EXPLICIT CONGESTION INDICATION FOR TCP OVER WIRELESS NETWORKS

A Thesis

by

SHISHIR

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2000

Major Subject: Computer Science

EXPLICIT CONGESTION INDICATION FOR TCP OVER WIRELESS NETWORKS

A Thesis

by

SHISHIR

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Nitin H. Vaidya
(Chair of Committee)

Riccardo Bettati
(Member)

A. L. Narasimha Reddy
(Member)

Wei Zhao
(Head of Department)

August 2000

Major Subject: Computer Science

ABSTRACT

Explicit Congestion Indication for TCP over Wireless Networks. (August 2000)

Shishir, B.Tech., Banaras Hindu University

Chair of Advisory Committee: Dr. Nitin H. Vaidya

Transmission control protocol (TCP), the widely used transport protocol in the Internet, assumes that packet losses are because of congestion. Wireless networks have higher error rates than wired networks; TCP misinterprets the error losses as congestion and wrongly invokes congestion control algorithms. In the proposed scheme, we show that by using congestion indication feedback from the network, performance of TCP can be improved in network paths containing wireless links. The routers in the network detect congestion and set a congestion indication bit on packets flowing in the forward direction. The congestion indication is communicated back to the users through the transport-level acknowledgement. When the sender encounters a packet loss, explicit congestion indication feedback received for the packets sent before and after the packet dropped are used to identify the state of the network at the time of drop. If the network is identified as not congested when the packet was lost and the recent congestion indication feedback is also low, then the packet is considered to be lost because of transmission error. We compared our scheme against TCP-Reno which assumes that all packet losses are because of congestion. Under low congestion in the network, our scheme can lead to significant throughput improvement.

To my father, mother and sister

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Vaidya for his guidance, comments and encouragement. Without his advice, enthusiasm and help, my graduate school experience would not have been fulfilling.

I thank Dr. Bettati and Dr. Reddy for being part of my advisory committee.

I would like to thank NSF and Department of Computer Science at Texas A&M University for supporting me financially to carry out my research.

I would like to thank my parents and my sister for their constant support and encouragement.

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

II RELATED WORK 3

- A. DECbit's ECN 3
- B. Random Early Detection 3
- C. Explicit Congestion Notification 5

III THE PROPOSED SCHEME 7

- A. ECI Implementation 7
- B. Role of the Router 9
- C. Receiver 10
- D. Sender's TCP Response 10

IV PERFORMANCE EVALUATION - SATELLITE LINK 13

- A. Simulation Model 13
- B. Methodology 14
- C. Simulation Results 14

V PERFORMANCE EVALUATION - CELLULAR LINK 24

- A. Simulation Model 24
- B. Methodology 24
- C. Simulation Results 25

VI CONCLUSION 34

- A. Advantages 34
- B. Disadvantages 34

REFERENCES 36

VITA 38

LIST OF TABLES

TABLE Page

I ECN Implementation 6

II ECI Implementation 8

LIST OF FIGURES

FIGURE Page

1 Random early detection 4

2 Topology-1: Satellite link r0-r1 14

3 Topology-1: Throughput connection 1 17

4 Topology-1: Aggregate throughput normalized by factor of 10 18

5 Topology-1: Number of wireless transmission error losses and number of wireless transmission error losses correctly identified . . . 20

6 Topology-1: Accuracy Aw 21

7 Topology-1: Number of congestion losses and number of congestion losses correctly identified 22

8 Topology-1: Accuracy Ac 23

9 Topology-2: Cellular link r1-d0 25

10 Topology-2: Throughput connection 1 27

11 Topology-2: Aggregate throughput normalized by factor of 10 28

12 Topology-2: Number of wireless transmission error losses and number of wireless transmission error losses correctly identified . . . 29

13 Topology-2: Number of congestion losses and number of congestion losses correctly identified 31

14 Topology-2: Accuracy Aw 32

15 Topology-2: Accuracy Ac 33

CHAPTER I

INTRODUCTION

Current data networking protocols have been optimized for networks that have packet losses primarily due to congestion. Optimizing these protocols for error-prone wireless links is an important issue for effectively utilizing the available capacity of wireless networks. TCP is a widely used protocol for reliable data delivery in the internet. However, in network paths containing wireless links, TCP performance is limited because of the presence of errors. The reason for this is the implicit assumption in TCP that all packet losses are due to congestion. On a packet loss, TCP reduces transmission rate to allow the network to recover. This behavior has proven beneficial in presence of congestion. However, wireless networks are prone to error, and TCP misinterprets the error losses as congestion and throttles throughput.

Ideally, it would help if a sender could differentiate between packet losses due to congestion from the packet losses due to wireless transmission errors using some end-to-end technique. Attempts to apply heuristics to distinguish between congestion and transmission errors have not been successful [1][2][3]. Other techniques of using Performance-Enhancing Proxies [4] at the boundary of wireless networks require TCP-level awareness by the intermediate nodes.

A Random Early Detection(RED) [5] enabled router detects congestion before the buffer overflows, based on a running average queue size, and drops or marks Explicit Congestion Notification(ECN) bit [8] in the packet probabilistically before the queue actually fills up. The probability of dropping or marking a new arriving packet increases as the average queue size increases above a minimum threshold,

---

[0] The journal model is *IEEE Transactions on Automatic Control.*

towards a maximum threshold. In the proposed scheme, RED gateways are extended to set a Explicit Congestion Indication in the IP header to indicate the congestion level at the router. RED gateways detect congestion and set a explicit congestion indication(ECI) bit in the forwarded packets (when the average queue size at the router becomes greater than the minimum threshold). Explicit congestion indication is echoed back by the receiver in the ack, similar to ECN.

Sender's TCP detects a packet loss in the network if three dupacks are received or a timeout of the retransmit timer is there. TCP assumes that all packet losses are because of congestion and invokes a severe congestion control algorithm. However with our scheme, sender's TCP congestion response to three dupacks depends on the state of the network when the packet was lost and the recent congestion indication feedback. Explicit congestion indication feedback received for the packets sent before and after the packet dropped are used to identify the state of the network when the packet was dropped. If high congestion indication feedback is received for either of these packets then it is assumed that packet was lost because of congestion. However, if both of the congestion indications received for these packets sent before and after the packet dropped are low, and recent congestion indication feedback is also low; then packet is assumed to be lost because of transmission errors, and a less severe congestion control algorithm is invoked.

The remainder of this thesis is organized as follows. Chapter II presents the related work and elaborates the motivation for this thesis. Chapter III presents the details of the scheme. Chapter IV presents the simulation results for a network with a satellite link. Chapter V presents the simulation results for a network with a cellular link. Chapter VI discusses the advantages and disadvantages of the proposed scheme.

CHAPTER II

RELATED WORK

In this chapter we briefly discuss DECbit's ECN bit[9], current RED[5] and ECN[8] mechanisms.

A. DECbit's ECN

In the DECbit congestion avoidance scheme, the gateway uses a congestion-notification bit in packet headers to provide feedback about congestion in the network. When the packet arrives at the gateway, the gateway calculates the average queue length. When the average queue size at the gateway exceeds one, the gateway sets the ECN bit in the packet header of the arriving packet.

The source uses window flow control, and updates its window once every two roundtrip times. If at least half of the packets in the last window had the ECN bit set, then the congestion window is decreased multiplicatively. Otherwise congestion window is increased additively.

The scheme proposed in this thesis borrows from DECbit the idea of marking explicit congestion indication bit in the arriving packet, when the average queue size exceeds the minimum threshold.

B. Random Early Detection

In Random Early Detection(RED) scheme, the gateway detects incipients congestion and implicitly signals the oversubscribing flow to slow down, by dropping its packets. A RED router randomly drops arriving packets, with the result that the probability of dropping a packet belonging to a particular flow is approximately proportional

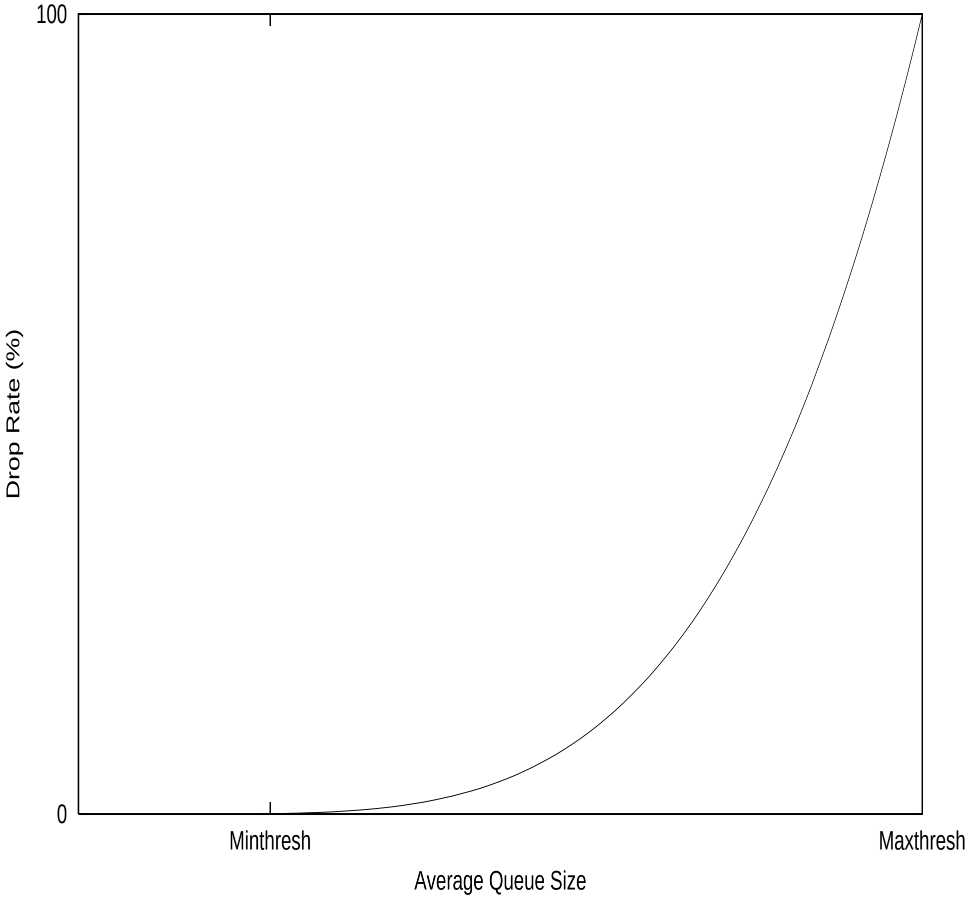

Fig. 1. Random early detection

to the flow's share of bandwidth. RED-enabled router detects incipient congestion before the buffer overflows, based on a running average queue size, and drops packets probabilistically before the queue actually fills up.

