SWINBURNE UNIVERSITY OF TECHNOLOGY

TCP/IP over IEEE 802.11b WLAN: the Challenge of Harnessing Known-Corrupt Data

Mattia Rossi

mrossi@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Abstract

- Transport protocols like DCCP and UDPLite make use of known corrupt packets
- Internet draft for SCTP exists to use this sort of packets
- TCP could make use of such packets to improve congestion control over lossy links (ELN)
- Measurements on 802.11 WLANs (typical lossy link) to check useability of ELN

Outline

Introduction
- Current Situation
- Solution Suggestions
- Test Case

System Description
- Corruption Detection and Notification
- Congestion Control Reaction to Corruption

Experimental Setup

Measurements
- Results
- Indoor Measurements
- Outdoor Measurements

Related Work

TCP in Wired Networks

- Low packet loss - only congestion related
- Good performance due to correct adaptation of transfer rate
- TCP congestion control was designed for wired networks
- Congestion control mechanisms are unaware of underlying technology

TCP in Wireless Networks

- High packet loss dependent on signal strength - non congestion related
- Many unnecessary congestion window reductions - bad performance
- No congestion control is designed for lossy (wireless) links
- TCP performance over wireless links can be improved

Performance Enhancing Proxies - PEP

- List of methods described in RFC 3135
- Transport layer proxies - application layer proxies
- Some suggest TCP spoofing - splitting the TCP connection into a local and a public connection
- Berkeley's SNOOP protocol is a PEP for WLANs (may also use ELN)

Explicit Loss Notification - ELN

- Information about type of packet loss has to be advertised distinguish congestion losses from corruption losses
- Adapt congestion control mechanisms to type of loss
- Corrupt packets are usually a sign of packet loss due to link problems (corruption notification)
- Various studies about ELN exist, most of them only used simulated environments (TCP_ELN, TCPHACK)

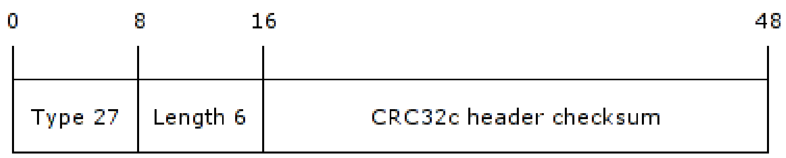
ELN in TCP over real WLAN Links

- Measurements about how often corrupt packets reach the transport layer with disabled link layer control
- Measurement of performance gain in TCP
- Using NewReno, SACK, corruption detection and corruption notification
- Using only a single direct connection in ad-hoc mode
- Results may also apply to DCCP and UDPLite (and SCTP)

TCP Corruption Detection Option

An IETF draft exists with the proposal to add corruption notification to TCP. The draft uses two TCP options to provide the necessary information:

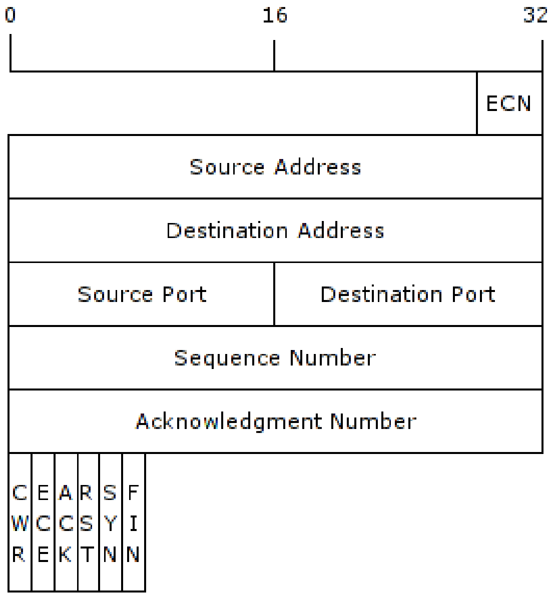
- The corruption detection option (CDO) contains a 32 bit CRC32c checksum over a pseudo header



