Trade-Off between Cost and Goodput in Wireless: Replacing Transmitters with Coding

MinJi Kim[1], Thierry Klein[2], Emina Soljanin[2], João Barros[3], and Muriel Médard[1]

[1] Research Laboratory of Electronics (RLE), MIT
Cambridge, MA USA 02139
{minjikim,medard}@mit.edu
[2] Alcatel-Lucent Bell Laboratories
Murray Hill, NJ USA 07974
thierry.klein@alcatel-lucent.com, emina@research.bell-labs.com
[3] Department of Electrical and Computer Engineering, University of Porto
Porto, Portugal
jbarros@fe.up.pt

Abstract. We study the cost of improving the *goodput*, or the useful data rate, to user in a wireless network. We measure the cost in terms of number of base stations, which is highly correlated to the energy cost as well as capital and operational costs of a network provider. We show that increasing the available bandwidth, or throughput, may not necessarily lead to increase in goodput, particularly in lossy wireless networks in which TCP does not perform well. As a result, much of the resources dedicated to the user may not translate to high goodput, resulting in an inefficient use of the network resources. We show that using protocols such as TCP/NC, which are more resilient to erasures and failures in the network, may lead to a goodput commensurate with the throughput dedicated to each user. By increasing goodput, users' transactions are completed faster; thus, the resources dedicated to these users can be released to serve other requests or transactions. Consequently, we show that translating efficiently throughput to goodput may bring forth better connection to users while reducing the cost for the network providers.

1 Introduction

Mobile data traffic has been growing at an alarming rate with some estimating that it will increase more than 25-folds in the next five years [1]. In order to meet such growth, there has been an increasing effort to install and upgrade the current networks. As shown in Figure 1, mobile service providers often install more infrastructure (e.g. more base stations) in areas which already have full coverage. The new infrastructure is to provide more bandwidth, which would lead to higher quality of experience to users. However, this increase in bandwidth comes at a significant energy cost as each base station has been shown to use 2-3 kilowatts (kW) [2]. The sustainability and the feasibility of such rapid development have been brought to question as several trends indicate that the technology efficiency improvements may not be able to keep pace with the traffic growth [2].

D. Pesch et al. (Eds.): MONAMI 2013, LNICST 125, pp. 1–14, 2013.

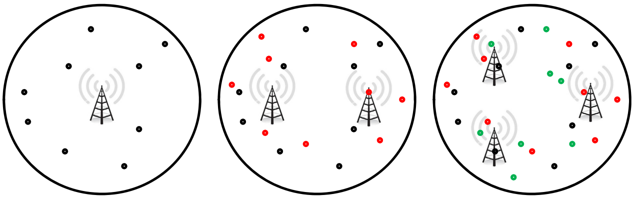


Fig. 1. As number of users in a given area grows, a service provider may add additional base stations not for coverage but for bandwidth. As red users join the network, a second base station may be necessary; as green users join the network, a third base station may become necessary in order to maintain a certain level of quality of service.

We show that maintaining or even improving users' quality of experience may be achieved without installing more base stations. In some cases, we show that the users' quality of experience may be improved while reducing the number of base stations. We measure users' quality of experience using the throughput perceived by the user or the application, i.e. *goodput*. We make a clear distinction between the terms goodput and *throughput*, where goodput is the number of *useful* bits over unit time received by the user and throughput is the number of bits transmitted by the base station per unit time. In essence, throughput is indicative of the bandwidth/resources provisioned by the service providers; while goodput is indicative of the user's quality of experience. For example, the base station, after accounting for the FEC overhead, may be transmitting bits at 10 megabits per second (Mbps), i.e. throughput is 10 Mbps. However, the user may only receive useful information at 5 Mbps, i.e. goodput is 5 Mbps.

There can be a significant disparity between throughput and goodput, particularly in lossy networks using TCP. TCP often mistakes random erasures as congestion [3, 4]. For example, 1-3% packet loss rate is sufficient to harm TCP's performance [3–6]. This performance degradation can lead to inefficient use of network resources and incur substantially higher cost to maintain the same goodput. There has been extensive research to combat these harmful effects of erasures and failures; however, TCP even with modifications does not achieve significant improvement. References [4, 7] give an overview of various TCP versions over wireless links.

This disparity between throughput and goodput can be reduced by using a transport protocol that is more resilient to losses. One method is to use multiple base stations simultaneously (using multiple TCP connections [8] or multipath TCP [9]). However, the management of the multiple streams or paths may be difficult, especially in lossy networks. Furthermore, each path or TCP stream still suffer from performance degradation in lossy environments [8, 9].

We propose TCP/NC [5, 10] as an alternative transport protocol. We provide an overview of TCP/NC in Section 1.1 TCP/NC may not be the only viable solution, and other transport protocols that can combat erasures may be used.