RED operates by maintaining two levels of thresholds minimum and maximum. The probability of dropping a new arriving packet increases as the average queue size increases from minimum threshold towards maximum threshold, as shown in Fig 1. When the average queue size exceeds max threshold all arriving packets are dropped. Dropping packets in this way ensures that when some subset of the source TCP packets get dropped and as these TCP sources invoke congestion avoidance algorithms; it will ease the congestion at the gateway. Since the dropping is distributed across flows, the problem of global synchronization is avoided.

C. Explicit Congestion Notification

Explicit Congestion Notification[8] is an extension to RED which marks the IP header instead of dropping the packets (when the average queue size is between min threshold and max threshold; above max threshold arriving packets are dropped as before). Upon receipt of a ECN marked packet, the TCP receiver informs the sender (in the subsequent ACK) about incipient congestion which will in turn trigger the congestion avoidance algorithm at the sender. Packets from flows that are not ECN capable are dropped by RED.

It is necessary that the router identifies that a packet is ECN capable, and should only mark packets that are from ECN capable hosts. This uses two bits in the IP header. The ECN Capable Transport (ECT) bit is set by the sender end system if both the end systems are ECN capable. In TCP this is confirmed in the pre-negotiation during the connection setup phase. Packets encountering congestion are marked by the router using the Congestion Experienced(CE). Bits 6 and 7 of the IPV4 header DSCP field are specified for the ECT and CE bits respectively (Table I).

Table I. ECN Implementation

ECT bit	*CE bit*	
0	0	valid state: not ECN capable
0	1	invalid state
1	0	valid state: CE bit not set
1	1	valid state: CE bit set

The proposal to add ECN to TCP specifies two new flags in the reserved field of the TCP header. Bit 9 in the reserved field of the TCP header is designated as the ECN-Echo (ECE) flag and Bit 8 is designated as the Congestion Window Reduced (CWR) flag. In the connection setup phase, the source and destination TCPs exchange information about their desire and capability to use ECN. This is done by setting both the ECN-Echo flag and CWR flag in the SYN packet of the initial connection phase by the sender; on receipt of this SYN packet, the receiver sets the ECN-Echo flag in the SYN-ACK response.

CHAPTER III

THE PROPOSED SCHEME

ECN is probabilistically marked in the subset of flows through a congested RED gateway (when the average queue size is between min threshold and max threshold). Hence even in networks with all ECN-capable routers, absence of ECN feedback at the sender does not imply absence of congestion. However, in the proposed scheme explicit congestion indication (ECI) bit is set for each packet flowing through the RED gateway, when the average queue size is between min threshold and max threshold. Explicit congestion indication marking explicitly identifies the state of the network. Absence of ECI marking is expected to typically imply an absence of congestion. This observation led us to consider a scheme that uses explicit congestion indication to distinguish error-related drops from congestion-related drops.

In the following sections we will explain ECI bit implementation in the IP header, the role of the router in setting the ECI bit, and sender's TCP response to packet loss.

A. ECI Implementation

This section considers the implementation of ECI bit in the IP header. Implementation of ECI requires one bit in the IP header. Explicit congestion indication (ECI) bit is set for each packet flowing through the RED gateway (when the average queue size is between min threshold and max threshold). However, it is necessary that all the routers in the pre-negotiation during connection setup phase, indicate their capability and desire to mark ECI. This can be done by setting one of the IP option flag in the SYN packet by the sender; if any of the routers in the connection path is not capable of marking ECI, it will reset the flag. The receiver will echo the flag in the ack sent for that packet. Once this agreement is reached, the sender will discriminate packet loss due to congestion from error drops using the ECI feedback.

Bits 6 and 7 of the IPV4 header DSCP field are specified for the ECT and CE bits respectively. ECN implementation (Table I) defines only three valid states. However, ECI requires two valid states to indicate low or high congestion in the network. ECI bit can be marked by changing the semantics of ECN fields, as shown in Table II.

Table II. ECI Implementation

ECT bit	*CE bit*	
0	0	`valid state: not ECN capable`
0	1	`valid state: ECI bit not set and ECN bit not set`
1	0	`valid state: ECI bit set and ECN bit not set`
1	1	`valid state: ECI bit set and ECN bit set`

ECI implementation defines four valid states. Instead of using a ECT bit to indicate that host is not ECN capable; in our scheme, routers consider a host is not ECN capable if both ECT and CE bits are not set. Host is considered to be ECN-capable for any other combination of ECT and CE bits. Note that due to the manner in which ECI and ECN bits are defined, ECI is always set whenever ECN is set.

Two bits in the TCP header are needed for ECN and ECI feedback in the outgoing ack. One of the bits can be ECN-echo flag (in the TCP header). Other bit required can be allocated from the reserved field of the TCP header. When a packet reaches a receiver, the receiver responds by copying ECT and CE bits in the ECN-echo flag and a reserved bit in the TCP header of the next outgoing ack of the flow. Sender TCP's response to ECI marking is discussed in section D.

B. Role of the Router

This section discusses the role of the router in setting the explicit congestion indication bit. This section considers IP networks with RED gateways, where the gateways monitor the average queue size and during congestion marks the explicit congestion indication bit in the arriving packets. Explicit congestion indication (ECI) bit is set in each forwarded packet if the average queue size is greater than minimum threshold at the router.

Routers also probabilistically set the ECN field in the packet header. RED gateways probabilistically drop packets or set ECN bit to punish flows to provide fair sharing of the bottleneck link and to provide early warning to reduce synchronization. However, explicit congestion indication bit is set for all TCP packets arriving at a RED gateway, if the average queue size is greater than minimum threshold at the router.

Algorithm for ECI marking

```
for each packet arrival
  • calculate the average queue size: avg
  • if ($min\_threshold < avg < max\_threshold$)
        mark ECI bit in the arriving packet
```

Explicit congestion indication marking can have two values: low and high. Low congestion indication marking indicates that buffers are available in the network.

C. Receiver

Receiver copies the explicit congestion indication bit in the packet to the corresponding ack or dupack generated on receiving this packet.

D. Sender's TCP Response

Sender detects a packet loss in the network if three dupacks are received or a timeout of the retransmit timer is there. Congestion indication feedback received for the packets sent before and after the lost packet give information about the state of the network just before and after the lost packet. For example, for the first packet lost in a window, latest congestion indication feedback before the drop is in the last ack; earliest congestion indication feedback after the drop is in the first dupack.

Congestion indication feedback received at the sender is used to estimate the average congestion indication (avg_cong_ind) as follows:

$$avg\_cong\_ind = (1 - w_{CI}) * avg\_cong\_ind + w_{CI} * new\_cong\_ind$$

where new_cong_ind is the ECI feedback received and weight w_{CI} determines the time constant of the high-pass filter.

If *avg_cong_ind* is greater than a *cong_threshold*, then recent congestion feedback is considered to indicate high congestion in the network.

If high congestion indication feedback is received for either of the packets sent before and after the lost packet, then it is assumed that packet was lost because of congestion. However, if low congestion indications feedback is received for the packets sent before and after the dropped packet, then it is assumed that there was low congestion in the network at the time of packet drop. Hence, the sender assumes that packet is lost because of transmission errors or some other non-congestion reasons.

If congestion indications feedback is low for the packets sent before and after the dropped packet and $avg\_cong\_ind <= cong\_threshold$ then a less severe congestion control algorithm is invoked. Specifically, congestion window is reduced to three-quarter of its present value. Otherwise packet is assumed to be lost because of congestion and a severe congestion control algorithm is invoked [6].

The condition for identifying a packet loss as a transmission error loss is conservative. The reason for this is that it is preferable for the sake of the network, to mistake a wireless loss for a congestion loss, rather than the opposite. With this scheme, some transmission error losses in a congested network are identified as congestion losses.

Sender's TCP Congestion Control Algorithm

```
for each ack received do
  1. estimate the average congestion indication as follows
       $avg\_cong\_ind = (1 - w_{CI}) * avg\_cong\_ind + w_{CI} * new\_cong\_ind$
  2. if (ack received with ECN-echo bit set) then
       • congestion response as suggested in [8]
  3. if (no. of dupacks received=3) then
       (a) if (Ack with ECN-echo bit set was received in the last RTT) [8]
             • congestion response as suggested in [8]
       (b) else
             • if ( congestion indication bit is not set in the feedback received for the packets sent before and after the dropped packet and $avg\_cong\_ind <= cong\_threshold$)
```

cwnd=ssthresh=3*cwnd/4

- else

  cwnd=ssthresh=cwnd/2

(c) do fast retransmit and fast recovery

CHAPTER IV

PERFORMANCE EVALUATION - SATELLITE LINK

To evaluate our scheme, we will measure the accuracy of discrimination of error losses and congestion losses using the following two metrics :

A_c : accuracy of congestion loss discrimination

A_w : accuracy of wireless loss discrimination

A_c is defined as the ratio of the number of congestion losses correctly identified over total number of congestion losses. A_w is similarly defined as the ratio of the number of wireless transmission error losses correctly identified over total number of transmission error losses.

A. Simulation Model

The simulation scenario consist of two RED gateways r_0 and r_1 as shown in the Fig 2. Each sender s_i is connected through a 1Mbps link to the gateway r_0. All the links in Fig 2 are labeled with a *(bandwidth, propagation delay)* pair. The link from gateway r_0 to node r_1 is the satellite link.

TCP ModReno is the modified version of TCP Reno. TCP ModReno congestion response is specified in Chapter III. Maximum congestion window size for the sender's TCP is set to 64kB. The weight w_{CI} is set to 0.25.

