Can you fool me? Towards automatically checking protocol gullibility

Milan Stanojević
Todd Millstein
UCLA

Ratul Mahajan
Madanlal Musuvathi
Microsoft Research

Protocol gullibility

Gullibility

tendency to believe too readily and therefore be easily deceived [thefreedictionary.com]

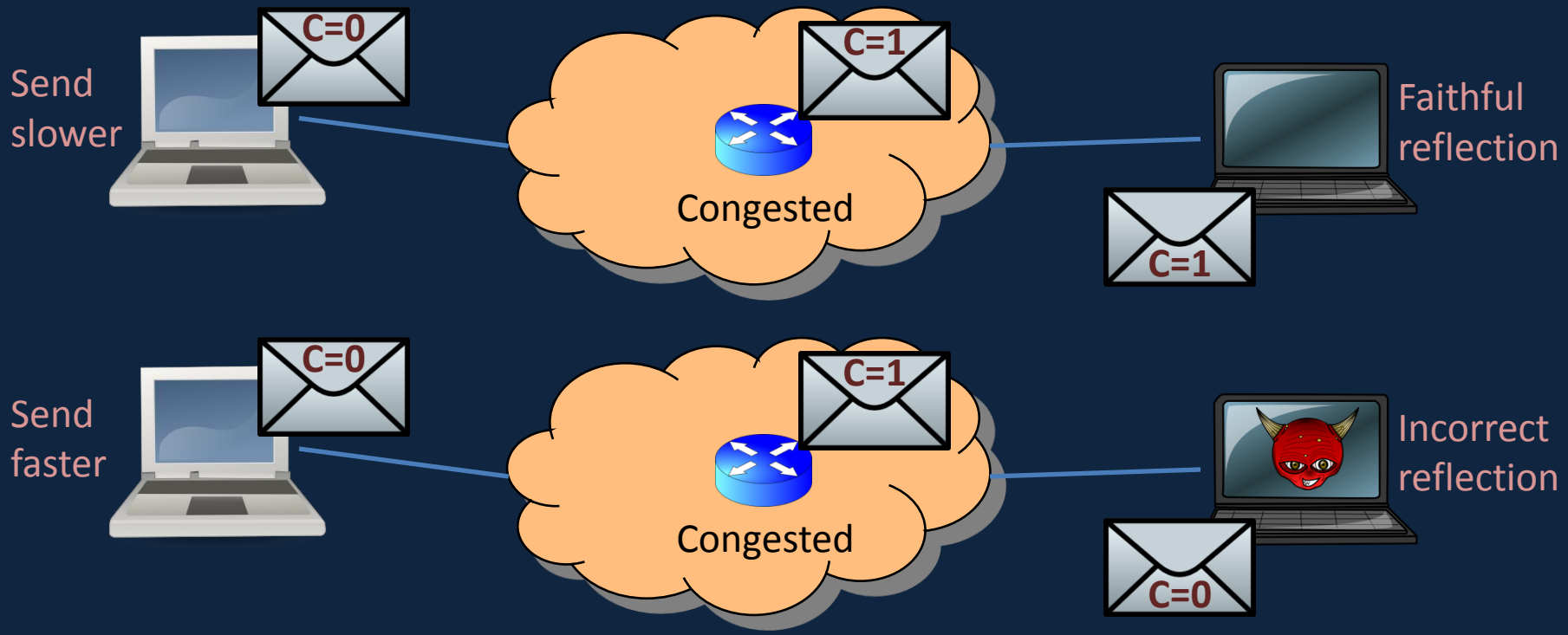
Protocol gullibility

tendency ***of the protocol participants*** to believe too readily and therefore be easily deceived ***by manipulative participants***

- The protocol can be subverted without the knowledge of the honest participants

Gullibility of the ECN protocol

ECN receiver can deceive the sender into sending faster [Wetherall 2001]



More examples of protocol gullibility

TCP receiver can deceive the sender into sending faster

- Three separate manipulation mechanisms [Savage 1999]

Nodes can lie about connectivity in routing protocols

Nodes can refuse to relay packets in multi-hop wireless networks

Several manipulation mechanisms exist for DHTs

Why care about protocol gullibility?

Gullibility is a different form of protocol weakness

- ≠ bugs
- ≠ security problems of auth., integrity, and privacy
- Manipulation by legitimate participants instead of external agents

Gullible protocols fail to achieve their goal in the presence of manipulation

- Modern protocols are regularly used between entities that should not trust each other
- Blind trust is foolhardy: hijacked or buggy participants

Our work

Develop methods to automatically uncover protocol gullibility

This paper represents a baby step:

- Poses and formalizes the problem
- Identifies key challenges and helpful techniques
- Implements a preliminary checker

Problem formulation

Two-player game between *angelic* and *demonic* components

The angelic component consists of honest participants

- Follows the protocol
- Non-determinism is allowed

The demonic component consists of manipulators

- Not limited by the protocol; can do anything
- Collusion is allowed

The protocol is gullible if is there exists a strategy for the demonic component that violates a desirable property

Challenges in determining gullibility

1. Practical search over demonic strategies
2. Determining when a strategy succeeds
3. Dependence on network conditions

Challenge 1: Practical strategy search

The space of demonic strategies

- Any bit-pattern can be sent in a packet
  - 2^{12000} possibilities for a 1500-byte packet
- Some strategies may involve packet sequences

Proposed techniques to make search tractable

- Consider only the header part of the packet
- Consider only syntactically correct packets
- Consider limited-history strategies
- Exploit independence of header fields
- Program analysis (in a few slides)

Challenge 2: Determining when a strategy has been successful

Want to go beyond non-binary properties but hard to predict protocol behavior in arbitrary conditions