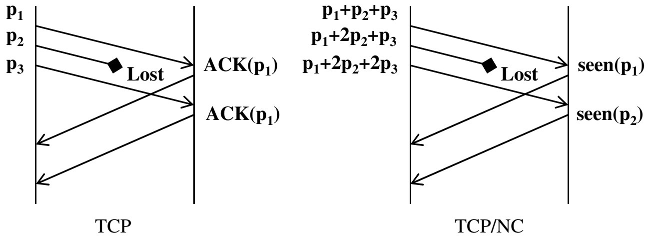


Fig. 2. Example of TCP and TCP/NC. In the case of TCP, the TCP sender receives duplicate ACKs for packet $\mathbf{p_1}$, which may wrongly indicate congestion. However, for TCP/NC, the TCP sender receives ACKs for packets $\mathbf{p_1}$ and $\mathbf{p_2}$; thus, the TCP sender perceives a longer RTT but does not mistake the loss to be congestion.

We use TCP/NC for its effectiveness and simplicity. TCP/NC allows a better use of the base stations installed, and can improve the goodput without any additional base stations. Improving the goodput with the same or a fewer number of base stations implies reduction in energy cost, operational expenses, capital expenses, and maintenance cost for the network provider. The results in this paper can also be understood as being able to serve more users or traffic growth with the same number of base stations. This may lead to significant cost savings, and may be of interest for further investigation.

We note that, to prevent TCP's performance degradation, cellular systems such as LTE have implemented various mechanisms (e.g. HARQ [11] and lower layer retransmissions) with stringent bit-error rates to reduce packet loss rate. Using a transport protocol that can combat erasures, e.g. TCP/NC, may relieve the lower layers from such stringent performance requirements. It would be interesting to study the effect of using erasure-resilient transport protocols on the lower layers' performance requirements, and the cross-layer optimization to improve the throughput and the energy cost of cellular systems.

1.1 Overview of TCP/NC

Reference [10] introduces a new *network coding* layer between the TCP and IP in the protocol stack. The network coding layer intercepts and modifies TCP's acknowledgment (ACK) scheme such that random erasures do not affect the transport layer's performance. To do so, the *encoder*, the network coding unit under the sender TCP, transmits R random linear combinations of the buffered packets for every transmitted packet from TCP sender. The parameter R is the *redundancy factor*. Redundancy factor helps TCP/NC to recover from random losses; however, it cannot mask correlated losses, which are usually due to congestion. The *decoder*, the network coding unit under the receiver TCP, acknowledges *degrees of freedom* instead of individual packets, as shown in Figure 2. Once enough degrees of freedoms are received at the decoder, the decoder solves the set of linear equations to decode the original data transmitted by the TCP sender, and delivers the data to the TCP receiver.

We briefly note the overhead associated with network coding. The main overhead associated with network coding can be considered in two parts: 1) the coding vector (or coefficients) that has to be shared between the sender and the receiver; 2) the encoding/decoding complexity. For receiver to decode a network coded packet, the receiver needs to know the coding coefficients used to generate the linear combination of the original data packets. The first overhead can be minimized with the sender including a seed for a pseudo-random number generator which allows the receiver to generate the coding coefficients in each coded packet. The second overhead associated with network coding is the encoding and decoding complexity, and the delay associated with the coding operations. Note that to affect TCP's performance, the decoding/encoding operations must take substantial amount of time to affect the round-trip time estimate of the TCP sender and receiver. However, we note that the delay caused the coding operations is negligible compared to the network round-trip time. For example, the network round-trip time is often in milliseconds (if not in hundreds of milliseconds), while encoding/decoding operations involve a matrix multiplication/inversion in a field (e.g. $\mathbf{F}_{256}$), which can be performed in a few microseconds.

In [10], the authors present two versions of TCP/NC – one that adheres to the end-to-end philosophy of TCP, in which coding operations are only performed at the source and destination; another that takes advantage of network coding even further by allowing any subset of intermediate nodes to re-encode. Note that re-encoding at the intermediate nodes is an optional feature, and is not required for TCP/NC to work. Here, we focus on TCP/NC with end-to-end network coding. However, a similar analysis applies to TCP/NC with re-encoding.

2 Model

Consider a network with n users. We assume that these n users are in an area such that a single base station can cover them as shown in Figure 1. If the users are far apart enough that a single base station cannot cover the area, then more base stations are necessary; however, we do not consider the problem of coverage.

The network provider's goal is to provide a *fair* service to any user that wishes to start a transaction. Here, by fair, we mean that *every user receives the same average throughput*, denoted as r_t Mbps. It would be interesting to extend and analyze TCP/NC or other alternative protocols under different notions of fairness as well as in networks with priority-based scheduling. However, in this paper, we use a simple definition of *fairness* in which all users receive the same throughput.

The network provider wishes to have enough network resources, measured in number of base stations, so that any user that wishes to start a transaction is able to join the network immediately and achieve an average throughput of r_t Mbps. We denote r_g to be the goodput experienced by the user. Note that $r_g \leq r_t$.