TCP Header Checksum

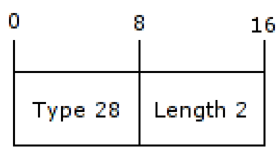
- Normal TCP checksum covers header and payload
- If packets are corrupt, TCP checksum fails and packet is discarded
- TCP header checksum should be tested if TCP checksum fails
- If header checksum is Ok, a corruption notification to the correct sender
- IP checksum also has to be Ok
- No authentication and encryption possible at link and network layer (WEP, WPA, IPSec)

TCP Pseudo Header for CDO



TCP Corruption Notification Option

- The corruption notification option (CNO) is small
- The CNO only contains the minimal necessary information to inform the source that a non-congestion-related packet loss occurred



Effects on Congestion Control

- Three sequences are defined:
  - Recalculate the RTT (Round Trip Time) and RTO (Retransmission Timeout) values
  - Retransmit selectively the lost segment
  - Update the congestion window size (cwnd)
- First two sequences are defined, third one is still a discussed issue

The Congestion Window Update Problem

- Standard TCP reacts to loss by halving cwnd
- Bad (wrong) reaction to non-congestion-related loss
- Little benefit for the network (none at all?)
- Right reaction might be not to reduce rate at all (TCPHACK)
- No agreement in IETF about that, as congestion on multiple connections over wireless links can create corrupt packets

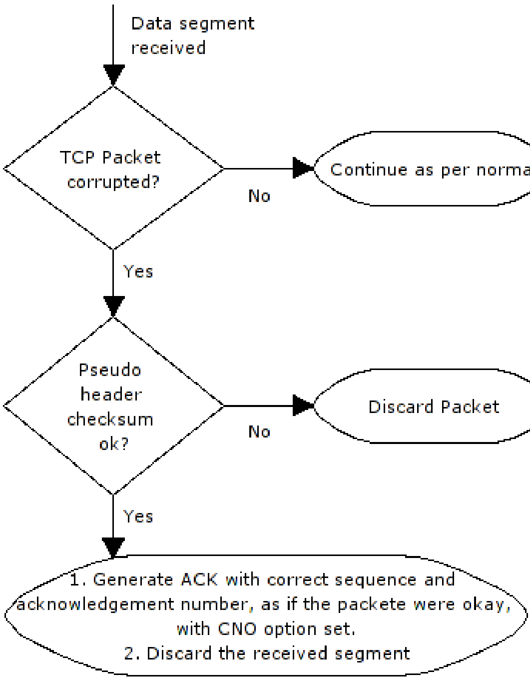
The Congestion Window Update Problem

- The CDO/CNO internet draft does not define how to update cwnd
- Neither does DCCP
- One idea is to react similarly to congestion-related loss
- Conservative, but has still benefits:
  - correctly updated RTO
  - earlier retransmission
  - retained information for connection setup or teardown,
  - possibility to react to congestion earlier when congestion is explicitly indicated via the “ECN-Echo” flag.
- More investigation at link layer level needed