RED Gateways are modified to set the congestion indication bit in the forwarded packets. The buffer size at the router r_0 is 100kB. Minimum threshold of the RED Gateways is set to 10kB and maximum threshold is set to three times the minimum threshold.

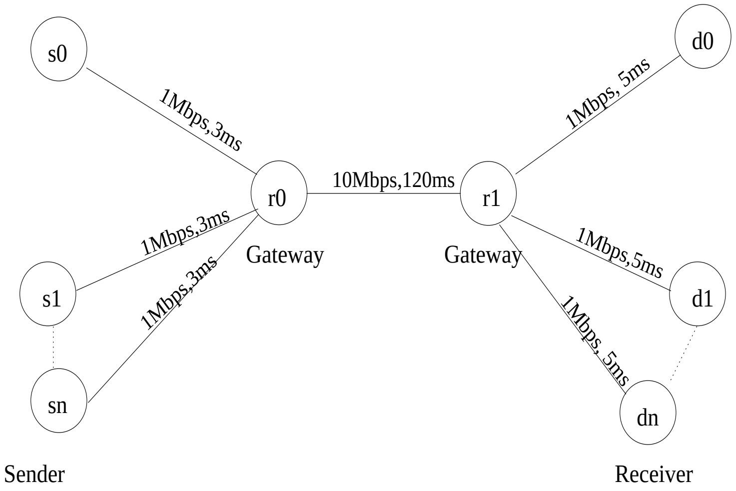

Fig. 2. Topology-1: Satellite link r0-r1

B. Methodology

The objective is to set a TCP connection between two fixed hosts across a satellite link. This connection has to contend with cross traffic to share the link $r_0 - r_1$. This contention will generate congestion losses.

TCP connection i is established from node s_{i-1} to d_{i-1}. TCP connection i starts at *100*i ms*. Simulation are run for 100 seconds. All TCP connections from s_i to d_i experience transmission error losses on the satellite link $r_0 - r_1$.

C. Simulation Results

Simulations were conducted by varying these parameters:

- number of TCP connection n are varied from 10, 12, 15, 20
- mean number of seconds between error T_w takes values 0.1 s, 0.2 s, 0.4 s, 0.8 s, infinity (no error)

- TCP versions TCP_V used are TCP-Reno, TCP-ModReno

For each set of parameters n, T_w and TCP version – TCP-ModReno we measured:

- aggregate throughput of n connections
- throughput of TCP connection #1 between nodes s_0 and d_0
- number of congestion losses for connection #1
- number of congestion losses correctly identified for connection #1
- number of wireless transmission error losses for connection #1
- number of wireless transmission error losses correctly identified for connection #1
- accuracy A_c for congestion losses for connection #1
- accuracy A_w for wireless transmission errors for connection #1

We can observe the following for the given simulation model:

1. Synchronization of flows can happen, if many TCP flows reduce there windows at the same time. If there are only TCP flows in the network, flows tend to get synchronized. Synchronization of flows through a router decreases as the error rate increases.

2. The error model used in the simulations is, mean number of seconds between errors. Hence for a given error rate, number of wireless transmission losses for a connection is dependent on the congestion at the router. As the congestion increases probability of wireless transmission error in a flow decreases.

3. Probability of multiple packet drop in a window for a flow increases, as the error rate increases.
4. Flows through a RED gateway can get synchronized, if not enough flows are marked ECN.
5. Accuracy A_c is dependent on accuracy of congestion indication received, which is dependent on the RED parameters like queue weight, minimum threshold.
6. Accuracy A_w is dependent on cross-traffic characteristics(congestion in the network) and and the accuracy of congestion indication.

Fig 3 presents the throughput for connection #1 with increasing cross-traffic. TCP-ModReno is performing much better than TCP-Reno for all error rates.

As the error rate increases the probability of multiple drops and timeout in a connection increases. However, the proposed scheme can improve performance only if dupack can be received. Therefore the improvement in performance for TCP-ModReno is limited at high error rates. We can observe that for high error rates, as the cross traffic increases, TCP-ModReno performance does not degrade. The reason for this is that when the error rate is high, congestion is low at the router.

We can observe for low error rates that, as the cross traffic increases TCP-ModReno performance decreases. The reason for this is that as the cross-traffic increases, average queue size fluctuate around minimum threshold and, TCP-ModReno starts identifying transmission error losses as congestion losses.

Fig 4 present the aggregate throughput of all connections with increasing cross-traffic. If there are only TCP flows in the network, flows tend to get synchronized. Synchronization of flows through a router decreases as the error rate increases. If the flows are not much synchronized then increase in cross traffic causes aggregate

