C_b + (C_m + C_k)S_b + \frac{S_b}{np}\left(a + \frac{nb+c}{\gamma^n}\right) \equiv \mathcal{C}(n). \quad (3)$$

In Equation (3), the host network-processing cost per data block due to TCP, IP, and network processing (the third term on the right-hand side) is calculated as follows. The cost of TCP processing equals a per segment, and the cost of IP and network-layer processing per TCP segment is bounded by nb, where the bound follows from Equation (1). The cost per TCP segment to process the incoming acknowledgments sent by the receiver is c. The average number of times a TCP segment is transmitted before it is received successfully equals $1/\gamma^n$, because when a network packet fails to reach its destination successfully, the entire TCP segment that contains it must be retransmitted.

To determine the optimal TCP segment size, we study the difference

$$C(1) - \mathcal{C}(n) = \frac{S_b}{np\gamma^n}[a(n-1)\gamma^n + n(b+c)\gamma^{n-1} - nb - c] \quad (4)$$

as a function of γ. We study the difference $C(1) - \mathcal{C}(n)$ rather than $C(1) - C(n)$ to make the problem easier to analyze. We determine the values of γ for which this difference is positive, which is equivalent to determining the values of network reliability for which it is more economical to transmit TCP segment sizes larger than the MTU size. To simplify the notation, we define

$$\Delta(n, \gamma) \equiv a(n-1)\gamma^n + (b+c)n\gamma^{n-1} - nb - c. \quad (5)$$

For any value of γ that makes $\Delta(n, \gamma)$ positive, i.e., $\mathcal{C}(n) < C(1)$, we study the behavior of the cost function $\mathcal{C}(n)$ as a function of n. Our goal is to determine the integer value of n, denoted n_0, that minimizes $\mathcal{C}(n)$ for the given value of γ. Note that the optimal number of network packets in each TCP segment must not be greater than S_b/p; hence, it is the smaller of n_0 and the floor of S_b/p [floor(x) denotes the largest integer $\le x$].

Problem analysis

In this section, we analyze the problem formulated in the previous section. Specifically, we establish the following.

Lemma 1
For every $n > 1$, there exists a unique $\gamma_n < 1$, such that $\Delta(n, \gamma_n) = 0$, where $\Delta(n, \gamma)$ is as given by Equation (5). Moreover, for all $\gamma > \gamma_n$, $\Delta(n, \gamma) > 0$ [i.e., $\mathcal{C}(n) < C(1)$], and for all $\gamma < \gamma_n$, $\Delta(n, \gamma) < 0$.

Lemma 2
The sequence $\{\gamma_n\}$ is strictly monotonically increasing in n, for $n > 1$.

Lemma 3
γ_n tends to 1 as $n \to \infty$.

Lemma 4

- For each fixed $\gamma < \gamma_2 \equiv \gamma_0$, $C(1) < \mathcal{C}(n)$ for all $n > 1$.
- For each fixed $\gamma \ge \gamma_0$, there exists an integer $n_0 \ge 2$ that minimizes $\mathcal{C}(n)$. Moreover, for all $n < n_0$, $\mathcal{C}(n)$ is decreasing, and for all $n > n_0$, $\mathcal{C}(n)$ is increasing.

In Lemma 1, we establish, for any number, n, of network packets in a TCP segment, the existence of a range of network reliability factors for which it is more economical to use a segment size larger than the MTU size. One important application of Lemma 2 is to demonstrate that an optimal TCP segment size larger than the MTU size exists if and only if the network reliability factor γ is greater than or equal to γ_0. Lemma 3 states that the network reliability threshold (the point that determines whether or not it is optimal to design a TCP segment size larger than the MTU size) increases to 1 as the TCP segment sizes become indefinitely large. Finally, Lemma 4 enables us to empirically calculate the optimal TCP segment size for a given network reliability factor.

Let us begin by proving Lemma 1 above.