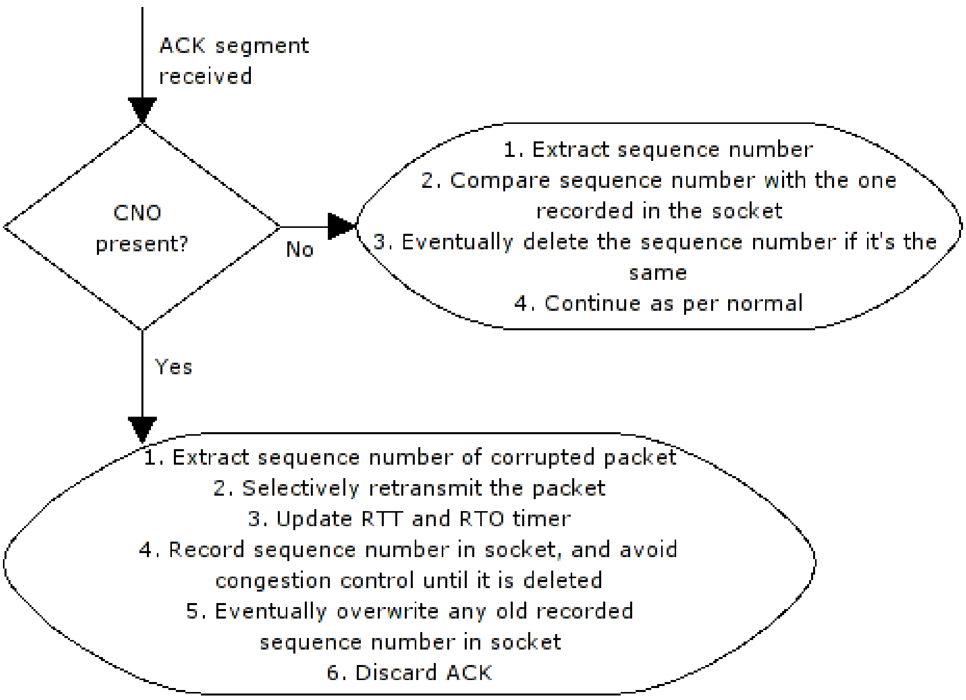
Equipment

- Hardware: WLAN card with Prism2.5 chipset (sender and receiver), WLAN card with Broadcom chipset (sender only)
- Software: Linux Kernel 2.6.17, HostAP WLAN driver for Prism2.5 based cards, Broadcom closed source Windows drivers via ndiswrapper for Broadcom based card
- Tweaks on HostAP driver:
  - Enabling promiscuous mode which allows reception of corrupt frames at link layer without switching off sending capability
  - Forward corrupt packets to the TCP/IP stack even if packet is corrupt
  - Count corrupt packets at arrival
  - Count total amount of sent and received packets

Flowchart CDO



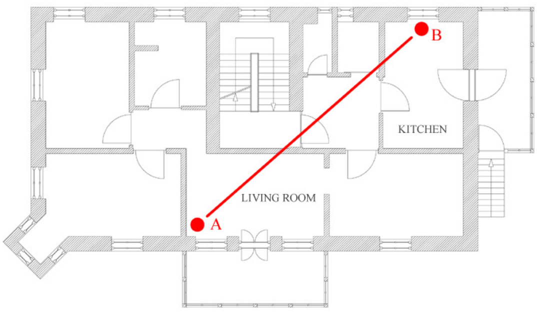
Flowchart CNO



Measurement proceedings

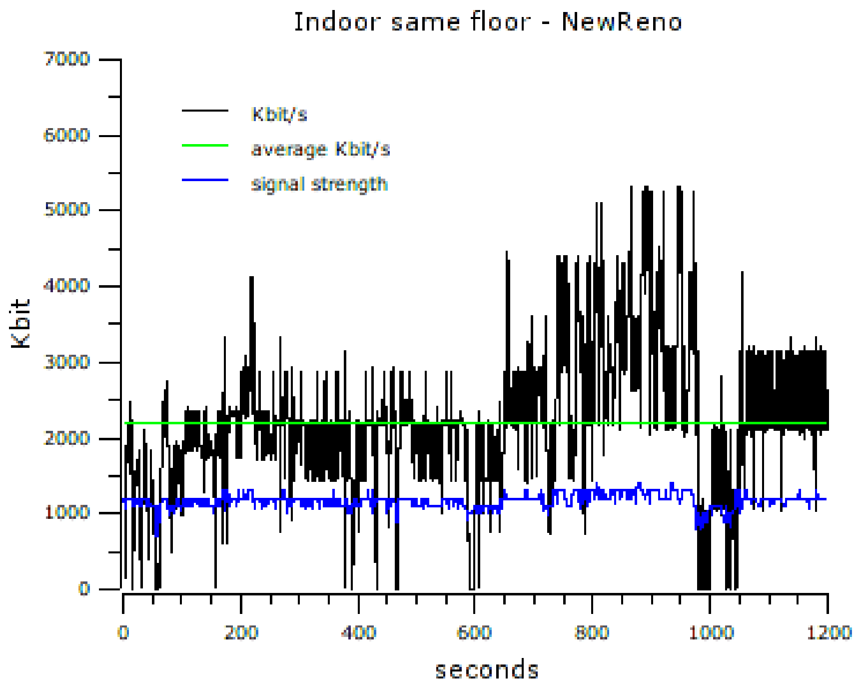
- Four different setups with four measurements of 20 min. each:
  - Indoor to indoor same floor, indoor to indoor different floor, indoor to outdoor, outdoor to outdoor
  - TCP with NewReno only, NewReno + SACK, NewReno + CNO, NewReno + SACK + CNO
- Iperf used to measure transferrate, packet status is logged to linux system log, signal strength is recorded
- Produced graph shows the stability of the TCP transferrate over the whole measurement
- Average transferrate is inserted

