



Flow-based TCP Connection State Detection

Tobias Limmer

Computer Networks and Communication Systems
Department of Computer Science
University of Erlangen-Nürnberg

limmer at informatik.uni-erlangen.de

30.10.2008



Introduction



- Essential question:
How can we distinguish successful from unsuccessful TCP connections in flow data?
- Interesting for security-related analysis
 - ➔ attack detection
 - ➔ malicious use
 - ➔ QoS
- very hard problem, even for packet-based analysis!



Flow Fields



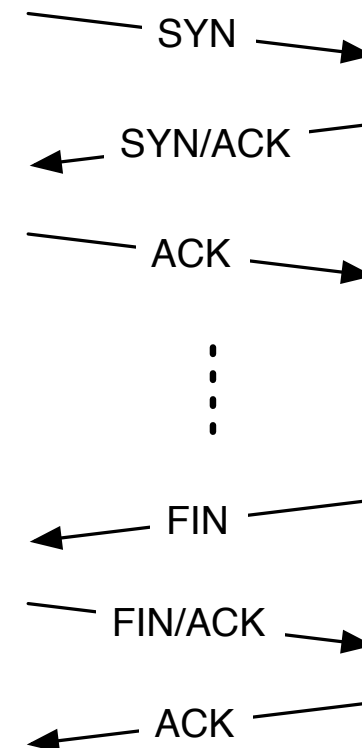
- 5-tuple fields

- ➔ srcIP, dstIP, srcPort, dstPort, protocol:
for connection identification

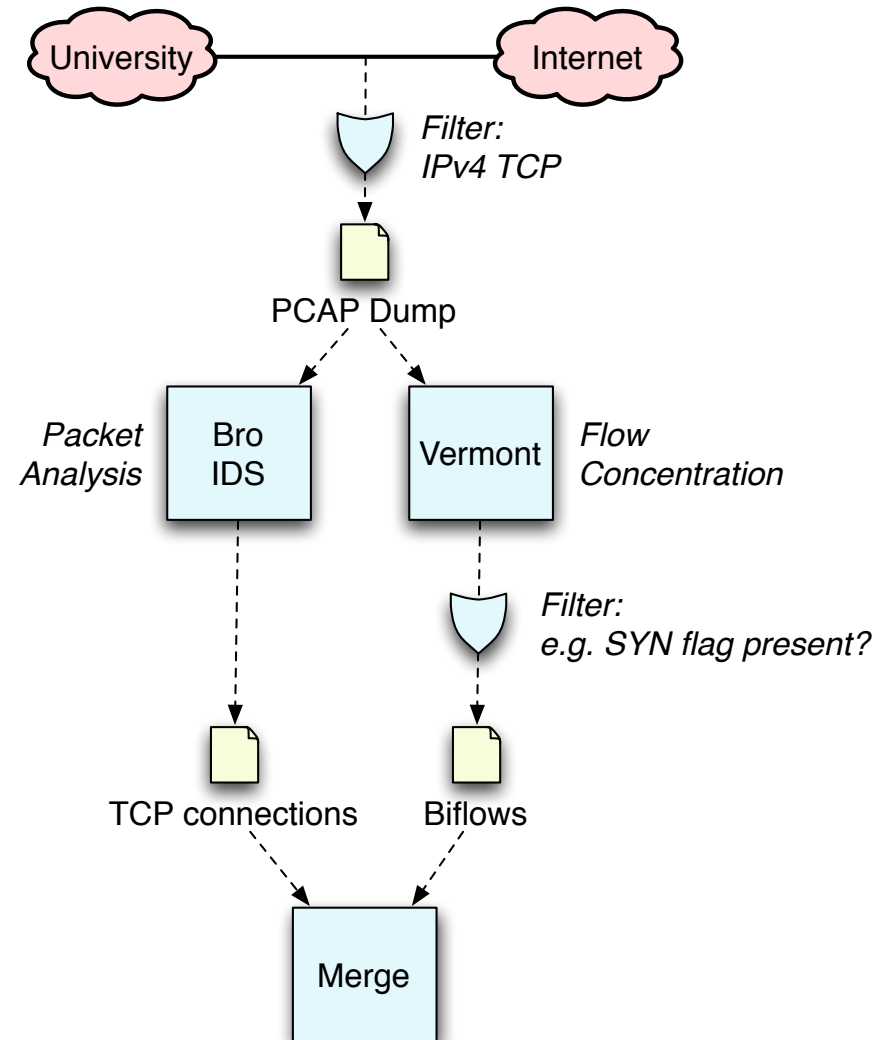
- aggregated/non-key fields
(in both directions)

