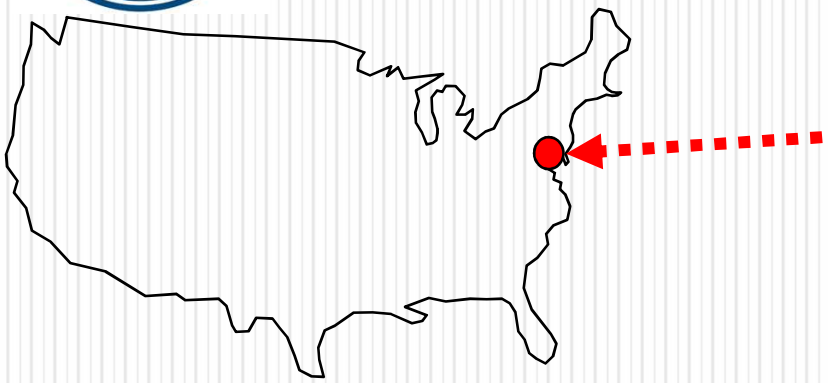
Misbehaviors in TCP SACK Generation

Nasif Ekiz, Abuthahir H. Rahman,
Paul D. Amer

Protocol Engineering Laboratory
Computer and Information Sciences,
University of Delaware



supported by CISCO

OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

TCP acknowledgements

- cumulative ACK *n* (for ordered data)

  bytes [*… to n-1*] [RFC 793]

- selective ACK (SACK) *m-n* (for out-of-order data)

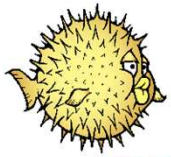
  bytes [*m* to *n-1*] [RFC 2018]

  - prevents unnecessary retransmissions during loss recovery
  - improves throughput when multiple losses in same window (SACK-based loss recovery [RFC 3517])

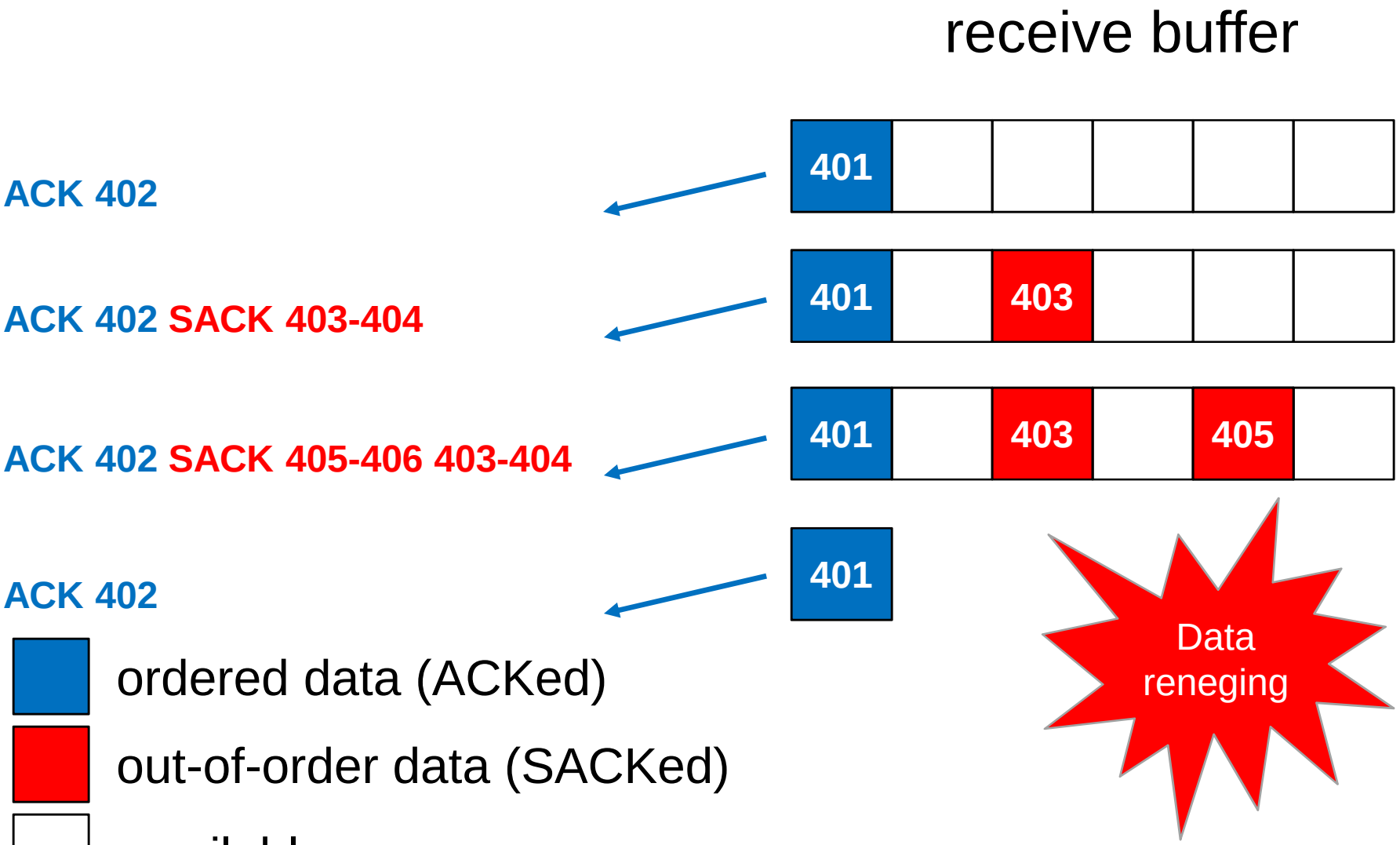
Deployment of SACK

- 2001: 41% of web servers do SACK
- 2005: 68% of web servers do SACK
- 2010: All recent versions of OSes do SACK
  - FreeBSD
  - Linux
  - Mac OS X
  - OpenBSD
  - Open Solaris
  - Solaris
  - Windows



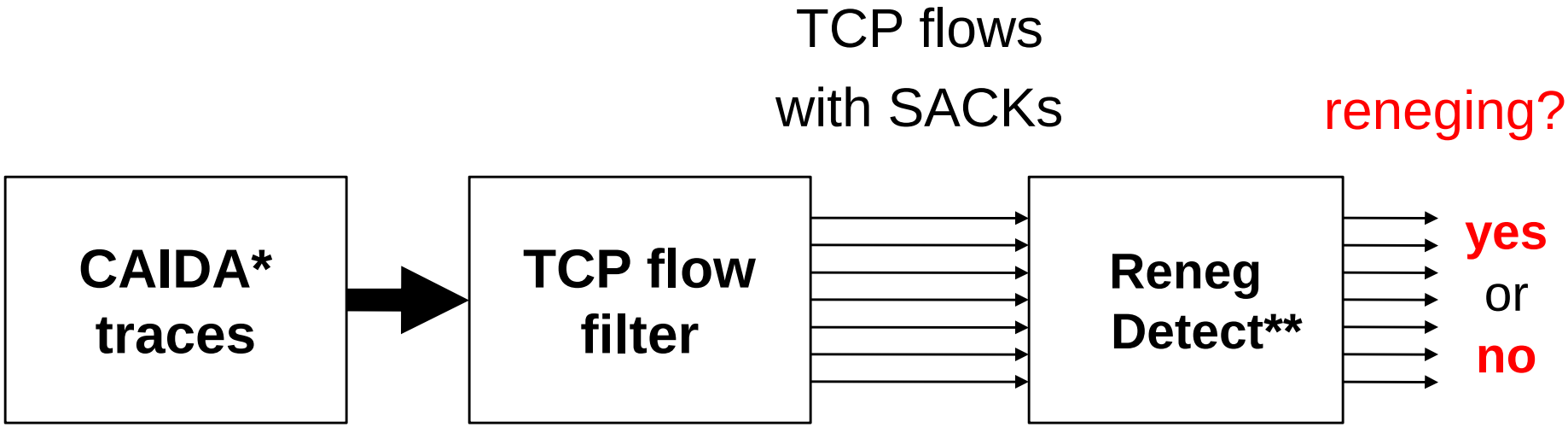
Data reneging



Data reneging

- TCP is designed to tolerate reneging
  - [RFC 2018]: “The SACK option is *advisory*, in that, while it notifies the data sender that the data receiver has received the indicated segments, the data receiver is permitted to later *discard* data which have been reported in a SACK option.”
  - TCP data sender retains copies of all SACKed data until ACKed

RenegDetect



* Cooperative Association for Internet Data Analysis

** Nasif Ekiz, Paul D. Amer, **"*A model for detecting transport layer data reneging*"**, PFLDNeT 2010, Lancaster, PA, 11/10

RenegDetect

- RenegDetect was tested with real TCP flows
  - at first, reneging seemed to occur frequently
  - on closer inspection, we found many unexpected behaviors !
    - sometimes SACKs that should have been sent were not
    - sometimes the wrong SACKs were sent
- these SACK misbehaviors wrongly gave the impression that data reneging was occurring

Research goal

- verify TCP SACK for a wide-range of OSes
- not measure degraded performance
  - misbehaviors can reduce the effectiveness of SACKs

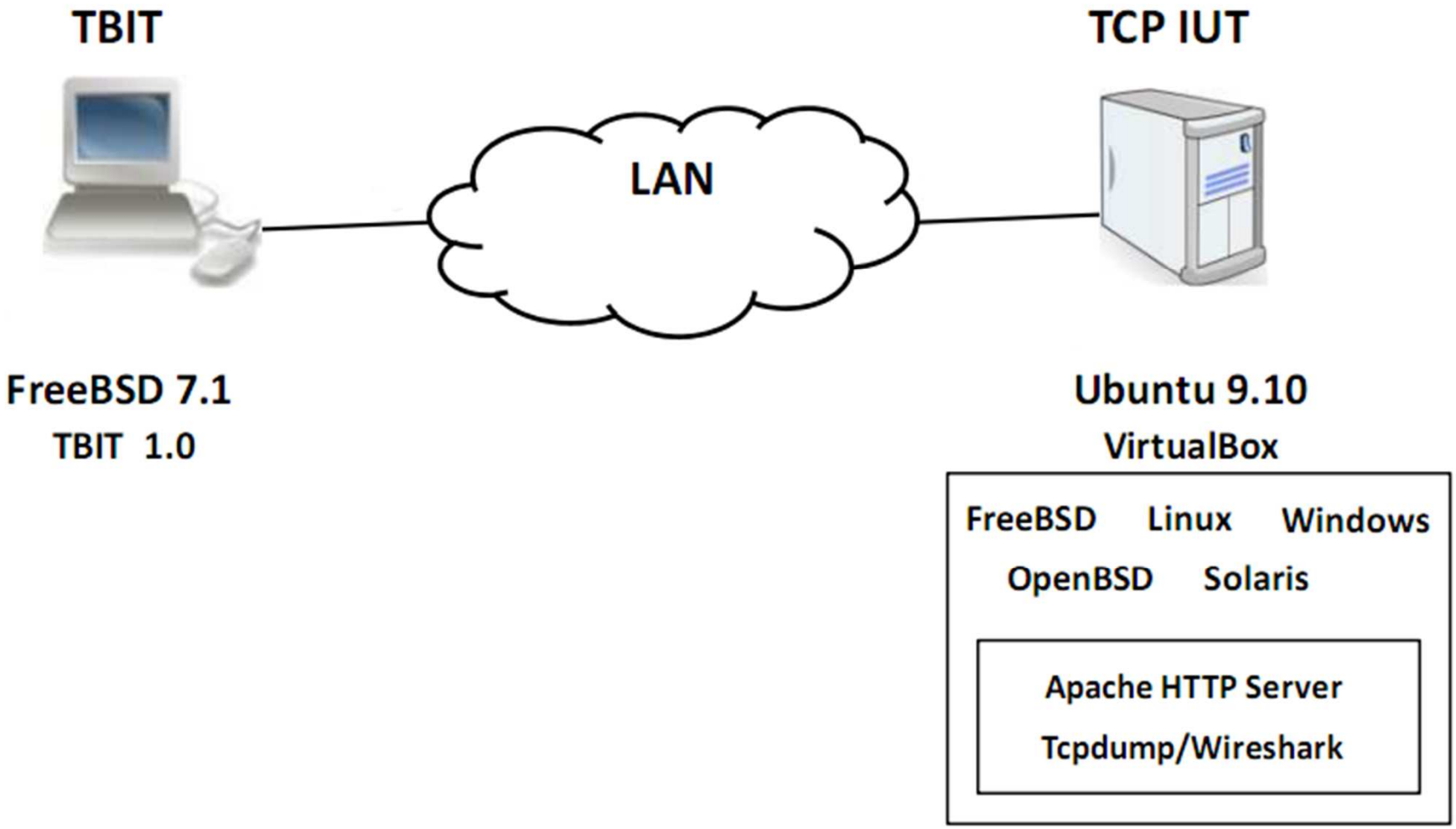
OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

Experimental design

TCP Behavior Inference Tool

TCP Implementation Under Test



OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

MISBEHAVIOR

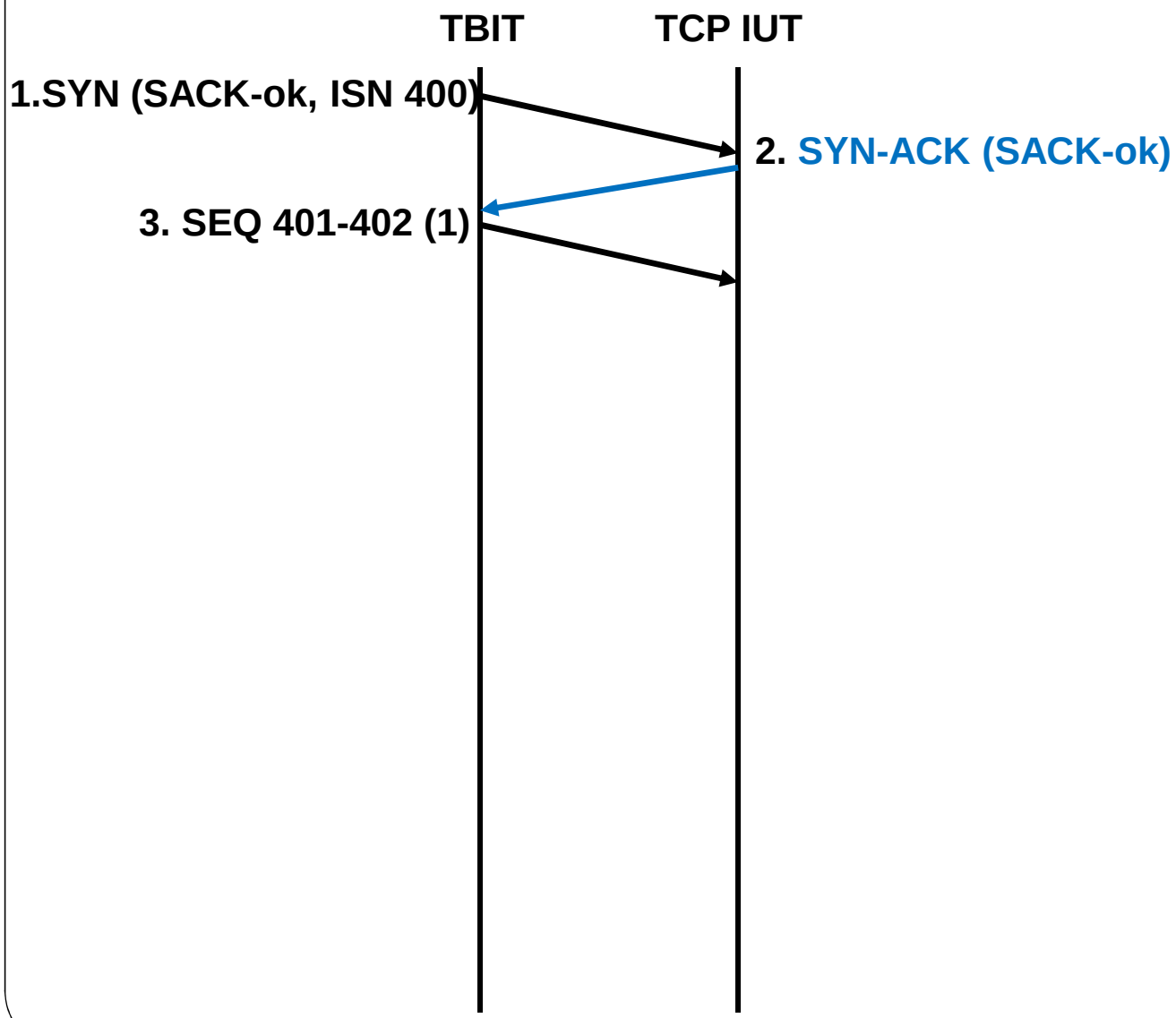
A

Misbehavior A:
fewer than maximum number of reported SACKs

- RFC 2018 section 3
  - "*the data receiver SHOULD include as many distinct SACK blocks possible in the SACK option,*" and
  - "*the 40 bytes available for TCP options can specify a maximum of four SACK blocks*"