We denote n_{bs} to be the number of base stations needed to meet the network provider's goal. We assume that every base station can support at most $R_{\max}$ Mbps (in throughput) and at most $N_{\max}$ active users simultaneously. In this paper, we assume that $R_{\max} = 300$ Mbps and $N_{\max} = 200$.

A user is *active* if the user is currently downloading a file; *idle* otherwise. A user decides to initiate a transaction with probability p at each time slot. Once a user decides to initiate a transaction, a file size of f bits is chosen according to a probability distribution P_f. We denote μ_f to be the expected file size, and the expected duration of the transaction to be $\Delta = \mu_f / r_g$ seconds. If the user is already active, then the new transaction is added to the user's queue. If the user has initiated k transactions, the model of adding the jobs into the user's queue is equivalent to splitting the goodput r_g to k transactions (each transaction achieves a rate of r_g/k Mbps).

We denote p_p to be the probability of packet loss in the network, and RTT to be the round-trip time. In a wireless, p_p and RTT may vary widely. For example, wireless connection over WiFi may have RTT ranging from tens of milliseconds to hundreds of milliseconds with loss rates typically ranging from 0-10%. In a more managed network (such as cellular networks), RTT are typically higher than that of a WiFi network but lower in loss rates.

3 Analysis of the Number of Base Stations

We analyze the number of base stations n_{bs} needed to support n users given throughput r_t and goodput r_g. We first analyze $P(\Delta, p)$, the probability that a user is active at any given point in time. Given $P(\Delta, p)$, we compute the expected number of active users at any given point in time and n_{bs} needed to support these active users.

Consider a user u at time t. There are many scenarios in which u would be active at t. User u may initiate a transaction at precisely time t with probability p. Otherwise, u is still in the middle of a transaction initiated previously.

To derive $P(\Delta, p)$, we use the Little's Law. For a stable system, the Little's Law states that the average number of jobs (or transactions in our case) in the user's queue is equal to the product of the arrival rate p and the average transaction time Δ. When $\Delta p \geq 1$, we expect the user's queue to have on average at least one transaction in the long run. This implies that the user is expected to be active at all times. When $\Delta p < 1$, we can interpret the result from Little's Law to represent the probability that a user is active. For example, if $\Delta p = 0.3$, the user's queue is expected to have 0.3 transactions at any given point in time. This can be understood as the user being active for 0.3 fraction of the time. Note that when the system is unstable, the long term average number of uncompleted jobs in the user's queue may grow unboundedly. In an unstable system, we assume that in the long term, a user is active with probability equal to one.

Therefore, we can state the following result for $P(\Delta, p)$.

$$P(\Delta, p) = \min\{1, \Delta p\} = \min\left\{1, \frac{\mu_f}{r_g} \cdot p\right\}. \tag{1}$$

Given $P(\Delta, p)$, the expected number of active users is $nP(\Delta, p)$. We can now characterize the expected number of base stations needed as

$$n_{bs} = nP(\Delta, p) \cdot \max\left\{\frac{r_t}{R_{\max}}, \frac{1}{N_{\max}}\right\}. \tag{2}$$

In Equation (2), $\max\{\frac{r_t}{R_{\max}}, \frac{1}{N_{\max}}\}$ represents the amount of base stations' resources (the maximum load $R_{\max}$ or the amount of activity $N_{\max}$) each active user consumes. The value of n_{bs} from Equation (2) may be fractional, indicating that actually $\lceil n_{bs} \rceil$ base stations are needed.

Note the effect of r_t and r_g. As shown in Equation (2), increasing r_t incurs higher cost while increasing r_g reduces the cost. Therefore, when a network provider dedicates resources to increase r_t, the goal of the network provider is to increase r_g proportional to r_t.

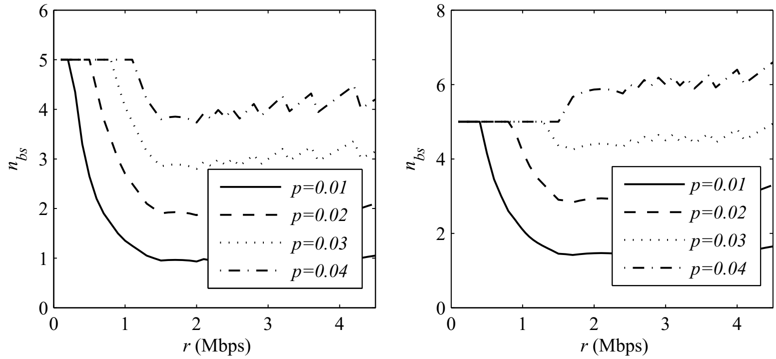
4 Best Case Scenario

In an ideal scenario, the user should see a goodput $r_g = r_t$. In this section, we analyze this best case scenario with $r = r_t = r_g$. Once we understand the optimal scenario, we then consider the behavior of TCP and TCP/NC in Section 5 where generally $r_g \leq r_t$.