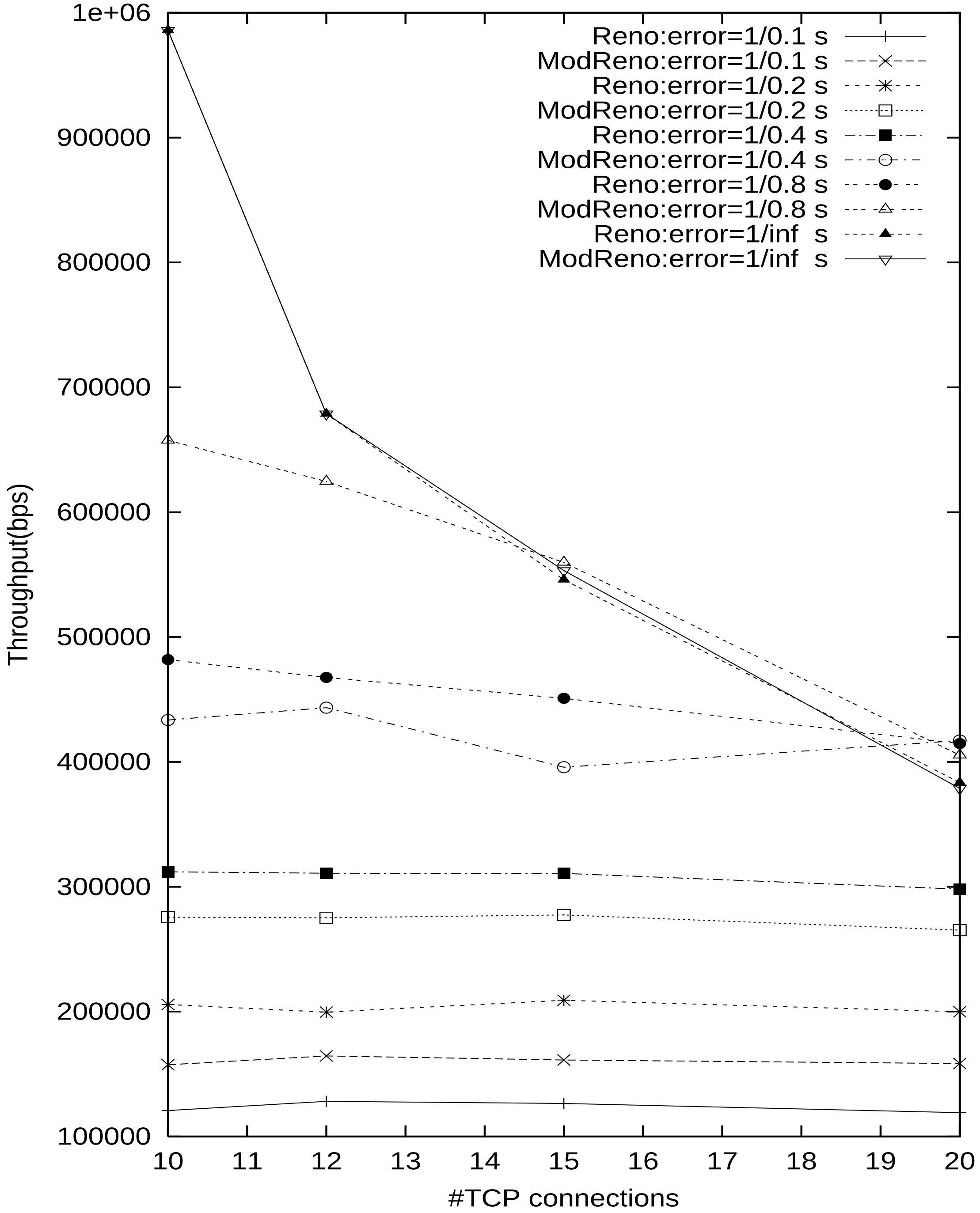

Fig. 3. Topology-1: Throughput connection 1

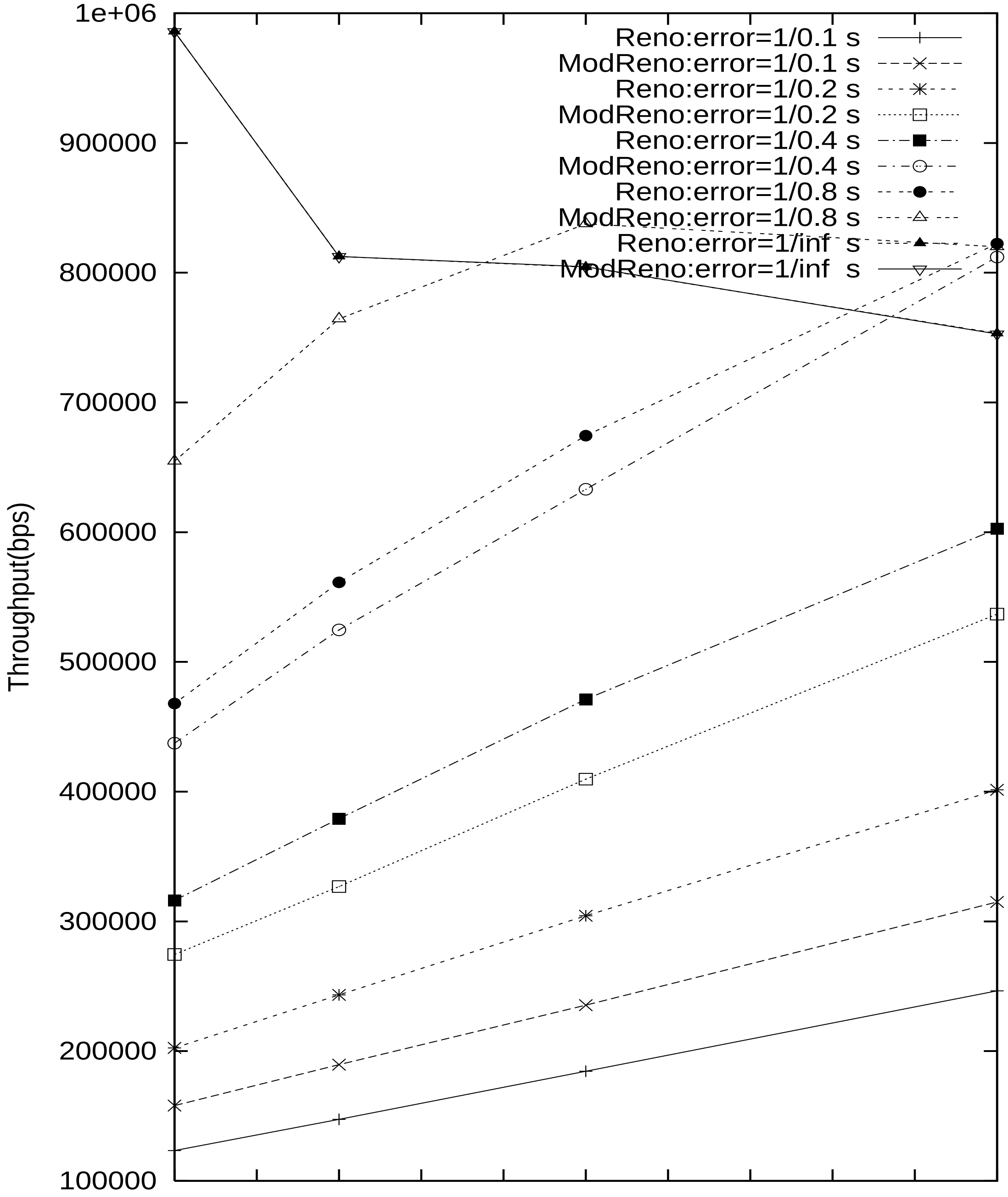

Fig. 4. Topology-1: Aggregate throughput normalized by factor of 10

throughput to increase.

Fig 5 present the number of wireless transmission error losses for connection #1 and number of wireless transmission error losses correctly identified by our TCP-ModReno at node s_0, with increasing cross-traffic.

Fig 6 present the Accuracy Aw for connection #1 with increasing cross-traffic.

Number of wireless transmission losses identified at sender s_0 with our scheme, are dependent on the cross-traffic characteristics(congestion in the network) and the accuracy of congestion indication. As the error rate increases the probability of multiple drops in a connection and timeout increases, which reduces the accuracy Aw.

The error model used in the simulations is, mean number of seconds between errors. Hence for a given error rate, number of wireless transmission losses for a connection is dependent on the congestion at the router. As the congestion increases in the network, number of wireless transmission error in a flow decreases. However when the error rate is high, congestion is low at the router.

Fig 7 present the number of congestion losses for connection #1 and number of congestion losses correctly identified by our scheme at node s_0 with increasing cross-traffic. As the error rate increases, congestion decreases at the router.

Fig 8 present the Accuracy Ac for connection #1 with increasing cross-traffic. Number of congestion losses identified at sender s_0 with our scheme, is dependent on accuracy of congestion indication received, which is dependent on the RED parameters like queue weight. As the scheme used for discriminating error losses from congestion losses is conservative, accuracy Ac is always high.

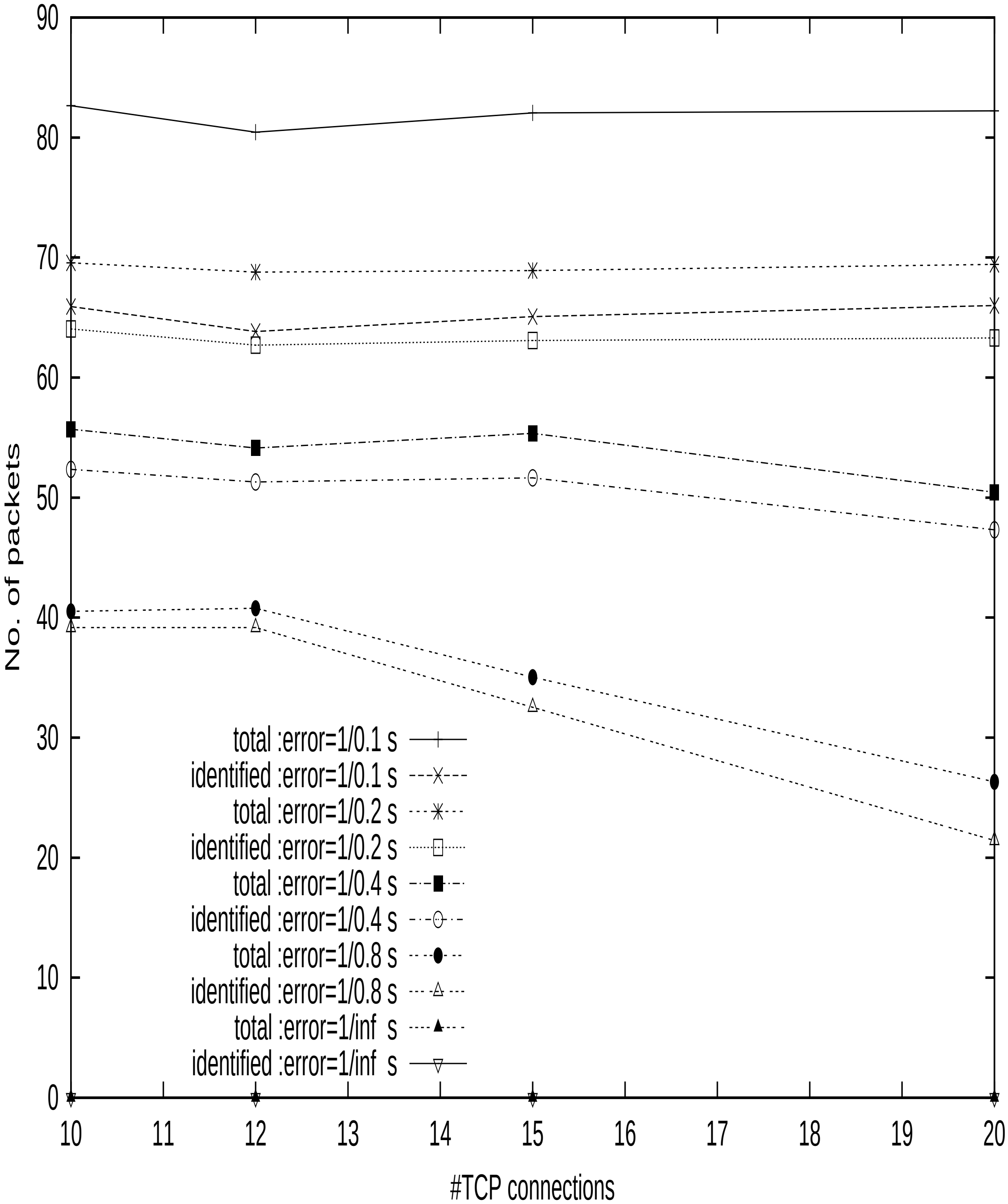

Fig. 5. Topology-1: Number of wireless transmission error losses and number of wireless transmission error losses correctly identified

