Enhancement of the TCP Module in the OMNeT++/INET Framework

UNIVERSITÄT DUISBURG ESSEN

Motivation

- The Internet
  - is the largest computer network in the world
  - most of the Internet traffic is handled by the Internet Protocol (IP) and the Transmission Control Protocol (TCP)
  - the evolution of TCP is still ongoing and there exist a variety of different TCP algorithm variants
- The INET Framework for OMNeT++
  - includes IP-based simulation models e.g. IP, UDP, TCP, SCTP
- The TCP Module
  - used a fixed TCP header length of 20 bytes and options (e.g. SACK) were not supported
  - The **SACK option** is part of common TCP implementations and is – according to RFC 4614 – a "recommended enhancement" for TCP
  - The following **Flow Control** related parts were missing:
    - A finite receive buffer size was not modelled
    - The data receiver was always offering the maximum receiver Window size
    - The data receiver was not able to send a Zero Window
    - The Persist Timer was missing
    - A function to send a Window Probe was implemented but has never been invoked

Enhancements

- Header Extension
  - Segment format has been changed to support **TCP options**
- SACK Option
  - Only if both nodes send a **SACK Permitted** option during connection setup, SACK (RFCs 2018, 2883) will be enabled
  - **SACK-based Loss Recovery** algorithm (RFC 3517) has been integrated into TCPReno
- Flow Control Enhancements
  - Added missing parts of the **Flow Control** mechanism
- TCPDump Module (to capture PCAP files)
  - has been extended to convert the TCP segment format of INET (including the newly implemented TCP options) from/to binary (network byte order) TCP segments

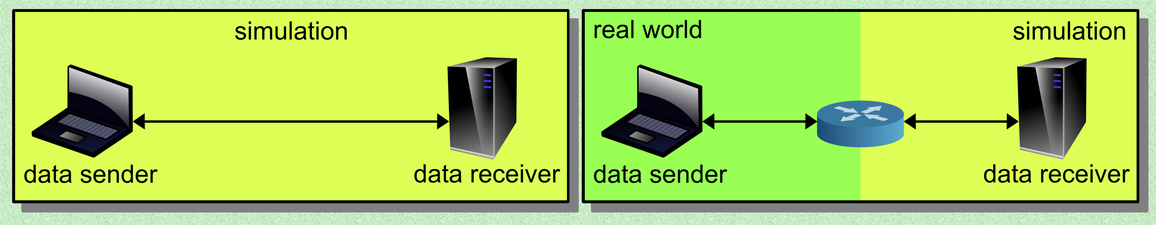
  → Allows for **TCP signalling analyses** with external tools
- ExtInterface Module (to connect network nodes)
  - has been extended to support TCP communications

  → Supports **connecting** the **TCP module with real, external TCP implementations**
- Further Enhancements
  - RFC 3042 – **Limited Transmit** (Loss Recovery algorithm)
  - RFC 3390 – Increasing TCP's **Initial Window**
  - RFC 3782 – **NewReno** (Loss Recovery algorithm)