4.1 Analytical Results

In Figures 3a and 3b, we plot Equation (2) with $\mu_f = 3.2$ MB and $\mu_f = 5.08$ MB for varying values of p. As r increases, it does not necessarily lead to increase in n_{bs}. Higher r results in users finishing their transactions faster, which in turn allows the resources dedicated to these users to be released to serve other requests or transactions. As a result, counter-intuitively, we may be able to maintain a higher r with *the same or a fewer* number of base stations than we would have needed for a lower r. For example, in Figure 3a, when $r < 1$ Mbps, the rate of new requests exceeds the rate at which the requests are handled; resulting in an unstable system. As a result, most users are active all the time, and the system needs $\frac{n}{N_{\max}} = \frac{1000}{200} = 5$ base stations.

There are many cases where n_{bs} is relatively constant regardless of r. For instance, consider $p = 0.03$ in Figure 3b. The value of n_{bs} is approximately 4-5 throughout. However, there is a significant difference in the way the resources are used. When r is low, all users have slow connections; therefore, the base stations are fully occupied not in throughput but in the number of active users. On the other hand, when r is high, the base stations are being used at full-capacity in terms of throughput. As a result, although the system requires the same number



(a) $\mu_f = 3.2$ MB (b) $\mu_f = 5.08$ MB

Fig. 3. The values of n_{bs} from Equation (2) with $n = 1000$ and varying p and r

of base stations, users experience better quality of service and users' requests are completed quickly.

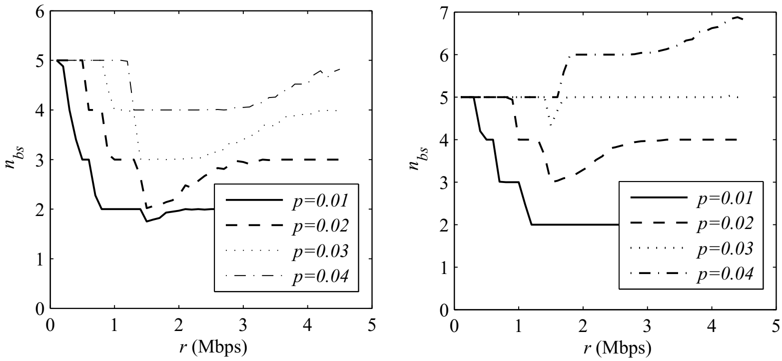
When p and r are high enough, it is necessary to increase n_{bs}. As demand exceeds the network capacity, it becomes necessary to add more infrastructure to meet the growth in demand. For example, consider $p = 0.04$ in Figure 3b. In this case, as r increases n_{bs} increases.

4.2 Simulation Results

We present MATLAB simulation results to verify our analysis results in Section 4.1. We assume that at every 0.1 second, a user may start a new transaction with probability $\frac{p}{10}$. This was done to give a finer granularity in the simulations; the results from this setup is equivalent to having users start a new transaction with probability p every second. We assume that there are $n = 1000$ users. For each iteration, we simulate the network for 1000 seconds. Each plot is averaged over 100 iterations.

Once a user decides to start a transaction, a file size is chosen randomly in the following manner. We assume there are four types of files: $f_{doc} =$ 8KB (a document), $f_{image} =$ 1MB (an image), $f_{mp3} =$ 3 MB (a mp3 file), $f_{video} =$ 20 MB (a small video), and are chosen with probability p_{doc}, p_{image}, p_{mp3}, and p_{video}, respectively. In Figure 4a, we set $[p_{doc}, p_{image}, p_{mp3}, p_{video}] = [0.3, 0.3, 0.3, 0.1]$. This results in $\mu_f = 3.2$ MB as in Figure 3a. In Figure 4b, we set $[p_{doc}, p_{image}, p_{mp3}, p_{video}] = [0.26, 0.27, 0.27, 0.2]$, which gives $\mu_f = 5.08$ MB as in Figure 3b.

The simulation results show close concordance to our analysis. Note that the values in Figures 4a and 4b are slightly greater than that of Figures 3a and 3b. This is because, in the simulation, we round-up any fractional n_{bs}'s since the number of base stations needs to be integral.



(a) $\mu_f = 3.2$ MB (b) $\mu_f = 5.08$ MB

Fig. 4. Average value of n_{bs} over 100 iterations with $n = 1000$ and varying p and r

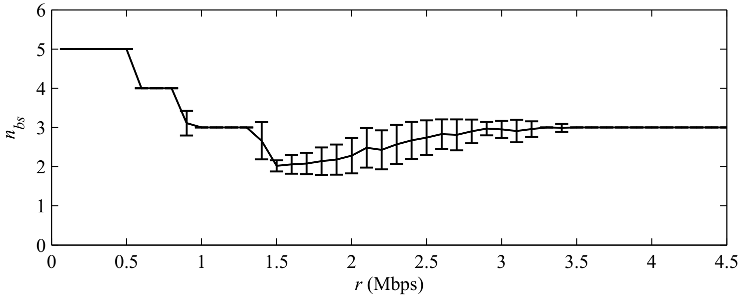


Fig. 5. Average and standard deviation of n_{bs} over 100 iterations with $\mu_f = 3.2$ MB and $p = 0.02$

In Figure 5, we show the average value of n_{bs} and its standard deviation δ for $\mu_f = 3.2$ MB and $p = 0.02$. A plot similar to that of Figure 5 can be obtained for different values of μ_f and p; however, we omit them for want of space. When $r < 0.5$ Mbps, $n_{bs} = 5$ and $\delta = 0$. This is because all users' connections are slow and all users are active; thus, $\frac{n}{N_{\max}} = 5$ base stations are always needed (resulting in $\delta = 0$).

