UNDERSTANDING SIP THROUGH MODEL-CHECKING

Pamela Zave

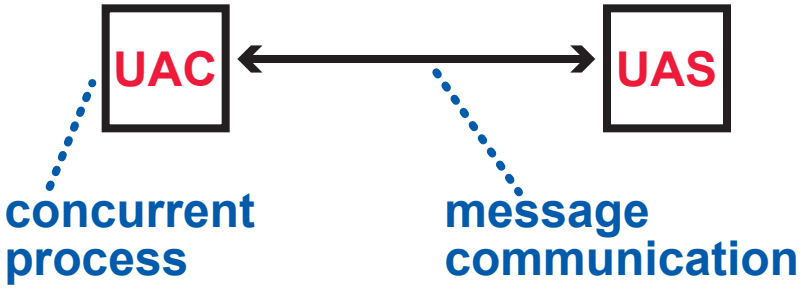
AT&T Laboratories—Research

Florham Park, New Jersey USA

OVERVIEW

MODELING

- wrote a formal model of SIP INVITE dialogs in Promela



- the model has a special emphasis on media control (offer/answer exchange)
- limitations and simplifications are documented carefully
- all versions of the model are available on my Web site

ANALYSIS

WHY?

Because there are thousands of pages of RFCs, scattered with rules such as:

"The UAS MUST NOT send a second reliable provisional response until the first is acknowledged."

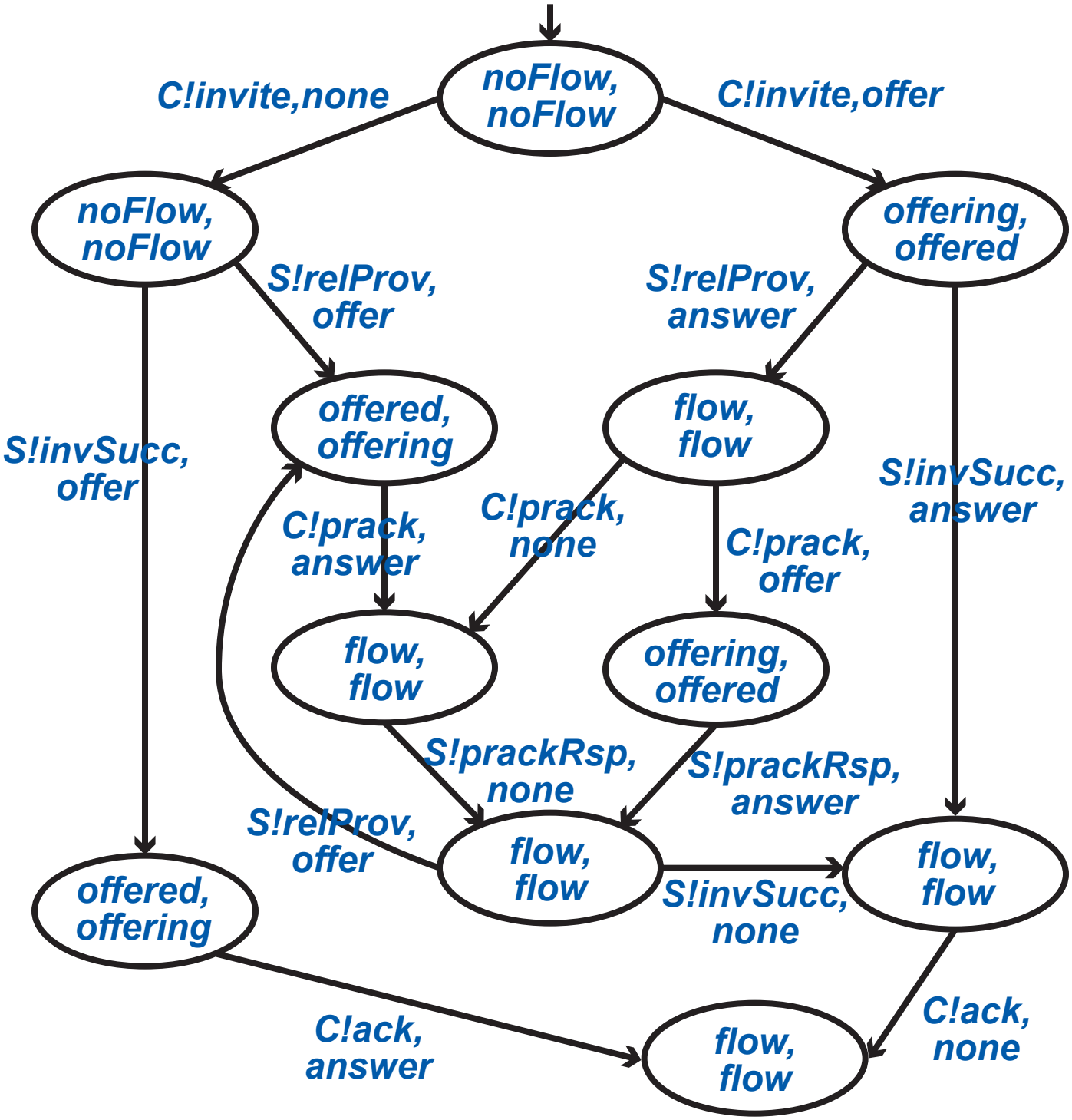
A state-oriented model pulls many of these rules together in this form:

