A cross-layer design for TCP end-to-end performance improvement in multi-hop wireless networks

Rung-Shiang Cheng[a], Hui-Tang Lib[b]

[a]Department of Electrical Engineering, National Cheng Kung University, Taiwan

[b]Department of Electrical Engineering and Institute of Computer Communication Engineering, National Cheng Kung University, Taiwan

Journal of Computer Communications, Volume 31, Issue 14, September 2008, Pages 3145-3152

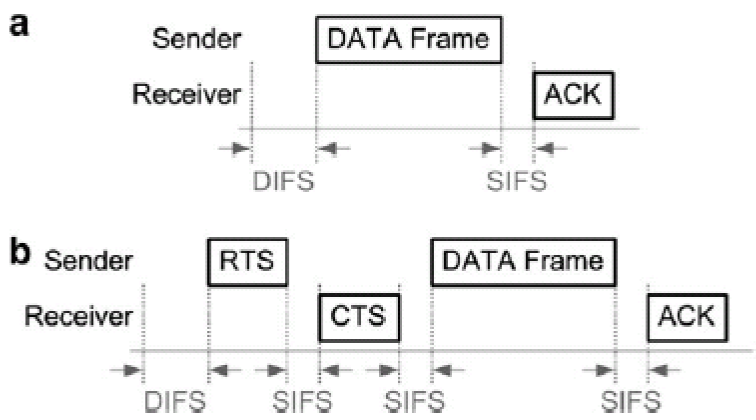
Outline

- Introduction
- Description of proposed method
  - Extension to IEEE 802.11 MAC
  - Modification to TCP
- Numerical results
- Conclusion

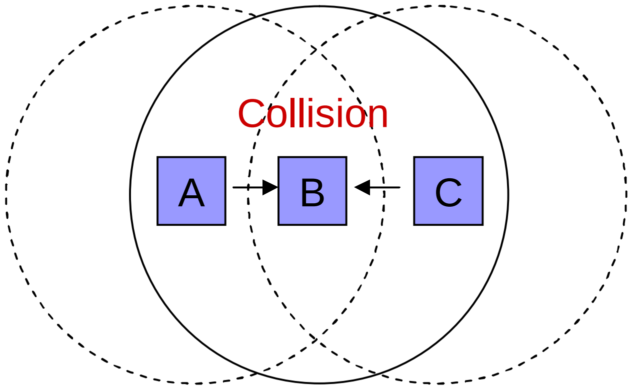
Introduction

- IEEE 802.11has emerged as the standard of choice for wireless networks
  - The standard defines the MAC layer and physical layer
- The MAC layer specifies two media access control services
  - Distributed coordination function (DCF)
    - Infrastructureless network (i.e. ad hoc networks)
  - Point coordination function (PCF)
- DCF is based on the conventional carrier sense multiple access with collision avoidance (CSMA/CA) protocol

Introduction (cont.)

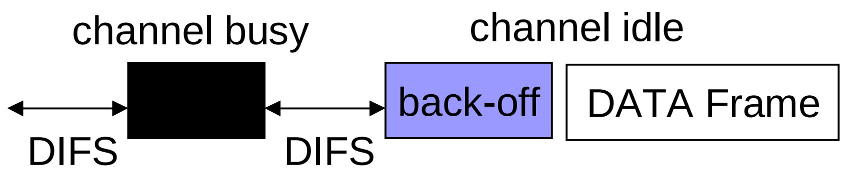


Media access mechanisms in IEEE 802.11 DCF
(a) Basic access mechanism
(b) Virtual carrier-sensing mechanism



Hidden-terminal problem

DIFS: Distributed inter-frame space
SIFS: Short inter-frame space
RTS: Request-to-send
CTS: Clear-to-send



back-off: 0 ~ contention window (integer)

Introduction (cont.)

- The CSMA/CA based protocol are limited to one-hop communication
  - With multi-hop communication being taken care of by the upper layer network protocol
- Multi-hop communication has a problem
  - The network performance is degraded significantly as the number of contending stations increases
- In wireless networks, channel access contentions may occur
  - Between different flows passing through the same vicinity
  - Between different packets within the same flow
    - TCP self-collisions in the MAC layer

Introduction (cont.)