Understanding the effect of the standard deviation δ is important. For example, when $r = 2$ Mbps, we have $n_{bs} = 2.28$ and $\delta = 0.2036$. Therefore, when $r = 2$ Mbps, we needed two base stations in most iterations, only occasionally three. This indicates that the third base station is needed to serve the occasional bursts of activities. Thus, to ensure a certain level of throughput to users, it is important to over-provision, e.g. install $\geq n_{bs} + 2\delta$ base stations to overcome the stochastic variations in activities. However, as r increases further (> 3 Mbps),

δ approaches zero. When $r > 3$ Mbps, bursty user activities do not lead to variations in n_{bs}; all user requests are completed quickly enough that bursty activities have negligible effect on n_{bs}. Therefore, when we consider the stochastic nature of user activities, it may be even more desirable to have large r.

5 Analysis for TCP/NC and TCP

We now study the effect of TCP and TCP/NC's behavior. We use the model and analysis from [5] to model the relationship between r_g and p_p for TCP and TCP/NC. We denote r_{g-nc} to be the goodput when using TCP/NC, and r_{g-tcp} to be that for TCP. We set the maximum congestion window, $W_{\max}$, of TCP and TCP/NC to be 50 packets (with each packet being 1000 bytes long), and their initial window size to be 1. We consider $RTT = 100$ ms and varying p_p from 0% to 5%. We note that, given r_t and p_p, $r_g \leq r_t(1-p_p)$ regardless of the protocol used.

In [5, 10], TCP/NC has been shown to be robust against erasures; thus, allowing it to maintain a high throughput despite random losses. For example, if the network allows for 2 Mbps per user and there is 10% loss rate, then the user should see approximately $2 \cdot (1-0.1) = 1.8$ Mbps. Reference [5] has shown, both analytically and with simulations, that TCP/NC indeed is able to achieve goodput close to 1.8 Mbps in such a scenario while TCP fails to do so.

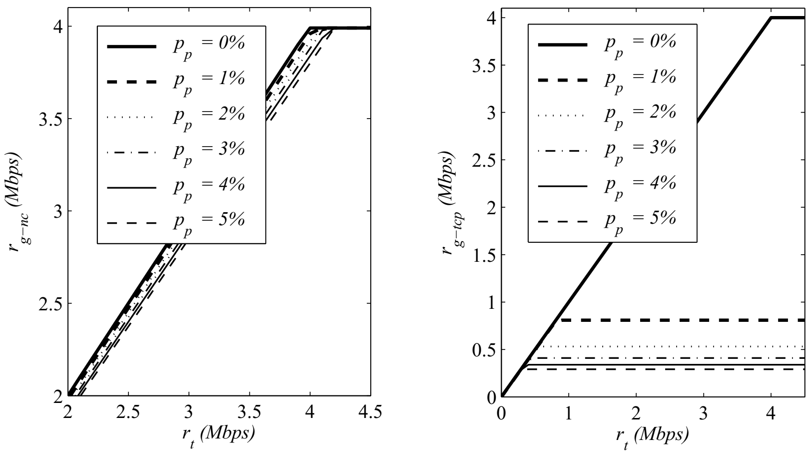
5.1 Behavior of r_{g-nc} with Varying p_p

Equation (20) from [5] provides the goodput behavior of TCP/NC, which we provide below in Equation (3).

$$r_{g-nc} = \frac{1}{tSRTT}\left(tW_{\max} - \frac{(W_{\max}-1)^2 + (W_{\max}-1)}{2}\right), \tag{3}$$

where $SRTT$ is the effective RTT observed by TCP/NC and increases with p_p and t represents the duration of the connection (in number of RTTs). Equation (3) shows the effect of network coding. The goodput of TCP/NC decreases with p_p; however, the effect is indirect. As p_p increases, the perceived RTT increases, which leads to TCP/NC reducing its rate.

Combining Equation (3) and $r_{g-nc} \leq r_t(1-p_p)$, we obtain the values of r_{g-nc} for various r_t, RTT, and p_p. In Figure 6a, the values of r_{g-nc} plateaus once r_t exceeds some value. This is caused by $W_{\max}$. Given $W_{\max}$ and RTT, TCP/NC and TCP both have a maximal goodput it can achieve. In the case with RTT = 100 ms, the maximal goodput is approximately 4 Mbps. Note that regardless of p_p, all TCP/NC flows achieve the maximal achievable rate. This shows that TCP/NC can overcome effectively the erasures or errors in the network, and provide a goodput that closely matches the throughput r_t.