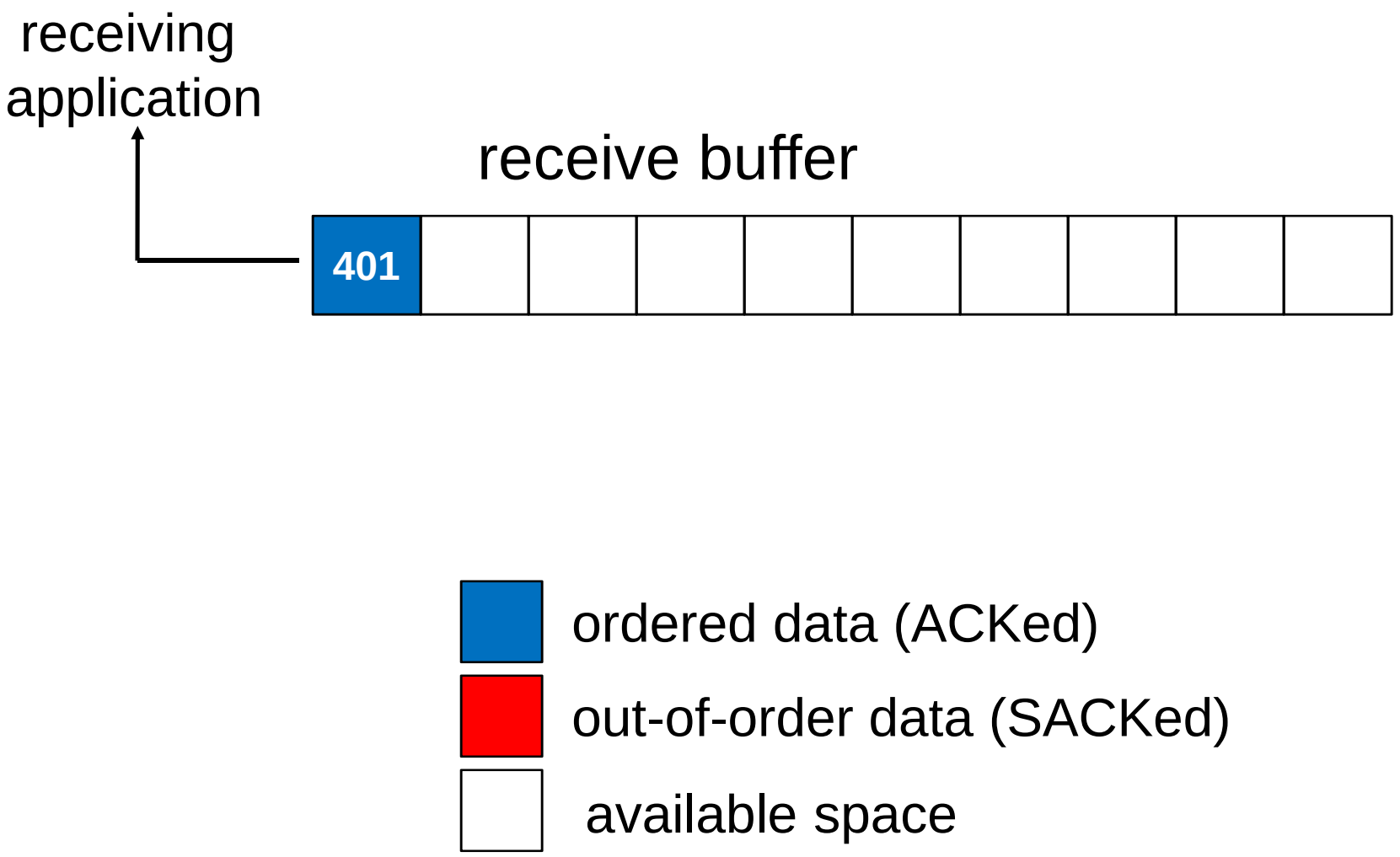
- for some TCP flows, only 2 or 3 SACK blocks are reported even when 4 are possible

Misbehavior A:
fewer than maximum number of reported SACKs

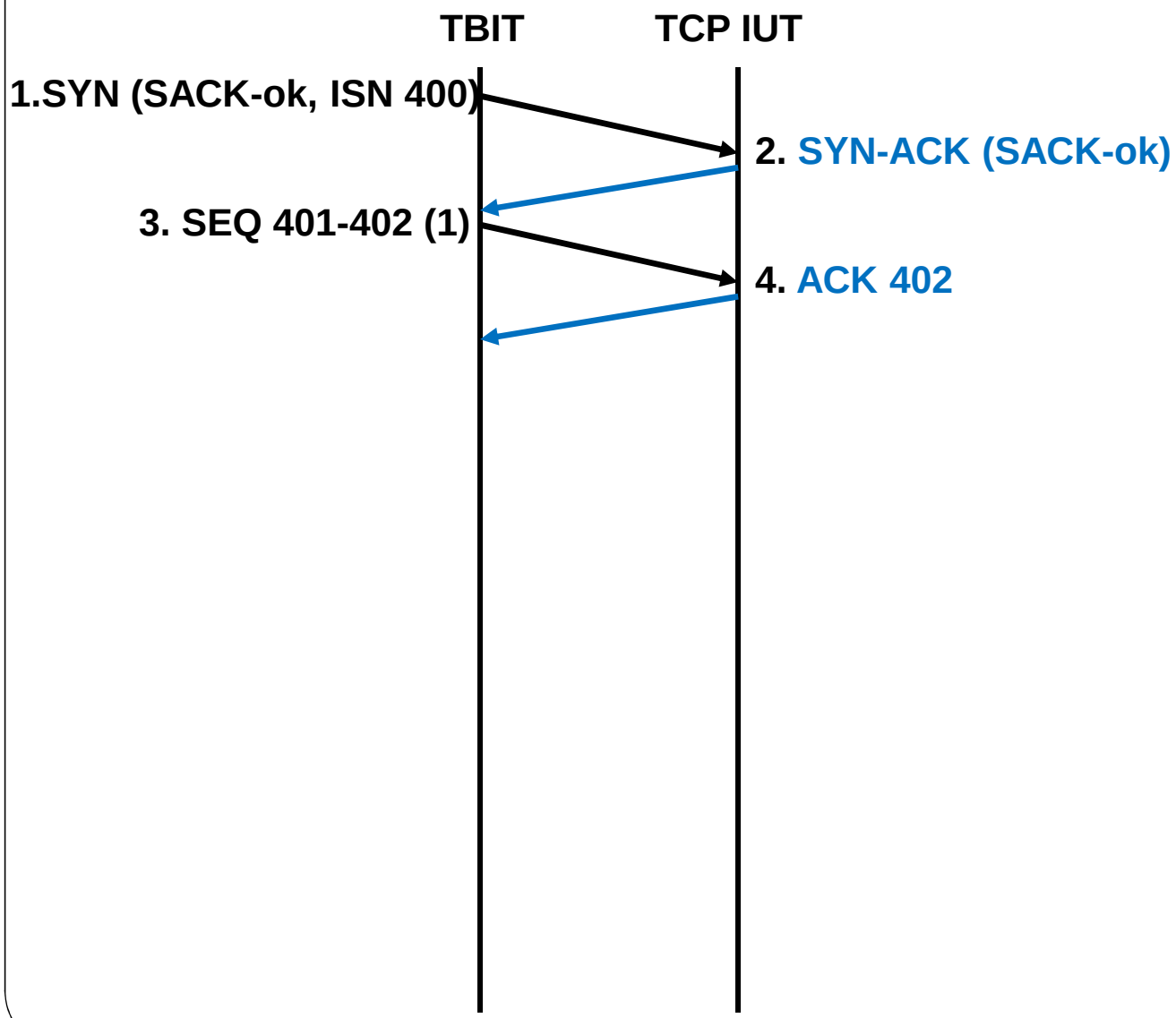


Misbehavior A:

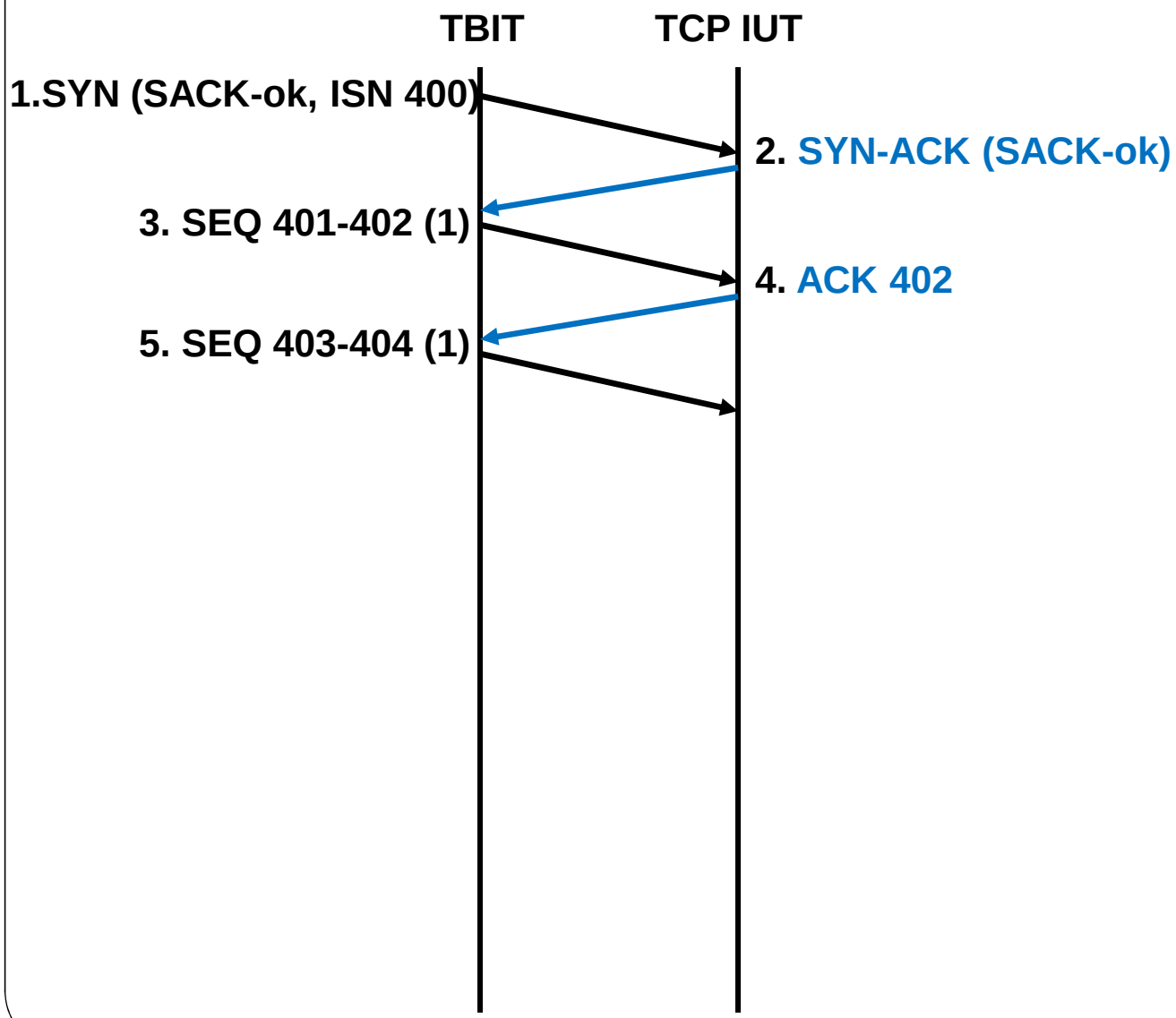
fewer than maximum number of reported SACKs



Misbehavior A:
fewer than maximum number of reported SACKs

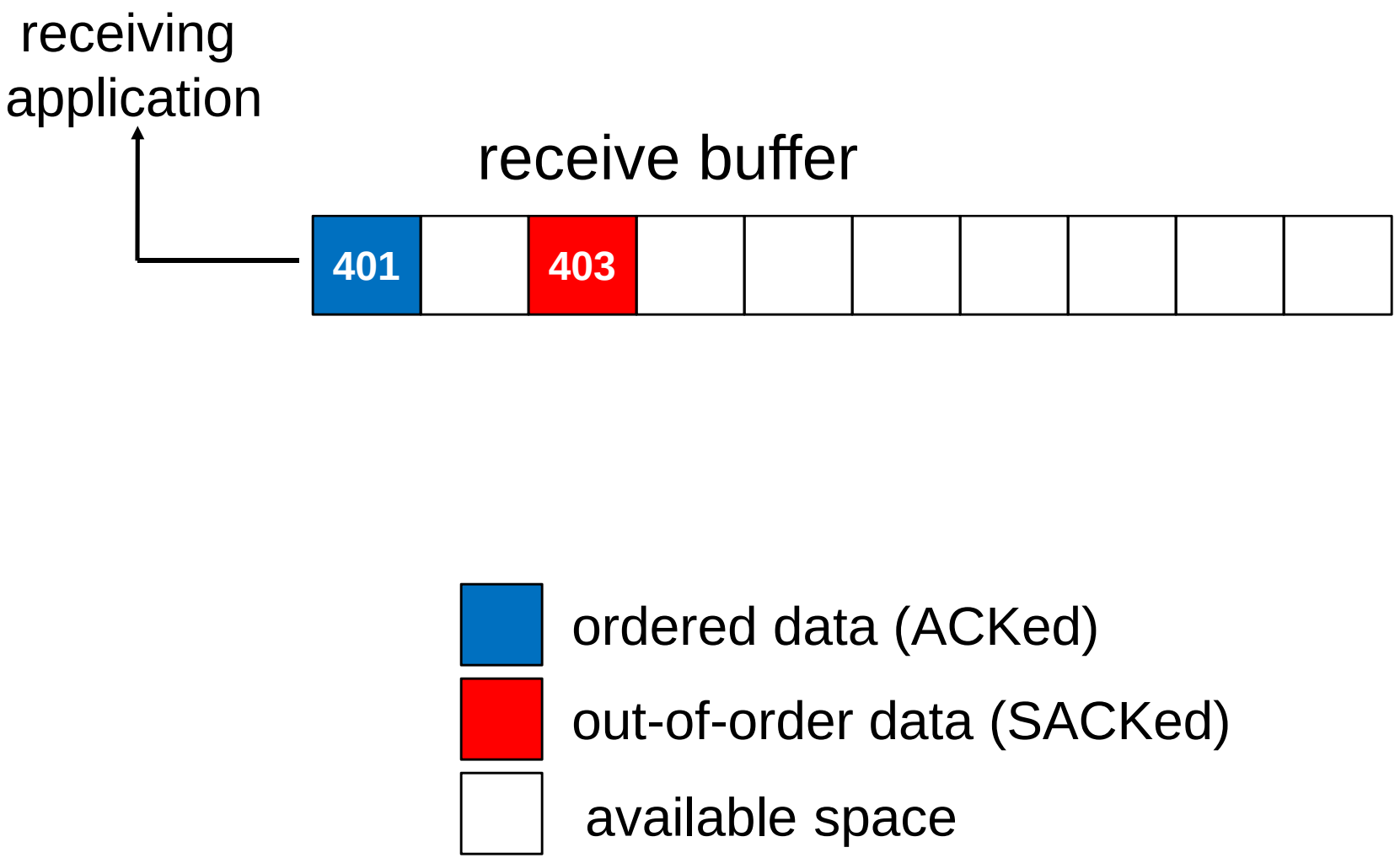


Misbehavior A:
fewer than maximum number of reported SACKs

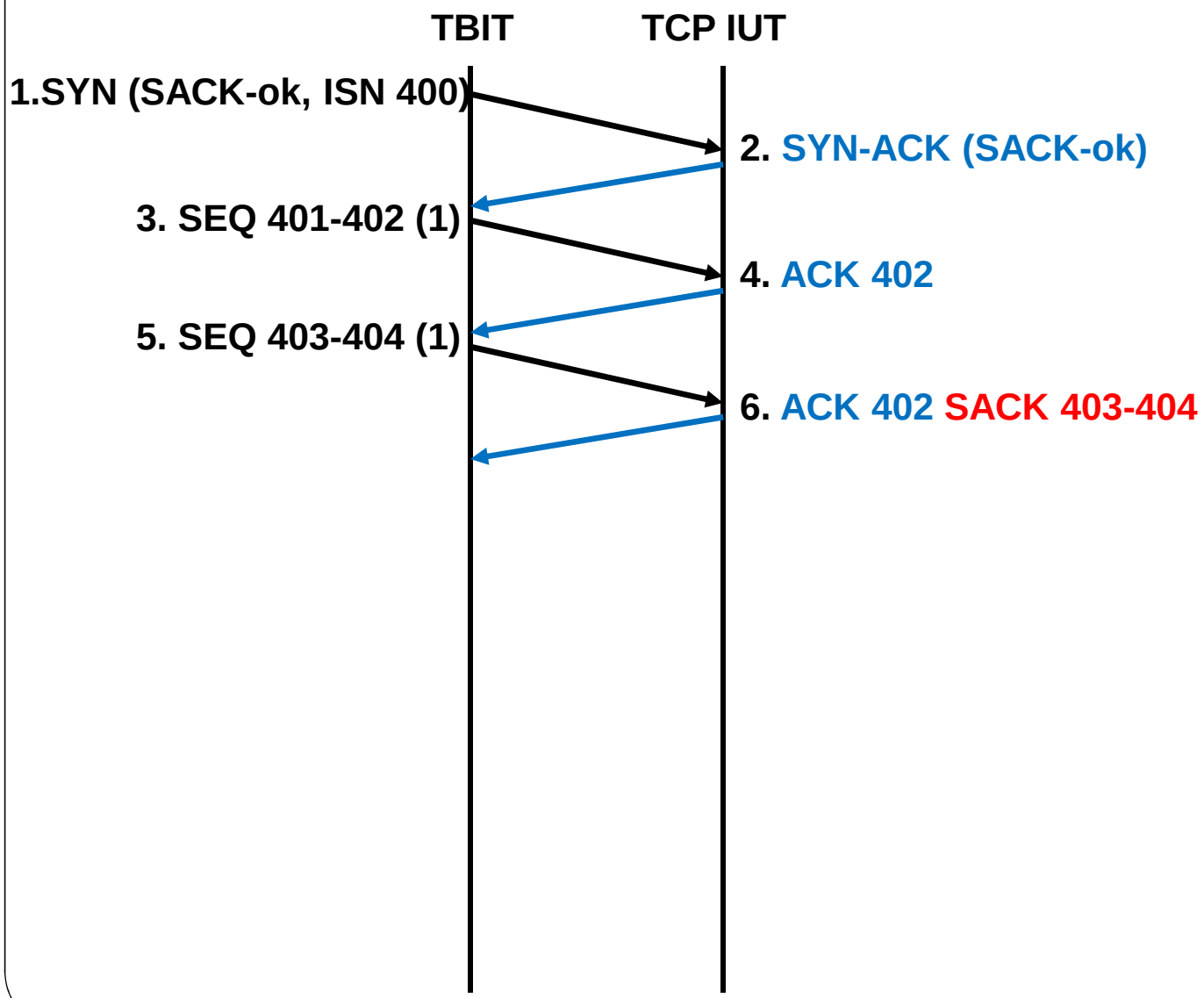


Misbehavior A:

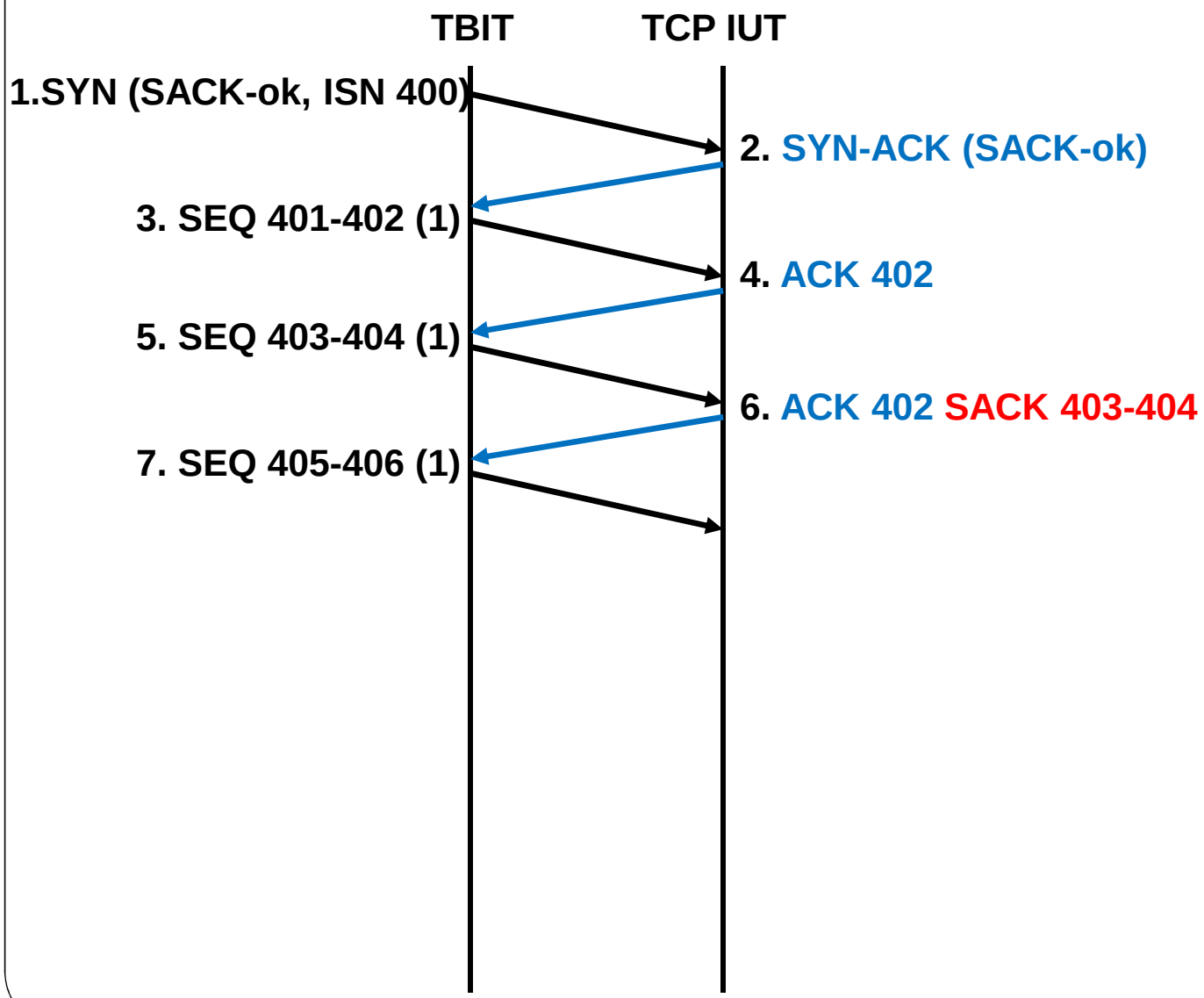
fewer than maximum number of reported SACKs



Misbehavior A:
fewer than maximum number of reported SACKs

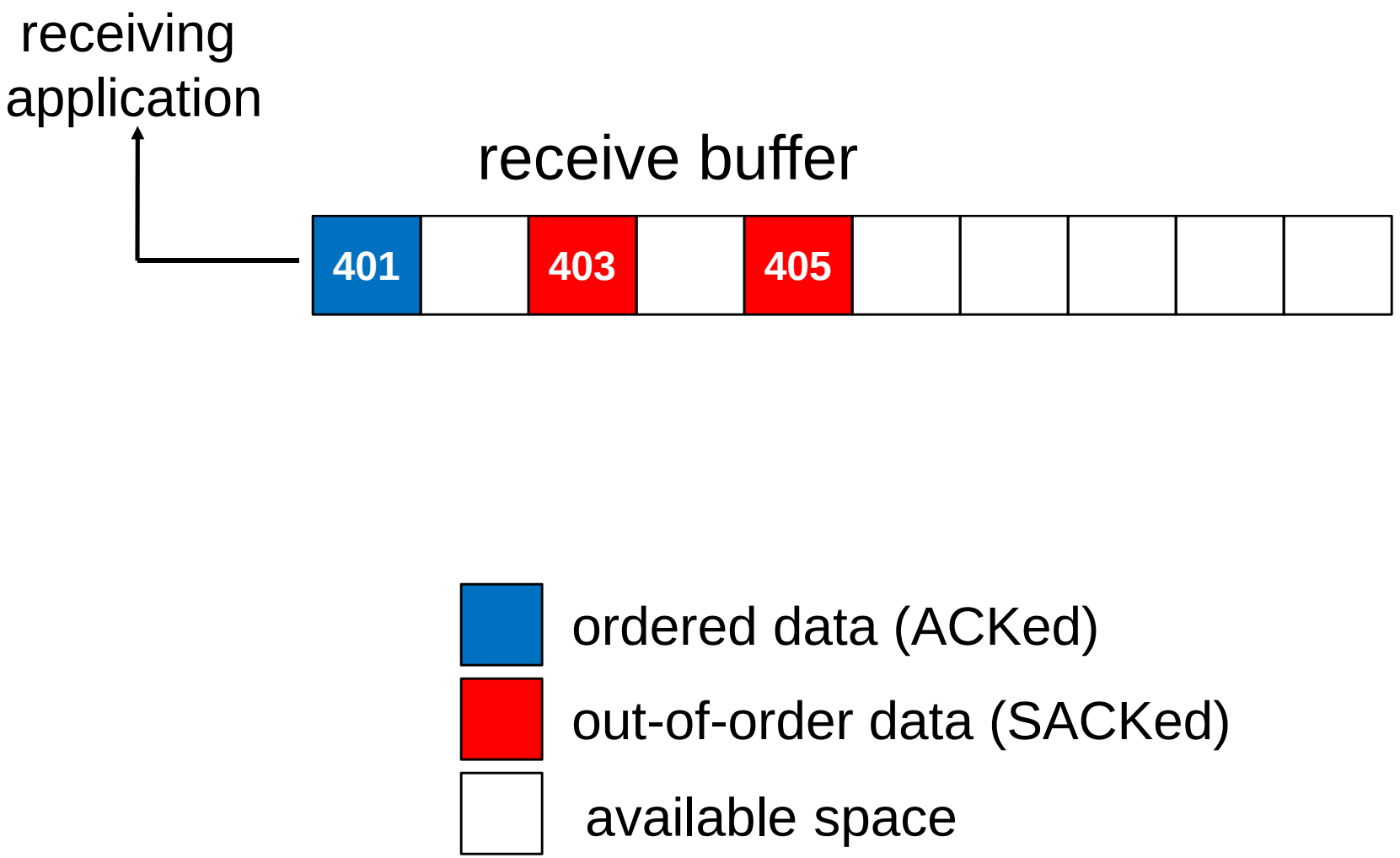


Misbehavior A:
fewer than maximum number of reported SACKs

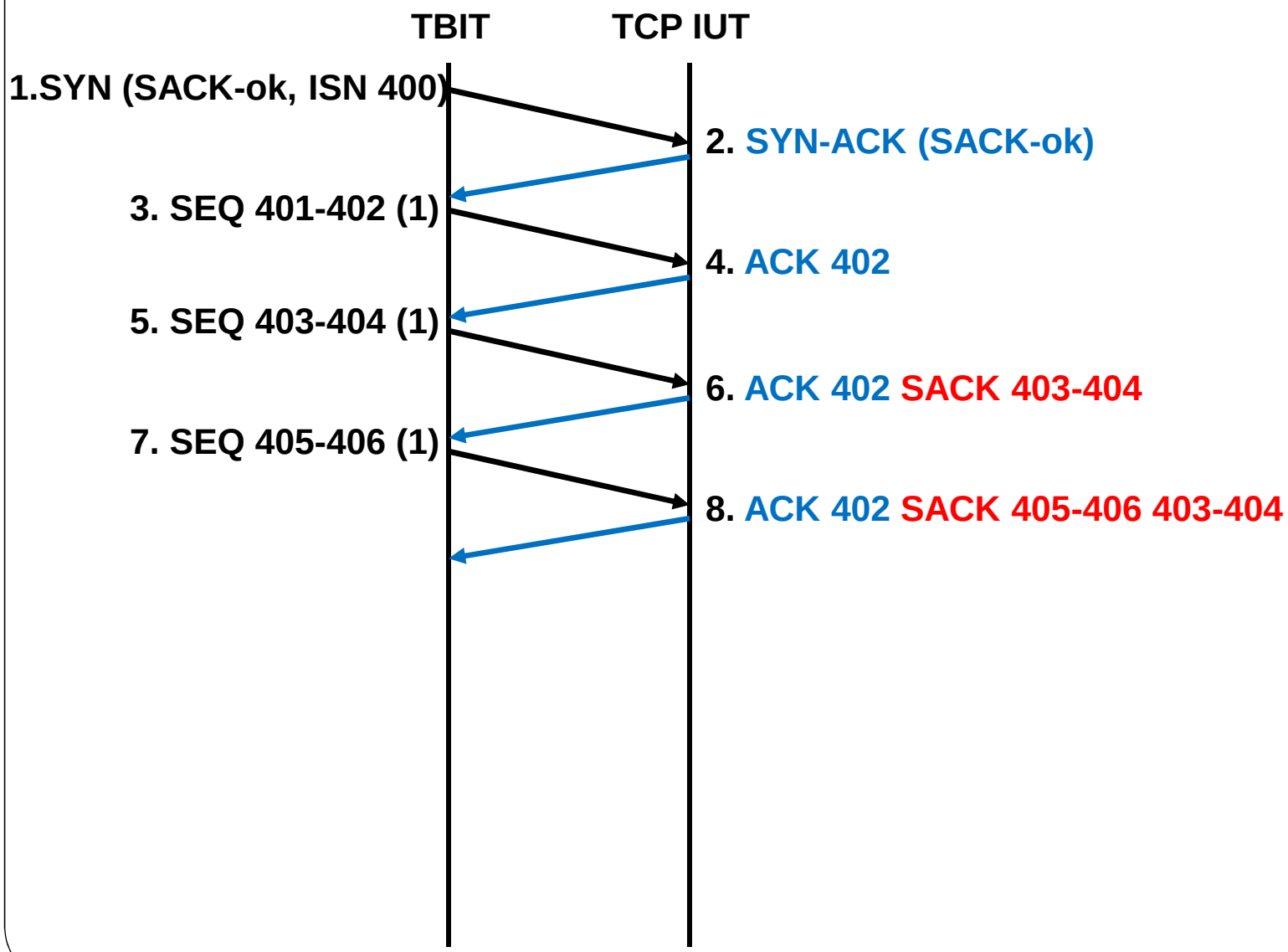


Misbehavior A:

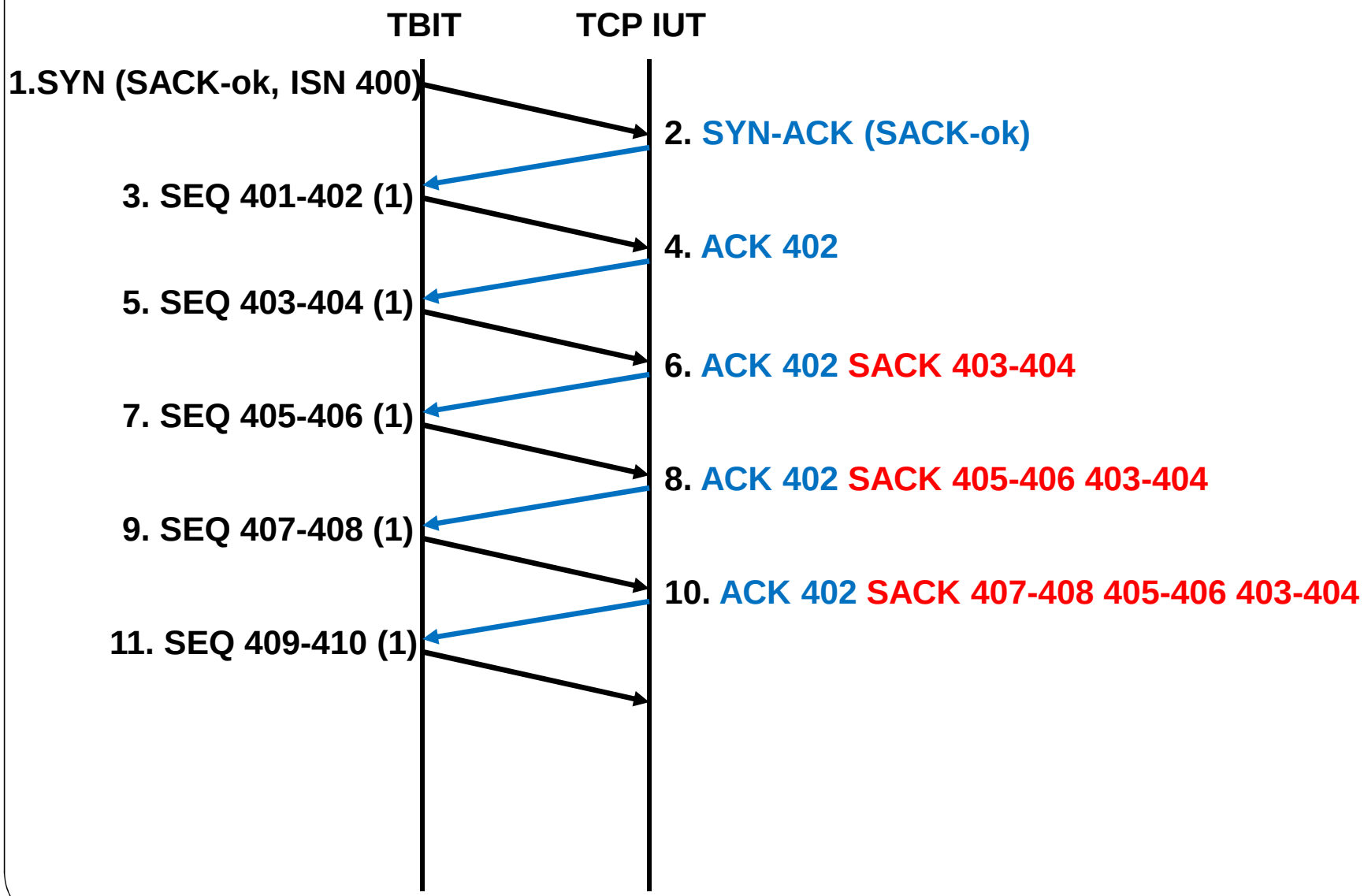
fewer than maximum number of reported SACKs