- The inability of TCP to determine whether a packet has been lost as a result of transmission errors or as a consequence of network congestion

- To enhance IEEE 802.11 MAC and TCP protocols
  - TCP is rendered capable of differentiating between corruption and congestion losses by using information received from the MAC layer

Description of proposed method
Extension to IEEE 802.11 MAC

- In standard IEEE 802.11 DCF schemes, whenever a node fails to transmit a frame
  - It retransmits that frame and then increases the value of the retry limit parameter (RET_L)
  - If the updated RET_L exceeds a retry threshold value
    - The MAC layer reports a link breakage to the network layer
    - Discards the frame and resets RET_L to 0

- The MAC layer may wrongly infer a link failure
  - When the link experiences a high degree of contention

- To introduce a new variable, retransmission limit (RET_F)
  - To record the number of retransmission attempts in the event of continuous transmission failures

Description of proposed method (cont.) Extension to IEEE 802.11 MAC

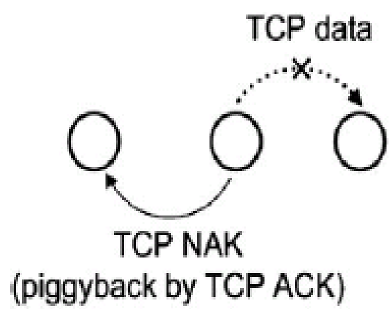
- If **RET_L > retry threshold**
  - If **RET_F < retransmission threshold** and **receipt of TCP ACK with the same flow**
    - Forward the TCP NAK piggybacks using the reverse TCP ACK along the end-to-end path
    - Increase the value of RET_F
  - Else discard the transmitted packet and then reset the contention window and RET_L
- Otherwise
  - Increase the contention window and than launch the back-off procedure
  - Increase the value of RET_L

RET_F: 0
retransmission threshold: n (default n = 1)
retry threshold: 7 for basic access mechanisms and 4 for virtual carrier-sensing mechanisms

Description of proposed method (cont.) Extension to IEEE 802.11 MAC

- The negative ACK (NACK) is triggered only when a frame is dropped as a result of transmission errors and is limited by the retransmission threshold

- If a TCP data frame is discarded
  - The MAC layer triggers the TCP NAK option in the TCP header associated with the sequence number of dropped packet
  - To piggyback the option using a reverse TCP ACK to notify the TCP sender



Services implemented in extended IEEE 802.11 MAC protocol

Description of proposed method (cont.) Modification to TCP

- To regulate the size of congestion window in the TCP layer based on the ACKs received from the receiver

- Receipt of new ACK
  - If **$W < W_t$**, set $W = W + 1$; slow-start phase
  - Else set $W = 1 + 1 / W$; congestion avoidance phase

- Receipt of NAK
  - Record the NAK sequence number, and then retransmit the "corrupted packet"

W: the current congestion window size
W_t: the slow-start threshold

Description of proposed method (cont.) Modification to TCP

- Receipt of duplicate ACK
  - Increment duplicate ACK count
  - When duplicate ACK count exceeds specified threshold value, retransmit “next expected packet
  - Set $W_t = W / 2$ and then set $W = W_t$ if the “next expected packet is not a “corrupted packet”
  - Resume congestion avoidance using new window size once retransmission has been acknowledged

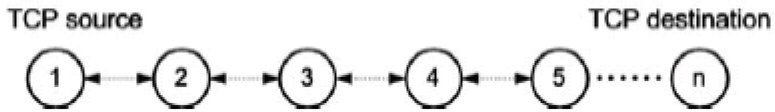
- Upon timer expiry
  - Set $W_t = W / 2$ and then set $W = 1$
  - Recover the “missing segments” from the slow-start phase

Numerical results

Parameter	802.11	802.11b	802.11g
SLOT	50 μs	20 μs	9 μs
SIFS	28 μs	10 μs	10 μs
DIFS	128 μs	50 μs	28 μs
PHY_{hdr}	128 bits	192 bits	192 bits
CW_{min}	32	32	32
CW_{max}	1024	1024	1024