Proof of Lemma 1
Calculating the values of $\Delta(n, \gamma)$ given by Equation (5) for $\gamma = 0$ and $\gamma = 1$, we have for every $n > 1$, $\Delta(n, 0) = -nb - c < 0$, and $\Delta(n, 1) = (n-1)(a+c) > 0$. Hence, by the continuity of $\Delta(n, \gamma)$ in γ, there exists at least one value of γ, denoted γ_n, in the interval $(0, 1)$ such that $\Delta(n, \gamma) = 0$. Since $\Delta(n, \gamma)$ is monotonically

increasing in γ, it follows that a) γ_n is unique; b) for all $\gamma > \gamma_n$, $\Delta(n, \gamma) > 0$; and c) for all $\gamma < \gamma_n$, $\Delta(n, \gamma) < 0$. This finishes the proof of Lemma 1.

Proof of Lemma 2

According to the definition of γ_n and Equation (5), we have

$$\Delta(n, \gamma_n) = a(n-1)\gamma_n^n + (b+c)n\gamma_n^{n-1} - nb - c = 0 \qquad (6)$$

and

$$\Delta(n+1, \gamma_n) = an\gamma_n^{n+1} + (b+c)(n+1)\gamma_n^n - (n+1)b - c < 0. \qquad (7)$$

For all $n > 1$, if $\Delta(n+1, \gamma_n) < 0$, it follows that $\gamma_{n+1} > \gamma_n$, since the function $\Delta(n+1, \gamma)$ is monotonically increasing in γ. Thus, we can prove the lemma by proving that $\Delta(n+1, \gamma_n) < 0$. We can express $\Delta(n+1, \gamma)$ as

$$\Delta(n+1, \gamma_n) = \gamma_n[a(n-1)\gamma_n^n + (b+c)n\gamma_n^{n-1}] + a\gamma_n^{n+1} + (b+c)\gamma_n^n - b - nb - c. \qquad (8)$$

We rewrite Equation (6) in the following two forms:

$$a(n-1)\gamma_n^n + (b+c)n\gamma_n^{n-1} = nb + c, \qquad (9)$$

and

$$a\gamma_n^n = \frac{nb + c - n(b+c)\gamma_n^{n-1}}{n-1}. \qquad (10)$$

Substituting Equations (9) and (10) for the first and second terms on the right-hand side of Equation (8), respectively, we obtain

$$\Delta(n+1, \gamma_n) = \gamma_n(nb+c) + \frac{(nb+c)\gamma_n - n(b+c)\gamma_n^n}{n-1} + (b+c)\gamma_n^n - b - (nb+c). \qquad (11)$$

After simplification, we have

$$(n-1)\Delta(n+1, \gamma_n) = b[n^2(\gamma_n - 1) + 1 - \gamma_n^n] + c[n(\gamma_n - 1) + 1 - \gamma_n^n]. \qquad (12)$$

Then, however,

$$n(\gamma_n - 1) + 1 - \gamma_n^n = (\gamma_n - 1)[n - (1 + \gamma_n + \gamma_n^2 + \cdots + \gamma_n^{n-1})], \qquad (13)$$

and

$$n^2(\gamma_n - 1) + 1 - \gamma_n^n = (\gamma_n - 1)[n^2 - (1 + \gamma_n + \gamma_n^2 + \cdots + \gamma_n^{n-1})]. \qquad (14)$$

Since $\gamma_n < 1$, it follows that $1 + \gamma_n + \gamma_n^2 + \cdots + \gamma_n^{n-1} < n < n^2$ for $n > 1$. Hence, the expressions on the right-hand sides of Equations (13) and (14) are strictly negative, from which it follows that the expressions on the left-hand sides of Equations (13) and (14) are strictly negative. Using these results in Equation (12) demonstrates that $\Delta(n+1, \gamma_n) < 0$. This finishes the proof.