WHY?

Because there are thousands of pages of RFCs, scattered with rules such as:

"The UAS MUST NOT send a second reliable provisional response until the first is acknowledged."

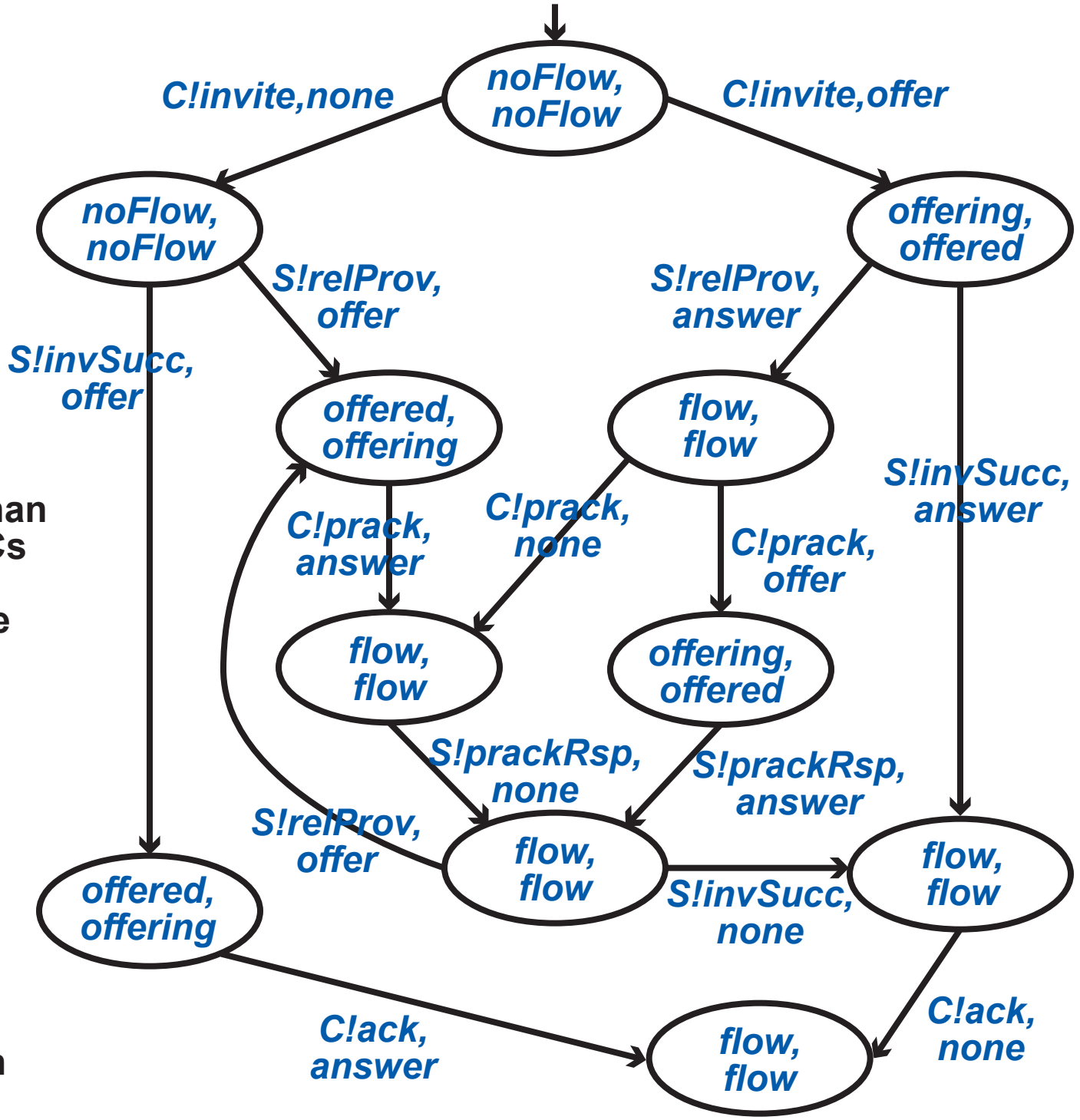
A state-oriented model pulls many of these rules together in this form:



Reliable provisional responses must be handled in exactly this way.

This state-oriented view has many advantages as supplementary documentation of SIP:

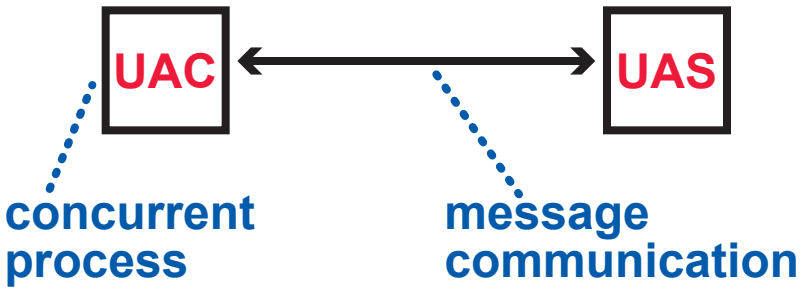
- centralized, rather than distributed over RFCs
- formal, and therefore unambiguous
- can be checked automatically
- shows the state information that user agents must maintain
- can be used for other purposes such as testing



OVERVIEW

MODELING

- wrote a formal model of SIP INVITE dialogs in Promela



- the model has a special emphasis on media control (offer/answer exchange)
- limitations and simplifications are documented carefully
- all versions of the model are available on my Web site

ANALYSIS

- analyzed the model using the model-checker Spin
- discovered a few problems in the SIP RFCs
- explored SIP issues using alternative models
- collected data on the analysis of several model versions

ANALYSIS: MODEL-CHECKING EXPLORES ALL POSSIBLE BEHAVIORS

case in confirmed state of UAC

embedded assertions record all expectations about state

Spin reports an error when an assertion is violated

```
:: irps?invSucc,sdp;
   assert(reInviting && ! reInvited && sdp != none);
   reInviting = false;
   if
   :: media == flow; assert(sdp == offer); ackc!ack,answer
   :: media == offering; assert(sdp == answer);
      ackc!ack,none; media = flow
   :: media == noFlow || media == offered; assert(false)
   fi
```