Default parameters for MAC and physical layers

- Transmission range (TX_{range}): 40m
- Physical carrier-sensing range (PCSrange): 85m
- The height of antenna: 1.5m
- To operate in the 2.4GHz band
- The nodes are assumed to be static and to be separated from their immediate neighbors by a distance of 30m

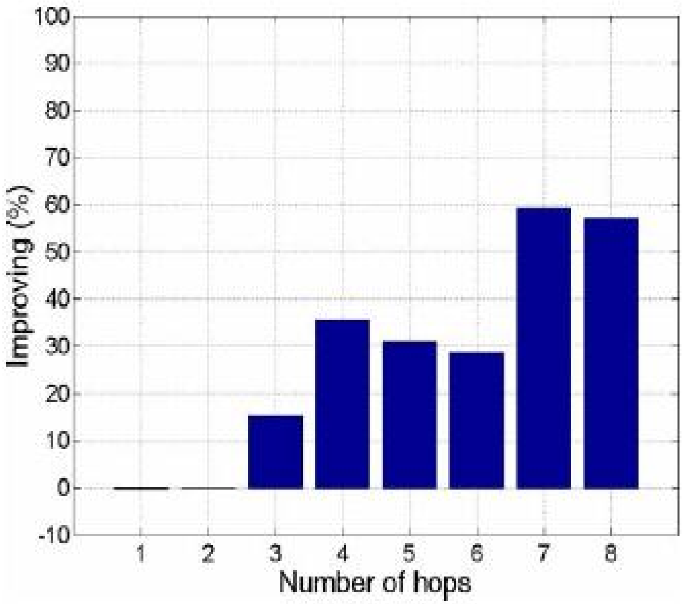


Chain network topology

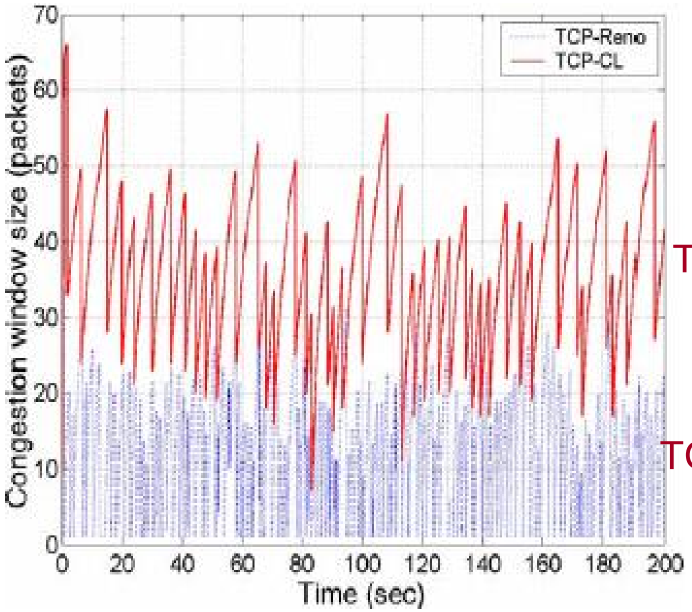
Numerical results (cont.)

Hop count	Goodput	
	TCP-Reno (Mbps)	TCP-CL (Mbps)
1	13.97	13.95
2	7.12	7.13
3	3.54	4.09
4	2.21	2.99
5	1.99	2.61
6	1.82	2.34
7	1.37	2.18
8	1.33	2.09

TCP goodput comparison



Goodput improvement obtained by TCP-CL in multi-hop scenario



Congestion window dynamics (in eight-hop path)

Numerical results (cont.)

- To use Gilbert-Elliot (GE) error model to mimic fading in the communication channel
- The frame error rate:
  - P_G: 0.0155, P_B: 0.25
  - P_{GG}: 0.94, P_{GB}: 0.06, P_{BB}: 0.85, P_{BG}: 0.15

Hop count	Goodput (Mbps)			
	TCP-Reno	TCP-CC	TCP-W	TCP-CL
1	12.70	12.74	13.02	12.87
2	6.32	6.26	6.33	6.32
3	2.72	2.76	2.76	3.67
4	1.53	1.59	1.66	2.30
5	1.19	1.28	1.29	2.02
6	1.03	1.12	1.14	1.86
7	0.83	0.85	0.84	1.66
8	0.73	0.80	0.76	1.54

TCP goodput comparison (with GE error model)

P_{GB}: The probability of the state transiting from a good state to a bad state
P_{GG}: 1 - P_{GB}

Numerical results (cont.)