Proof of Lemma 3

The limit of the sequence $\{\gamma_n\}$ as $n \to \infty$ exists because it is monotonically increasing and upper bounded by 1. From Equation (6), we have, by using the inequality $\gamma_n^n < \gamma_n^{n-1}$ twice,

$$\left[\frac{nb+c}{a(n-1)+(b+c)n}\right]^{1/(n-1)} < \gamma_n < \left[\frac{nb+c}{a(n-1)+(b+c)n}\right]^{1/n} \quad \text{for } n > 1. \qquad (15)$$

Using the inequality $\gamma_n^n < \gamma_n^{n-1}$ twice with respect to the left-hand side of Equation (9), we have

$$a(n-1)\gamma_n^n + (b+c)n\gamma_n^n < a(n-1)\gamma_n^n + (b+c)n\gamma_n^{n-1} < a(n-1)\gamma_n^{n-1} + (b+c)n\gamma_n^{n-1}.$$

Replacing the middle sum with the right-hand side of Equation (9) and dividing all three sums by $a(n-1) + (b+c)n$ produces

$$\gamma_n^n < \frac{nb+c}{a(n-1)+(b+c)n} < \gamma_n^{n-1}.$$

Let us designate the center term by X, a positive quantity less than 1 that approaches $b/(a+b+c)$ as $n \to \infty$. We then have

$$X^{1/(n-1)} < \gamma_n < X^{1/n}.$$

Since both $X^{1/(n-1)}$ and $X^{1/n}$ tend to 1 as $n \to \infty$, it follows that $\gamma_n \to 1$ as $n \to \infty$. This finishes the proof.

Proof of Lemma 4

Because $\Delta(n, \gamma)$ is strictly monotonically increasing in γ, we have for all $n > 1$ and $\gamma < \gamma_2$ that $\Delta(n, \gamma) < \Delta(n, \gamma_2)$. Because of Lemma 2, we have $\Delta(n, \gamma_2) \leq \Delta(n, \gamma_n)$. Finally, because of Lemma 1, we have $\Delta(n, \gamma_n) = 0$. Thus, $\Delta(n, \gamma) < 0$, so $C(1) < \mathcal{C}(n)$ for all $n > 1$ and $\gamma < \gamma_2$. For $\gamma \geq \gamma_2$, again since $\Delta(2, \gamma)$ is monotonically increasing in γ, we have $\Delta(2, \gamma) \geq \Delta(2, \gamma_2) = 0$; hence, $\mathcal{C}(2) \leq C(1)$ and there exists $n \geq 2$ such that $\mathcal{C}(n) \leq C(1)$. We now show that $\mathcal{C}(n)$ has a global minimum at one point (designated n_0) or two adjacent points. Let us extend the domain of n to the set of real numbers, and let $\zeta(n)$ be the first-order derivative of $C(1) - \mathcal{C}(n)$ with respect to n, which equals the derivative of $\Delta(n, \gamma)/n\gamma^n$. Calculating this quantity leads to the following expression:

$$(n^2\gamma^n)\zeta(n) = a\gamma^n + b(\log \gamma)n^2 + c(\log \gamma)n + c. \qquad (16)$$

The first term on the right-hand side of Equation (16) is monotonically decreasing in n, has value a at $n = 0$, and

approaches 0 as $n \to \infty$. The remaining three terms form a quadratic in n that has a global maximum at some $n < 0$, has value c at $n = 0$, and approaches $-\infty$ as $n \to \infty$. Thus, the quadratic is decreasing for $n \geq 0$. Consequently, the right-hand side of Equation (16), which is the sum of the first term and the remaining quadratic, must be zero at some point, denoted n_0, in the range $(0, \infty)$. Moreover, for $n > n_0$, the right-hand side is negative, and for $n < n_0$, the right-hand side is positive. Hence, $\mathcal{C}(1) - \mathcal{C}(n)$ has a global maximum at $n = n_0$, from which it follows that $\mathcal{C}(n)$ has a global minimum at $n = n_0$. This finishes the proof.

Bounds for optimal transmission

In this section,

1. We determine a sufficient condition on the network reliability factor γ for the optimal TCP segment size to be greater than the MTU size (the condition is that the network reliability factor be greater than some function of n).
2. We derive a lower bound of the optimal TCP segment size, for any given network reliability factor γ which is a function of the given network reliability factor γ.
3. We show that the cost function $\mathcal{C}(n)$ is concave upward.