Misbehavior A:
fewer than maximum number of reported SACKs

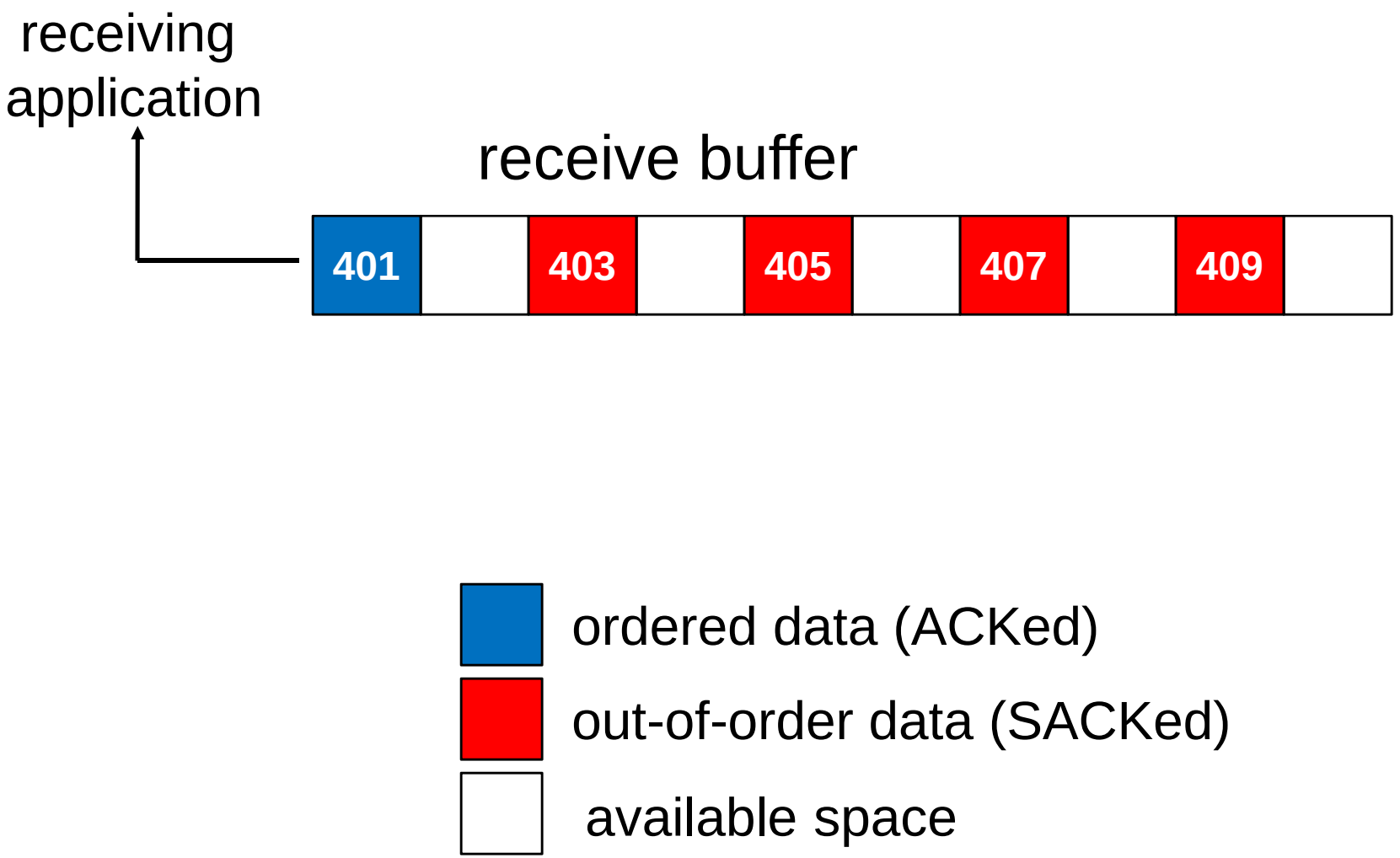


Misbehavior A:
fewer than maximum number of reported SACKs

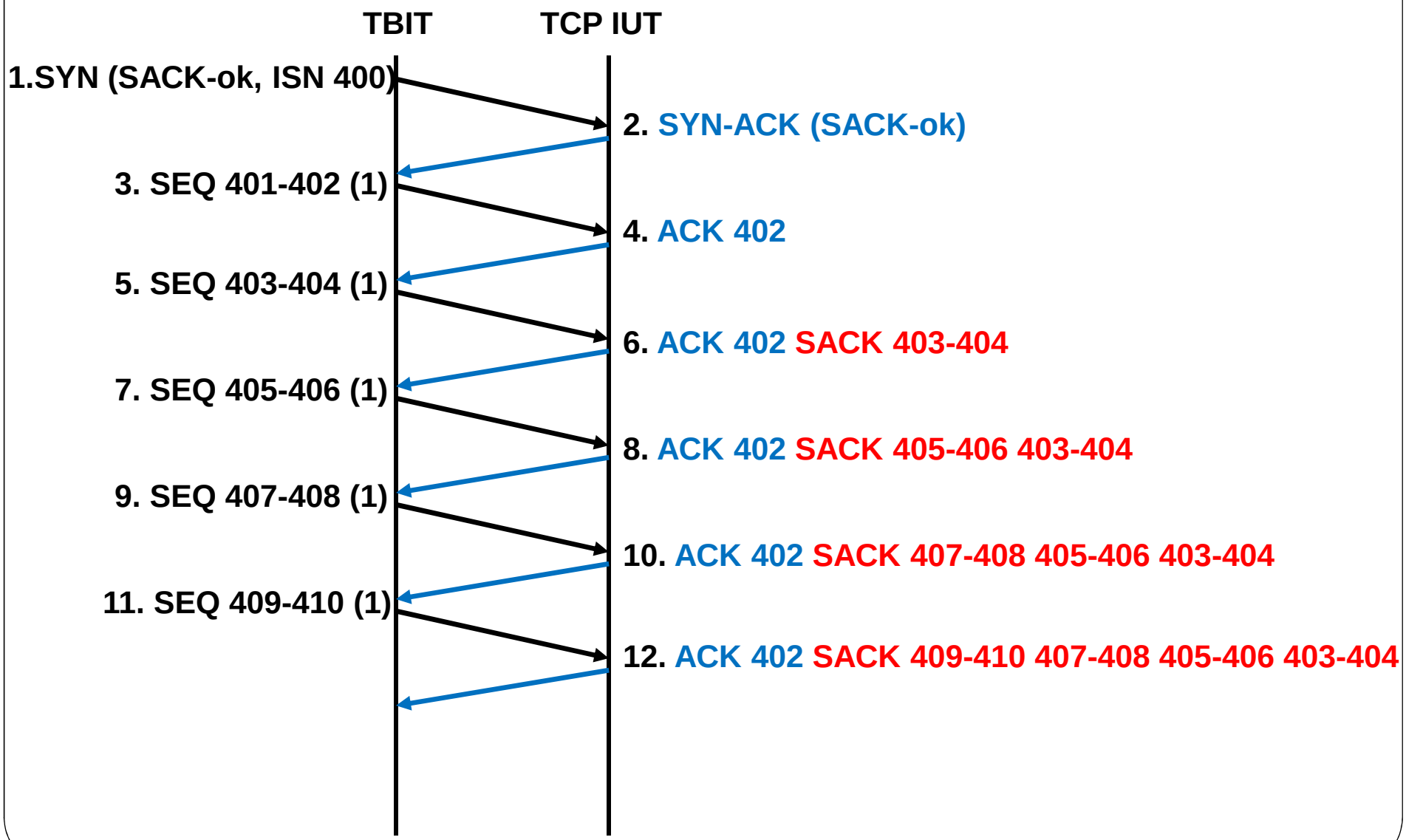


Misbehavior A:

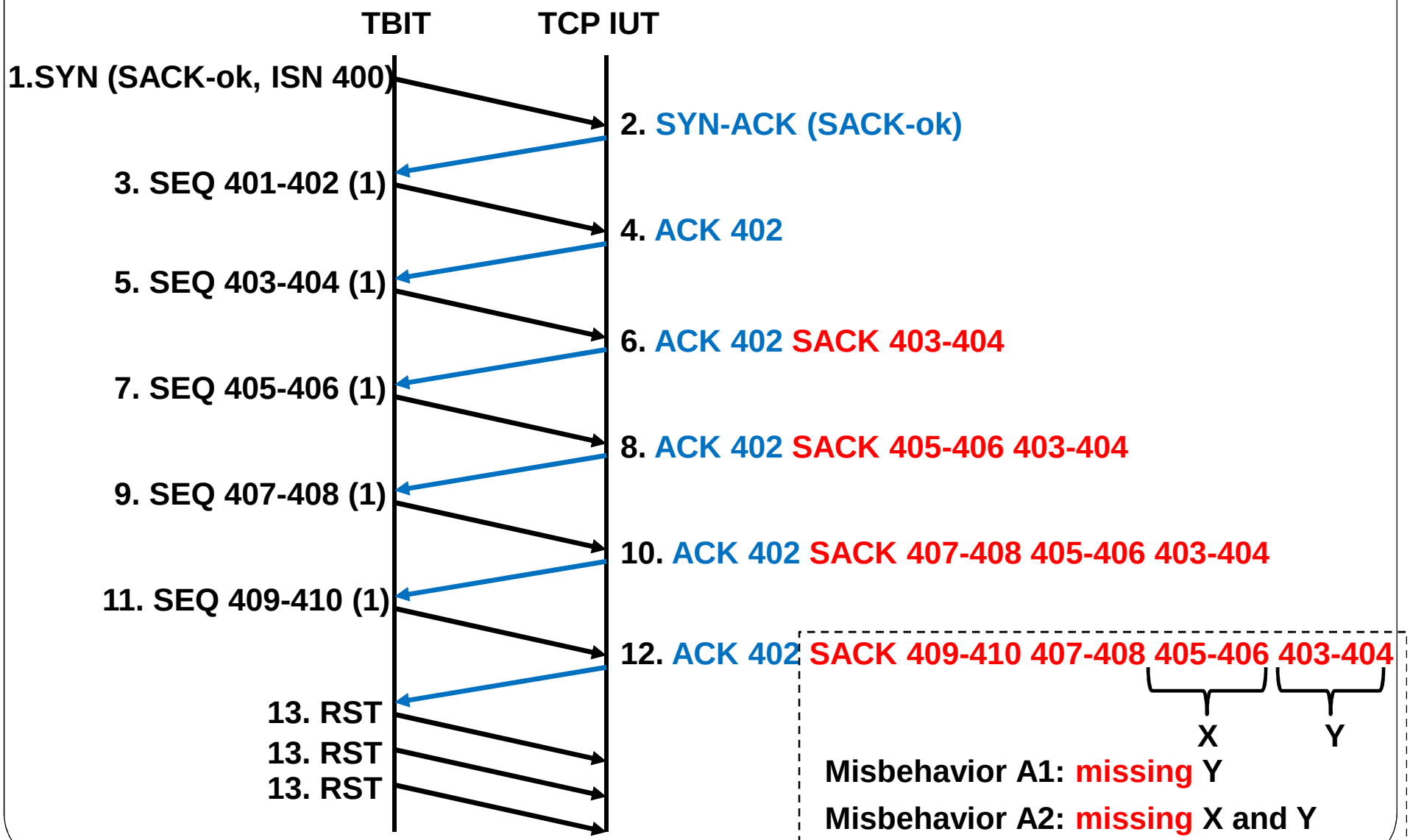
fewer than maximum number of reported SACKs



Misbehavior A:
fewer than maximum number of reported SACKs

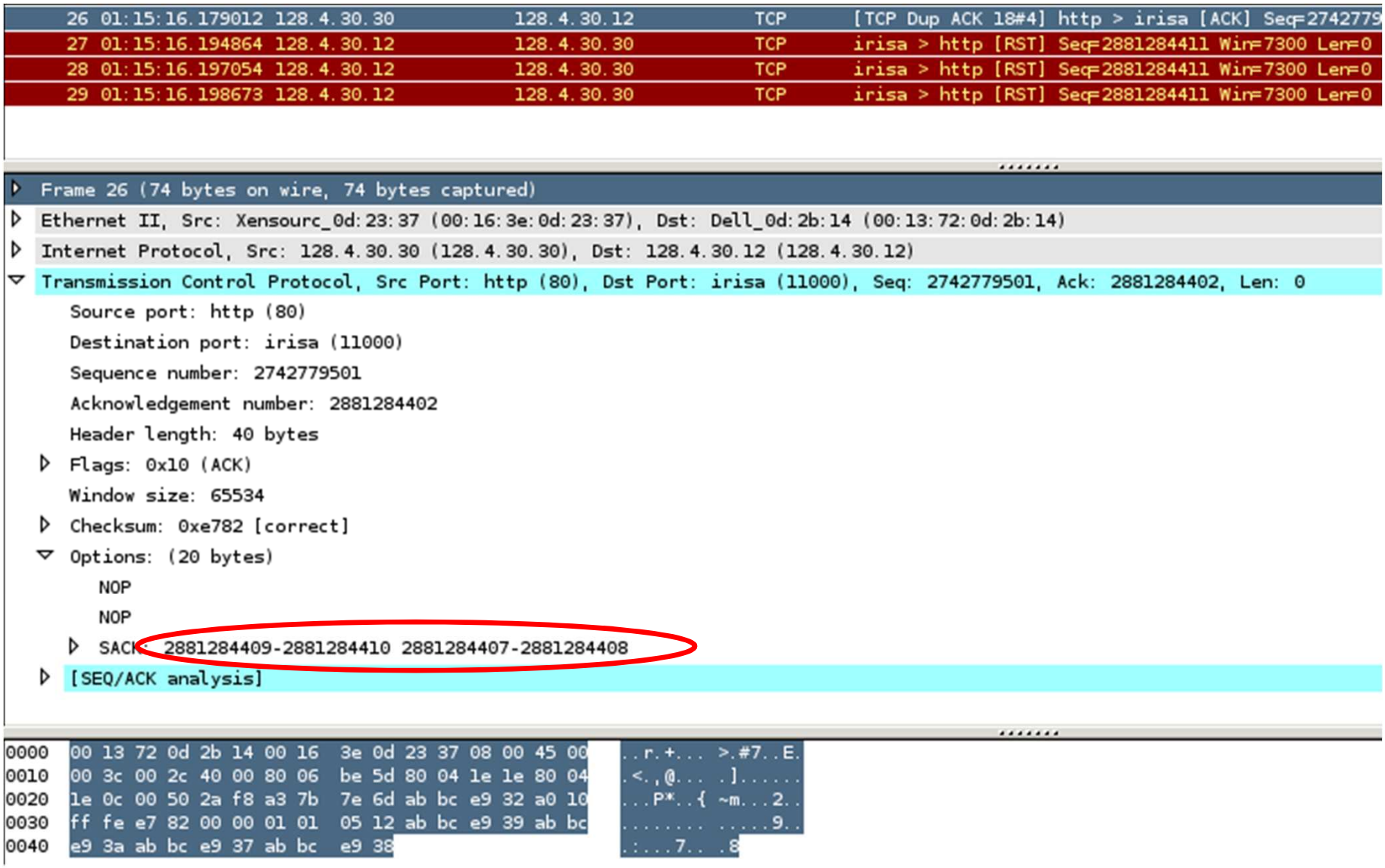


Misbehavior A:
fewer than maximum number of reported SACKs



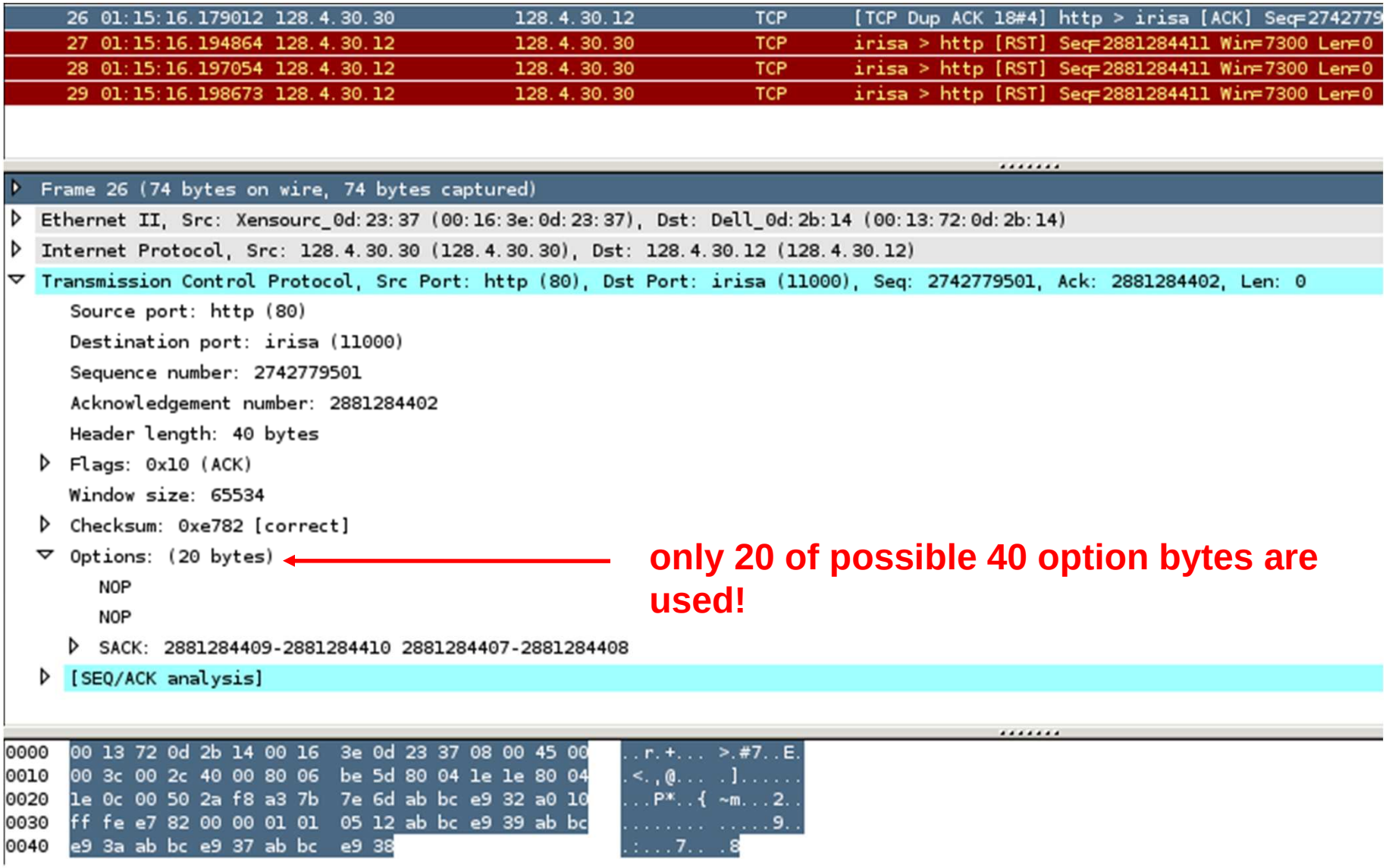
Misbehavior A:

fewer than maximum number of reported SACKs



Misbehavior A:

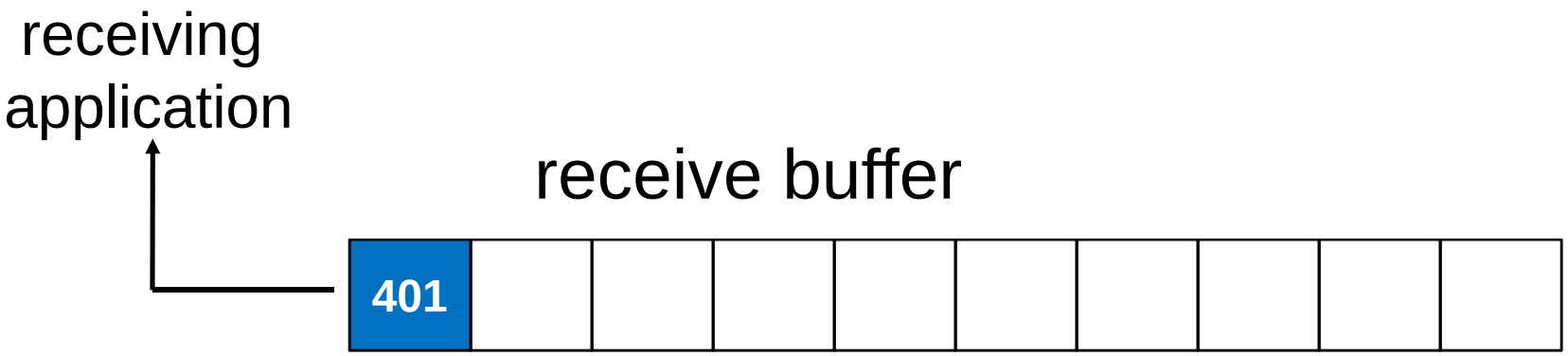
fewer than maximum number of reported SACKs