Indoor Measurements

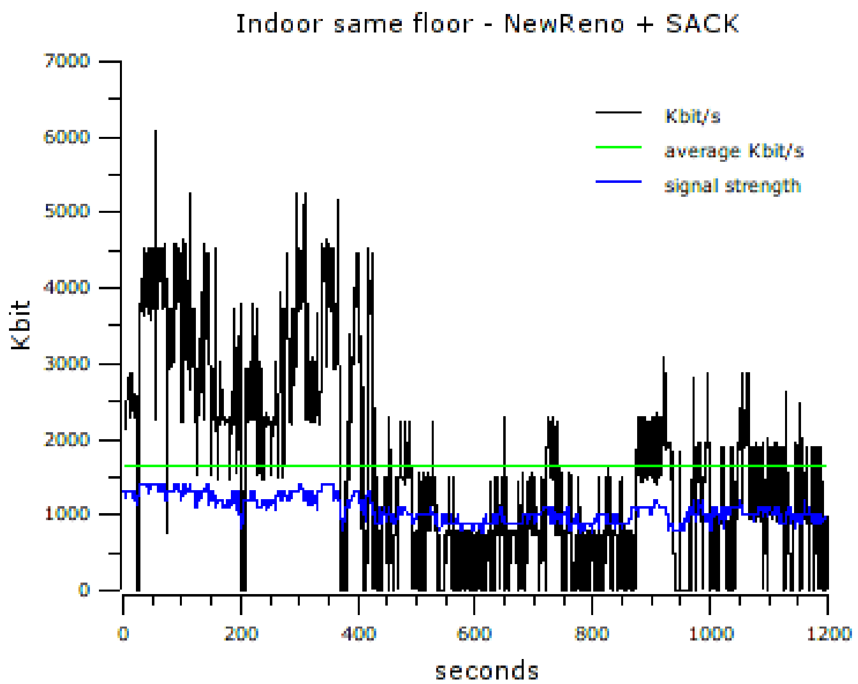


Indoor to indoor -same floor - measurement setup

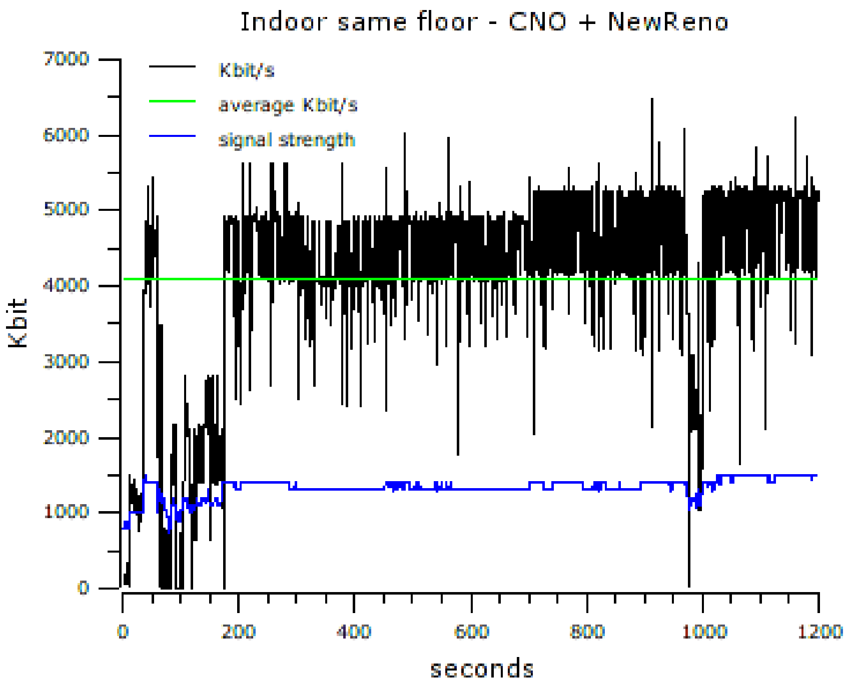
Indoor Measurements



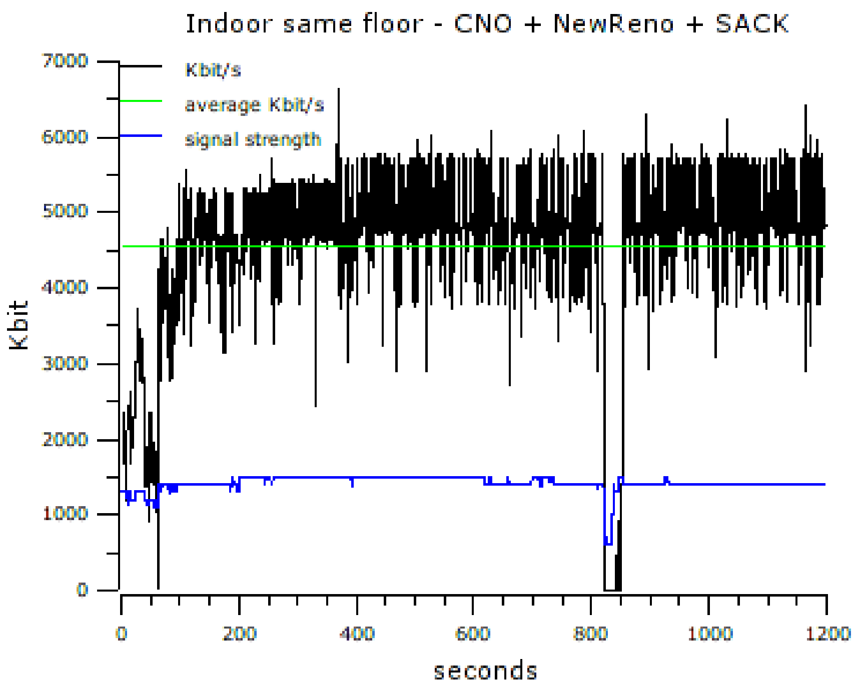
Indoor Measurements



Indoor Measurements



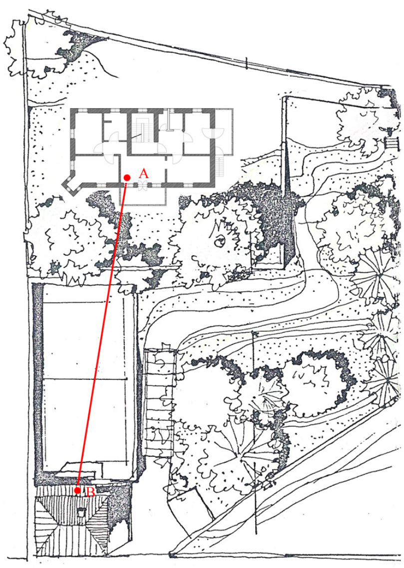
Indoor Measurements



Indoor Measurements