We make only the reasonable assumption that the TCP processing cost per segment equals the IP processing cost per network packet—i.e., $a = d$ (defined in the section on problem formulation). We normalize the cost difference [Equation (5)] by dividing it by the TCP and IP processing costs. This leads to a cost difference that is a function of the ratio of the network-layer processing cost to the TCP and IP processing costs, r, and the number of times the network layer is invoked to drive two network packets out to the network, m. (These two latter variables are also defined in the section on problem formulation). The concavity property of $\mathcal{C}(n)$ enables us to determine how fast $\mathcal{C}(n)$ decreases as a function of n, and thus how sensitive the choice of the optimal TCP segment size is with respect to the network reliability factor.

We now proceed to accomplish our first goal described above. Using the definitions of b, c, $C_{2\text{TCPs}}(1)$, m, and r with $a = d$, one derives in a straightforward manner

$$\frac{b}{a} = 1 + mr \tag{17}$$

and

$$\frac{c}{a} = 1 + r. \tag{18}$$

Using these two equalities in Equation (5), we obtain

$$\frac{\Delta(n, \gamma)}{a} = (n-1)\gamma^n + [2 + (m+1)r]n\gamma^{n-1} - (1 + mr)n - (1 + r). \tag{19}$$

Using Equation (19) and $\Delta(n, \gamma_n) = 0$, we derive the following equality:

$$\frac{(n-1)\gamma_n^n + 2n\gamma_n^{n-1} - (n+1)}{r} = 1 + mn - (m+1)n\gamma_n^{n-1}. \tag{20}$$

Also, using Equation (19) and $\Delta(n, \gamma_n) = 0$, for $n > 1$, we have the following nontrivial equality, derived by adding $[1 + (m+1)r](n-1)\gamma_n^n + [2 + (m+1)r]n\gamma_n^{n-1}$ to both sides of $\Delta(n, \gamma_n) = 0$ and rearranging terms:

$$(n-1)\gamma_n^n + 2n\gamma_n^{n-1} - (n+1) = \frac{[1 + (m+1)r][(n-1)\gamma_n^n + n\gamma_n^{n-1} + 1]}{2 + (m+1)r} + \frac{n(1 + \gamma_n^{n-1}) + (n-1)r}{2 + (m+1)r}. \tag{21}$$

Since all terms on the right-hand side of Equation (21) are positive, the left-hand side, which is the numerator of the left-hand side of Equation (20), must be strictly positive. Consequently, the right-hand side is strictly positive. For $n > 1$, we then deduce that

$$\gamma_n < \left[\frac{1 + mn}{(m+1)n}\right]^{1/(n-1)}. \tag{22}$$

From Lemma 1, we have that a sufficient condition for $\Delta(n, \gamma) > 0$ is

$$\gamma \geq \left[\frac{1 + mn}{(m+1)n}\right]^{1/(n-1)}. \tag{23}$$

We next derive the following sufficient condition, which is independent of n, for the optimal TCP segment size to be larger than the MTU size:

$$\gamma \geq \frac{1 + 2m}{2(m+1)}. \tag{24}$$

To demonstrate Equation (24), we define the sequence

$$x_n = \left[\frac{1 + mn}{(m+1)n}\right]^{1/(n-1)}.$$

Then x_n is the unique root, in the range (0, 1), of the monotonically increasing function in x, $(m+1)nx^{n-1} - nm - 1$. This latter function equals $\Delta(n, x)$ given by Equation (5) if $a = 0$, $b = m$, and $c = 1$. We follow exactly the same steps as those in the proof of Lemma 2, with x_n replacing γ_n, to prove that x_n is strictly monotonically increasing. Hence $x_n \geq x_2$ for all $n > 1$. Equation (24) then follows from Equation (23). This establishes our first goal stated in the introduction of this section.