- ➔ start/end flow time
- ➔ packet count
- ➔ byte count
- ➔ TCP flags

*TCP flags for successful
connection:*



- Source: network traffic of Uni Erlangen's internet connection
 - 2 hours
 - 290 MBit/s, 48 000 pkts/s, 770 biflows/s (only TCP)
- Flow concentrator: Vermont
 - timeouts: 1h active, 5min idle
- TCP state analyzer: Bro IDS
- Custom merging script
 - assigned bro results to biflows

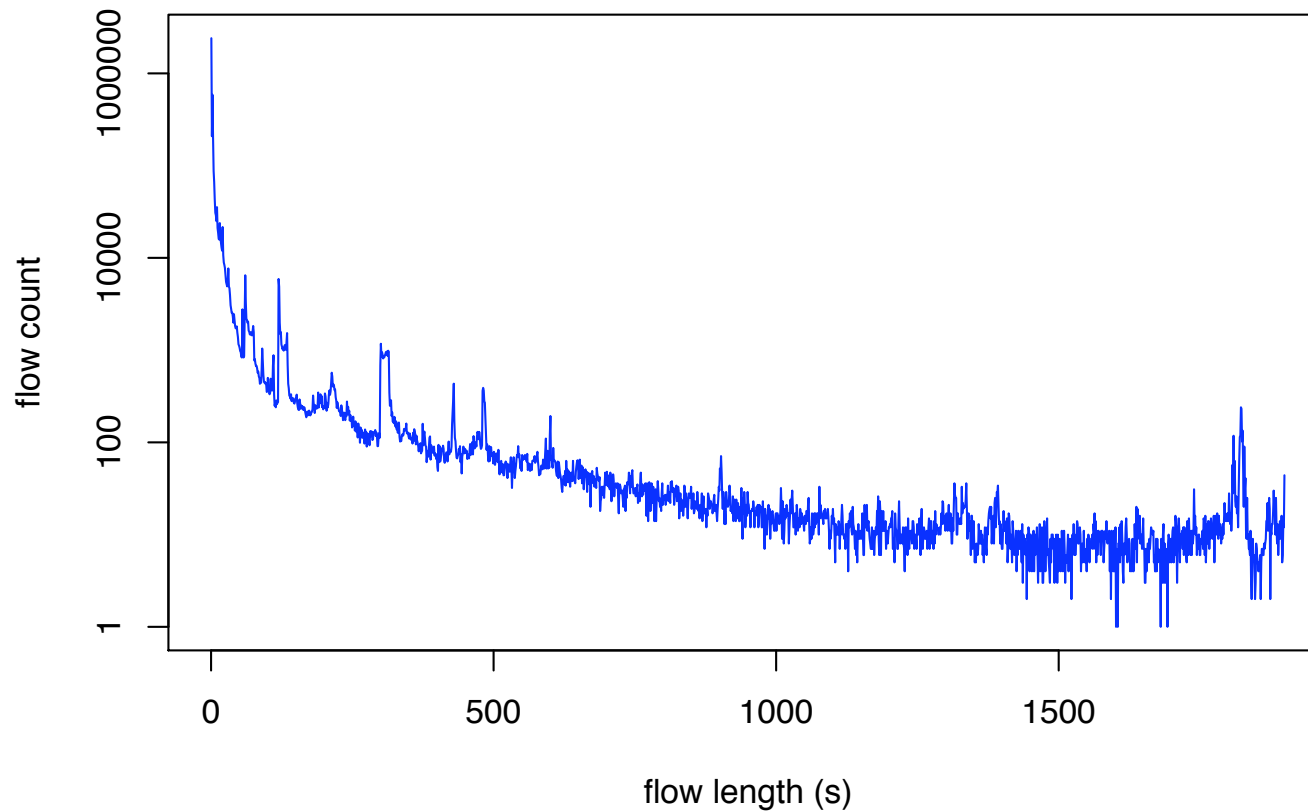




Capturing whole TCP connections



- new test setting: idle/active timeout: 7200s
- connection length of successful TCP connections:

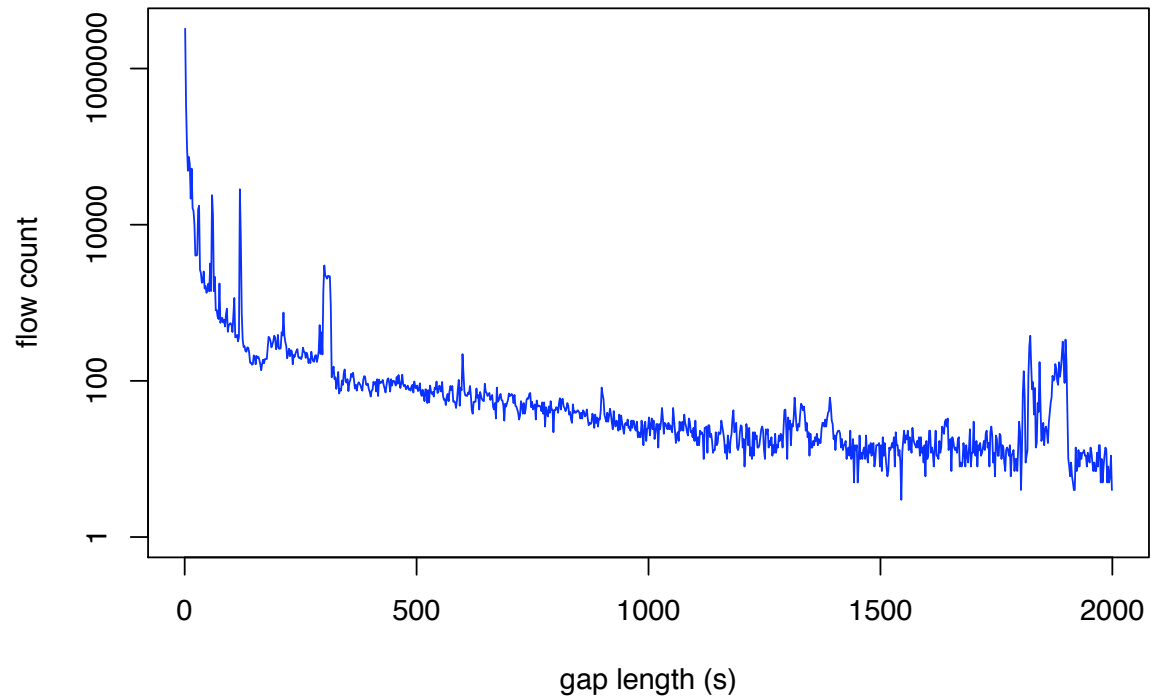
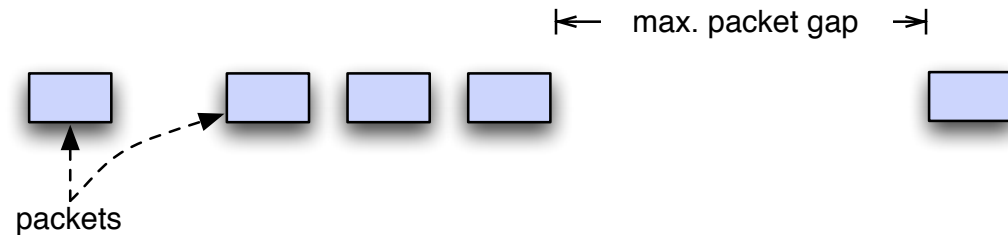




