A Wireless Transmission Control Protocol for CDPD

Prasun Sinha Narayanan Venkitaraman Thyagarajan Nandagopal
Raghupathy Sivakumar Vaduvur Bharghavan
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801
Email:{prasun, murali, thyagu, sivakumr, bharghav}@timely.crhc.uiuc.edu

Abstract— **Many current wireless wide area packet data networks are characterized by very low and variable bandwidths, very high and variable delays, significant non-congestion related loss, asymmetric uplink and downlink channels, and occasional blackouts. Additionally, the majority of the latency in a WWAN connection is incurred over the wireless link. Under such operating conditions, most contemporary wireless TCP algorithms do not perform very well.**

In [7], we presented WTCP, a reliable transport protocol that is designed to operate efficiently and fairly over wireless wide area networks. WTCP is rate-based, uses only end-to-end mechanisms, performs rate control at the receiver, and uses the ratio of sending rate to receiving rate as as the primary metric for rate control. In this paper, we evaluate the performance of WTCP over CDPD networks.

We have implemented and evaluated WTCP over the CDPD network, and also simulated it in the ns-2 simulator. Our performance results indicate that WTCP can improve on the performance of comparable algorithms such as TCP-NewReno and TCP-Vegas by between 20% to 200% for typical operating conditions.

I. Introduction

Recent years have witnessed an explosive growth in the use of wireless wide-area networks (WWANs) such as CDPD, RAM and Ardis [1], with industry projections topping $2.5 billion by the year 2002. In the typical WWAN deployment scenario, a mobile user connects over the WWAN network to a dedicated proxy in the corporate backbone, which then acts as the service point for all the requests from the mobile user. Providing efficient and reliable connectivity between the proxy and the mobile host over commercial wide-area wireless networks is thus becoming a critical issue.

Despite the enormous commercial interest, a typical connection over a CDPD network today observes low and highly variable throughput (between $O(1Kbps) - O(10Kbps)$), high and highly variable latency (between $O(400ms) - O(4sec)$), possibly bursty and random packet losses unrelated to congestion (1% - 10%), and occasional blackouts exceeding 10 seconds. Under such conditions, standard transport protocols such as TCP compute a large RTO for loss recovery and wrongly throttle the congestion window upon random packet loss resulting in extremely poor performance.

In [7], we designed a transport protocol called the *Wireless Transmission Control Protocol (WTCP)* to address the primary causes of performance degradation of TCP in wireless wide area networks. The goal of the WTCP protocol is to be deployable, robust, fair and efficient as a reliable transport protocol for commercial WWANs. In this paper, we perform simulation studies to evaluate the performance of WTCP over a CDPD network. In particular, we focus on the issues of throughput improvements and fairness of the WTCP protocol, and also look at the impact of channel contention on the performance of flows in a CDPD network with hosts running WTCP.

The rest of the paper is organized as follows. Section II identifies the typical characteristics of CDPD environments, and how they adversely affect the performance of TCP. Section III gives a brief description of the algorithms for congestion control and reliability in WTCP. Section IV evaluates the performance of the WTCP through simulations in ns-2. Section V concludes this paper.

II. Characteristics of CDPD Environments

In this section, we describe the nature of the CDPD environment, and identify certain characteristics that cause performance degradation with TCP and similar transport protocols.

CDPD [2] is a packet data network that overlays the AMPS cellular telephone infrastructure. It uses the available channels in the AMPS system to send data between a mobile host and the base station. The CDPD channel is 'full duplex'; communication between the base station and the mobile is done through a pair of unidirectional channels, each with a raw capacity of 19.2Kbps. A set of channels may be dedicated for data transmission, or CDPD users may be dynamically assigned to channels that are preferentially shared by cellular phone calls. The data is compressed and encrypted using Reed-Solomon(63,47) error-correcting code. This results in an actual data rate of around 9.6Kbps, which can be shared by upto 30 users. In CDPD, the base station transmits on the 'forward' channel, while on the 'reverse' channel, mobile hosts contend for access. The medium access control protocol for the mobiles is called Digital Carrier Sense Multiple Access with Collision Detection or DSMA/CD. Here, mobile hosts sense the forward channel for a flag bit from the base station which indicates that the reverse channel is free. If the flag bit is set, then hosts contend using the CSMA/CA protocol. The base station informs the successful host that it can transmit data. Unsuccessful hosts perform binary exponential backoff.