- E.g., throughput of a TCP receiver
- Solution: Compare with reference behavior under the same network conditions

Non-determinism precludes direct one-to-one comparison against reference behavior

- E.g., TCP throughput depends on exact packets lost
- Solution: Statistical comparison over multiple runs

Challenge 3: Dependence on network conditions

Some strategies succeed only under particular network conditions

- E.g., the ECN manipulation has no impact in the absence of congestion

Proposed solution:

- Search over the space of network conditions
  - Assume paths between pairs of participants are independent
- Check if the strategy succeeds under any condition

Our preliminary gullibility checker

Checks protocol implementations

- Assumes all headers fields are independent
- Single-step strategies (no history)
- Built on top of MACE [Killian 2005]

Explores angelic non-determinism using simulation

Explores demonic strategies as modifications of the reference implementation itself

- Modify outgoing packets (implemented)
- Add or drop packets (not yet implemented)

Inputs to the checker

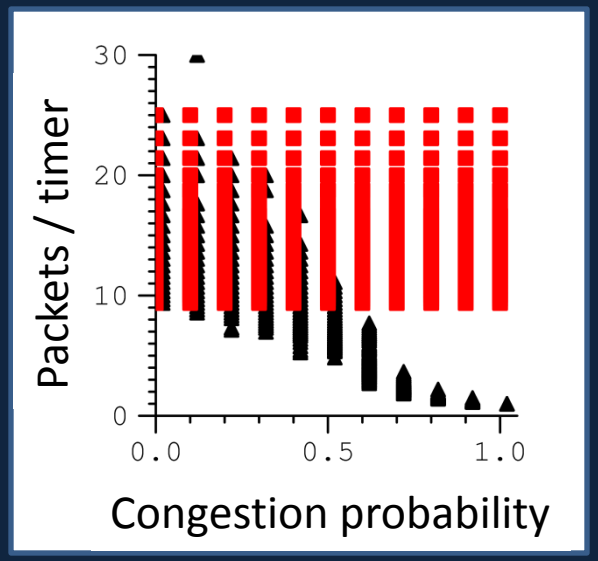
1. Network configuration
2. Properties to be checked
   - Specified in terms of implementation variables
3. Protocol header format
4. A packet modifier class

Field Type	Default strategies for demonic component
Fixed, CkSum	None
Enum, Range	Try each value, pick at random
SeqNum	Subtract or add a constant
Id	Pick at random
Other	User-specified

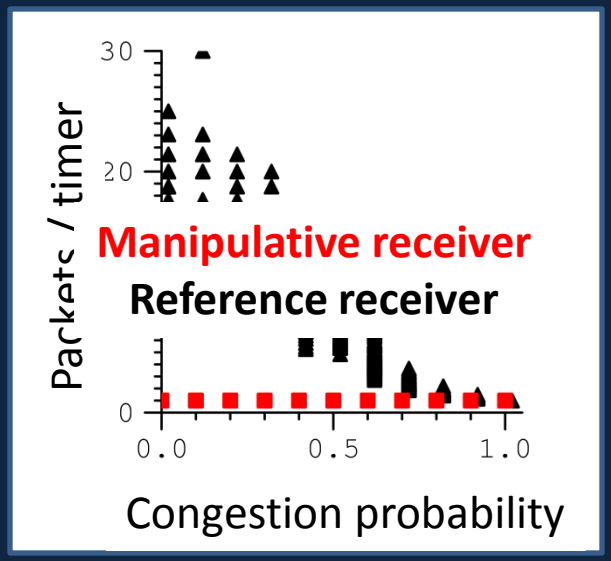
Case study: ECN

We implement a version of the ECN protocol in MACE

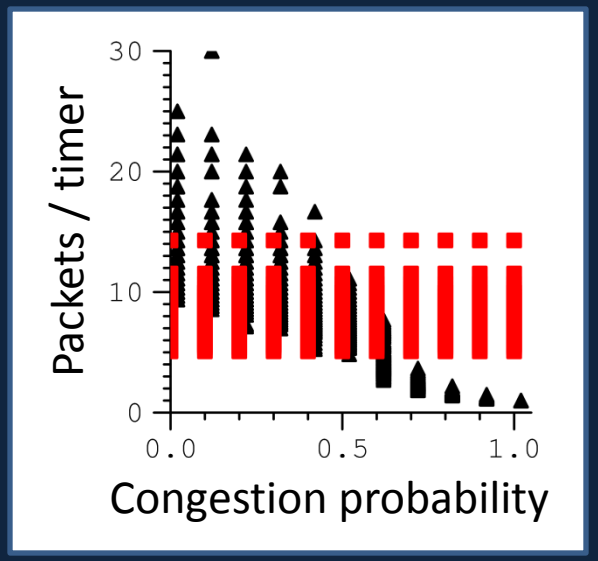
Specify throughput as the property to be preserved



Set bit to 0



Set bit to 1



Set bit randomly

Next: Program analysis to further reduce search space

Infer independent header fields

Infer inputs that are ignored by honest participants

- E.g., If (IP.version != 4) ignore;
- E.g., If (Ack.SeqNum NOT IN CongWin) ignore;

Infer inputs that impact relevant state variables

- E.g., If (Ack.SeqNum > LastSeqNum) pktsSent++;

Hope to leverage work on taint analysis, directed random testing, and symbolic execution

Conclusions and future work

Gullibility is a major vulnerability in modern protocols

- Important to develop methods for automatic detection

Our work scratches the surface of the problem

- Poses the problem and outlines the challenges
- Our preliminary methods show promise

Future work:

- Evaluate more complex protocols
- Design principles for non-gullibility