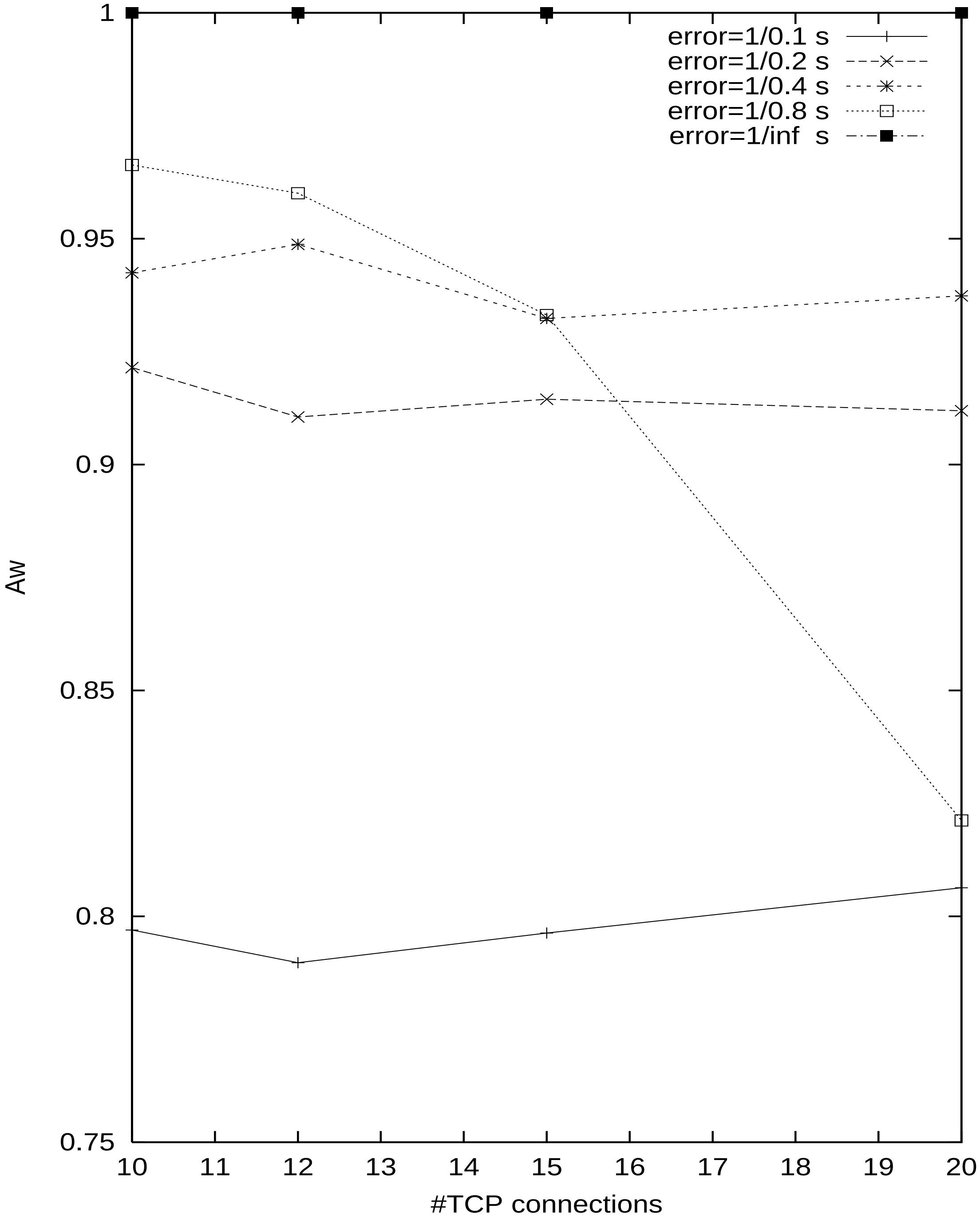

Fig. 6. Topology-1: Accuracy Aw

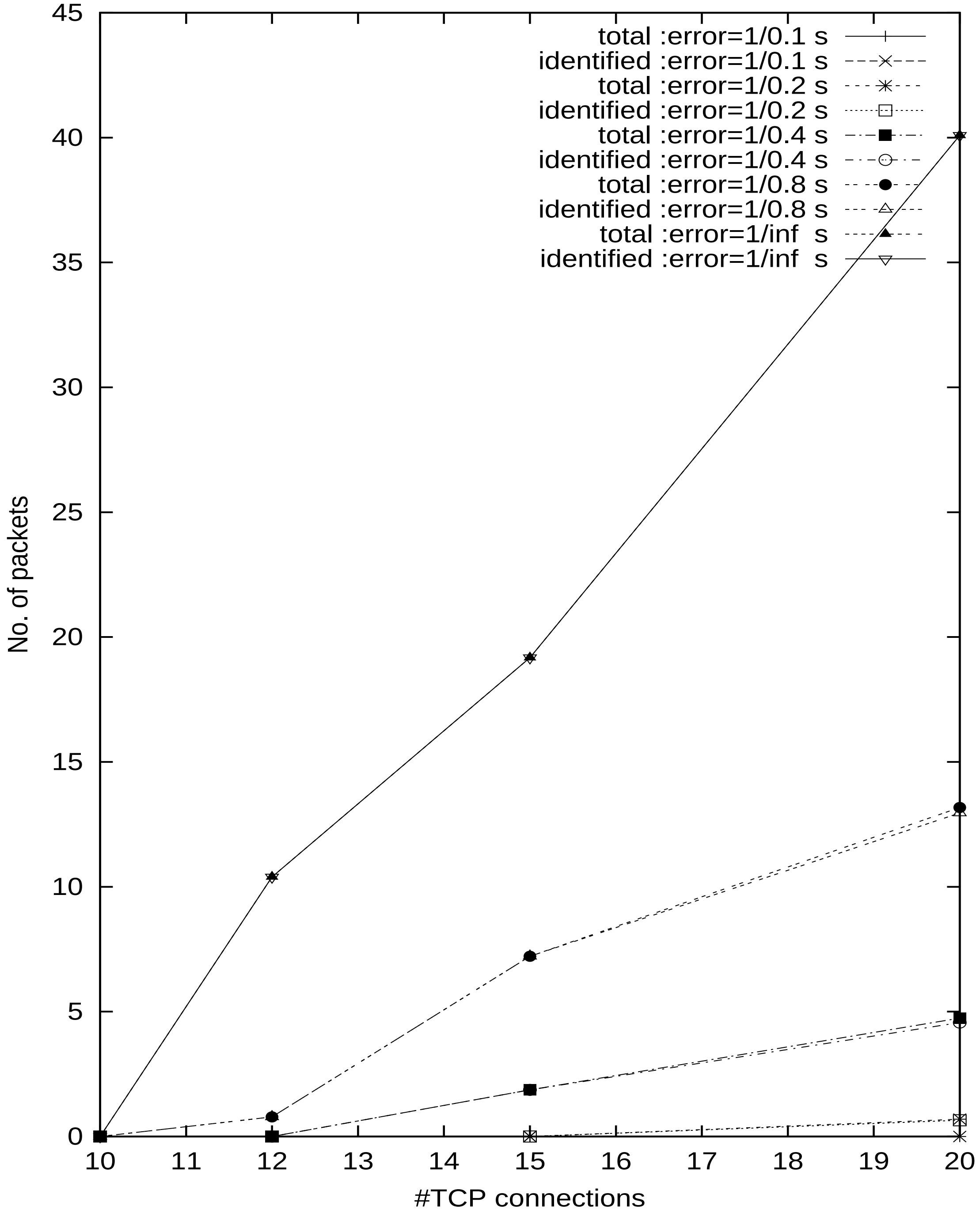

Fig. 7. Topology-1: Number of congestion losses and number of congestion losses correctly identified

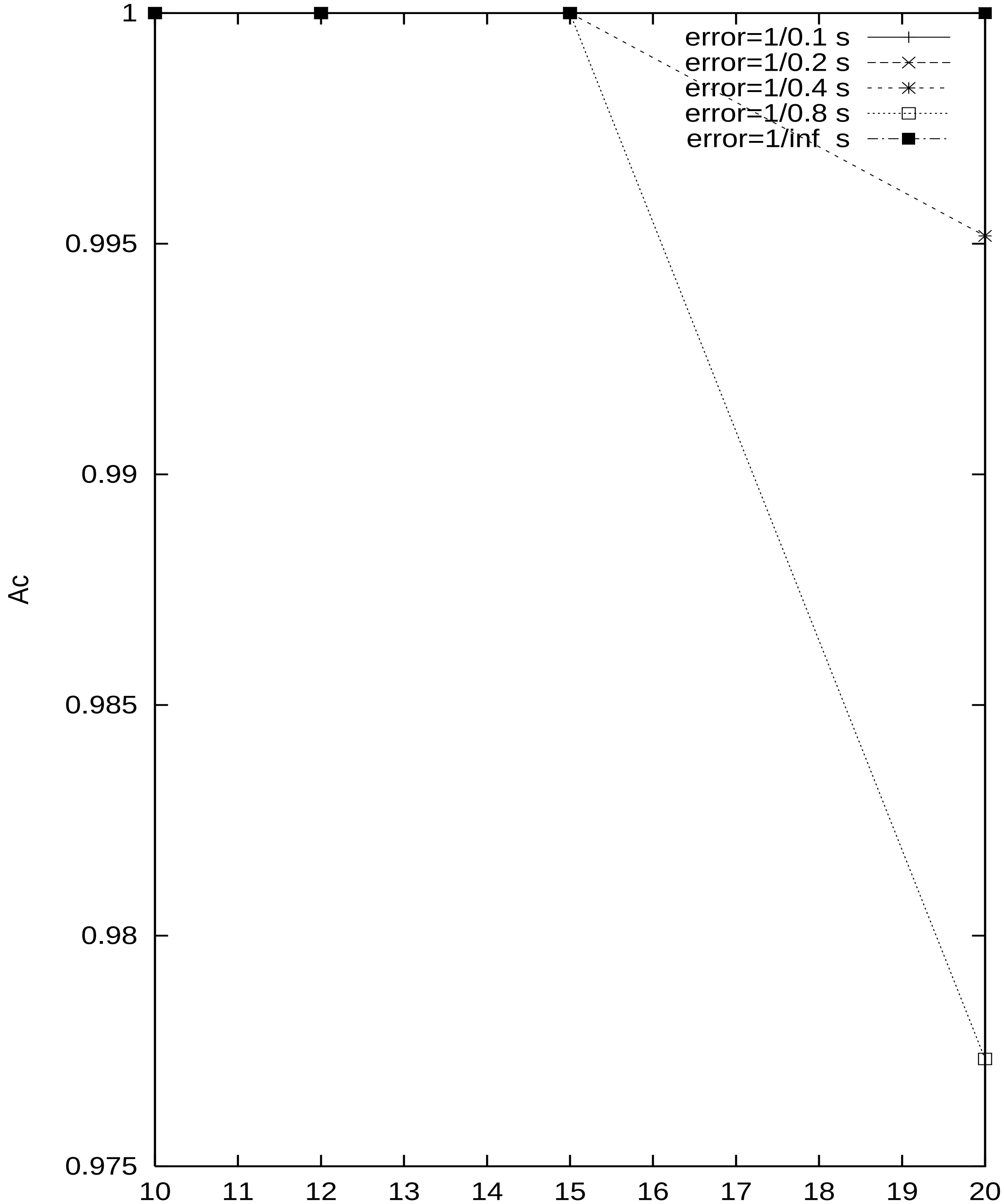

Fig. 8. Topology-1: Accuracy Ac

CHAPTER V

PERFORMANCE EVALUATION - CELLULAR LINK

This chapter discusses the performance evaluation of the scheme for the network with a cellular link.

A. Simulation Model

The proposed scheme is evaluated using the simulation tool *ns-2*[7]. The simulation scenario consist of two RED gateways r_0 and r_1 as shown in the Fig 9. Each sender s_i is connected through a 1Mbps link to the gateway r_0. All the links in Fig 9 are labeled with a *(bandwidth, propagation delay)* pair. The link from gateway r_1 to node d_0 is wireless.

TCP ModReno the modified version of TCP Reno. TCP ModReno congestion response is specified in Chapter III. Maximum congestion window size for the sender's TCP is set to 64kB. The weight w_{CI} is set to 0.25.

RED Gateways are modified to set the congestion indication bit in the forwarded packets. The buffer size at the router r_0 is 100kB. Minimum threshold of the RED Gateways is set to 10kB and maximum threshold is set to three times the minimum threshold.

B. Methodology

The objective is to set a TCP connection between a fixed host and wireless host. This connection has to contend with cross traffic to share the link $r_0 - r_1$. This contention will generate congestion losses.