From the above description of the CDPD network, we can identify the following factors that cause significant degradation in performance of TCP.

Due to extremely low bandwidths available to the user, the delay-bandwidth product of a connection maybe small (typically 2 or 3 packets). This can affect the congestion control and fast retransmit mechanisms of TCP adversely. TCP sometimes observes artificially larger congestion windows as a result of deep buffering in the CDPD network. While this allows a connection to pump in more packets into the network, sending back to back packets on a slow link artificially increases the round trip time and adversely affects TCP performance in case of a timeout.

CDPD suffers from the well known 'capture syndrome' of binary exponential backoff [**?**], in which a highly loaded shared medium ends up bursting the queued packet transmissions of each contending host in turn. This results in bunching of ACKs in the other direction, skewing the round-trip time and causing the sender to burst out packets.

Typical round trip times observed in CDPD are between 800ms to 4sec. A large fraction of this time of this time is due to transmission on the wireless link (e.g. transmitting a 512 byte packet at 12Kbps takes 300ms), and over 75% of the latency is typically incurred in the mobile switching station↔mobile host segment of the connection. In TCP, since the sender bursts out packets, this causes the deviation in round-trip time to be very high (e.g. 8 sec). As a result, the retransmission timeout of TCP can become very large (e.g. 32 sec).

The Mobile Data Base station(MDBS) in CDPD avoids collisions with AMPS traffic via sniffing and channel hopping. It sniffs a low-frequency RF signal from the transmit path of the voice network and analyzes it for voice activity. It utilizes two forms of channel hopping, planned - which occurs after the MDBS has been transmitting for a pre-defined period of time which is determined by the network manager and forced - which occurs when the sniffer detects AMPS energy and the MDBS shuts down the CDPD forward channel and tunes to another idle channel to resume data transmission. Hence, CDPD users can experience blackouts when there are no channels available. In addition to non-availability of channels, prolonged fades and sudden degradation in signal quality such as traveling through a tunnel or between non-overlapping base stations, and temporary lack of available channels (when cellular phone calls are occupying the channels) can cause blackouts lasting 10 seconds or more, and result in the back-to-back loss of a sequence of packets. Traveling at 55mph, we observed several blackouts ranging from 10 seconds to 10 minutes during the course of a day. In Figure 1, we sent 100 KB of data from the fixed host. On an average with a packet loss rate of about 4%, TCP takes 134 seconds to send 100KB of data from the fixed host to the mobile. There are two sequences of losses one from 13 sec to 26 sec and the other from 42 sec to 120 sec. During these blackout periods, TCP reacts using the RTO mechanism and backs off its RTO value resulting in poor performance.

In addition to the problems mentioned above, that are exclusively due to the nature of the CDPD network, wireless channel error causes significant reduction in throughput of TCP. TCP treats random losses on the wireless channel as congestion losses and tries to reduce its rate resulting in reduced throughput. We have observed that a 5% packet loss causes a 66% reduction in effective TCP throughput.

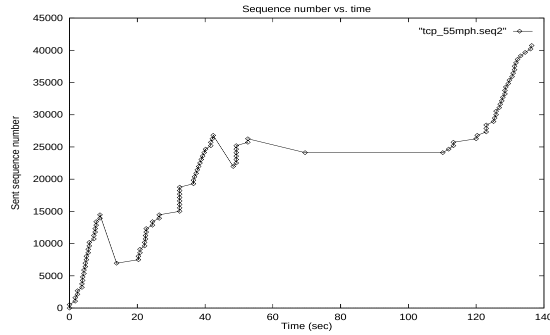


Fig. 1. Performance of TCP on CDPD while moving at 55 mph in a blackout period.

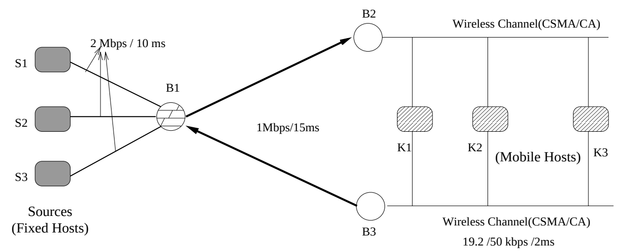


Fig. 2. DSMA/CD Model

To solve the above mentioned issues, we presented the WTCP protocol [7]. In the following sections, we evaluate the WTCP protocol in light of these characteristics of CDPD and explain the improvements obtained over TCP.