- To use the delayed ACK at the receiver end
  - Delayed packet number , n = 2
  - Delayed time interval, t = 0.1s

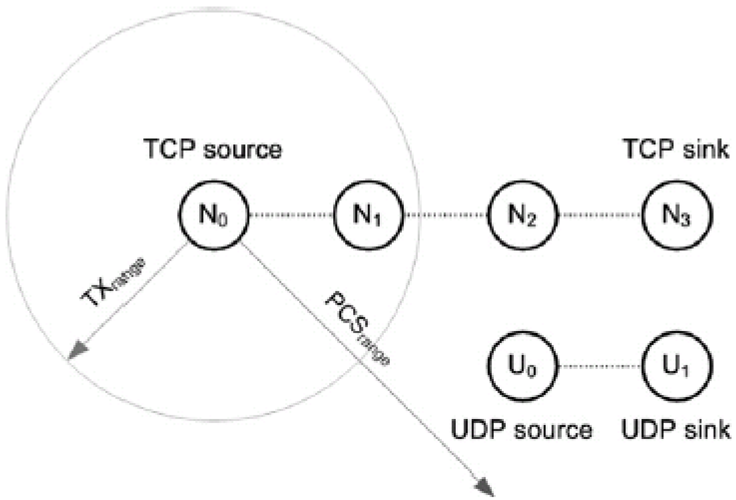
- The delayed ACK scheme alleviates the TCP self-collision problem in wireless link

Hop count	Goodput (Mbps)			
	TCP-Reno	TCP-CC	TCP-W	TCP-CL
1	14.21	14.27	14.25	14.28
2	7.12	7.16	7.14	7.16
3	3.34	3.34	3.25	4.15
4	1.98	1.93	2.19	2.69
5	1.57	1.53	1.63	2.31
6	1.38	1.38	1.43	2.06
7	0.95	0.97	0.95	1.88
8	0.85	0.93	0.91	1.79

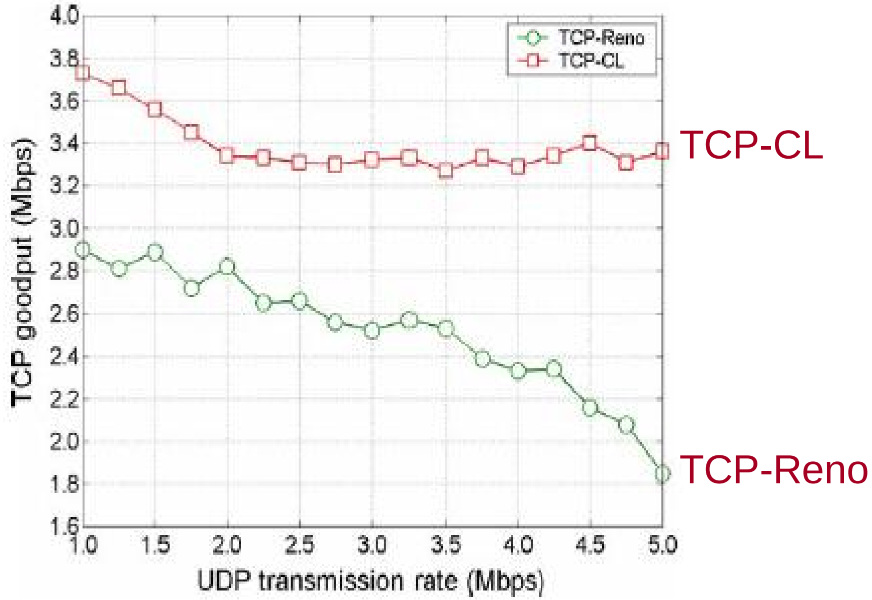
TCP goodput comparison (with delayed ACK)

Numerical results (cont.)