this should be unreachable

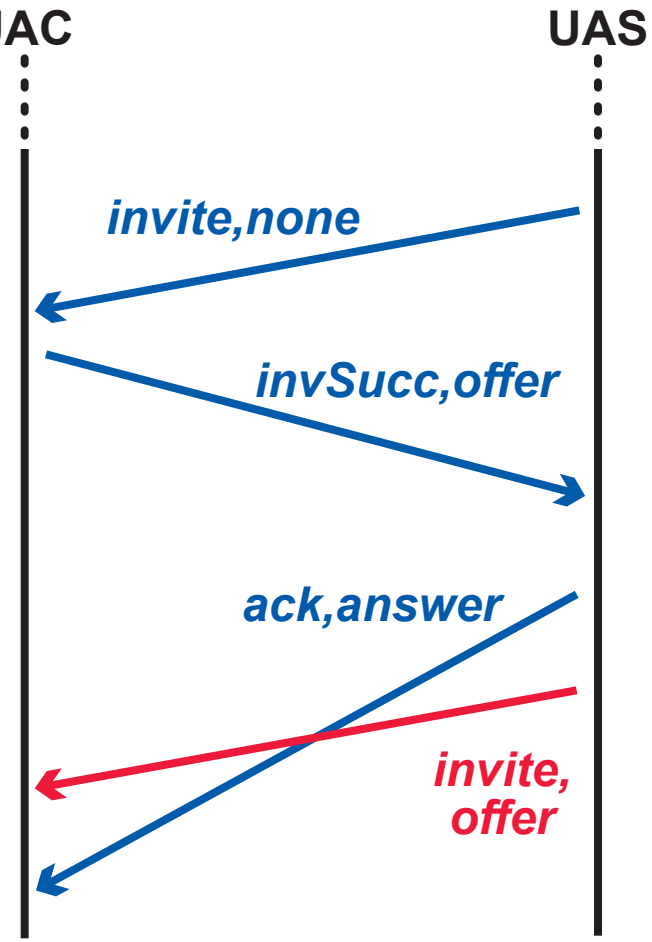
this case statement is guaranteed to be exhaustive

Spin reports an error when there is deadlock, or unreachable code is reached

THE RE-INVITE PROBLEM

within an invite dialog, consider *all* the messages sent from one UA to the other: they are not guaranteed to arrive in FIFO order

here the second *re-invite* cannot be handled when it is received because there is an ongoing offer/ answer exchange



the basic SIP model uses the obvious workaround of buffering the *re-invite* in the UAC or UAS until it can be processed

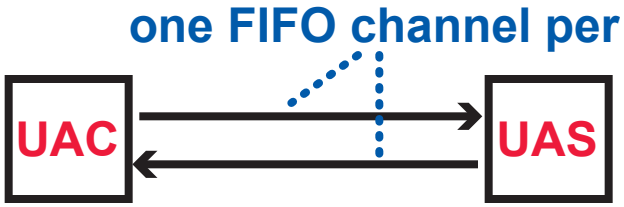
WHAT IS THE COST OF THIS WORKAROUND?

the same basic problem occurs in other scenarios, with different messages

later, another example

WHAT IF SIGNALING IN AN INVITE DIALOG WERE FIFO?

THE "FIFO" MODEL EXPLORES THIS POSSIBILITY



IT MAKES A HUGE DIFFERENCE!

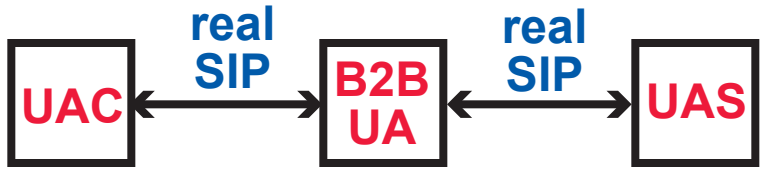
performance measure	basic model	FIFO model
lines of code	404	300
analysis memory (megabytes)	20,904	308
analysis time (seconds)	4,200	38

DOES MODEL COMPLEXITY MATTER?

In another study with similar model-checking and a related protocol, we had configurations like the ones here . . .

. . . and also configurations like this:

If we compare these two configurations and apply the ratios to the SIP numbers, we arrive at this guesstimate:



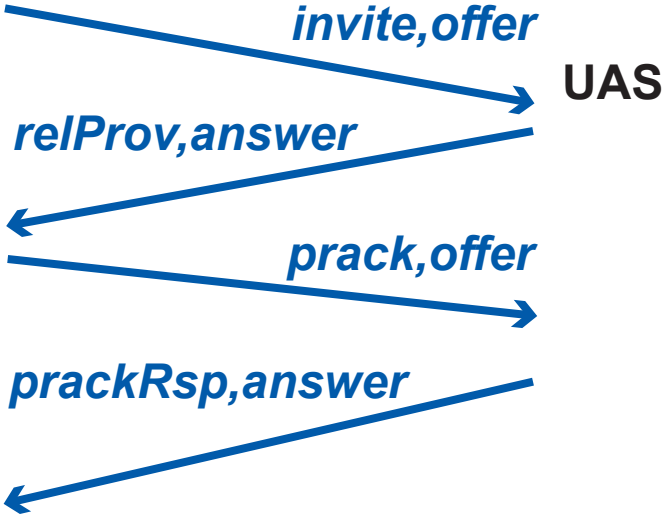
6 terabytes analysis memory

1200 hours analysis time

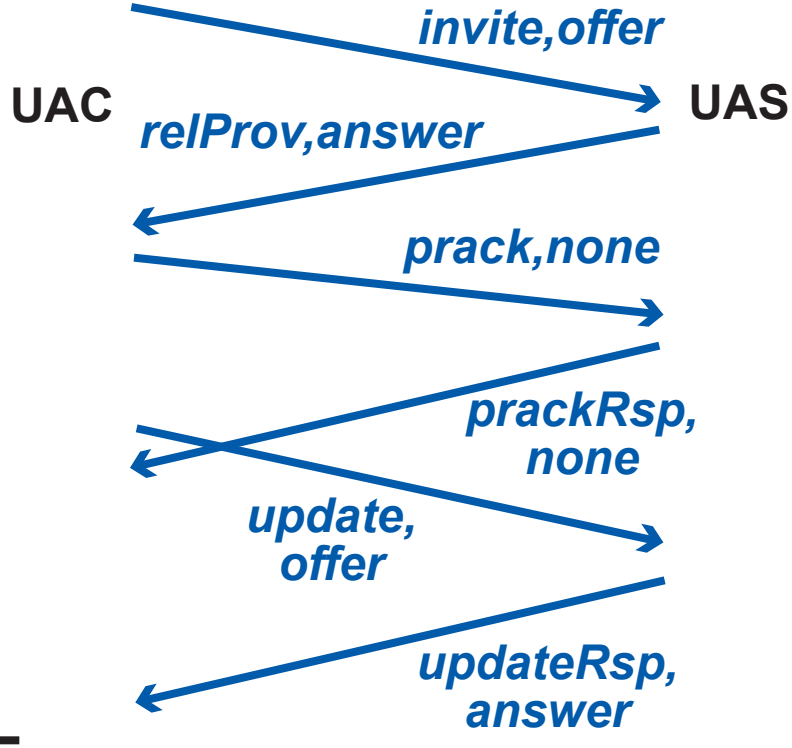
we are not confident in our ability to build a correct B2BUA for non-FIFO SIP

WHAT IF REDUNDANT CAPABILITIES WERE NOT USED?

when reliable provisional responses were added, *prack,offer* gave UAC a new capability



later updates were added, making *prack,offer* redundant



performance measure	basic model	FIFO model	pruned model
lines of code	404	300	266
analysis memory (megabytes)	20,904	308	105
analysis time (seconds)	4,200	38	13

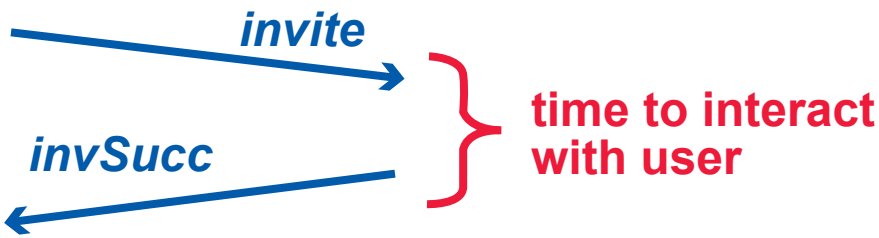
pruning a few redundant capabilities reduces analysis resources by another factor of 3, for a total reduction of 300