Modeling of CDPD environment

Each "full duplex channel" in CDPD, is a pair of shared unidirectional channels. The base station has exclusive access to the downlink 'forward' channel, while the mobile hosts contend for the uplink 'reverse' channel. We simulate the CDPD channel as shown in Figure 2. We simulate the uplink/downlink channels by having two simplex links ($b1 \rightarrow b2, b3 \rightarrow b1$) with a CSMA/CA LAN attached to the two end points ($b2, b3$). All mobile hosts are linked to both the LANs. This model approximates the DSMA/CD based contention mechanism of the CDPD network. We also assume that there exist at least two channels for dedicated data transfer in the cellular voice network. Wireless channel error is modeled through an exponential loss model, where the packet loss distribution on each unidirectional channel is exponentially distributed. Though there are sophisticated models for modeling wireless channel error, we choose this model for its simplicity. We use the topology described in Figure 2 for all our simulations.

III. The WTCP Protocol

A. Design Features

Any reliable transport protocol must provide the following functions: (a) connection management, (b) congestion control, (c) flow control, and (d) reliability. The flow control and connection management in WTCP are similar to the standard TCP mechanisms. The details of the congestion control and reliability mechanisms are presented in [7]. To give an insight to the reader, we briefly mention the design aspects of WTCP here.

The key aspects of congestion control in WTCP are that it is rate based, uses inter-packet delay as the primary mechanism to determine rate adaptation, performs the rate adaptation computations at the receiver, predicts the cause of packet loss and reacts accordingly, and varies the granularity of rate increase/decrease depending on the type of congestion observed. Additionally, WTCP also tailors its startup behavior to work well for short-lived flows. It does not use retransmit timeouts, and it tunes the frequency of sending acknowledgments to the dynamic network conditions.

B. The WTCP Algorithm

We give a short description of the rate control mechanism in WTCP for purposes of understanding. The algorithm is discussed much more elaborately in [7].

The rate control mechanism in WTCP is based on the Linear Increase Multiplicative Decrease paradigm. Specifically, the receiver maintains two ratios: (a) long-term (*lavg_ratio*) and short-term running averages (*savg_ratio*) of the ratio of the observed rate at the receiver to the actual rate of the sender. In addition, it also maintains four threshold parameters $\alpha+, \beta+, \alpha-, \beta-$. If the received packet is the next expected packet, the long term and the short term averages are compared with the constants $\alpha+$ and $\beta+$ respectively. If (*lavg_ratio* $>$ α_+) and (*savg_ratio* $>$) β_+, then the receiver increases the rate and sends it as feedback to the sender (Note that the receiver computes the new rate instead of the sender). The short term average is used to capture the effect of increase in available bandwidth, but to avoid a false increase, the long term average is also required to be above α_+, which is very close to one.

For the decrease phase, WTCP takes a two-pronged approach. First, it decreases the rate multiplicatively when it detects that the receiving rate has fallen below the sending rate by a significant amount (i.e. if (*lavg_ratio* $< \alpha_-$) or (*savg_ratio* $< \beta_-$)). If not, it maintains the rate at the current value. The decrease percentage is according to a *decrease* variable δ maintained at the receiver (δ is initialized to 20%). However, WTCP decreases this decrease variable for two consecutive decreases. Thus, it takes at most one extra RTT, compared to TCP Reno, to drop to 50% (decrease by 20%, followed by a decrease by 40%, as δ is doubled after a decrease, is approximately equivalent to a single drop of 50%). Second, if WTCP detects a congestion loss, then it immediately decreases its rate by half. The receiver, however, needs to distinguish congestion loss from random losses over the wireless channel. For this purpose, the receiver maintains a variable for the maximum expected number of losses in a given window of packets. In a given time window, if the total number of losses exceeds this variable, then the receiver treats this as onset of congestion, and reduces the rate by half.

If the long term and short term averages indicate that no increase or decrease needs to be done, then the receiver simply maintains the current rate, and resets its decrease variable δ to 20%. The expected number of random losses are updated only when the receiver is in this state.