Capturing whole TCP conn. 2



- What about idle timeouts?
 - ➔ Maximum packet gap in TCP connections!
- to capture X% conns, idle time-out needs to be Y seconds
 - ➔ 95% -> 16s
 - ➔ 98.5% -> 153s
 - ➔ 99.9% -> 1909s

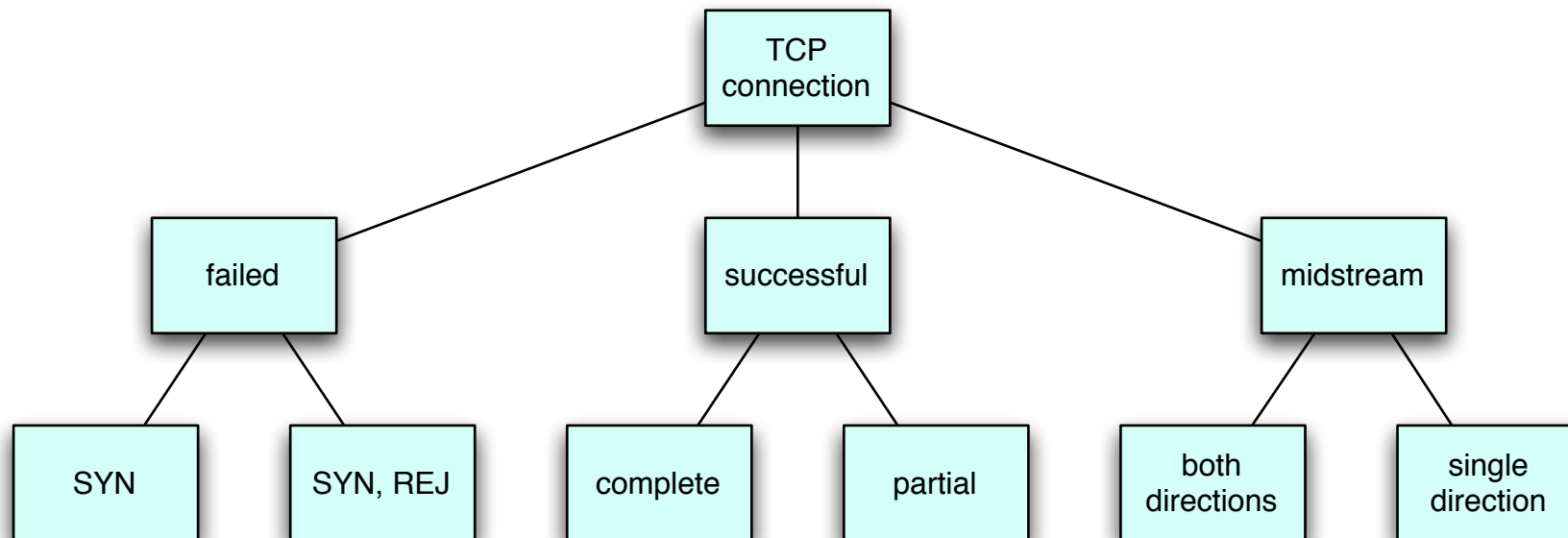




TCP State Analysis



- possible states:

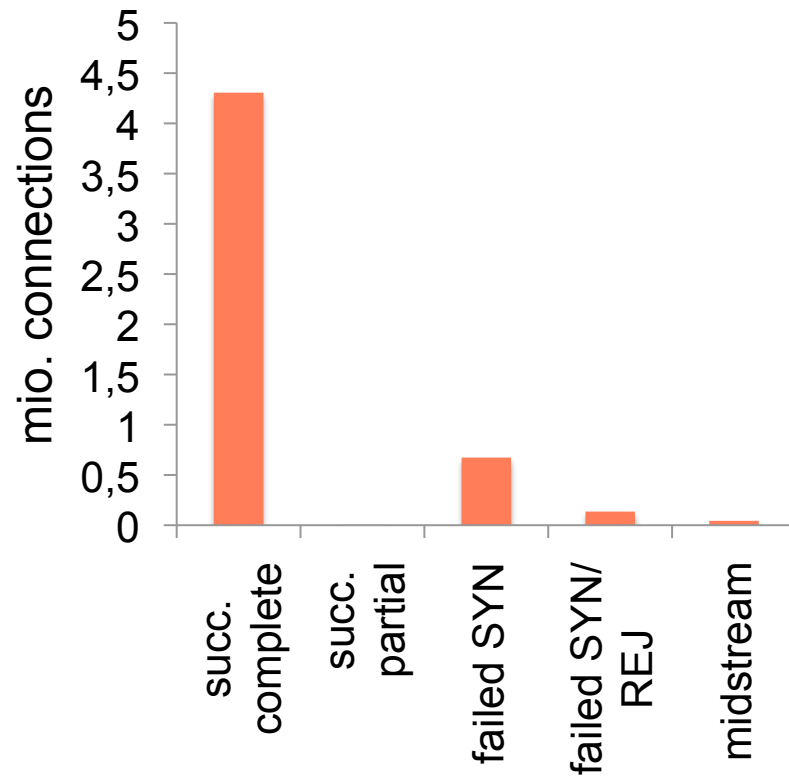




Results

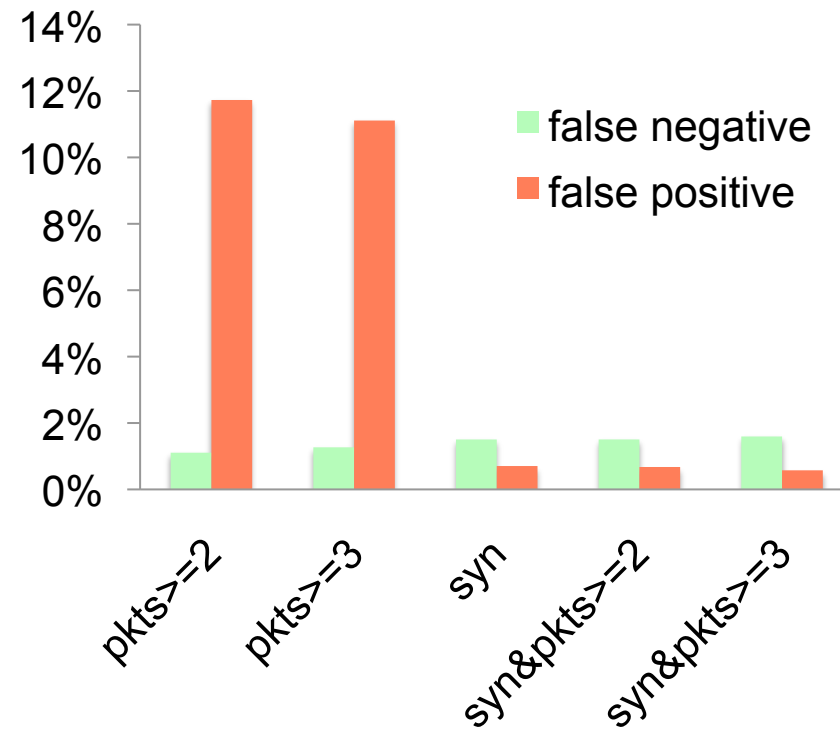


- Amount of classified connections:



- Filtered flows

➔ relative to “real” amount of succ./ failed connections





Conclusion



- results are moderate
 - ➔ but was to be expected
- still to do:
 - ➔ sampled data
 - ➔ include other relevant non-key fields
 - ➔ UDP
 - ➔ evasion techniques



Conclusion



anonymized flow dumps? anyone?



The End



Thanks for your attention!

Questions?