CONCLUSIONS OF THIS STUDY

THERE ARE MANY INTERESTING THINGS TO BE LEARNED BY STUDYING THESE MODELS

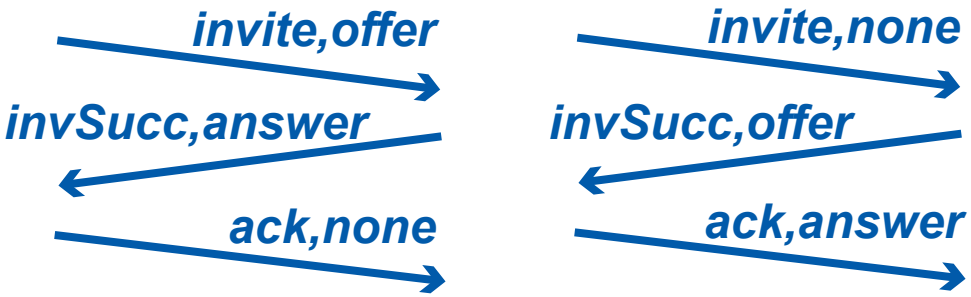
Example: Why is re-invite more powerful than update?

RFC 3311 (UPDATE) says "because UPDATE rules out user approval."



This is not the only difference!

A re-invite transaction allows the offer to come from either direction, which is critical to third-party call control.



THE UTILITY OF STATE-ORIENTED MODELS AND MODEL-CHECKING ARE INDISPUTABLE

- **they provide a new view of SIP**
- **considering the thousands of hours of labor that have gone into the SIP RFCs, this is a quick and cost-effective way to debug the protocol and its specification**
- **this view should influence the future evolution of SIP**

in particular, . . .

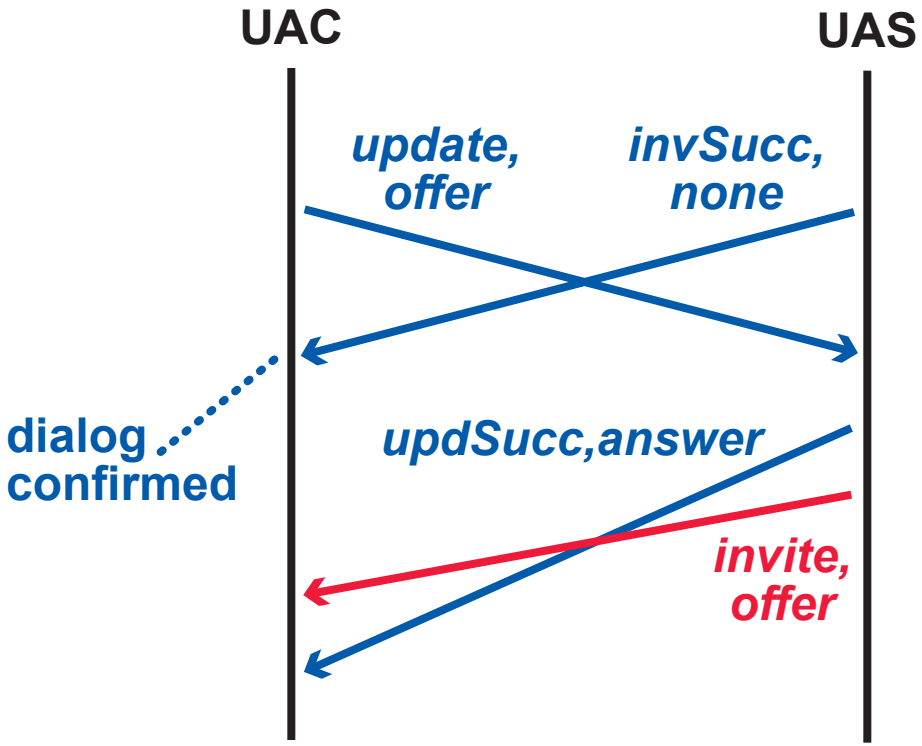
FUTURE WORK: HOW CAN WE GET THE BENEFITS OF FIFO SIGNALING?

with Greg Bond, Eric Cheung, Hal Purdy, Tom Smith

It is reasonable to assume TCP signaling.

RFC 3261 recommends at most two TCP connections at a time, one for transactions initiated in each direction.

However, this is not strong enough to ensure that messages arrive in FIFO order.



The number of TCP connections per dialog appears to be an overconstrained problem.

toward fewer connections:

- **SIP constrains port use**
- **setup of a secure connection is expensive, so fewer connections means less overhead**

toward more connections:

- **shorter-duration connections are more secure**
- **more connections minimizes congestion at port level**

we hope to find a way through this maze