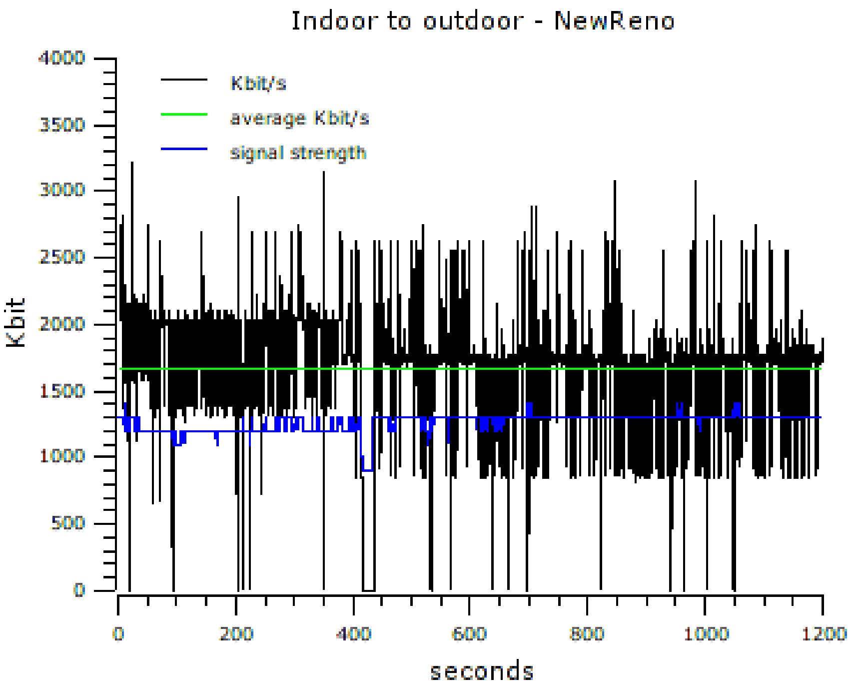
	CNO		no CNO	
	NewReno	SACK	NewReno	SACK
Tx packets	**448318**	**497769**	**336058**	**254395**
Rx packets	**430254**	**478987**	**228324**	**171933**
Lost packets	**18064**	**18782**	**107734**	**82462**
Corrupt packets	**632**	**178**	**1435**	**722**
Corrupt payload	**16**	**14**	**0**	**0**