MISBEHAVIOR

B

Misbehavior B:
filling gap between Cum ACK and first SACK

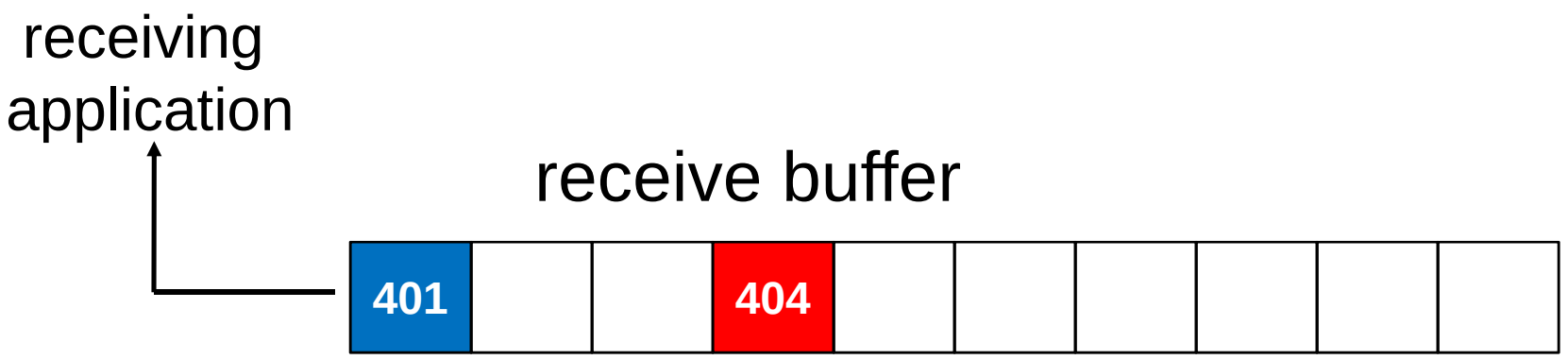


ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior B:
filling gap between Cum ACK and first SACK

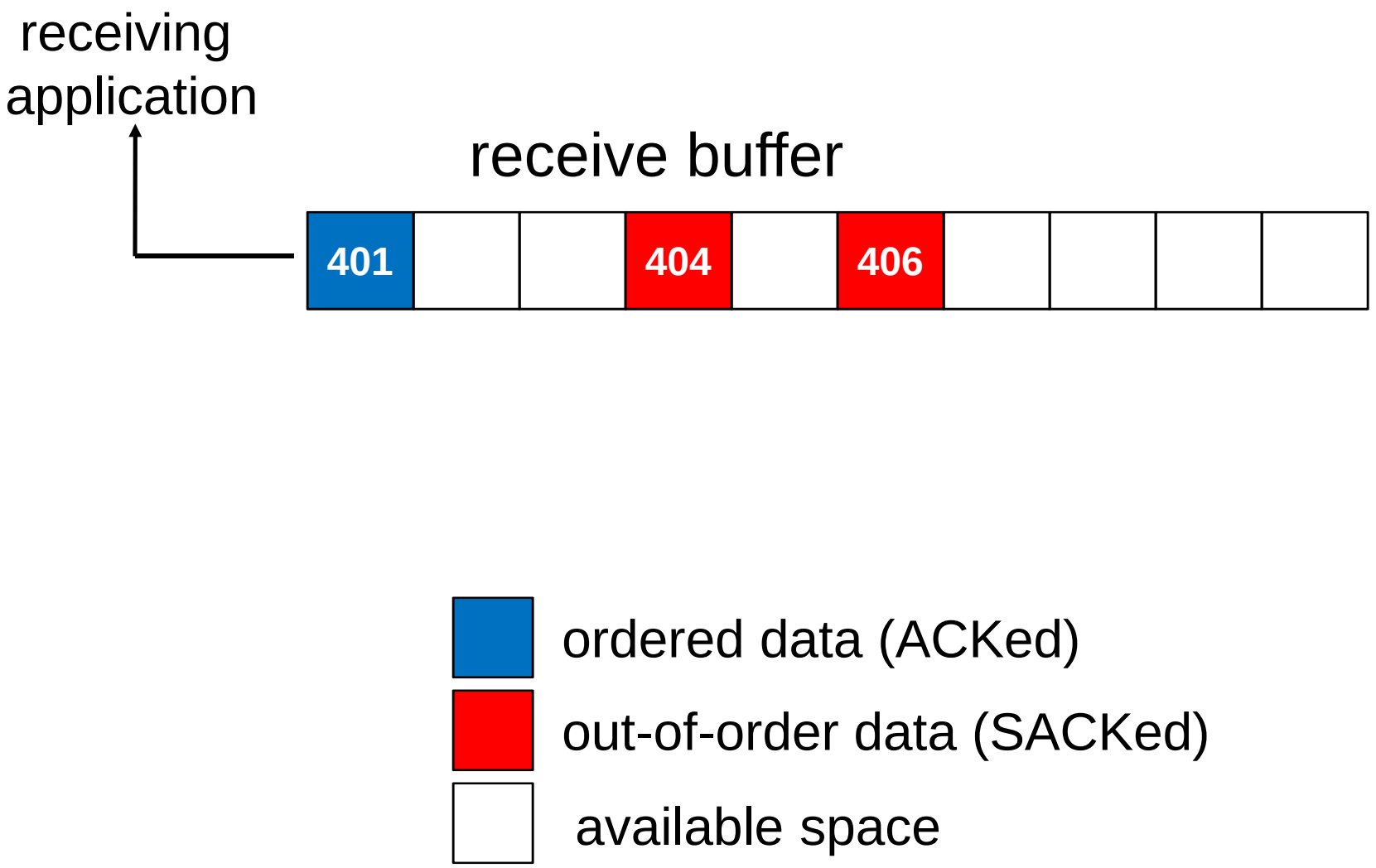


ordered data (ACKed)

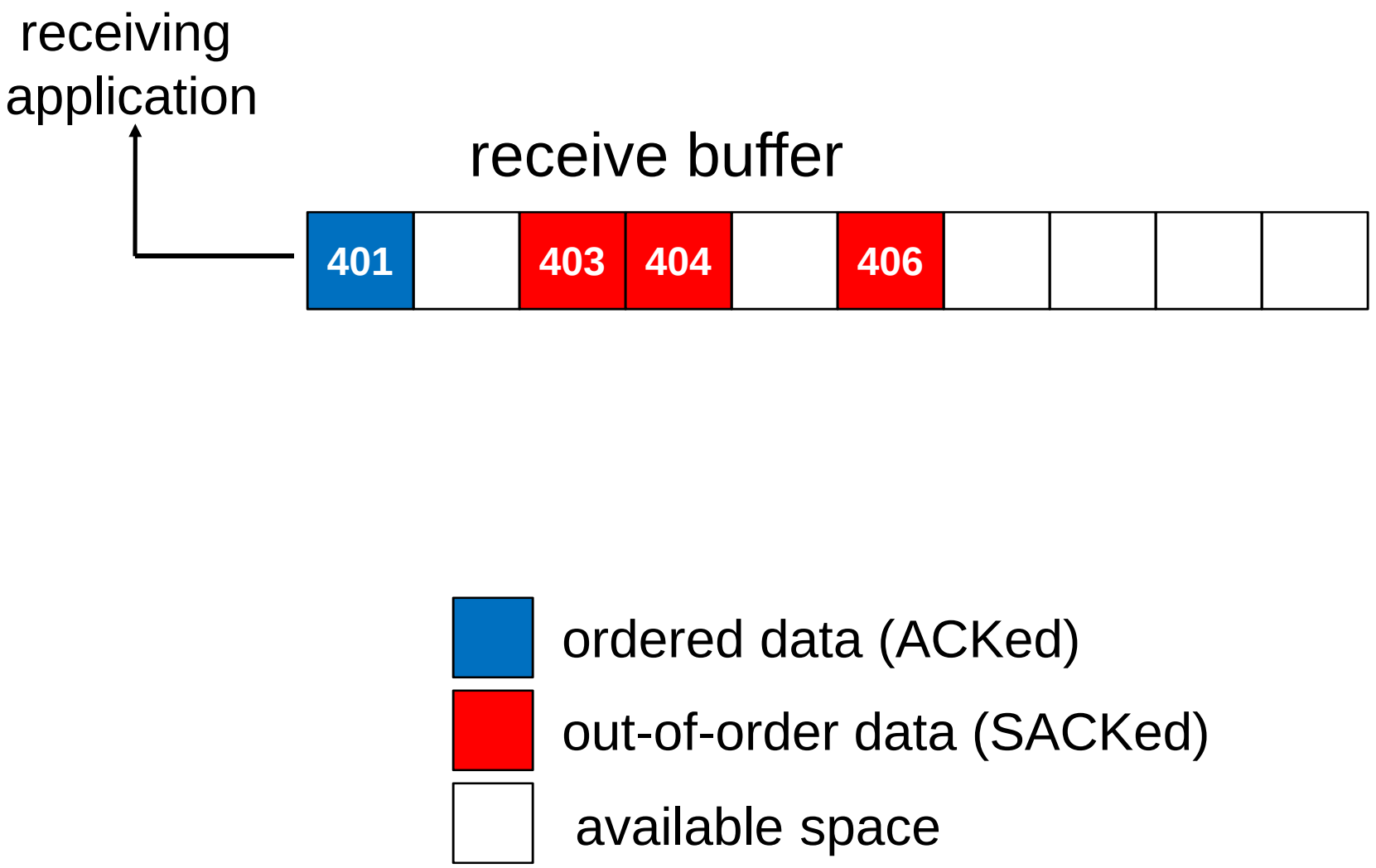
out-of-order data (SACKed)

available space

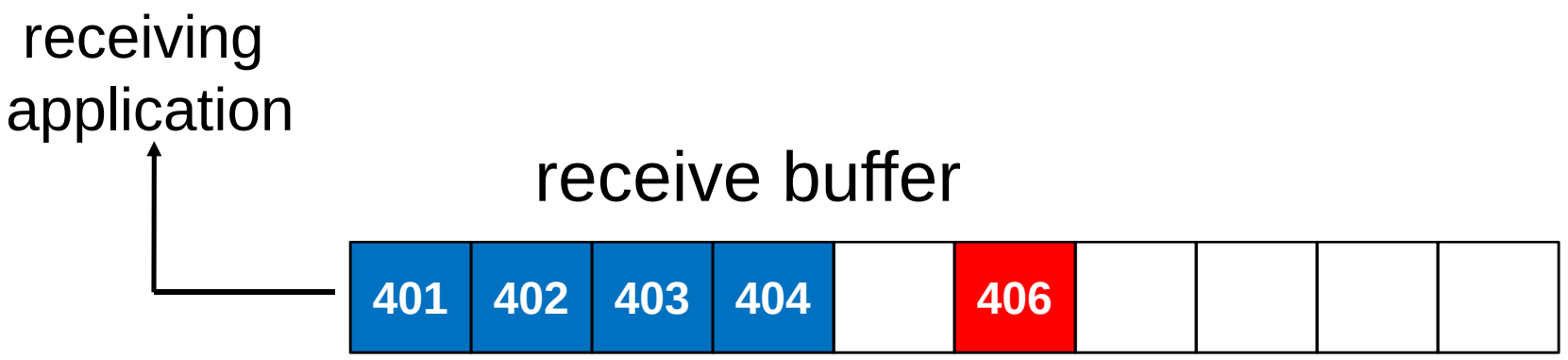
Misbehavior B:
filling gap between Cum ACK and first SACK



Misbehavior B:
filling gap between Cum ACK and first SACK



Misbehavior B:
filling gap between Cum ACK and first SACK

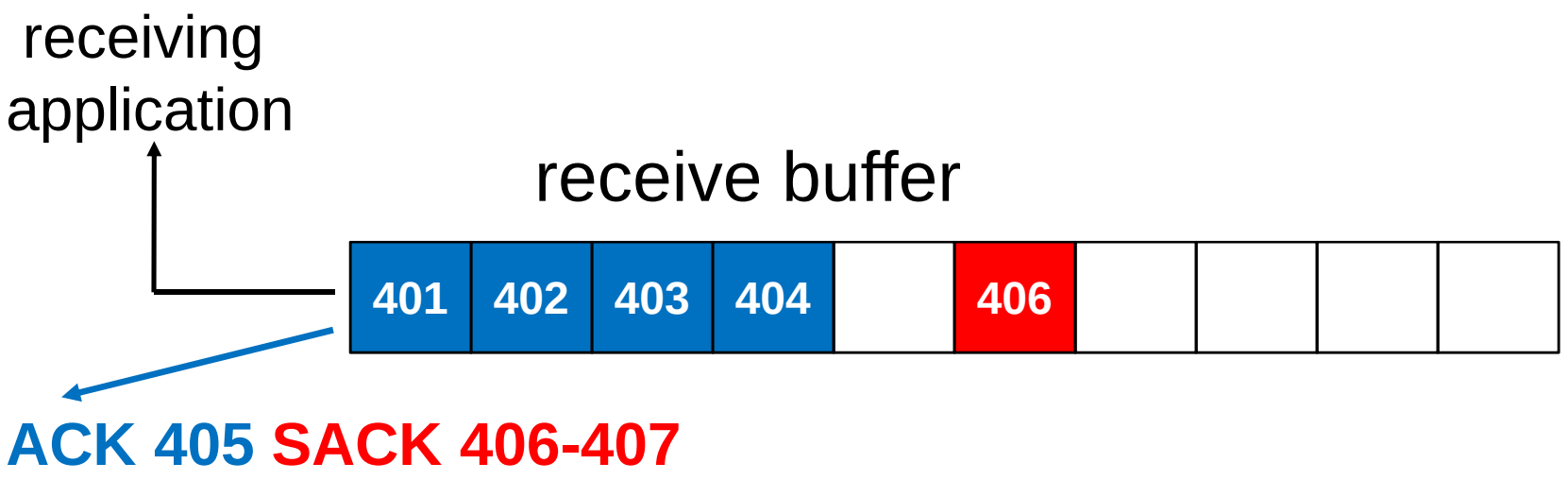


ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior B:
filling gap between Cum ACK and first SACK

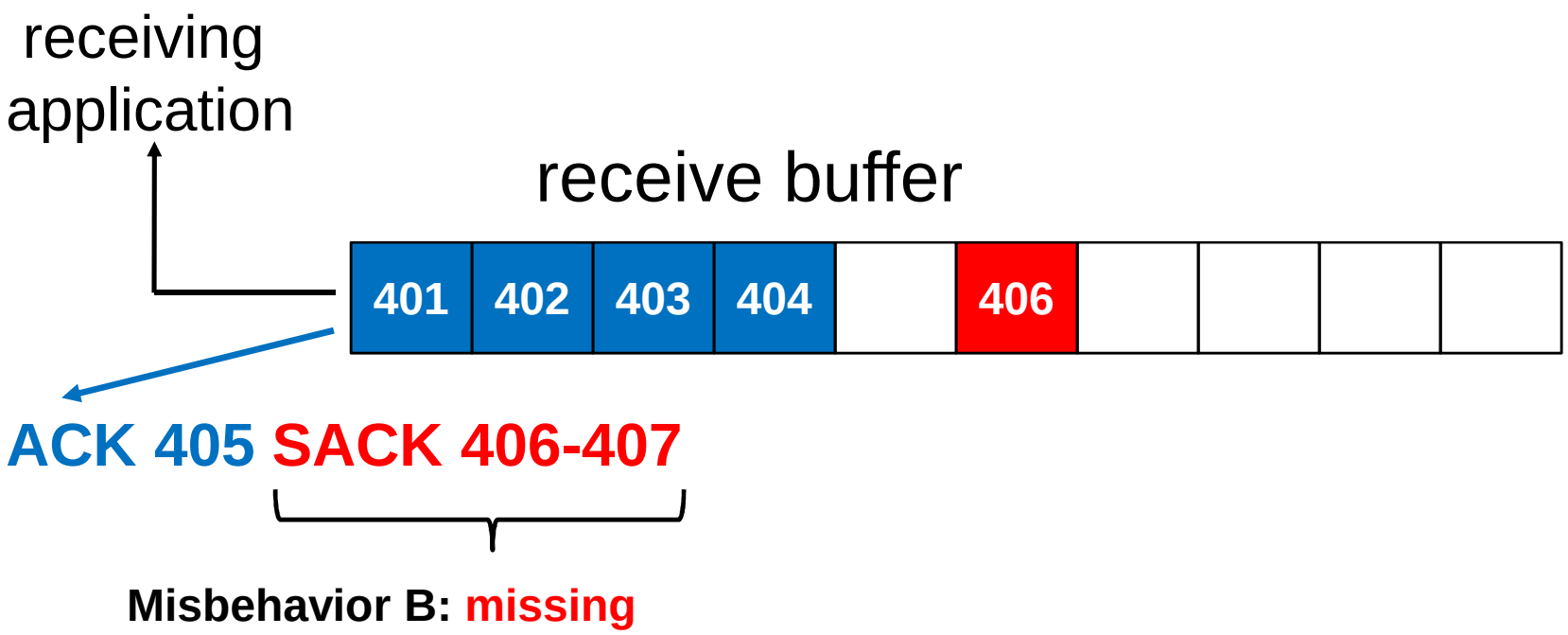


ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior B:
filling gap between Cum ACK and first SACK