TCP connection i is established from node s_{i-1} to d_{i-1}. TCP connection i starts

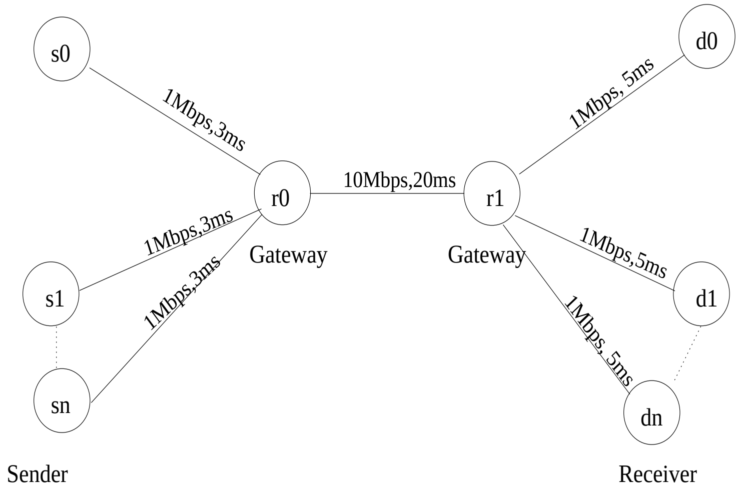

Fig. 9. Topology-2: Cellular link r1-d0

at *100*i ms*. Simulation are run for 100 seconds. TCP connection #1 from s_0 to d_0 experiences transmission error losses on the cellular link $r_1 - d_0$.

C. Simulation Results

Simulations were conducted by varying these parameters:

- number of TCP connection n are varied from 10, 12, 15, 20
- mean number of seconds between error T_w takes values 0.1 s, 0.2 s, 0.4 s, 0.8 s, infinity (no error)
- TCP versions TCP_V used are TCP-Reno, TCP-ModReno

For each set of parameters n, T_w and TCP version – TCP-ModReno we measured:

- aggregate throughput of n connections
- throughput of connection #1

- number of congestion losses for connection #1
- number of congestion losses correctly identified for connection #1
- number of wireless transmission error losses for connection #1
- number of wireless transmission error losses correctly identified for connection #1
- accuracy A_c for congestion losses at sender s_0
- accuracy A_w for wireless transmission errors at sender s_0

Fig 10 present the throughput for connection #1 with increasing cross-traffic over the wireless link. Fair share of the aggregate bandwidth at the router exponentially decreases as the cross-traffic increases. When the cross traffic is less, TCP-ModReno is performing much better than TCP-Reno for all error rates. But as the cross traffic increases TCP-ModReno performance is comparable with TCP-Reno. The reason for this is that as the cross-traffic increases, average queue size fluctuate around minimum threshold and, TCP-ModReno start identifying transmission error losses as congestion losses.

Fig 11 present the aggregate throughput of all connections with increasing cross-traffic. As there is only TCP traffic in the network, connections tend to get synchronized. Hence the aggregate throughput goes down from 10 to 20 connections. This synchronization happens inspite of using a RED Gateway and staggered start time of TCP connections, as not enough flows are getting ECN marked. max_p parameter in RED was set to 0.15. Min threshold was set to 10 packets. Min threshold need to be set lower and max_p need to be set higher to prevent synchronization.

Aggregate bandwidth of TCP-Reno and TCP-ModReno connections is comparable.