Outdoor Measurement

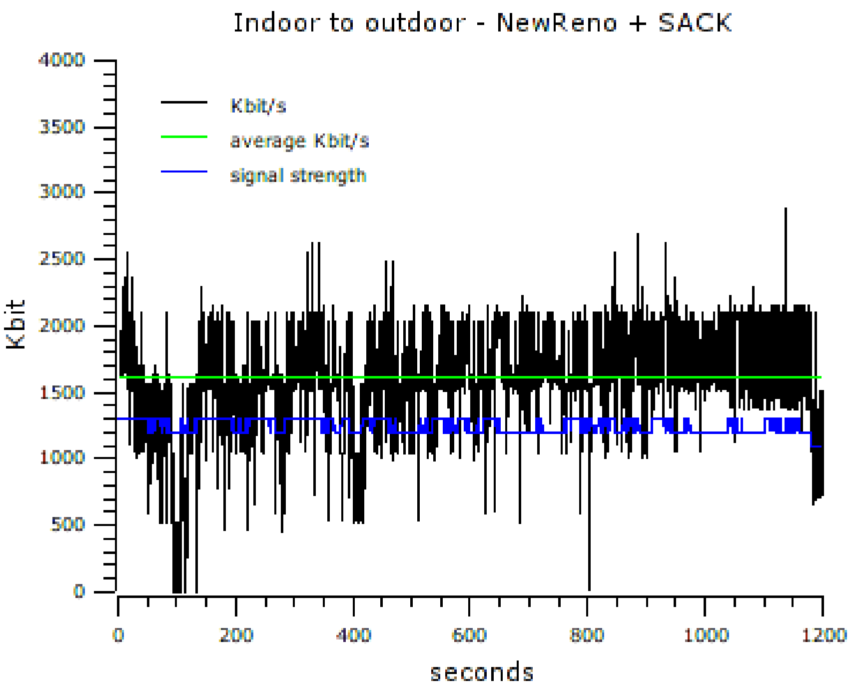


Indoor to outdoor measurement setup

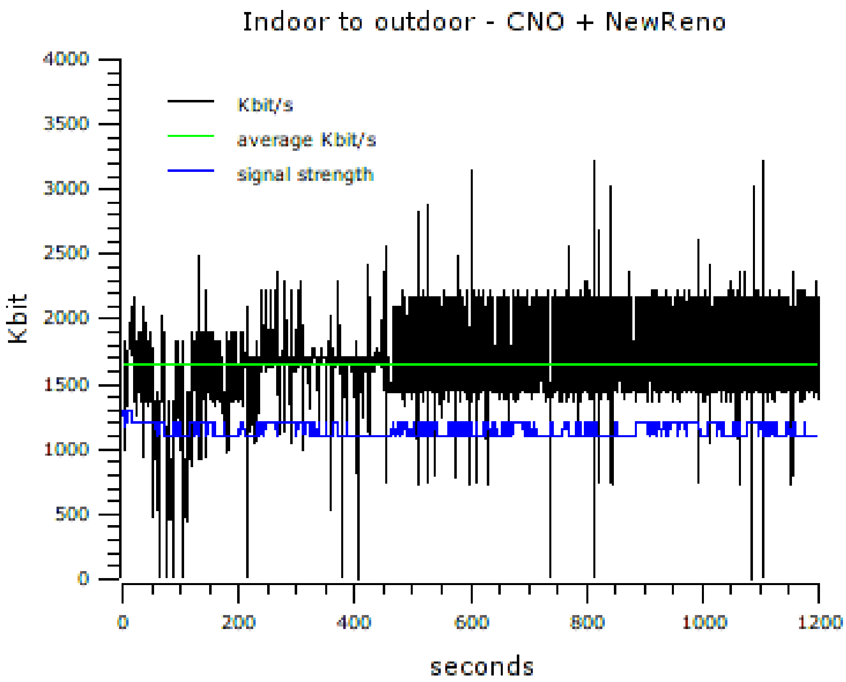
Outdoor Measurement



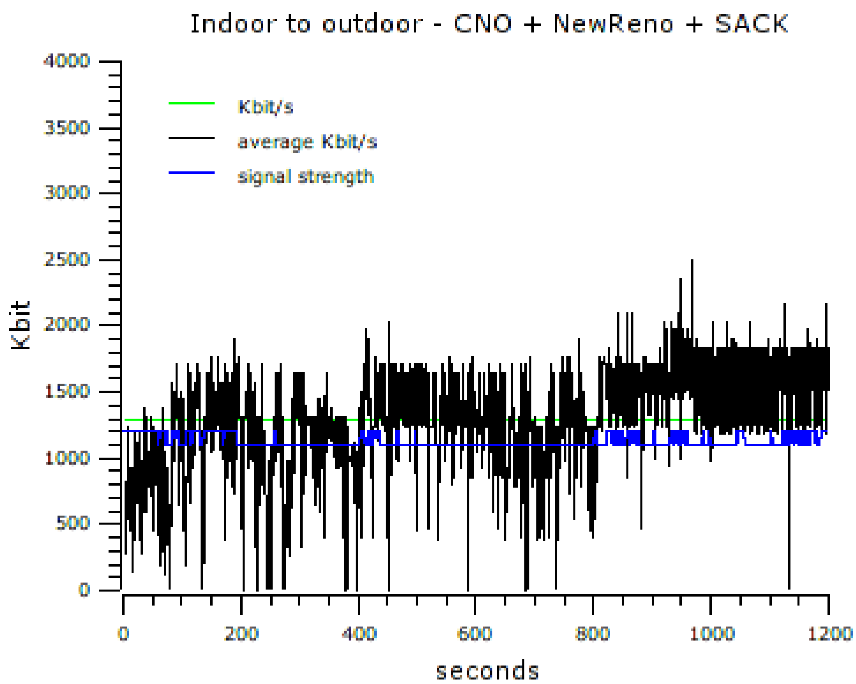
Outdoor Measurement



Outdoor Measurement



Outdoor Measurement



Outdoor Measurement

	CNO		no CNO	
	NewReno	SACK	NewReno	SACK
Tx packets	**258324**	**246158**	**263442**	**253713**
Rx packets	**174651**	**135941**	**175170**	**169781**
Lost packets	**83673**	**110217**	**88252**	**83932**
Corrupt packets	**1475**	**2081**	**1277**	**1479**
Corrupt payload	**109**	**93**	**0**	**0**

Related Work

- Additionally to TCP-ELN, TCPHACK, DCCP and UDPLite there are a lot of other related efforts to exploit corrupt data on transport layer
- Results depend mainly on the link layer technology used
- Similar research for GPRS showed much better results
- Other approaches show better results
- LT-TCP for example uses forward error correction to improve troughput

Conclusions

- No improvement for IEEE 802.11 WLAN, packet loss too high, CNO triggers too low
- Possible improvements on other wireless link layer technologies (e.g. UMTS, GPRS)
- Better results on lower link layer connection speed (explains good results on GPRS?)
- Interferences happen for certain periods of time
- Higher connection speed means higher modulation rate and equals to higher packet loss
- Hardware and drivers could have affected the outcome

Questions

THANK YOU FOR YOUR ATTENTION
Questions?