(a) r_{g-nc} (b) r_{g-tcp}

Fig. 6. The value of r_{g-nc} and r_{g-tcp} against r_t for varying values of p_p. We set $RTT = 100$ ms.

5.2 Behavior of r_{g-tcp} with Varying p_p

Equation (16) from [5] provides the goodput behavior of TCP, which we provide below in Equation (4).

$$r_{g-tcp} \approx \min\left(\frac{W_{\max}}{RTT}, \frac{1-p_p}{p_p}\frac{1}{RTT\left(\frac{5}{3}+\sqrt{\frac{2}{3}\frac{1-p_p}{p_p}}\right)}\right). \tag{4}$$

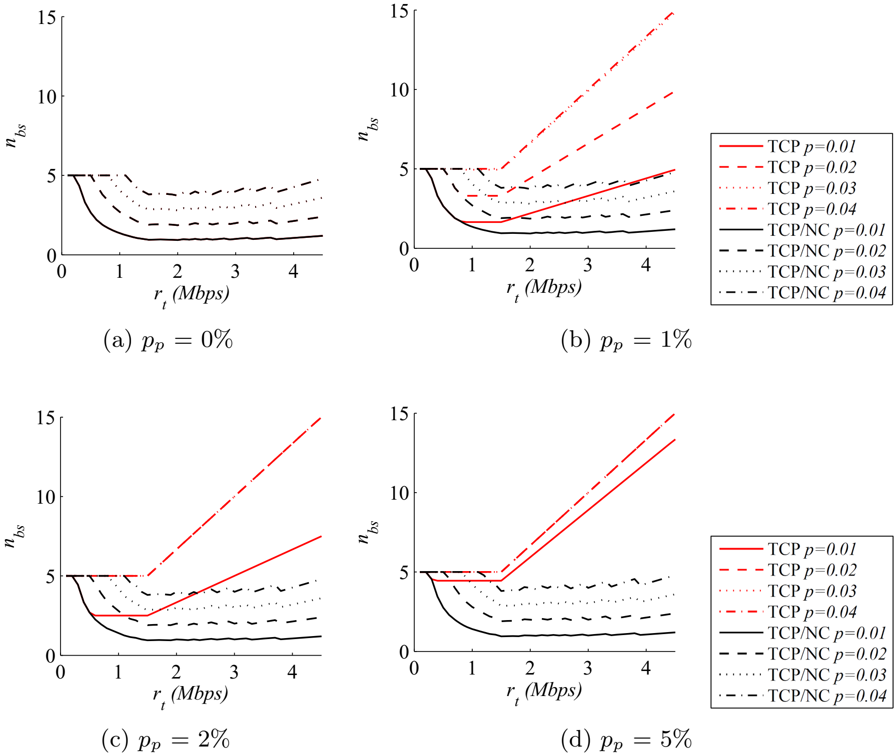
Note that unlike TCP/NC, TCP performance degrades proportionally to $\sqrt{\frac{1}{p}}$.

Combining Equation (4) and $r_{g-tcp} \leq r_t(1-p_p)$, we obtain the values of r_{g-tcp} for various r_t, RTT, and p_p as shown in Figure 6b. As in Figure 6a, the values of r_{g-tcp} are also restricted by $W_{\max}$. However, TCP achieves this maximal goodput only when $p_p = 0\%$. This is because, when there are losses in the network, TCP is unable to recover effectively from the erasures and fails to use the bandwidth dedicated to it. For $p_p > 0\%$, r_{g-tcp} is not limited by $W_{\max}$ but by TCP's performance limitations in lossy wireless networks.

5.3 The Number of Base Stations for TCP/NC and TCP

We use the values of r_{g-nc} and r_{g-tcp} from Sections 5.1 and 5.2 to compare the number of base stations for TCP/NC and TCP using Equation (2). We assume that $SRTT = RTT$. In general, $SRTT$ is slightly larger than RTT.

Figures 7 and 8 show n_{bs} predicted by Equation (2) when $RTT = 100$ ms. TCP suffers performance degradation as p_p increases; thus, n_{bs} increases rapidly with p_p. Note that increasing r_t without being able to increase r_g leads to inefficient