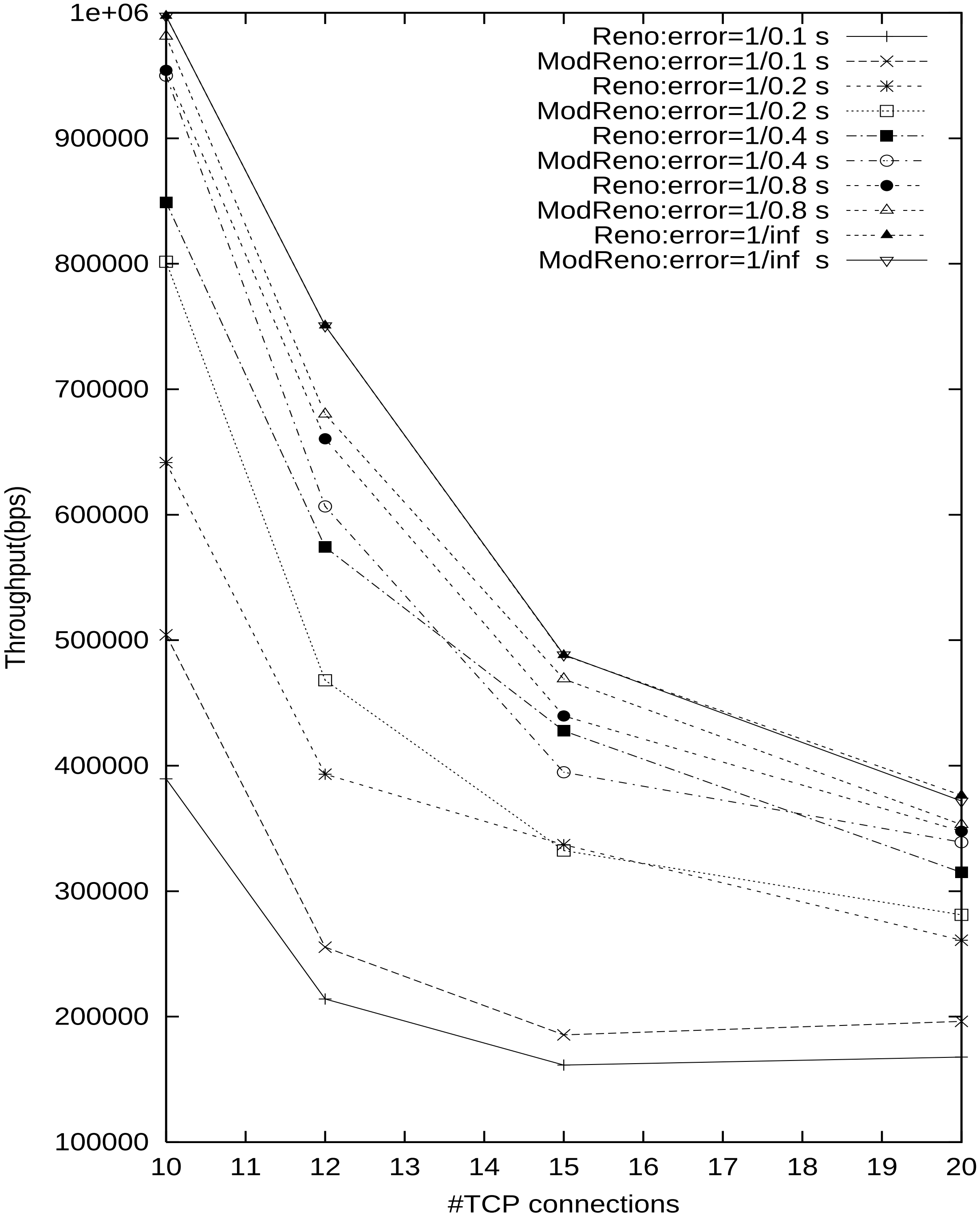

Fig. 10. Topology-2: Throughput connection 1

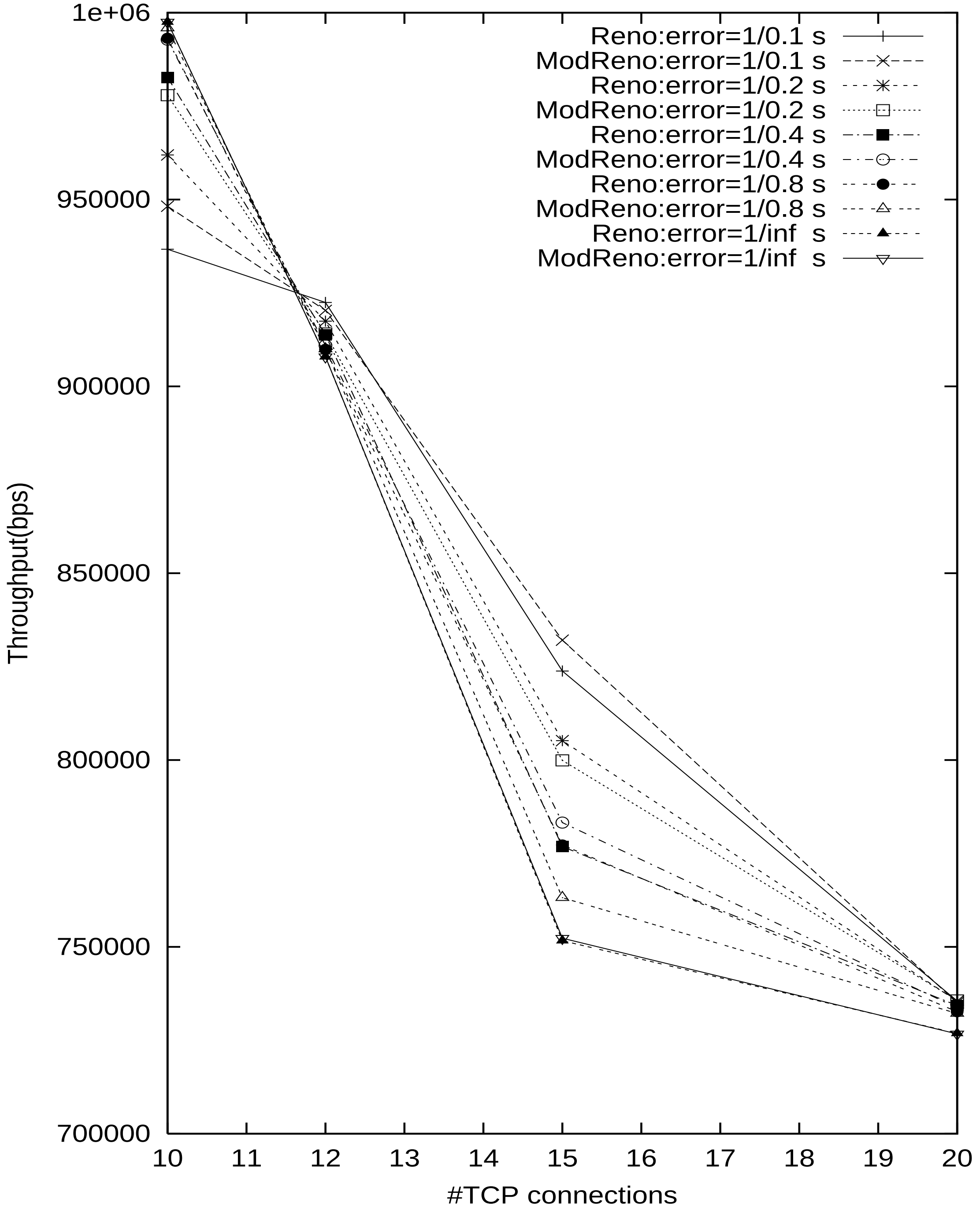

Fig. 11. Topology-2: Aggregate throughput normalized by factor of 10

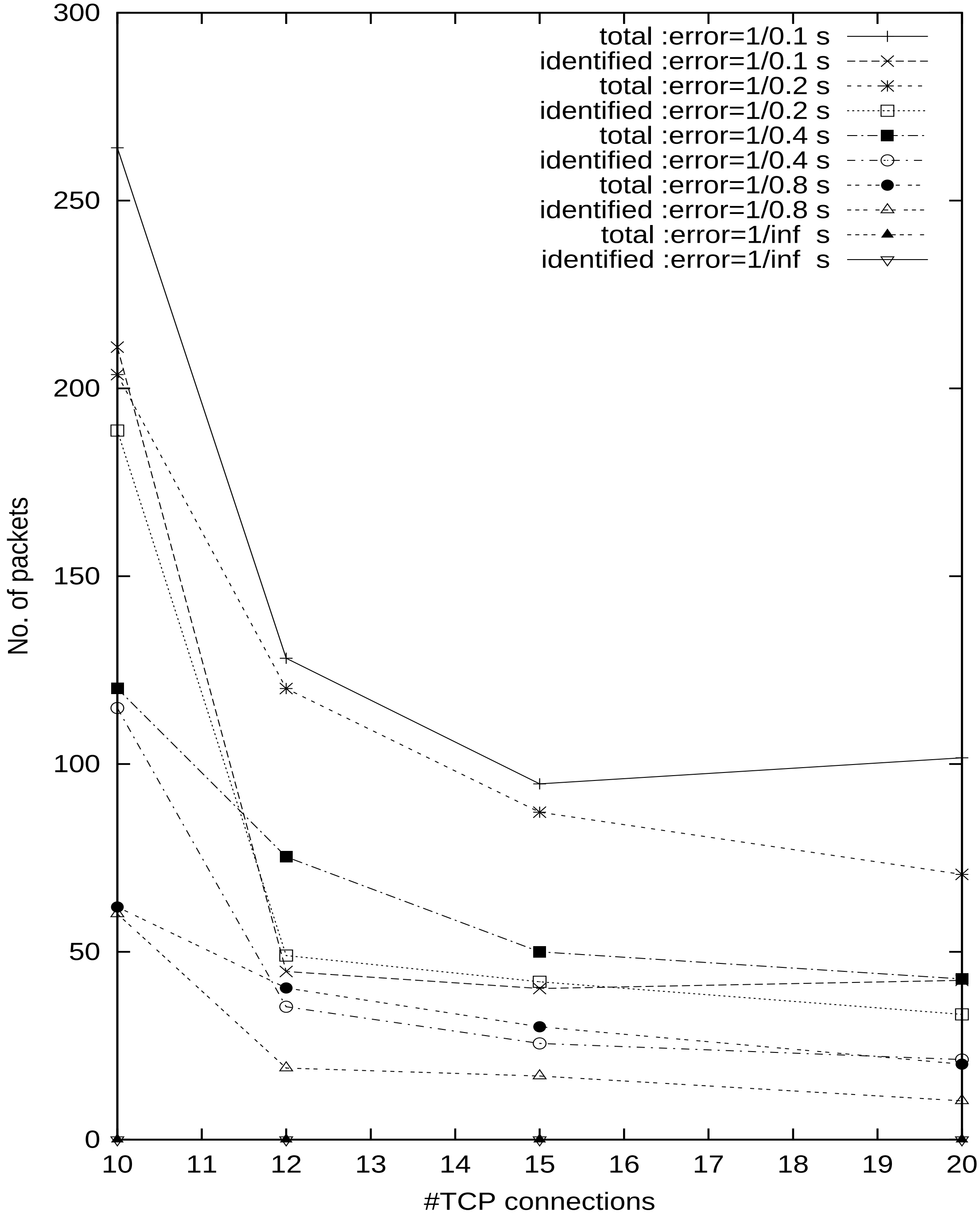

Fig. 12. Topology-2: Number of wireless transmission error losses and number of wireless transmission error losses correctly identified

Fig 12 present the number of wireless transmission error losses for connection #1 with increasing cross-traffic and number of wireless transmission error losses correctly identified at sender s_0 by TCP-ModReno.

Fig 13 present the number of congestion losses for connection #1 with increasing cross-traffic and number of congestion losses correctly identified at sender s_0 by TCP-ModReno.

Fig 14 present the Accuracy Aw for connection #1 with increasing cross-traffic. Number of wireless transmission losses identified at sender s_0 by TCP-ModReno, is dependent on cross-traffic characteristics(congestion in the network), the accuracy of congestion indication (fluctuations in the average queue size). As the the probability of multiple drop in a window decreases with decreasing error rate, accuracy Aw improves.

Fig 15 present the Accuracy Ac for connection #1 with increasing cross-traffic. Number of congestion losses identified at sender s_0 by TCP-ModReno, is dependent on accuracy of congestion indication received, which is dependent on the RED parameters like queue weight. As the scheme used for discriminating error losses from congestion losses is conservative, accuracy Ac is always high.

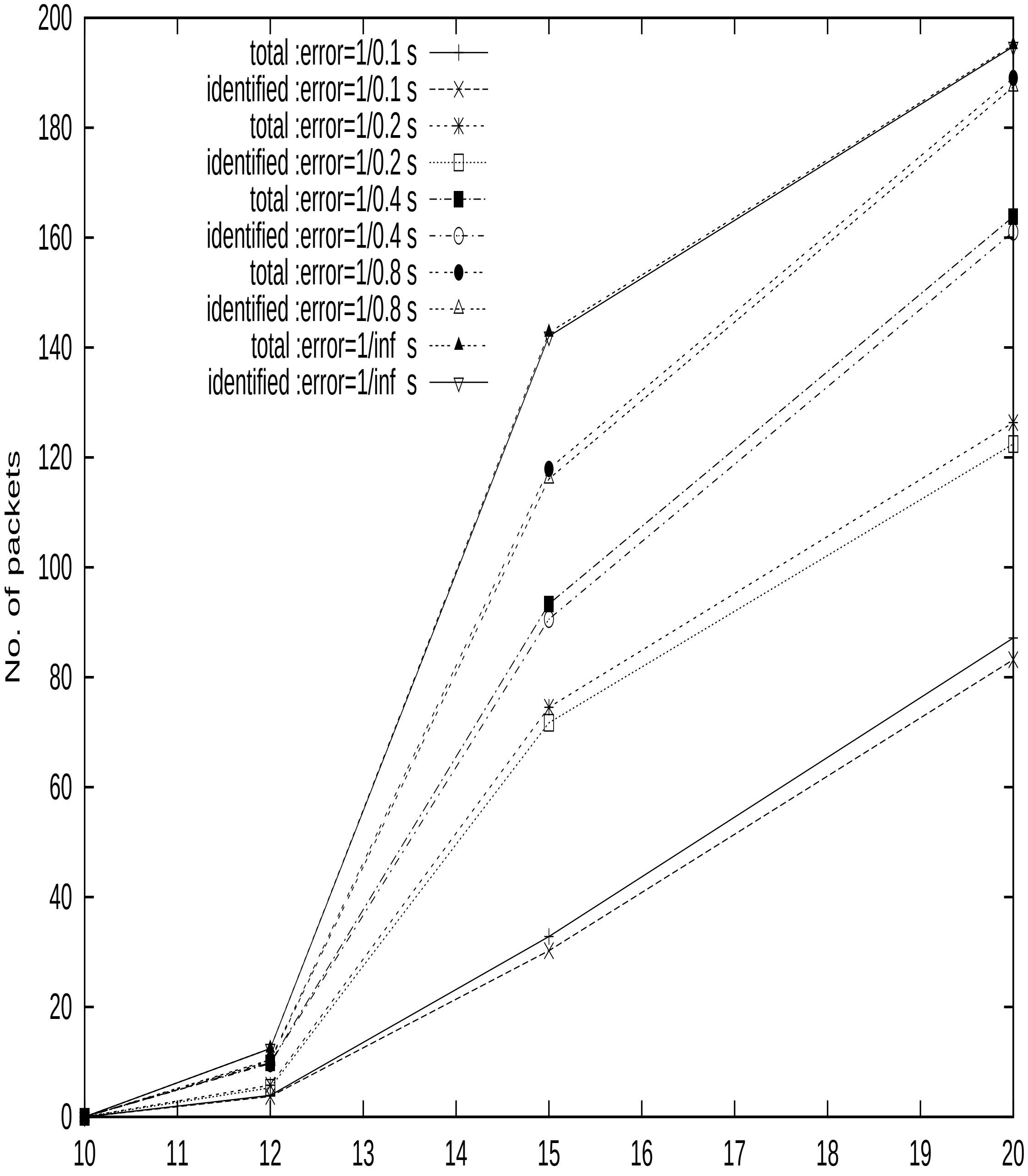

Fig. 13. Topology-2: Number of congestion losses and number of congestion losses correctly identified