We next derive a lower bound for the optimal TCP segment size. Denoting by n_0 the solution of $\zeta(n) = 0$ in Equation (16), and using equalities (17) and (18), we have, by a straightforward manipulation of terms,

$$\frac{1+mr}{1+r}(\log\gamma)n_0^2 + (\log\gamma)n_0 + 1 = \frac{-\gamma^{n_0}}{1+r}. \quad (25)$$

Since $m \geq 1$, we have $(1 + mr)/(1 + r) \leq m$. This and the facts that $\log\gamma < 0$ and $\gamma > 0$ lead to

$$m(\log\gamma)n_0^2 + (\log\gamma)n_0 + 1 \leq \frac{1+mr}{1+r}(\log\gamma)n_0^2 + (\log\gamma)n_0 + 1 = \frac{-\gamma^{n_0}}{1+r} < 0. \quad (26)$$

The feasible solution set of $m(\log\gamma)n_0^2 + (\log\gamma)n_0 + 1 < 0$ is $n_0 > \nu$, where

$$\nu = \frac{-\log\gamma - \sqrt{(\log\gamma)^2 - 4m\log\gamma}}{2m\log\gamma}. \quad (27)$$

We remark at this point that n_0, the solution of Equation (25), is a real number and not necessarily an integer. The integer solution that minimizes the processing cost $\mathcal{C}(n)$ is either floor(n_0) or ceil(n_0). Since it is a difficult task to determine in every case the exact integer that minimizes $\mathcal{C}(n)$, we use the approximation round(n_0) for the integer solution, where round(x) = integer part of $(x + 1/2)$. From $n_0 > \nu$, it follows that round$(n_0) \geq$ round(ν), and we may thus use round (ν) as an approximate lower bound for the optimal TCP segment size. The sufficient condition (24) enables us to check whether the optimal TCP segment size is greater than the MTU size, independently of the host-processing-cost parameters. When condition (24) is satisfied, the lower bound round (ν) is easy to calculate from Equation (27) and, again, does not require knowledge of the host-processing-cost parameters. In many cases, $m = 1$; thus Equation (24) becomes $\gamma \geq 0.75$, which is probably satisfied by many networks. For $m = 1$, approximate values for the optimal TCP segment size are given in **Table 1** as a function of γ. (In the table, $[a, b)$ is the notation for $a < \gamma \leq b$.) We also graph the optimal TCP segment size as a function of γ in **Figure 2**.

It is worth mentioning at this point that as ν increases, the average throughput (the number of bytes successfully received per unit time) of the file transfer decreases. The relative decrease in throughput [with respect to the case when the TCP segment size equals the MTU size ($\nu = 1$)] is $1 - \gamma^{n-1}$ (this does not take into account other factors governing the throughput, such as the sender's ability to recover from the loss of a packet without having to resort to a timeout). It is therefore important to consider the effects of the number of packets per TCP segment on the average throughput of the file transfer.

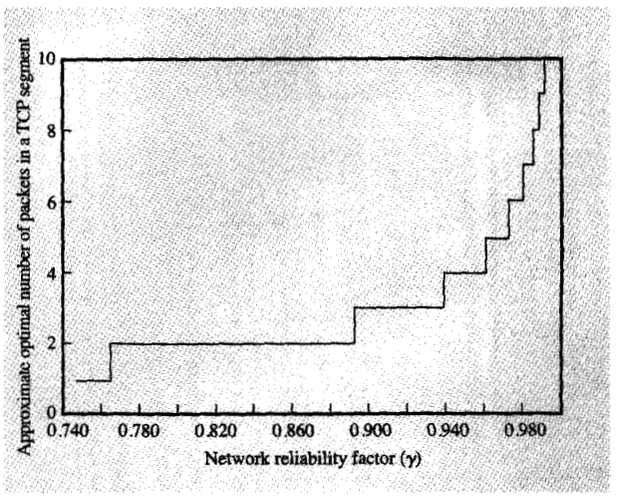

Figure 2

Approximate optimal TCP segment size as a function of the network reliability factor.

Table 1 Optimal TCP segment size (number of network packets) for various ranges of network reliability factor ($m = 1$).