(a) $p_p = 0\%$

(b) $p_p = 1\%$

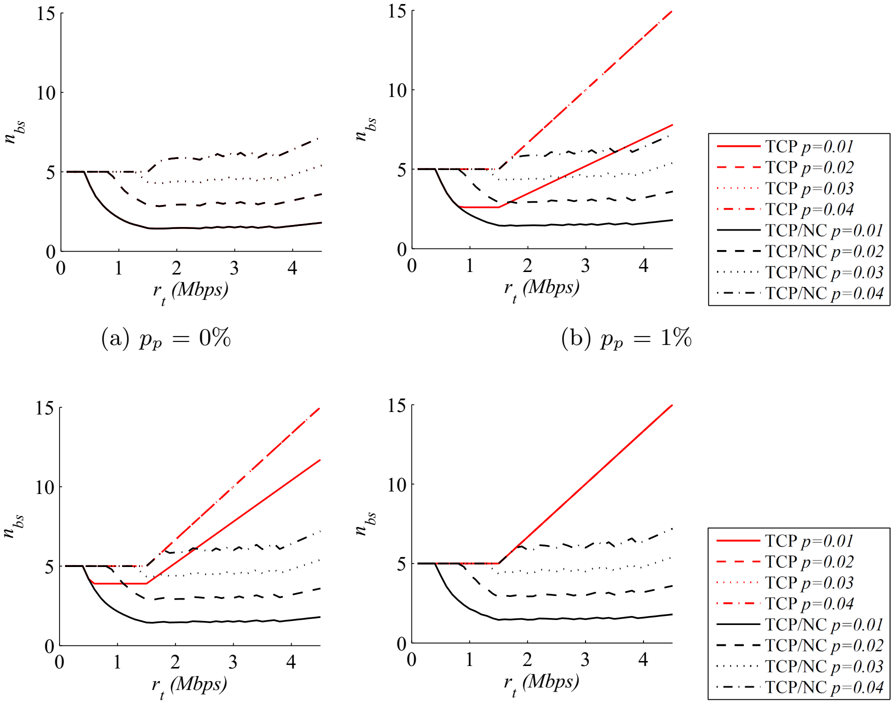
(c) $p_p = 2\%$

(d) $p_p = 5\%$

Fig. 7. The value of n_{bs} from Equation (2) for TCP and TCP/NC with varying p_p and p. Here, $RTT = 100$ ms, $W_{\max} = 50$, $n = 1000$, and $\mu_f = 3.2$ MB. In (a), $p_p = 0$ and both TCP and TCP/NC behaves the same; thus, the curves overlap. Note that this result is the same as that of Figure 3a. In (b), the value of n_{bs} with TCP for $p = 0.03$ and 0.04 coincide (upper most red curve). In (c) and (d), the values of n_{bs} with TCP for $p > 0.01$ overlap.

use of the network, and this is clearly shown by the performance of TCP as r_t increases with $p_p > 0\%$.

However, for TCP/NC, n_{bs} does not increase significantly (if any at all) when p_p increases. As discussed in Section 3, TCP/NC is able to translate better r_t into r_{g-nc} despite $p_p > 0\%$, i.e. $r_t \approx r_{g-nc}$. As a result, this leads to a significant reduction in n_{bs} for TCP/NC compared to TCP. Note that n_{bs} for TCP/NC is approximately equal to the values of n_{bs} in Section 3 regardless of the value of p_p. Since TCP/NC is resilient to losses, the behavior of r_{g-nc} does not change as dramatically against p_p as that of r_{g-tcp} does. As a result, we observe n_{bs} for TCP/NC to reflect closely the values of n_{bs} seen in Section 3, which is the best case with $r_t = r_g$.

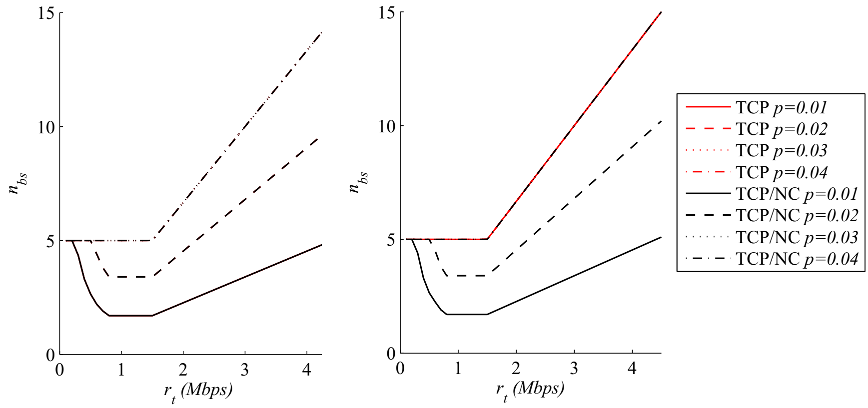


(a) $p_p = 0\%$ (b) $p_p = 1\%$

(c) $p_p = 2\%$ (d) $p_p = 3\%$

Fig. 8. The value of n_{bs} from Equation (2) for TCP and TCP/NC with varying p_p and p. Here, $RTT = 100$ ms, $W_{\max} = 50$, $n = 1000$, and $\mu_f = 5.08$ MB. In (a), the results for TCP and TCP/NC are the same. Note that this result is the same as that of Figure 3b. In (b) and (c), the value of n_{bs} with TCP for $p > 0.01$ coincide (upper red curve). In (d), the values of n_{bs} with TCP for any p all overlap. We do not show results for $p_p = 4\%$ or 5% as they are similar to that of (d).