As said before, in WTCP, the receiver computes the desired sending rate via its rate control mechanisms, and notifies this rate to the sender in the ACK packets. ACKs thus carry both reliability information (SACK) and rate control information. The sender monitors the reception of ACKs, and adjusts its rate accordingly. It also monitors the ACKs to tune the ACKing frequency, which it then notifies to the receiver in future data packets. If the sender does not receive an ACK for a threshold period of time, it goes into blackout mode and periodically sends probe packets to elicit ACKs from the receiver and recover from the blackout. The probe packet mechanism is also used for loss recovery, eliminating the need for timeout-based retransmission in WTCP.

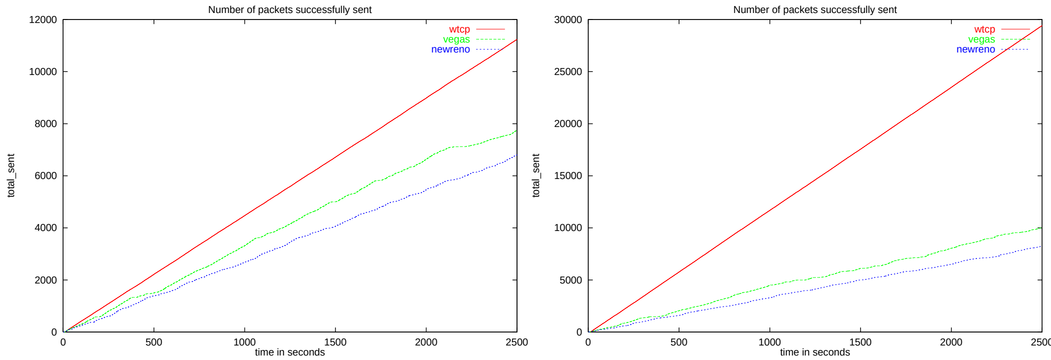
IV. Performance Evaluation

In this section we evaluate the performance of WTCP using simulations in the *ns-2* [6] simulator. The performance measures that we use to evaluate WTCP are: 1. *Channel Utilization:* How effectively does WTCP utilize the scare resources available? 2. *Impact of losses:* How does the presence of significant non-congestion related losses affect the performance of WTCP? 3. *Fairness:* How do multiple flows using the same channel share the available bandwidth? We also compare the performance of the congestion control algorithms in WTCP, TCP-Newreno [4] and TCP-Vegas [3] in terms of their channel utilization and susceptibility to non-congestion related losses.

The network topology used in the simulations is shown in Figure 2. While CDPD currently offers only 19.2Kbps raw bandwidth, it is expected to grow by three or four times in the near future [1]. So results have been presented using both 19.2Kbps and 50Kbps bandwidth for the wireless channel. All the simulations use a queue size of 15 packets and a fixed packet size of 500 bytes. As congestion control is primarily concerned with the total number of packets successfully sent and not the specific sequence number of the packet, we have used the total number of packets successfully transmitted as the primary metric to compare their performances. Also, in most of the simulations we have performed the transfer of data from the fixed host to the mobile host because it is the most common direction of data flow in a CDPD environment.

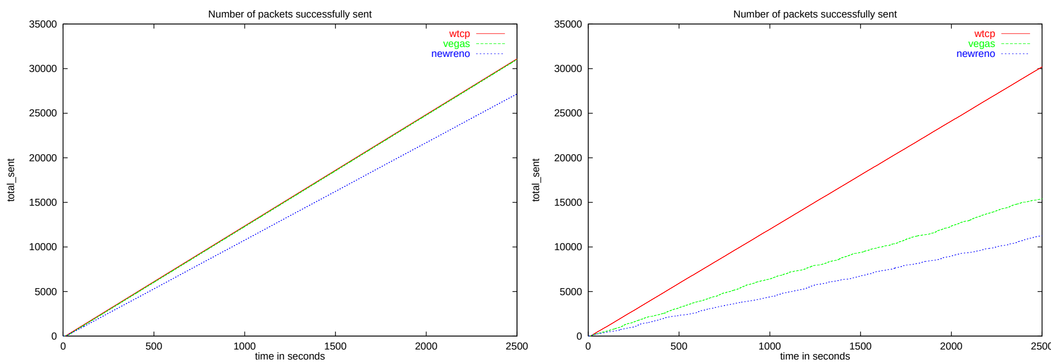
A. Channel Utilization

In this scenario we use a single flow to (a) compare the channel utilization of WTCP and the different versions of TCP that we have considered and (b) Study the effect of increasing er-