Range of network reliability factors γ	*Approximate optimal TCP segment size round* (ν)
[0.7500, 0.7659)	1
[0.7659, 0.8920)	2
[0.8920, 0.9385)	3
[0.9385, 0.9604)	4
[0.9604, 0.9724)	5
[0.9724, 0.9797)	6
[0.9797, 0.9844)	7
[0.9844, 0.9877)	8
[0.9877, 0.9900)	9
[0.9900, 0.9918)	10

We now show that the cost function $\mathcal{C}(n)$ is concave upward in n. A necessary and sufficient condition for this to be the case is that the second derivative of $\mathcal{C}(n)$ be non-negative for all values of $n \geq 1$. If we denote this second derivative by $\mathcal{C}''(n)$ and note that Equation (16) is the first derivative of $-\mathcal{C}(n)$ (ignoring a positive constant factor), we have

$$\mathcal{C}''(n) = \frac{2c - 2a\gamma^n + b(\log\gamma)^2 n^3 + c(\log\gamma)^2 n^2 + 2c(\log\gamma)n}{n^3\gamma^n}. \quad (28)$$

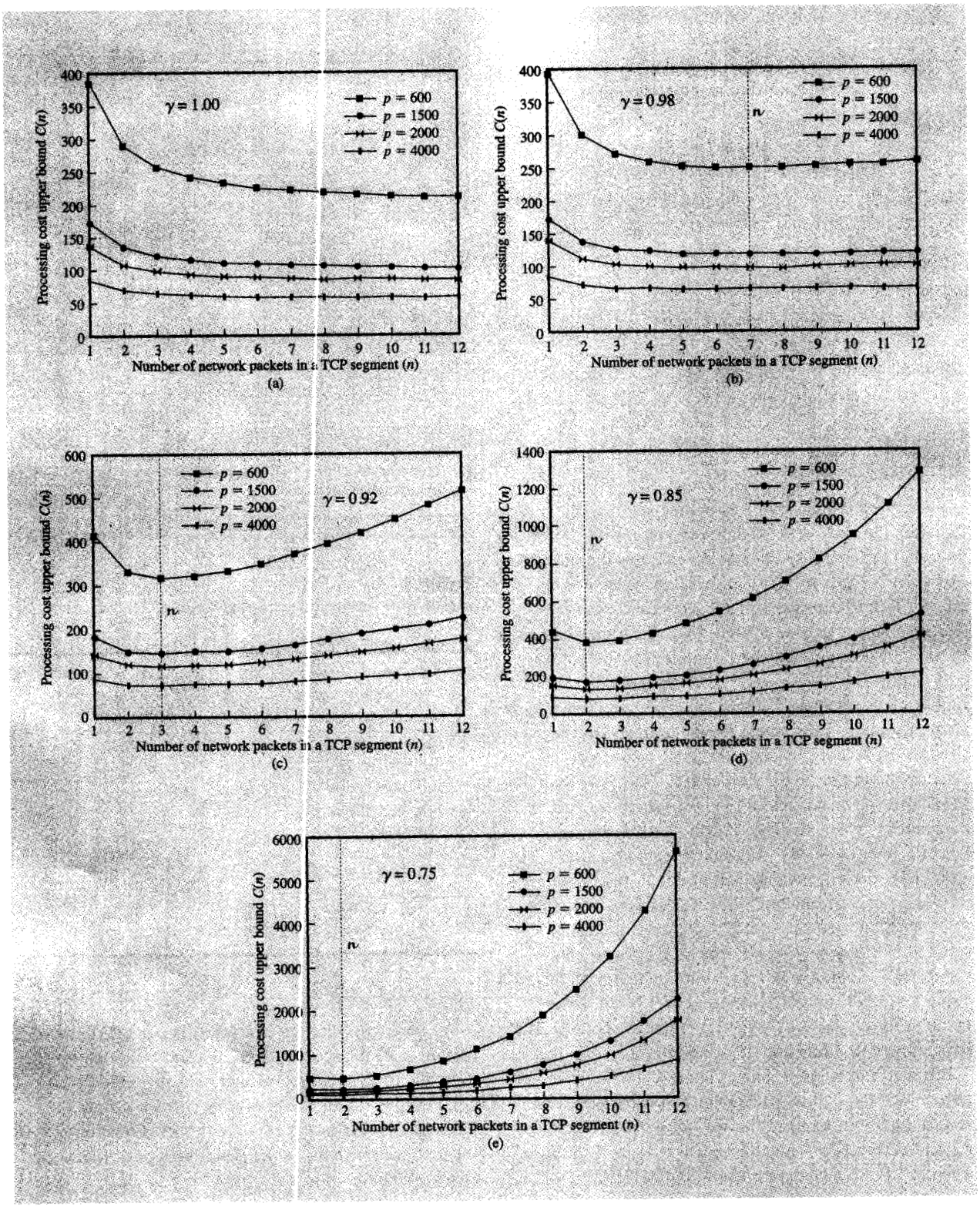

Figure 3

Processing cost bound as a function of number of packets per TCP segment, for various packet sizes p (bytes). This is for the example presented in the text, with network reliability factor γ = (a) 1.00, (b) 0.98, (c) 0.92, (d) 0.85, and (e) 0.75.

We show that the numerator of Equation (28) is non-negative. Since $m \geq 1$, we have from Equations (17) and (18) that $b \geq c > a$. Also, because $\gamma \leq 1$, it follows that $2c - 2a\gamma^n > 0$. It thus suffices to show that $b(\log \gamma)n^2 + c(\log \gamma)n + 2c$ is non-positive for $n \geq 1$, since $(\log \gamma)n < 0$. The two roots of this quadratic in n are $[c(\log \gamma) \pm \sqrt{A}]/(-2b \log \gamma)$, where $A \equiv c^2(\log \gamma)^2 - 8bc(\log \gamma)$. The root for $-\sqrt{A}$ is clearly negative (since $\log \gamma < 0$), and if we show that the other root is negative, that will demonstrate our claim (because the quadratic has the same sign as $b(\log \gamma) < 0$ outside the roots). But since $\log \gamma < 0$, it follows that $-bc \log \gamma > 0$, and $A > c^2(\log \gamma)^2$, which implies that $\sqrt{A} \geq |c \log \gamma| = -c \log \gamma$. This finishes the proof.

We now present some examples of optimal TCP segment size.

Example

