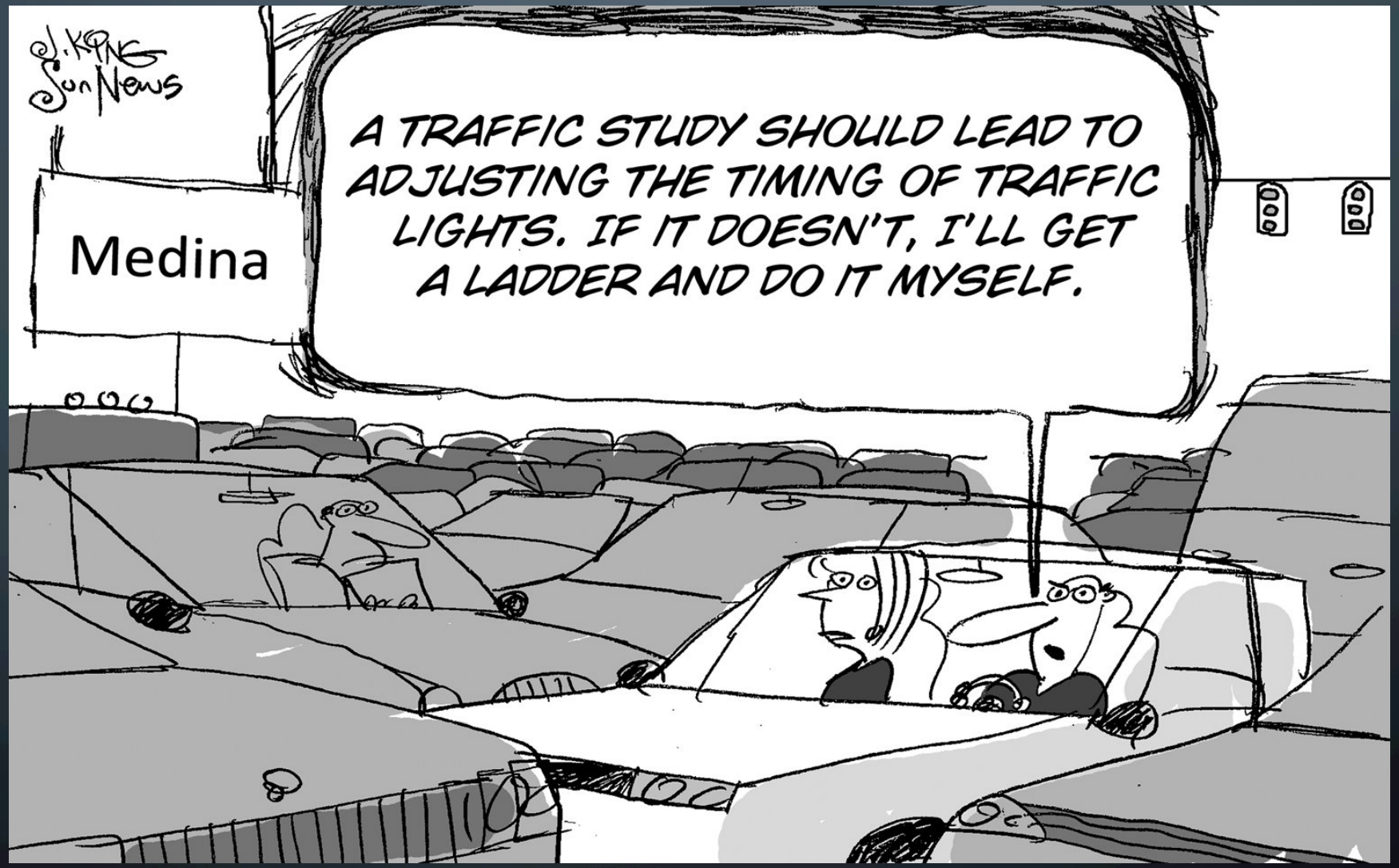
DESIGNING DCCP: CONGESTION CONTROL WITHOUT RELIABILITY

PAPER BY: EDDIE KOHLER, MARK HANDLEY, SALLY FLOYD

PRESENTED BY: ABRAR SALMAN

UNIVERSITY OF WATERLOO, 20490103



From Cleveland.com, By Sun News staff

OVERVIEW

- In the past few years, the use of internet applications has increased tremendously
- As we know, applications either prefer reliability (TCP) or timeliness (UDP)
- TCP is a great reliable protocol but lacks timeliness
- UDP is great for large data but provides no reliability

INTRODUCTION

- The idea of using some aspects from TCP and UDP and put them together to form a new protocol that provides congestion control from TCP and unreliability from UDP
- Applications like streaming media, video conferencing prefer timeliness over reliability.
- DCCP was introduced to solve this matter

INTRODUCTION, CONT'D

- TCP is not a good fit for 2 reasons:

  1- Congestion Control limits performance

  2- Congestion Control is not easy to implement

- DCCP designers found that providing unreliability is easier than providing reliability

DCCP OVERVIEW

- It is a unicast, connection oriented with bi-directional data flow
- Just like TCP, it starts and end the connection with 3 way handshake
- Header size is 16 bytes