(a) (b)

Fig. 3. Total number of packets sent successfully from fixed host with 6% error rate and bandwidths (a) 19.2Kbps and (b) 50Kbps



(a) (b)

Fig. 4. Total number of packets sent successfully from fixed host with 50Kbps bandwidths and error rates of (a) 0% and (b) 4%

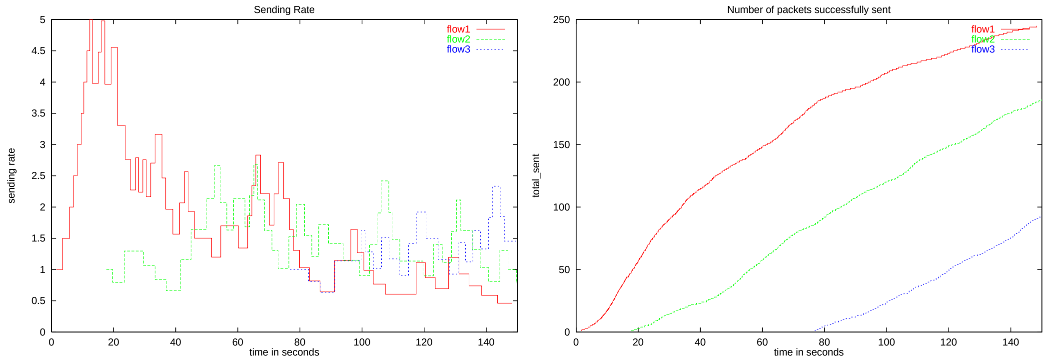
ror rates on WTCP and the TCP versions. Figure 3 (a) and (b) shows the total number of packets sent successfully from fixed host to a mobile host with 6% packet error rate and bandwidths of 19.2Kbps and 50Kbps respectively. It clearly shows that the WTCP achieves maximum channel utilization with both the bandwidth values and that the utilization of Newreno and Vegas are very much below the maximum possible value. In fact, the figure indicates that for a 50Kbps channel with an error rate of 6%, WTCP provides about 200% improvement in performance over the other versions of TCP we have considered. The difference in performance between WTCP and other mechanisms, though still significant, is less pronounced with a smaller bandwidth because for a single flow, the packets already in the queue keep the channel occupied for a larger period of time, thereby reducing the impact of the drastic reduction of the sender's congestion window.

B. Impact of channel error

The results for this scenario shown in Figures 4 (a), (b) and Figure 3, use a wireless bandwidth of 50Kbps and packet error rates of 0%, 4% and 6% respectively. Figure 4 (a) demonstrates that WTCP provides the same throughput as TCP-Vegas when the channel is error free. The next two graphs show that the performance of Newreno and Vegas degrade significantly with increasing error rates. Also note that with high error rates, the maximum channel utilization that can be achieved by TCP is bounded by the error rate itself [5]. The results in this and the previous section reflect that fact. These results indicate that in CDPD networks WTCP clearly outperforms its wire-line end to end counterparts.