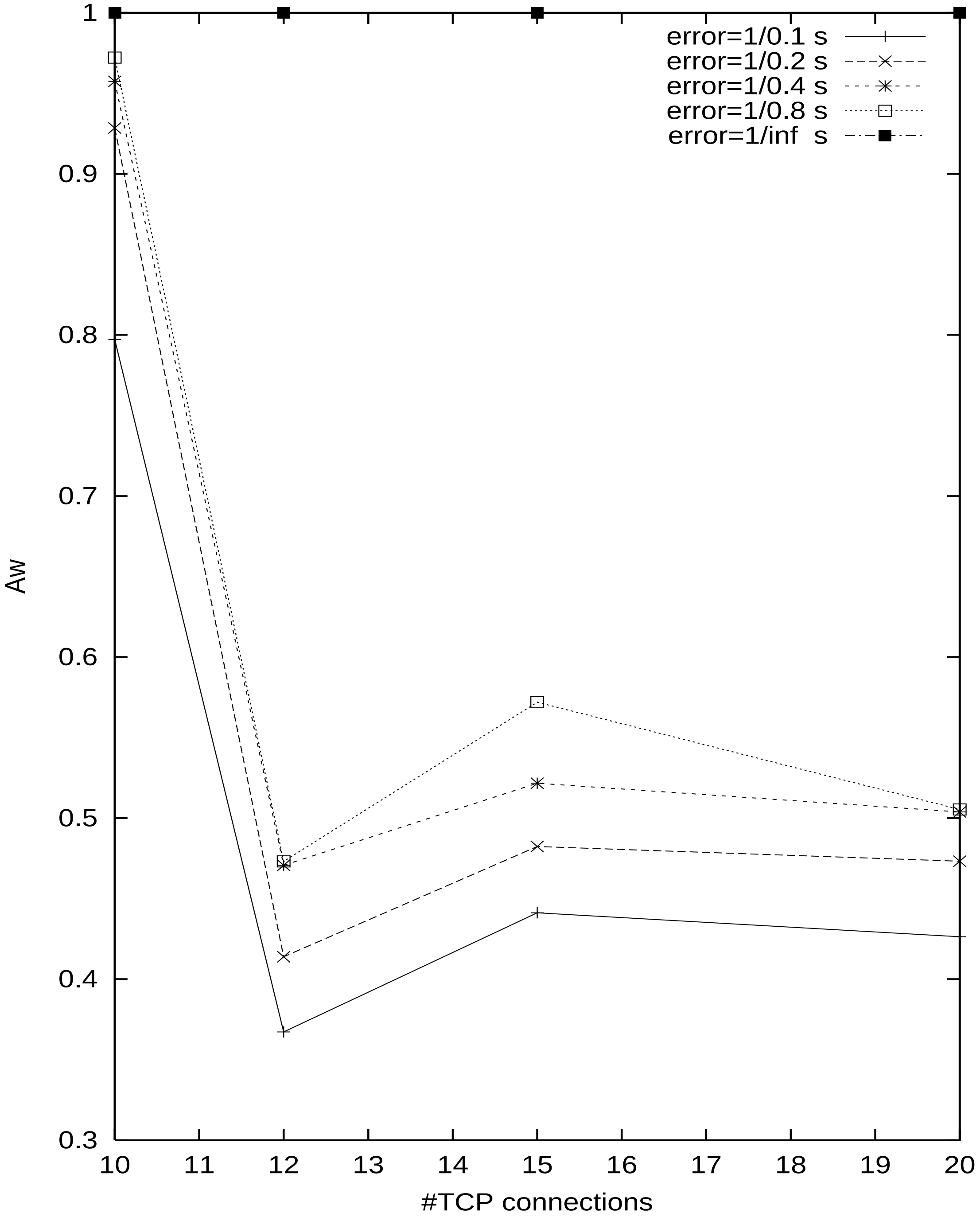

Fig. 14. Topology-2: Accuracy Aw

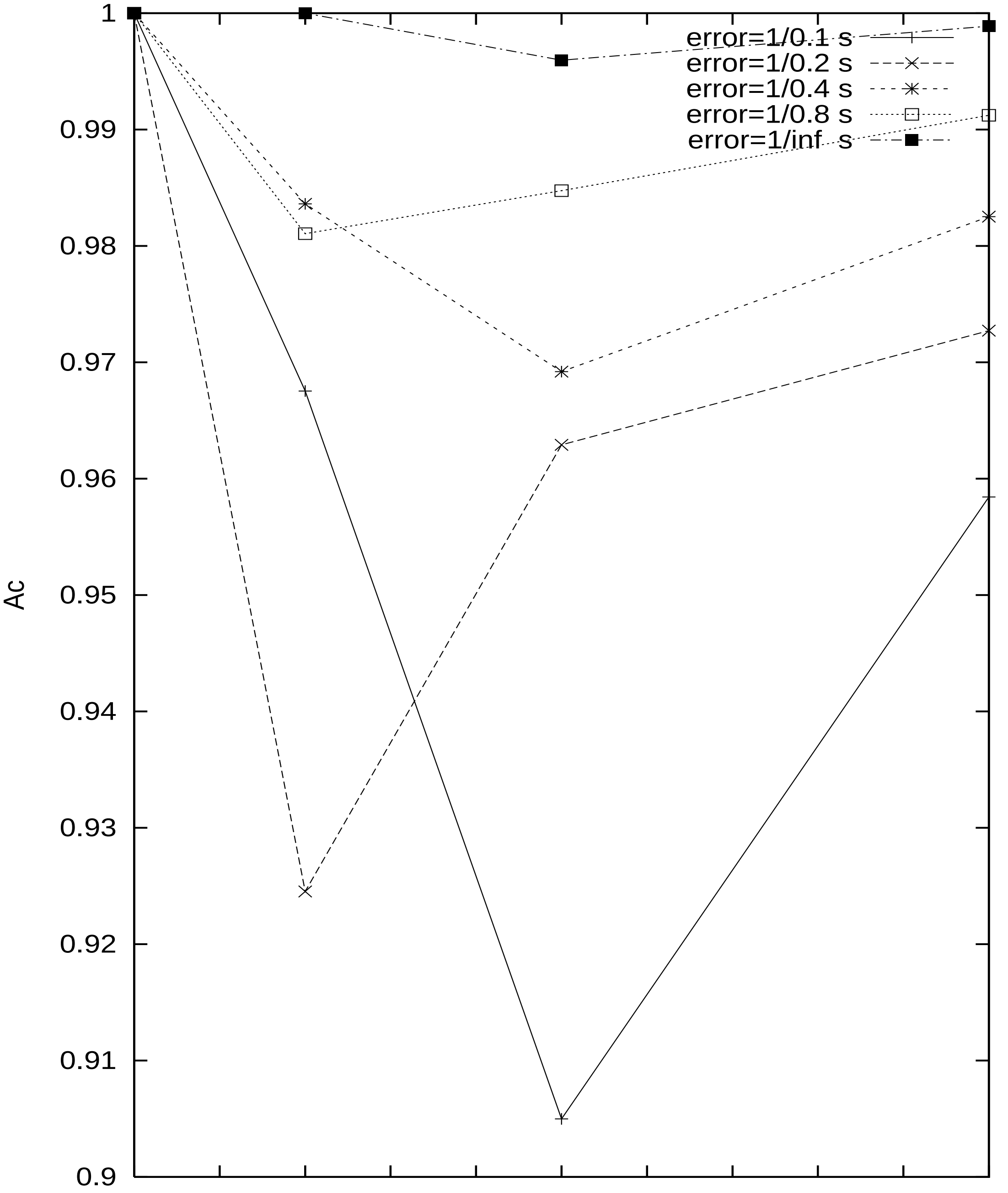

Fig. 15. Topology-2: Accuracy Ac

CHAPTER VI

CONCLUSION

In this chapter we discuss the advantages and disadvantages of the proposed Explicit Congestion Indication (ECI) mechanism, when used in conjunction with ECN.

A. Advantages

ECN marking is received by the subset of flows through a congested RED gateway but congestion indication is received for each TCP packet flowing through the gateway. Hence even in networks with all ECN-capable routers, absence of ECN does not imply absence of congestion. Thus each sender require explicit congestion indication from the network about its state to distinguish error related drops from congestion related drops.

Implementation of Explicit Congestion Indication is not difficult for RED Gateways as they already monitor the average queue size. Generation and communication of the ECI feedback itself do not consume any significant additional resources.

Moreover most of the other schemes designed for improving TCP performance in wireless network assume that wireless link is the last hop. Our scheme can work even in scenarios where wireless hop is not the last hop.

B. Disadvantages

Congestion indication is an unnecessary overhead for wired networks, and it requires changing the semantics of header bits.

This scheme will not work if any of the router on the network path is non-compliant in marking explicit congestion indication.

This scheme can improve performance only if dupacks are received by the TCP sender. This scheme therefore may not improve performance if multiple packets are dropped in the window and there is timeout of the retransmit timer.

Congestion indication is subject to noise if queue size at the RED gateway fluctuate too much. Although average queue size can filter some of the effect of queue oscillations, but still congestion indication depends on other parameters like queue weight, minimum threshold and maximum threshold. Actual setting of these parameters needs to be further investigated. For interactive traffic, congestion indication feedback in the last ack should be aged after some delay as the relevance of congestion indication decreases with time.

REFERENCES

[1] S. Biaz and N. Vaidya, “Using end-to-end statistics to distinguish congestion and corruption losses: a negative result,” Technical Report 97-009, Texas A&M University, August 1997.

[2] S. Biaz and N. Vaidya, “Sender-based heuristics for distinguising congestion losses from wireless transmission losses,” Technical Report 98-013, Texas A&M University, June 1998.

[3] S. Biaz and N. Vaidya, “Discriminating congestion losses from wireless losses using inter-arrival times at the receiver,” Technical Report 98-014, Texas A&M University, June 1998.

[4] J. Border, M. Kojo, J. Griner, and G. Montenegro, “Performance implications of link-layer characteristics: performance enhancing proxies,” Internet Draft URL:http://www.ietf.org/internet-drafts/draft-ietf-pilc-pep-01.txt, March 2000.

[5] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, V.1 N.4, pp. 397-413, August 1993.

[6] V. Jacobson, “Congestion avoidance and control,” in *Proc. of SIGCOMM 1988*, pp. 314-329, August 1988.

[7] “Network simulator,” URL:http://www-mash.cs.berkeley.edu/ns/ns.html [Accessed April 2000].

[8] K. K. Ramakrishnan and S. Floyd, “A proposal to add explicit congestion notification(ECN) to IP,” RFC 2481, URL:http://info.internet.isi.edu/in-notes/rfc/files/rfc2481.txt, January 1999

[9] K. K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks," *ACM Trasactions on Computer Systems*, Vol. 8 No. 2, pp. 158-181, 1990.

VITA

Shishir received his Bachelor of Technology in Computer Science from Banaras Hindu University, India in May 1997. He joined the graduate program in Computer Science at Texas A&M University, in September 1998. His research has been in mobile computing and TCP over wireless. His address is Department of Computer Science, Texas A&M University, College Station, TX 77843-3112.

The typist for this thesis was Shishir.