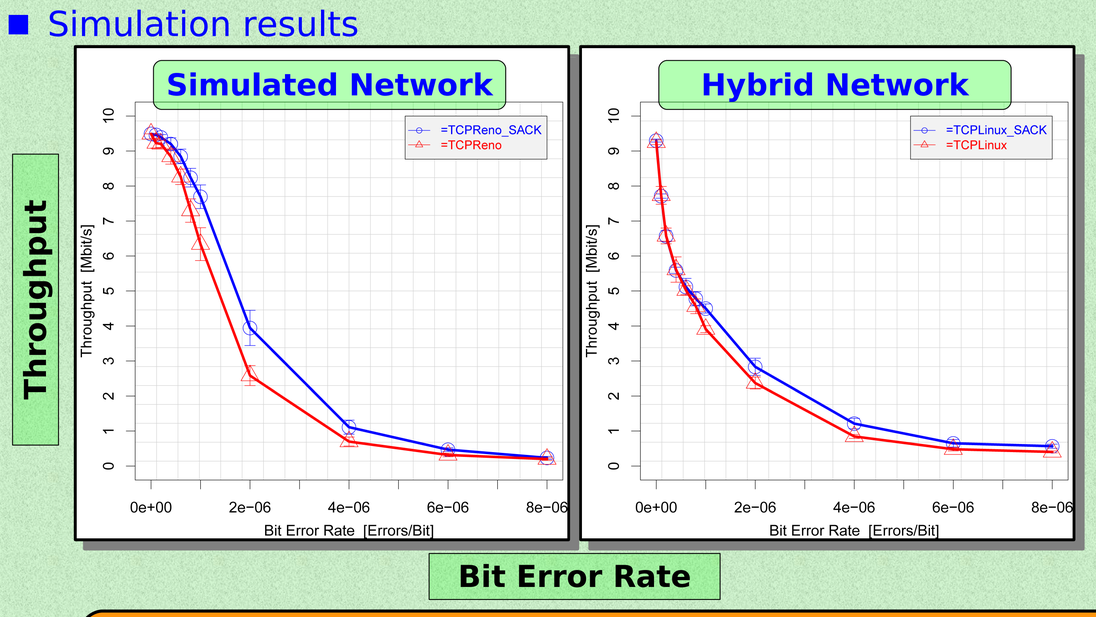
TCP Module

Performance

- Network setup
  - MTU = 1500B
  - MSS = 1452B
  - user data = 100MiB
  - test duration = 60s
  - data rate = 10Mbit/s
  - delay = 0.565µs
  - **bit error rate = variable**

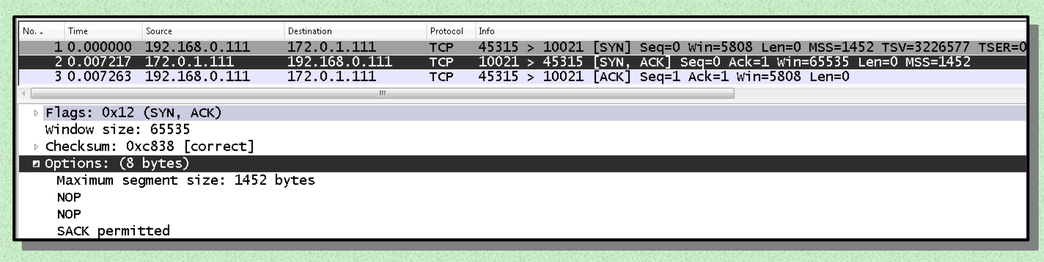


- Simulation results

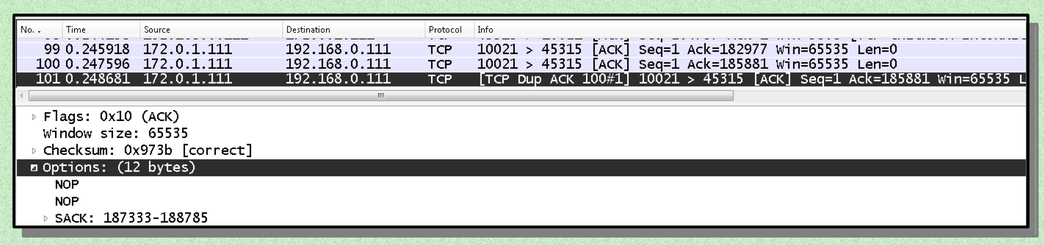


Interoperability

- **Captured TCP signalling logs** demonstrate the interoperability of the enhanced TCP module with real TCP implementations (e.g. Linux)



- The SYN,ACK segment contains a **SACK-Permitted** option to enable SACK



- Duplicate ACKs contain **SACK options** to allow TCP sender to recover more effectively when multiple segments are lost

- The enhanced TCP module has already been **integrated** into the INET framework and is **accessible** at:

  http://github.com/inet-framework/inet

Some important features of modern TCP implementations – particularly SACK and a complete Flow Control – have been added to INET's TCP module

Our Research

University of Duisburg-Essen

Flow Routing

- Research
  - QoS Device Concept
- Contributions to Standardization
  - IETF (Flow Identification)
  - ITU-T and ETSI (QoS Signalling)

SCTP

- Evaluation, Optimization and Improvement
  - Concurrent Multipath Transfer (CMT)
  - Performance and Security
- Open Source Implementation *SCTPLIB*
- Contributions to IETF Standardization
  - Secure-SCTP (Individual Submission)
  - TLS over SCTP (RFC 3436)

RSerPool

- Evaluation, Optimization and Improvement
- Various Contributions to Major Conferences
- Open Source Implementation *RSPLIB*
- Contributions to IETF Standardization
  - RFC 5351, RFC 5356, RFC 5525
  - Multiple Internet Drafts

Thomas Reschka, T.Reschka@gmx.de