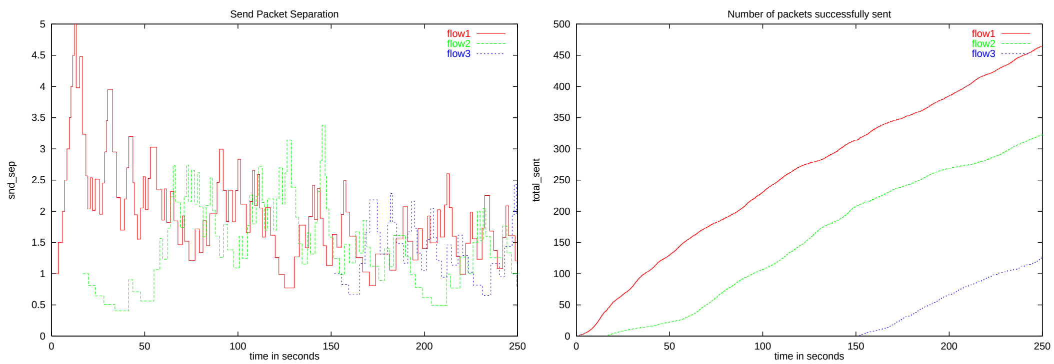
C. Fairness

In this section we study the fairness characteristics of WTCP by observing its behavior when multiple flows using WTCP share a channel. We show the results in the for both the cases where (a) packets are sent from the fixed host to the mobile over the CDPD network and (b) when packets are sent from the mobile host to the fixed host. For all the cases considered we have used a bandwidth of 19.2 Kbps. Figure 5 corresponds to case (a). Figure 5 (a) and (b) show the current rate of a flow and the total number of packets successfully sent respectively. In this case, two new flows are introduced into the system at times $t = 15$ and $t = 150$ *seconds* respectively. The introduction of a new flow results in an increase in the receive packet separation for the existing flows. As a result, the existing flows increase their send packet separation thereby providing more room for the new flow. The figure indicates that the new flows manage to obtain the fair share of the wireless bandwidth in a relatively short period of time. Note that the when flow 2 starts with a rate of 1 packet/sec, flow 1 has a rate of 5 packets/second. As the rates are so low and the round trip times are of the order of 1 second, flow 2 takes a longer time to converge to the fairness



(a) (b)

Fig. 5. WTCP with three flows starting at different times. (a) Rates of individual flows and (b) Total number of packets sent successfully from fixed host



(a) (b)

Fig. 6. WTCP with three flows starting at different times. (a) Rates of individual flows and (b) Total number of packets sent successfully from mobile host

point. However, when flow 3 joins the system, flows 1 and 2 have rates around 2 packets/second, hence, flow 3 catches up with the rate of the other flows very soon. In all the simulations there were no packet losses.

Figure 6 corresponds to the case (b), with multiple uplink flows. As in the previous example, Figures 6 (a) and (b) correspond to the sending rates, number of packets successfully sent by the flows. The new flows are introduced at times $t = 15$ and $t = 30$. We see that the flows converge to fairness even in the downlink case. We expect the contention among different users to cause a significant degradation in the throughput. However, the link utilization is still high, resulting in a high throughput. Note that, WTCP does congestion avoidance with out causing any packet losses and so it does not burst out packets. Resultantly contention does not have a significant effect on the flows. The parallel lines clearly show that the wireless bandwidth is equally divided between the new and the old flows. The results in this section indicates that the WTCP congestion control algorithm is fair.

V. Conclusion

WTCP is rate-based and the rate adjustment is performed at the receiver; consequently, WTCP does not burst packets, overcomes the problems of inaccurate round-trip time computations, and handles asymmetric channels. WTCP uses inter-packet separation at the receiver as the primary metric for rate control with congestion-related loss detection as the backup mechanism; responding early to incipient congestion helps to keep the algorithm stable and at the same time, handling congestion-related losses causes WTCP to react correctly and fairly in worst-case scenarios of sudden congestion peaks. WTCP uses SACK and no retransmission timers for loss recovery; this enables efficient loss recovery without going into prolonged timeouts in the worst case. In summary, WTCP handles most of the problems of CDPD networks, including those traditionally ignored by related work such as large and varying round trip times, and the significant fraction of the delay being incurred in the wireless segment.

References

[1] Wireless Data Forum. http://www.wirelessdata.org/.

[2] J. Agosta and T. Russle. *CDPD: Cellular Digital Packet Data Standards and Technology*. McGraw Hill, New York, NY, 1997.

[3] L. Brakmo and L. Peterson. End to end congestion avoidance on a global internet. *IEEE JSAC*, 13(8), October 1995.

[4] S. Floyd and T. Henderson. The newreno modification to tcp's fast recovery algorithm. Internet Request for Comments 2582, April 1999.

[5] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. Networking*, June 1997.

[6] UCB/LBNL/VINT Network Simulator - ns (version 2), *http://www-mash.CS.Berkeley.EDU/ns/*.

[7] P. Sinha, N. Venkitaraman, R. Sivakumar and V. Bharghavan. WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. In *Proc. of ACM Mobicom '99*, August 1999.