Consider a file server with the following processing costs and parameters for outbound data transfer. The goal is to show to the reader how to calculate the optimal TCP segment size and to point out the host-processing savings, in CPU cycles, by using the larger TCP segment size. Assume the following values:

- a = 500 instructions per segment TCP processing cost.
- b = 3000 instructions per packet IP and network-layer processing cost.
- c = 3000 instructions per TCP acknowledgment in processing cost.
- d = 500 instructions per packet IP processing cost.
- C_m = 0.15 instructions per byte data-moving cost.
- C_k = 0.628 instructions per byte TCP checksum cost.
- C_b = 5000 instructions, fixed processing cost to transfer a data block from the FTP layer to the TCP layer.
- S_{IPb} = 65536 bytes in the network-layer buffer.
- S_b = 32768 bytes in each data block transferred between FTP and TCP.

In **Figure 3**, we graph the upper bound on cost function, $\mathcal{C}(n)$, for various values of γ, the probability that a network packet is received successfully, and for a selected set of network packet sizes p. Note that in Figure 3(a), for $\gamma = 1$, the optimum TCP segment size is infinite, since the penalty for retransmission is never incurred. In all cases presented, the greatest improvement in processing cost (measured in CPU cycles) comes from increasing the TCP segment size from p to $2p$. This is because the cost function $\mathcal{C}(n)$ is concave upward in n, so when the function is decreasing, the greatest cost savings come from increasing the segment size from p to $2p$. Intuitively, we reason that $\mathcal{C}(n)$ is concave upward because as the segment size increases, the cost of retransmission of network packets increases faster than the cost of processing TCP segments and acknowledgments decreases. Hence, the percentage of cost savings (in CPU cycles consumed) decreases as the TCP segment size increases. The cost reduction due to choosing a non-optimal TCP segment size (at least $2p$ in size) is not overly sensitive to the network reliability factor γ, and it is not a major problem if the sender does not have accurate knowledge of γ.

Table 2 Value of n for various network reliability factors.