- One TCP connection and One UDP connection
- TX_{range}: 40m, PCS_{range}: 85m
- The nodes are assumed to be static and to be separated from their immediate neighbors by a distance of 30m
- UDP transmission rate: 1 to 5 Mbps



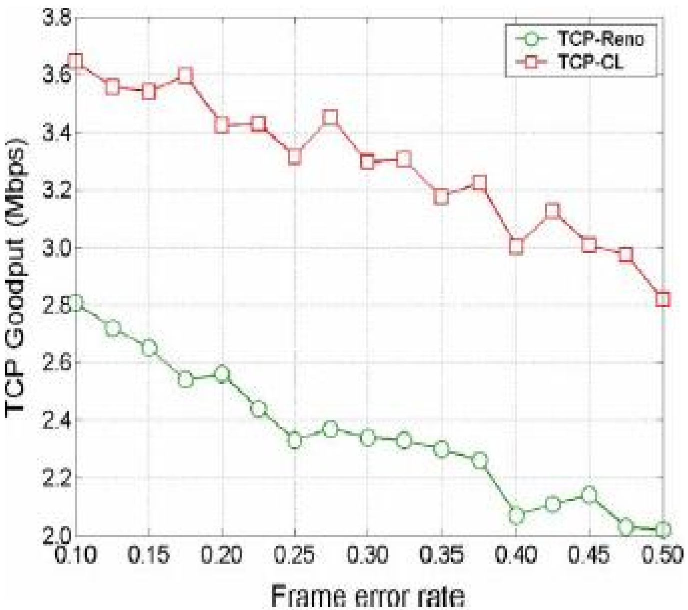
TCP evaluation topology2



TCP goodput comparison (UDP interference)

Numerical results (cont.)

- UDP transmission rate: 1Mbps
- The frame error rate
  - P_G: 0.015, P_B: 0.1 to 0.5
  - P_{GG}: 0.9, P_{GB}: 0.1, P_{BB}: 0.85, P_{BG}: 0.15



TCP-CL

TCP-Reno

Variation of TCP goodput with FER

Numerical results (cont.)

- A series of simulations is conducted on randomly generated ad hoc network topologies
- Network region: 100m x 100m
- Wireless station: 20
  - TX_{range}: 40m
  - PCS_{range}: 85m
- TCP connection: 2 to 9
  - TCP source and TCP destination of each connection are randomly selected

Numerical results (cont.)

Number of connection	Goodput (Mbps)			
	TCP-Reno	TCP-CC	TCP-W	TCP-CL
2	4.05	4.15	4.05	4.55
3	3.21	3.30	3.28	3.71
4	2.68	2.77	2.79	3.16
5	2.34	2.39	2.37	2.90
6	2.25	2.29	2.26	2.65
7	2.11	2.18	2.15	2.53
8	1.90	2.04	1.94	2.23
9	1.86	1.91	1.92	2.20

TCP goodput comparison (two hops)

Number of connection	Goodput (Mbps)			
	TCP-Reno	TCP-CC	TCP-W	TCP-CL
2	2.89	2.93	3.04	3.28
3	2.41	2.54	2.40	2.72
4	2.09	2.12	2.14	2.45
5	1.93	1.96	2.01	2.18
6	1.82	1.85	1.83	2.04
7	1.68	1.74	1.77	1.96
8	1.61	1.66	1.72	1.90
9	1.51	1.60	1.64	1.85

TCP goodput comparison (three hops)

Number of connection	Goodput (Mbps)			
	TCP-Reno	TCP-CC	TCP-W	TCP-CL
2	11.58	11.93	11.82	14.82
3	9.76	10.63	10.66	12.56
4	8.75	8.87	8.82	10.91
5	7.24	8.29	8.40	9.34
6	6.53	7.17	7.27	8.90
7	6.50	7.14	7.16	8.39
8	6.13	6.27	6.57	7.58
9	5.96	6.12	6.26	7.37

TCP goodput comparison (path of all lengths)

Conclusion

- The paper has proposed a cross-layer approach to enhance the performance of TCP in multi-hop wireless networks
  - The MAC layer and the transport layer
  - To provide explicit corruption loss information

- The proposed scheme has a number of advantages
  - A more efficient treatment on frequent transmission losses
  - A faster reaction to corruption losses
  - The ability to distinguish between congestion error and transmission error