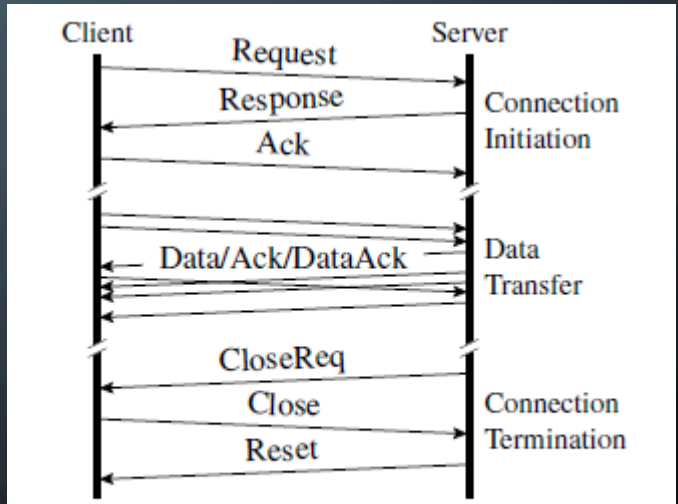


Figure 1: DCCP packet exchange overview.

GOALS OF DCCP

- DCCP's desire was to provide generality and minimality

1- Minimalism:

Minimal functionality means that it goes in-line with end to end processes, while the minimal mechanism provides the DCCP with more than one feature to solve more than one problem at the same time

GOALS OF DCCP, CONT'D

- 2- Robustness: is the robust behaviour of the protocol in the presence of attacks. It does not however provide cryptographic guarantees, attackers must know the initial sequence numbers to get some probability of success.

So, there was a way to provide security to an unreliable protocol like not including network addresses in packet payloads or cryptographically signed data

GOALS OF DCCP, CONT'D

- 3- A framework for modern congestion control

DCCP is to support various applications not limited to file transfer but telephony and streaming media too. It provides congestion control, Selective Acknowledgement SACK, Enhanced Congestion Notification ECN, and acknowledgement verification.

DCCP provides a framework that can be implemented on multiple applications and not a single fixed algorithm

GOALS OF DCCP, CONT'D

- 4- Self-sufficiency

DCCP implementation is to be able to manage congestion control without the aid of applications

Senders should calculate the right sending rates, while receivers should be able to detect congestions without application intervention

GOALS OF DCCP, CONT'D

- 5- Support timing-reliability tradeoffs:

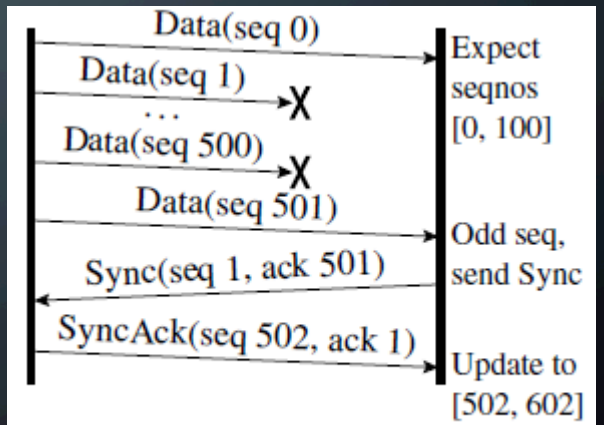
DCCP supports buffering to smooth out scheduling bumps. But when the buffer gets overwhelmed, a smart application should detect which packets has priority over the others, for example audio is preferred over video or the new packets over the old ones

SEQUENCE NUMBERS

- DCCP header include sequence numbers
- Sequence numbers measures datagrams not bytes because unreliable applications send datagrams instead of portions of bytes
- DCCP contains acknowledgement numbers which in turn reports the latest packet received

SYNCHRONIZATION

- What happens when a network gets interrupted?
- Because DCCP does not support retransmission, any packet sent out during the outage will receive a new sequence number
- Now, eliminating the old sequence number can cause losing the main line of network defence
- The receiver receiving a packet with a different sequence number than expected, will notify the sender to confirm the sequence number to SYNC numbers on both ends



(a) Synchronization

ACKNOWLEDGMENT

- TCP acknowledges every packet sent
- DCCP does not support retransmission, so the DCCP will have to compromise a little efficiency to support timeliness-reliability tradeoff concept
- DCCP Acknowledgement number reports packets received, processed and valid and enqueued its data for future delivery ?
- And, Data Dropped option indicates the acknowledged packet's data was not delivered because it was dropped in the receiver buffer

ROBUSTNESS

- DCCP may use a 24 bit sequence number, making it easier to inject data into the stream (this can be considered a disadvantage)
- This can be avoided from the application's side not to ask for short sequence number, making the injection less dangerous

CONNECTION MANAGEMENT

- Many applications uses single bidirectional connection like in streaming media, most of the data sent from server to client, after the initial setup all client's packets are acknowledged
- DCCP overcomes this issue by dividing the connection into two half-connection; one contains data packets when the other carries acknowledgements

CONGESTION CONTROL

- DCCP uses either CCID 2 or CCID 3 to overcome the congestion issue
- CCID 2 is a TCP-like congestion control with the difference that TCP does not enforce any congestion control on acknowledgments coming from the other end point
- CCID 3 uses LOSS INTERVALS that calculates the loss events rate

CONCLUSION

- Designing an unreliable protocol acts like TCP is not an easy thing to do
- DCCP features sounds promising
- Some NATs and other application does not use DCCP as a main protocol

QUESTIONS?