As shown in Figure 9, we observe a similar behavior for other values of RTT as we did for $RTT = 100$ ms. The key effect of the value of RTT in the maximal achievable goodput. For example, if $W_{\max}$ is limited to 50, the maximal achievable goodput is approximately 0.8 Mbps when $RTT = 500$ ms, which is much less than the 4 Mbps achievable with $RTT = 100$ ms. As a result, for $RTT = 500$ ms, neither r_{g-nc} nor r_{g-tcp} can benefit from the increase in r_t beyond 0.8 Mbps. Despite this limitation, TCP/NC still performs better than TCP when losses occur. When demand exceeds the maximal achievable goodput, n_{bs} increases for both TCP/NC and TCP in the same manner.



(a) $p_p = 0\%$ (b) $p_p = 1\%$

Fig. 9. The value of n_{bs} from Equation (2) for TCP and TCP/NC with varying p_p and p. Here, RTT = 500 ms, $W_{\max}$ = 50, n = 1000, and μ_f = 3.2 MB. In (a), the results for TCP and TCP/NC are the same. The curves for $p = 0.03$ and $p = 0.04$ are the same both for TCP and TCP/NC. In (b), the value of n_{bs} with TCP for any p all overlap, while the TCP/NC curves are the same as in (a). We do not show results for $p_p > 1\%$ as they are similar to that of (b).

6 Conclusions

In wireless networks, the solution to higher demand is often to add more infrastructure. This is indeed necessary if all the base stations are at capacity (in terms of throughput). However, in many cases, the base stations are "at capacity" either because they are transmitting redundant data to recover from losses; or because they cannot effectively serve more than a few hundred active users. This may be costly as base stations are expensive to operate. One way to make sure that wireless networks are efficient is to ensure that, whenever base stations are added, they are added to effectively increase the goodput of the network.

We studied the number of base stations n_{bs} needed to improve the goodput r_g to the users. It may seem that higher r_g necessarily increases n_{bs}. Indeed, if there are enough demand (i.e. r_g, p, or μ_f are high enough), we eventually need to increase n_{bs}. However, we show that this relationship is not necessarily true. When r_g is low, each transaction takes more time to complete and each user stays in the system longer. This degrades the user experience and delays the release of network resources dedicated to the user. This is particularly important as the number of active users each base station can support is limited to the low hundreds. We observed that, given r_t, achieving low r_g may lead to a significant increase in n_{bs} and an ineffective use of the network resources; while achieving high r_g may lead to reduction in n_{bs}.

We showed that, in lossy networks, the goodput r_g observed may not closely match the amount of resources dedicated to the user, e.g. $r_g \ll r_t$. This is due to the poor performance of TCP in lossy networks. To combat these harmful effects, network providers dedicate significant amount of resources, e.g. retransmissions and error corrections, to lower the loss rates. This, however, results in the base station transmitting at high throughput r_t but little translating to goodput r_g. We showed that TCP/NC, which is more resilient to losses than TCP, may better translate r_t to r_g. Therefore, TCP/NC may lead to a better use of the available network resources and reduce the number of base stations n_{bs} needed to support users at a given r_g.

References

1. Cisco, Cisco visual networking index: Global mobile data traffic forecast (2011)
2. Kilper, D., Atkinson, G., Korotky, S., Goyal, S., Vetter, P., Suvakovic, D., Blume, O.: Power trends in communication networks. IEEE Journal of Selected Topics in Quantum Electronics 17(2), 275–284 (2011)
3. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP throughput: A simple model and its empirical validation. In: Proceedings of the ACM SIGCOMM (1998)
4. Balakrishnan, H., Padmanabhan, V.N., Seshan, S., Katz, R.H.: A comparison of mechanisms for improving TCP performance over wireless links. IEEE/ACM Transactions on Networking 5 (December 1997)
5. Kim, M., Médard, M., Barros, J.: Modeling network coded TCP throughput: A simple model and its validation. In: Proceedings of ICST/ACM Valuetools (May 2011)
6. Cáceres, R., Iftode, L.: Improving the performance of reliable transport protocols in mobile computing environments. IEEE Journal on Selected Areas in Communications 13(5) (June 1995)
7. Tian, Y., Xu, K., Ansari, N.: TCP in wireless environments: Problems and solutions. IEEE Comm. Magazine 43, 27–32 (2005)
8. Hacker, T.J., Athey, B.D., Noble, B.: The end-to-end performance effects of parallel TCP sockets on a lossy wire-area network. In: Proceedings of the IEEE IPDPS (2002)
9. Ford, A., Raiciu, C., Handley, M., Barre, S., Iyengar, J.: Architectural guidelines for multipath tcp development. IETF, Request for Comments, no. 6182 (March 2011)
10. Sundararajan, J.K., Shah, D., Médard, M., Jakubczak, S., Mitzenmacher, M., Barros, J.: Network coding meets tcp: Theory and implementation. Proceedings of IEEE 99, 490–512 (2011)
11. Lott, C., Milenkovic, O., Soljanin, E.: Hybrid ARQ: Theory, state of the art and future directions. In: Proceedgins of IEEE Information Theory Workshop (ITW), Bergen, Norway (July 2007)