γ	n	$round(n)$
0.98	6.553	7
0.92	2.999	3
0.85	2.030	2
0.75	1.430	1

Using the parameters for this example in Equation (6), we calculate γ_0 ($\equiv \gamma_2$) to be 0.73. All of the values of γ used in the example are greater than 0.73, which guarantees that the optimal TCP segment size is greater than the MTU size. Note also that the sufficient condition of Equation (24), $\gamma > 0.75$, is satisfied in each case. The values of n given by Equation (27) are shown in **Table 2**. Note that the absolute value of the difference of $round(n)$ and the optimal segment size (in Figure 3) is less than or equal to 1.

Conclusion

In this paper, we studied the problem of selecting the optimal TCP segment size in order to minimize the TCP/IP processing cost for file transfer from hosts to clients. In the literature, the TCP segment size is usually designed to maximize file-transfer throughput. However, some file servers cannot afford to consume many CPU cycles on network-communications processing. There is thus a need to minimize TCP/IP processing for such file servers. We formulated the sender processing cost as a function of the TCP segment size and of the acknowledgments processing. The parameters of this cost function are costs for the TCP, IP, and network-layer processing, checksum calculation, and data moves, for processing both inbound (acknowledgments) and outbound data. Other parameters are the TCP window size, the maximum network packet size, and the network reliability factor. The variable in this cost function is the TCP segment size.

We proved the existence of a network reliability factor γ_0 with the following property. For any network reliability factor $\gamma \leq \gamma_0$, the optimal TCP segment size equals the maximum network packet size (less the TCP and IP headers). In this case, the processing-cost function increases with the segment size. For $\gamma > \gamma_0$, the cost

function has a global minimum at a segment size greater than the MTU size. Also, the optimal TCP segment size increases with γ because, as the TCP network reliability factor increases, the average number of retransmitted TCP segments decreases. We also derived a sufficient condition on the network reliability factor for the optimal TCP segment size to be greater than the MTU size. In this case, we obtained a lower bound for the optimal TCP segment size as a simple function of the network reliability factor.

It is worth mentioning that the amount of TCP/IP processing at the receiving node also decreases as a result of larger TCP segment sizes. This is because, just as in the sender case, the network-layer processing cost decreases as a result of a larger batch of network packet arrivals, and the TCP cost decreases as a result of processing fewer incoming segments and outgoing acknowledgments. The receiver TCP/IP, however, must efficiently implement the reassembly of IP datagrams into TCP segments. Otherwise, the CPU savings in cycles due to larger segments will be lost because of the expensive processing cost of TCP-segment reassembly.

*Trademark or registered trademark of International Business Machines Corporation.

References

1. W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley Publishing Co., Reading, MA, 1994.
2. K. Bharat-Kumar, "Optimum End-to-End Flow Control in Networks," *Proceedings of the International Conference on Communications*, Seattle, WA, June 1980, pp. 23.3.1–23.3.6.
3. M. Schwartz, "Routing and Flow Control in Data Networks," *Research Report RC-8353*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, July 1980.
4. R. M. Bournas, "Effects of TCP and IP Tuning Parameters on Network Throughput," *Technical Report TR-29.1830*, IBM Networking Software Division, Research Triangle Park, NC, 1994.
5. R. M. Bournas, "Bounds for Optimal Flow Control Window Size Design with Application to High-Speed Networks," *J. Franklin Institute* **332B,** No. 1, 77–89 (1995).

Received February 2, 1996; accepted for publication September 16, 1996

Redha M. Bournas *IBM Software Solutions Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709 (redha_bournas@vnet.ibm.com).*

Dr. Bournas received the B.S. degree in computer science and mathematics with honors in 1980, the M.S. degree in electrical engineering from the University of Pittsburgh in 1981, and the Ph.D. degree in electrical engineering systems from the University of Michigan in 1990. He contributed to the design and development of the IBM 4381 and 9370 processors, and has been working on TCP/IP performance design, development, analysis, and modeling since he joined Networking Systems in 1991. His research interests include multidimensional queueing systems, performance modeling, and analysis of communication protocols and networks. Dr. Bournas was awarded a three-year IBM graduate fellowship from 1987 to 1990. He received the graduate distinguished achievement award in electrical engineering systems at the University of Michigan in April 1990.