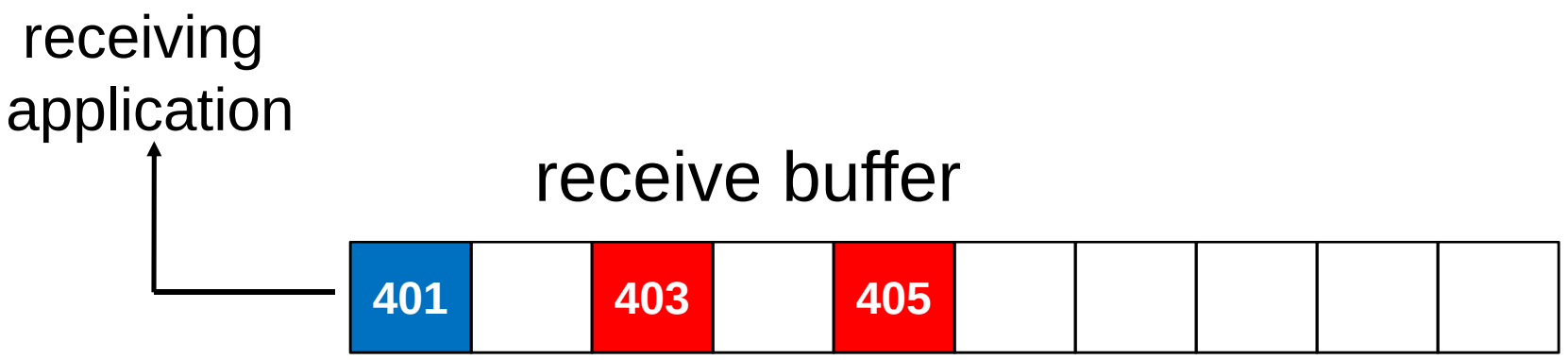


MISBEHAVIOR

C

Misbehavior C:

filling gap between two previously reported SACKs



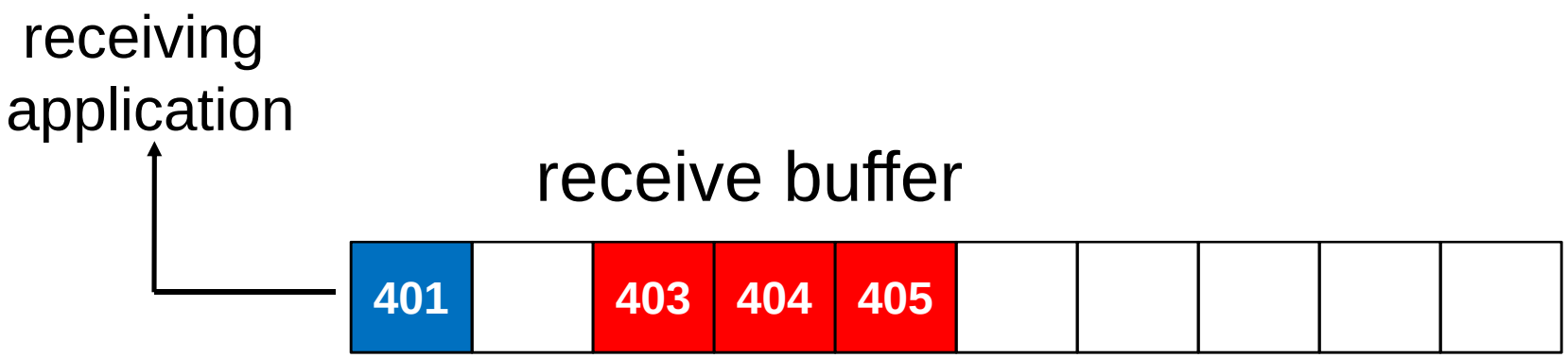
ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior C:

filling gap between two previously reported SACKs



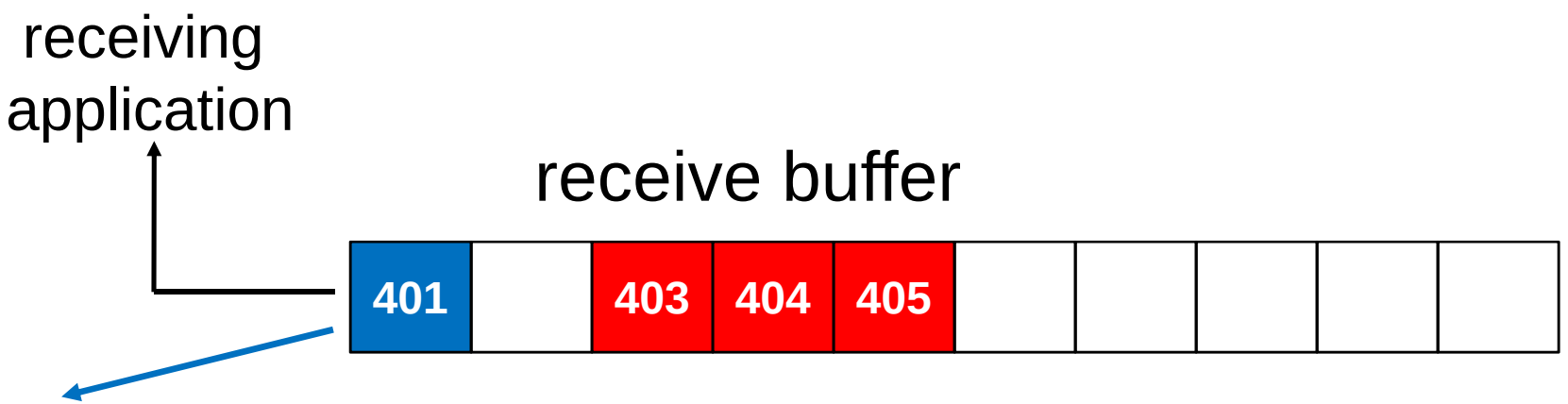
ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior C:

filling gap between two previously reported SACKs



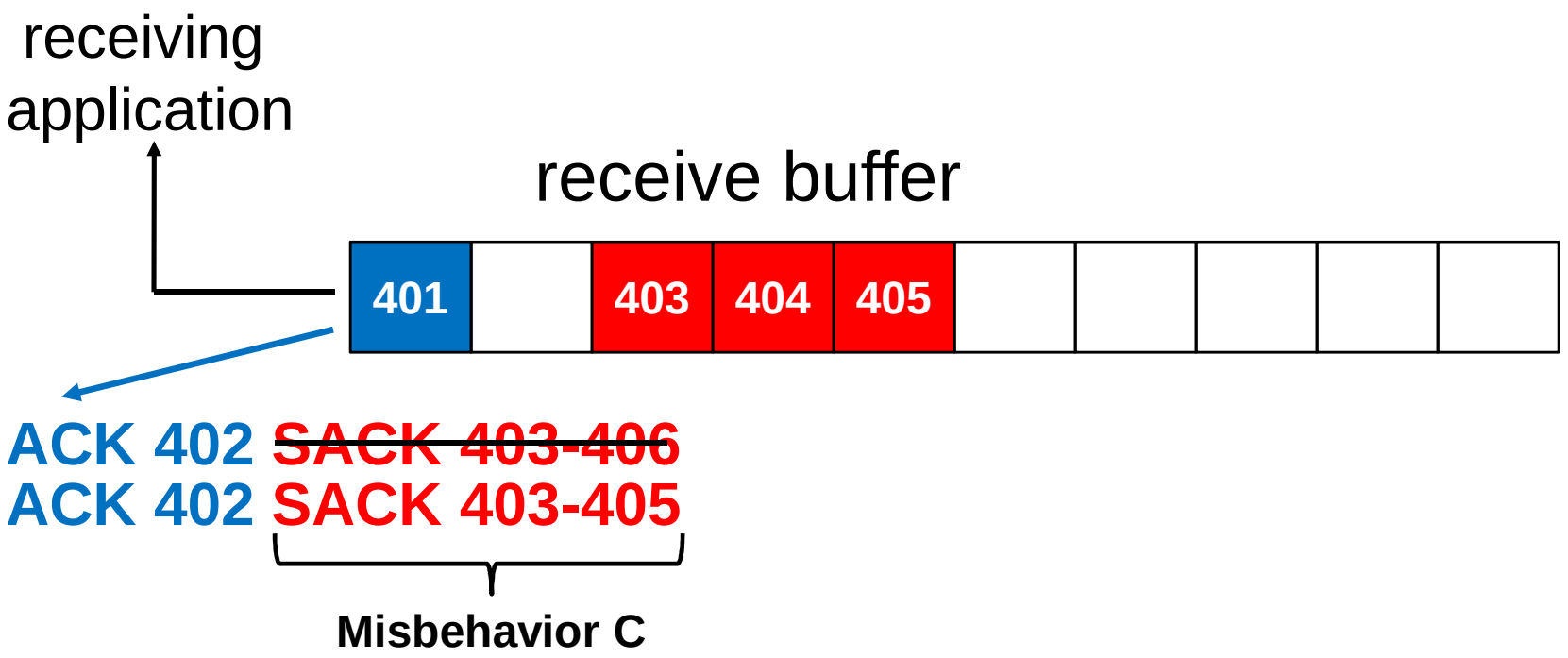
ordered data (ACKed)

out-of-order data (SACKed)

available space

Misbehavior C:

filling gap between two previously reported SACKs

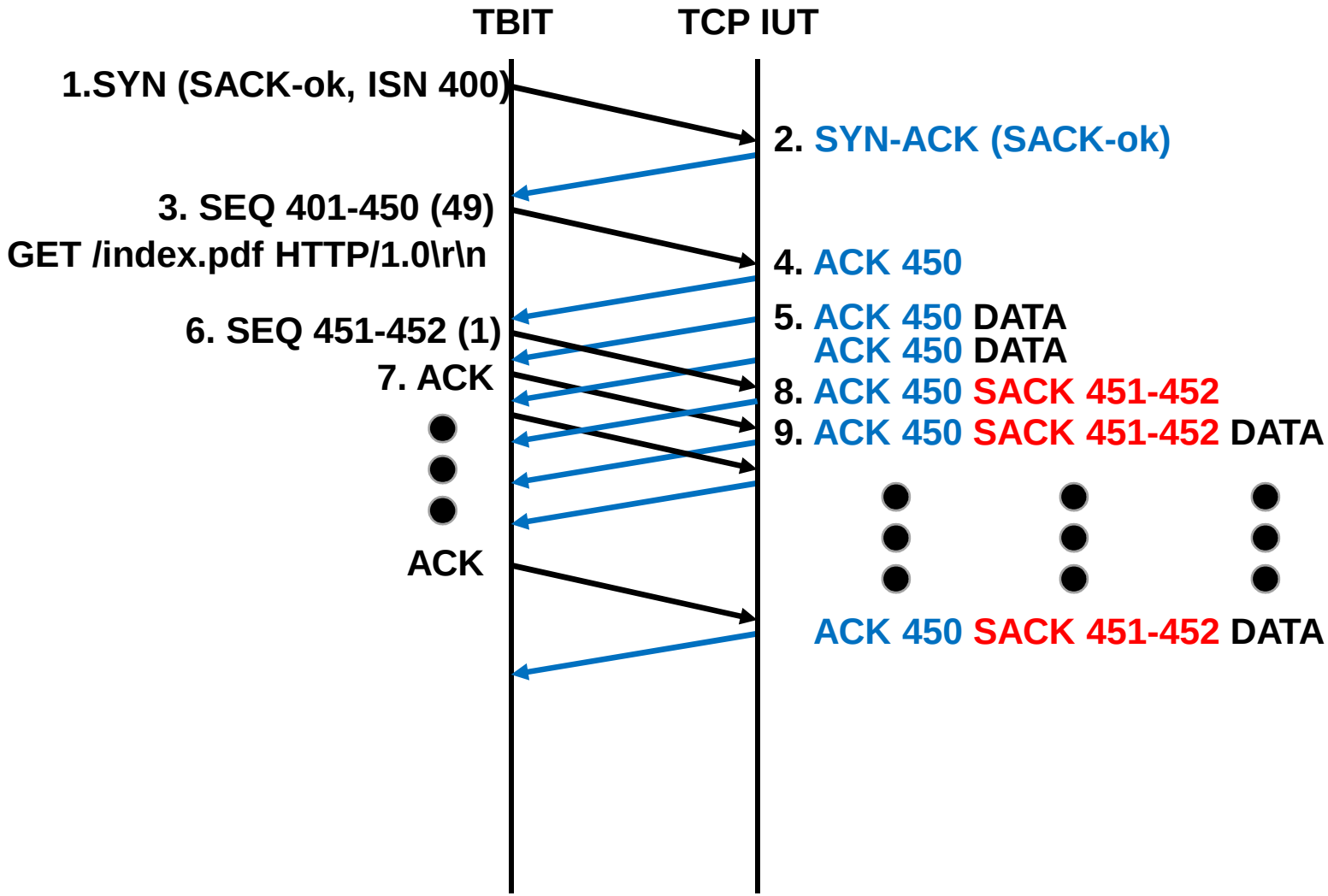


MISBEHAVIOR

D

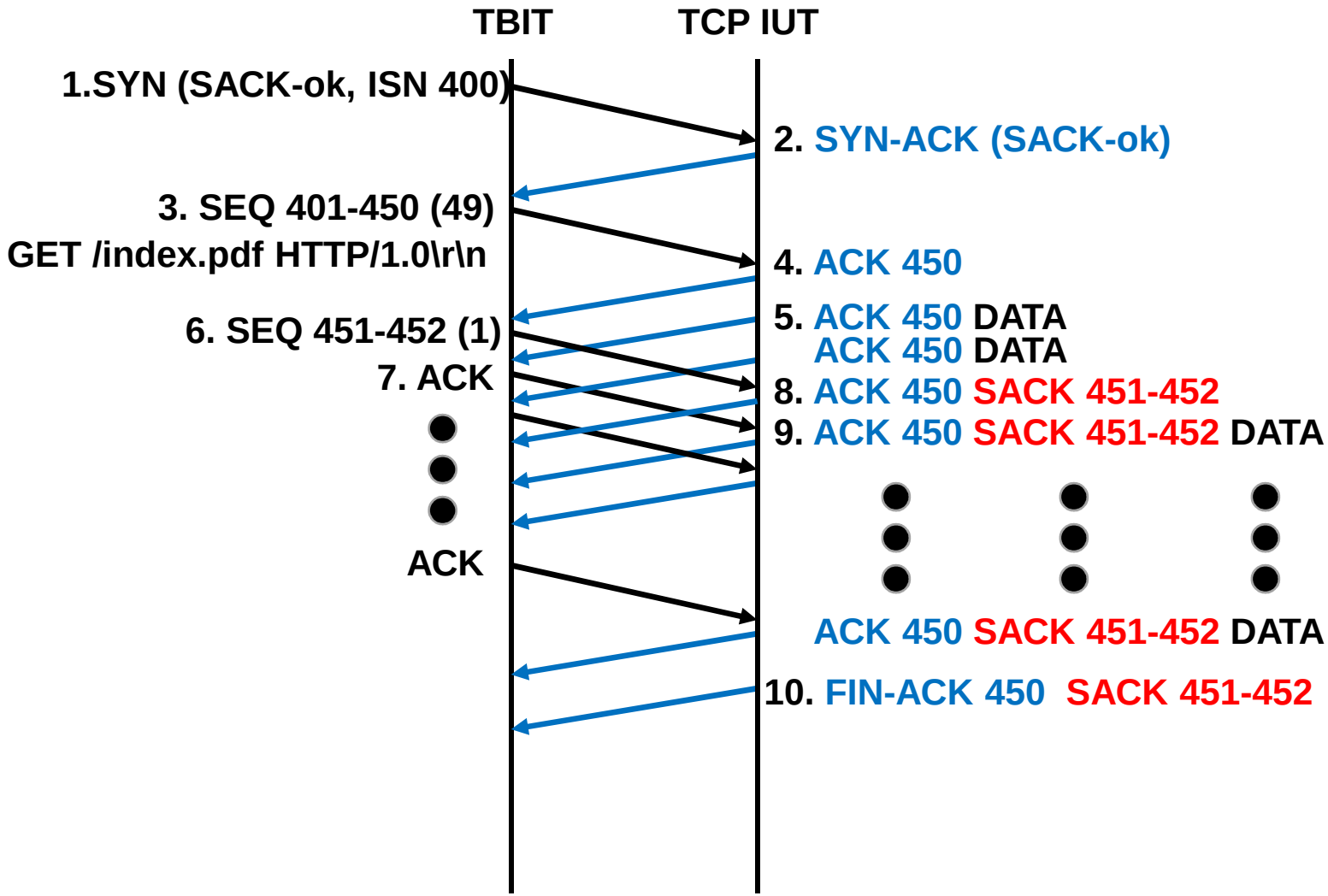
Misbehavior D:

failure to report SACKs in FIN segments



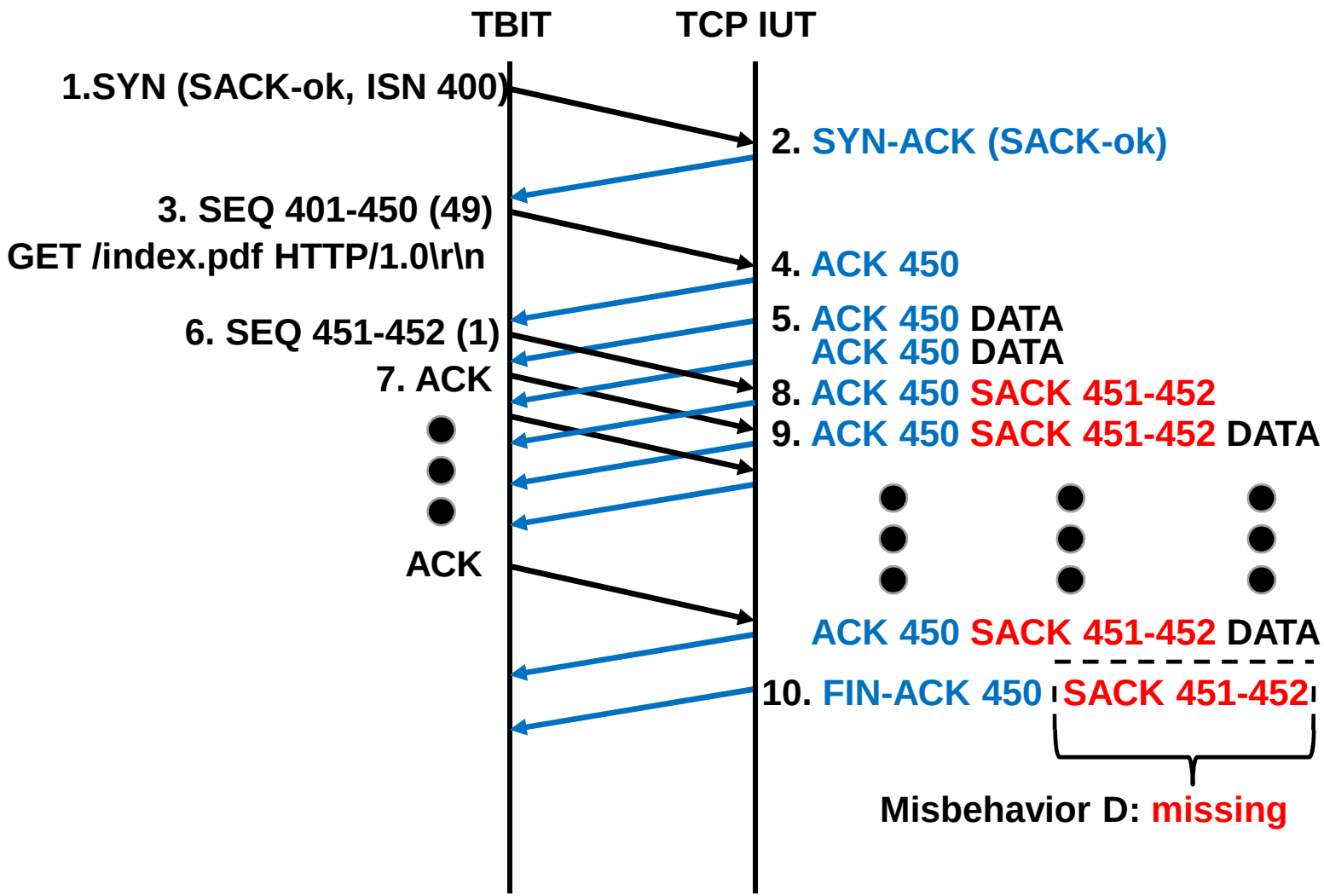
Misbehavior D:

failure to report SACKs in FIN segments



Misbehavior D:

failure to report SACKs in FIN segments

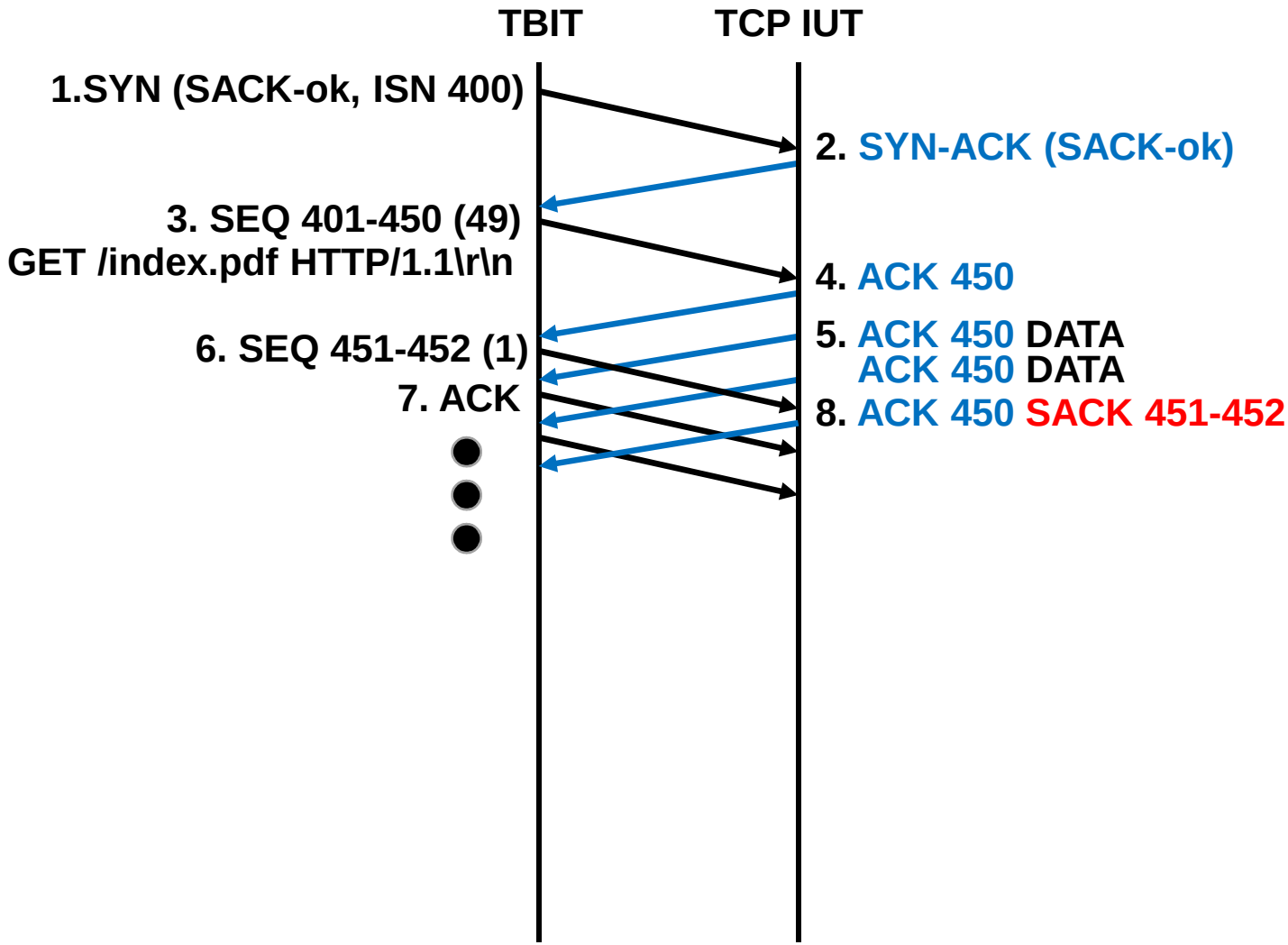


MISBEHAVIOR

E

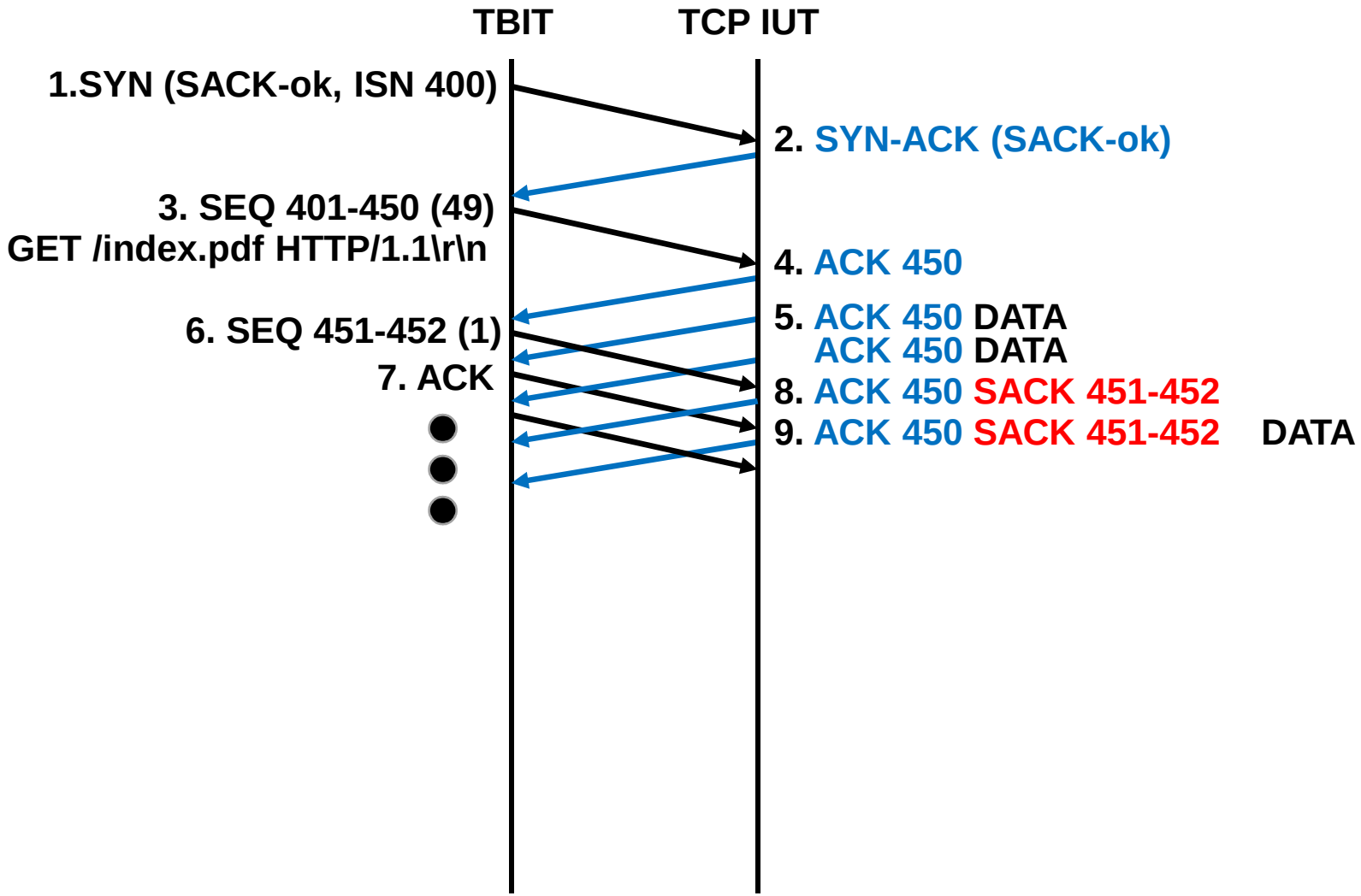
Misbehavior E:

failure to report SACKs in bidirectional data flow



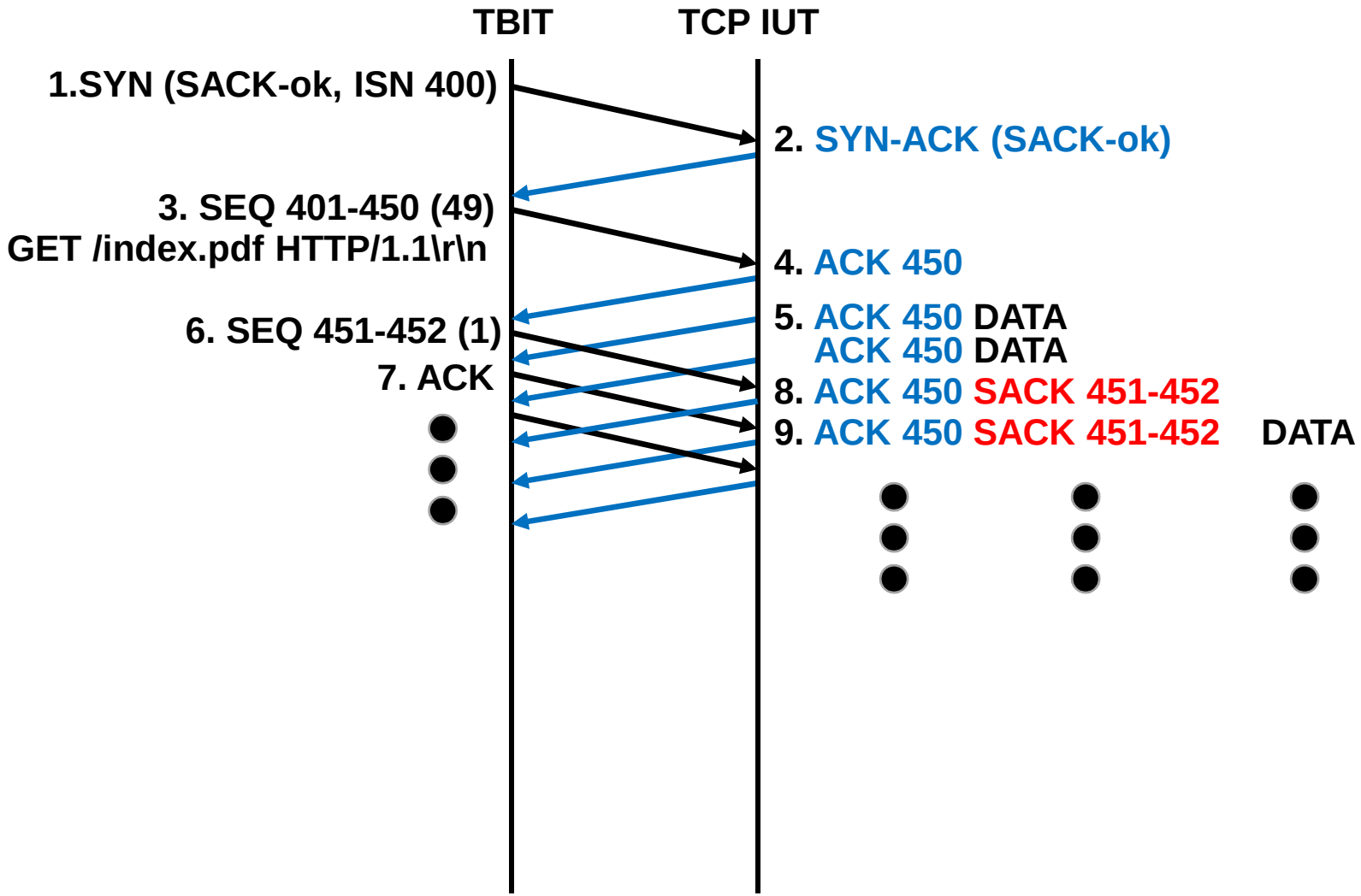
Misbehavior E:

failure to report SACKs in bidirectional data flow



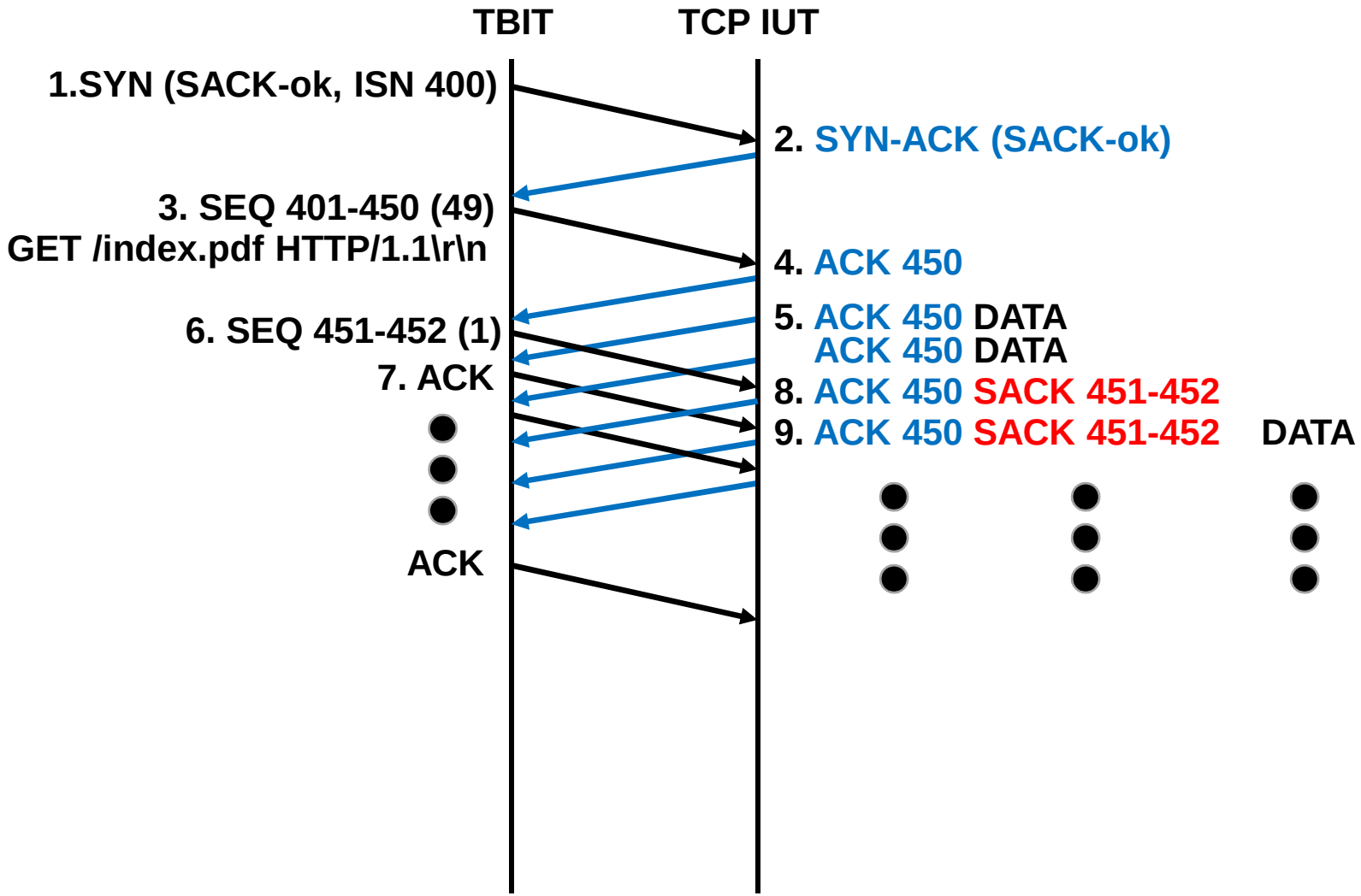
Misbehavior E:

failure to report SACKs in bidirectional data flow



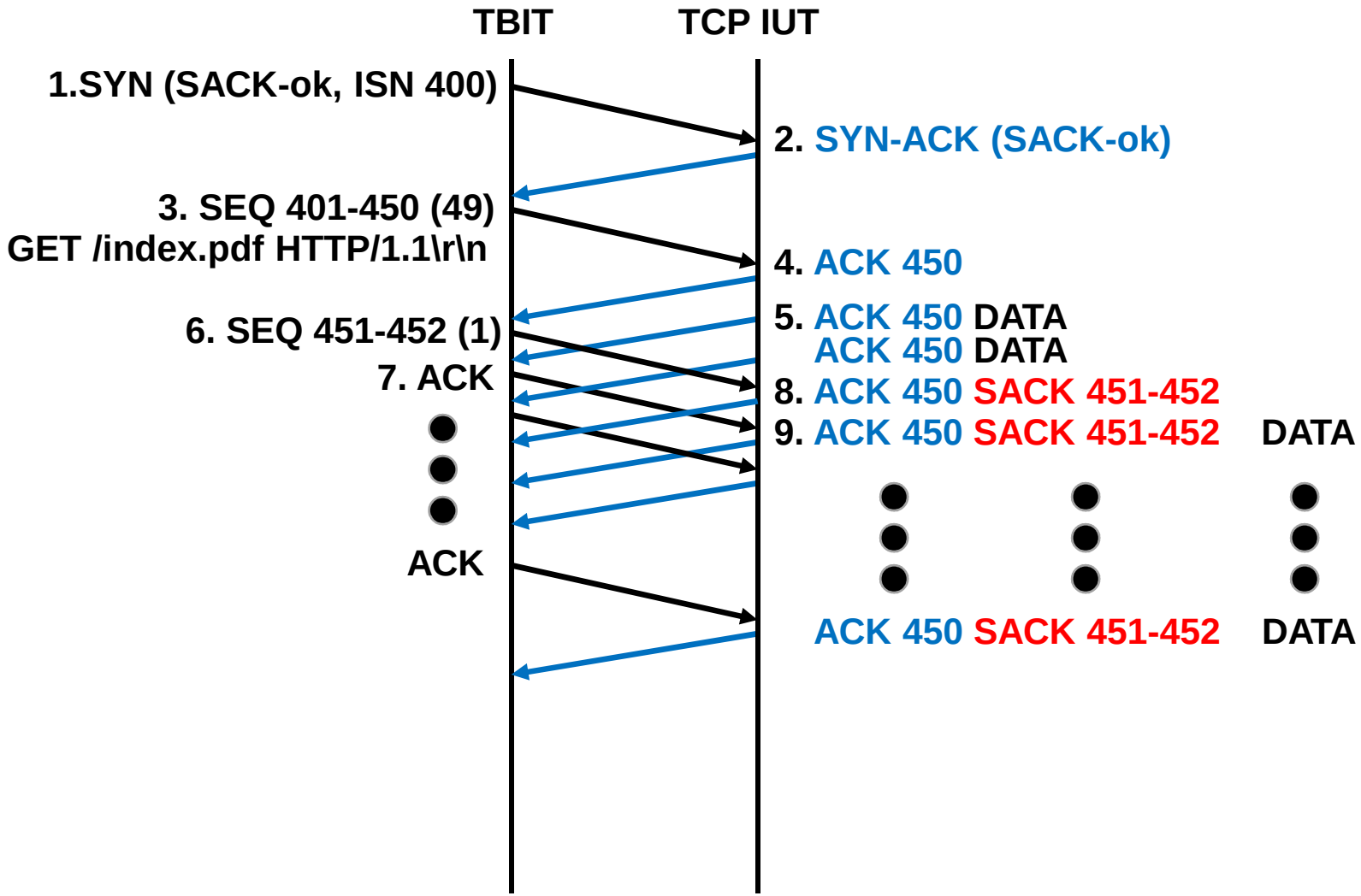
Misbehavior E:

failure to report SACKs in bidirectional data flow



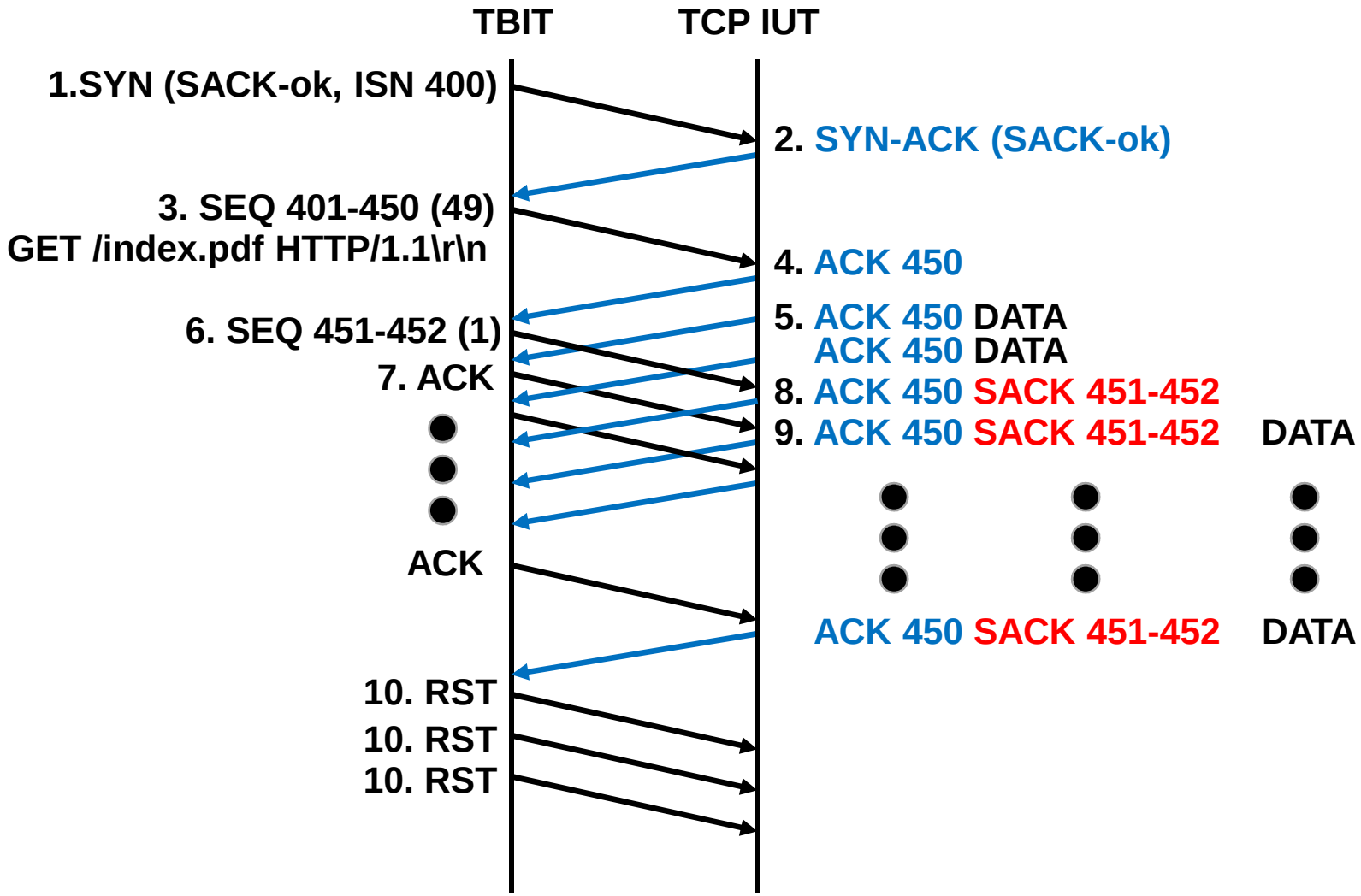
Misbehavior E:

failure to report SACKs in bidirectional data flow



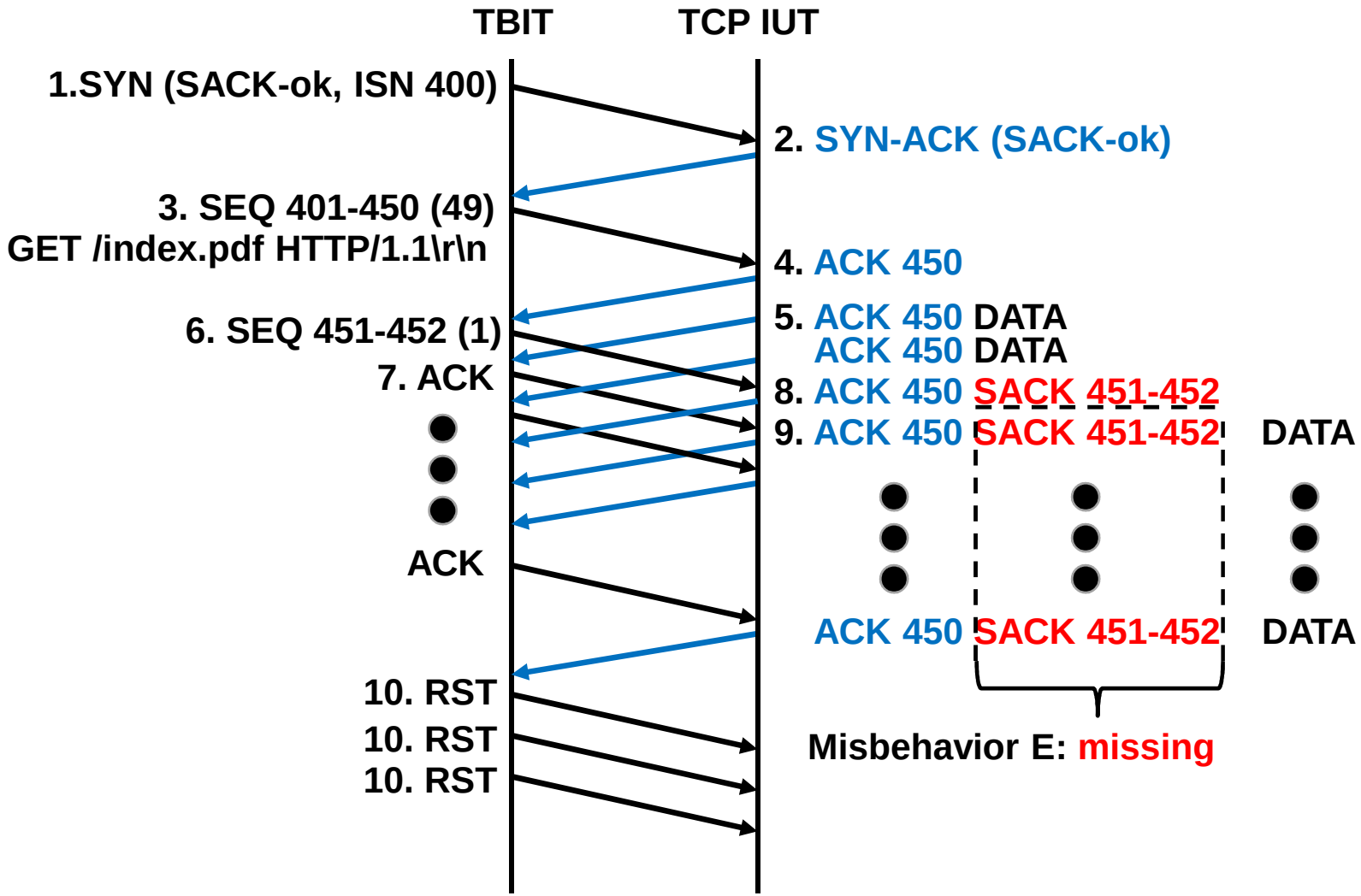
Misbehavior E:

failure to report SACKs in bidirectional data flow



Misbehavior E:

failure to report SACKs in bidirectional data flow

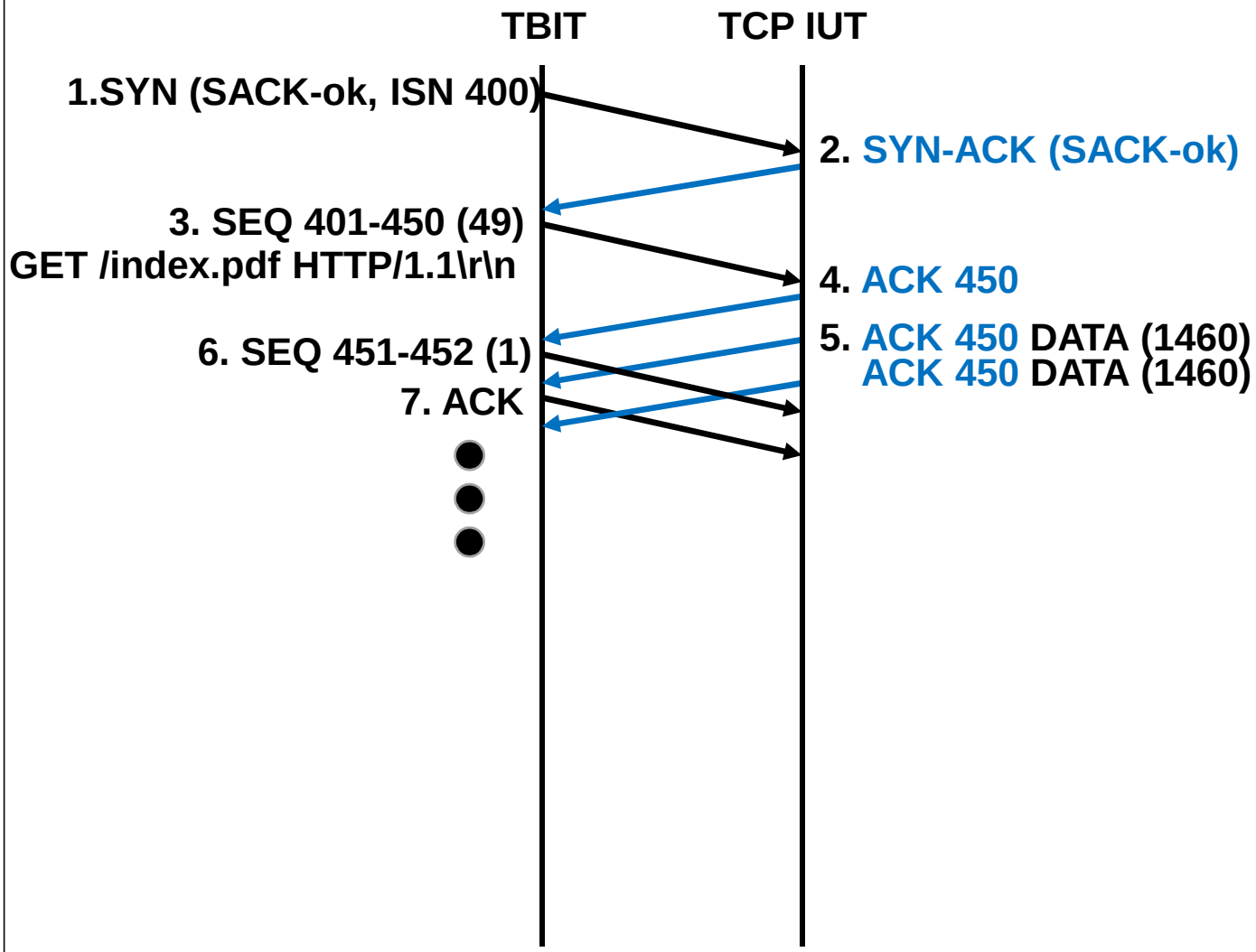


MISBEHAVIOR

F

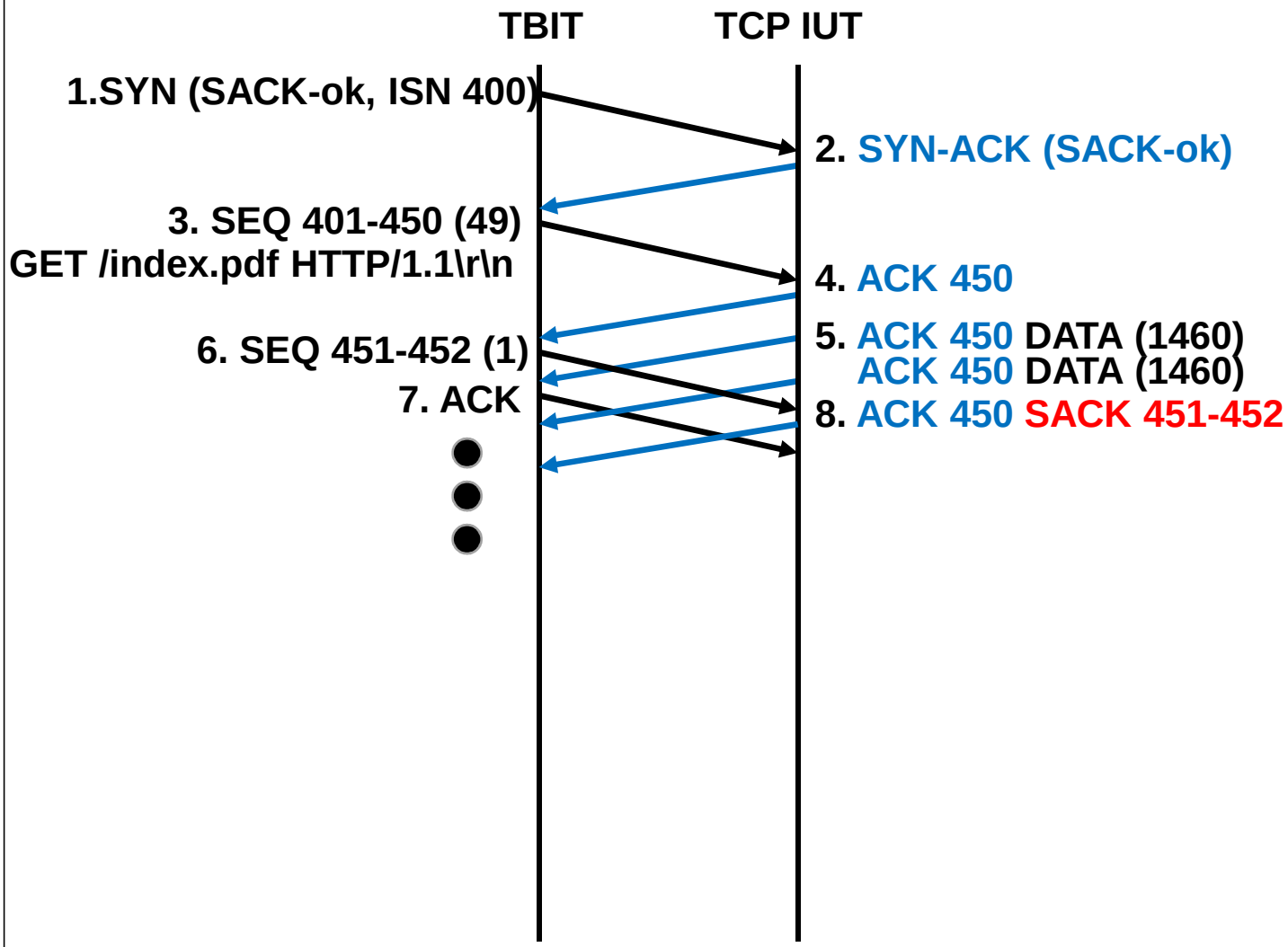
Misbehavior F:

mishandling of data due to SACK processing



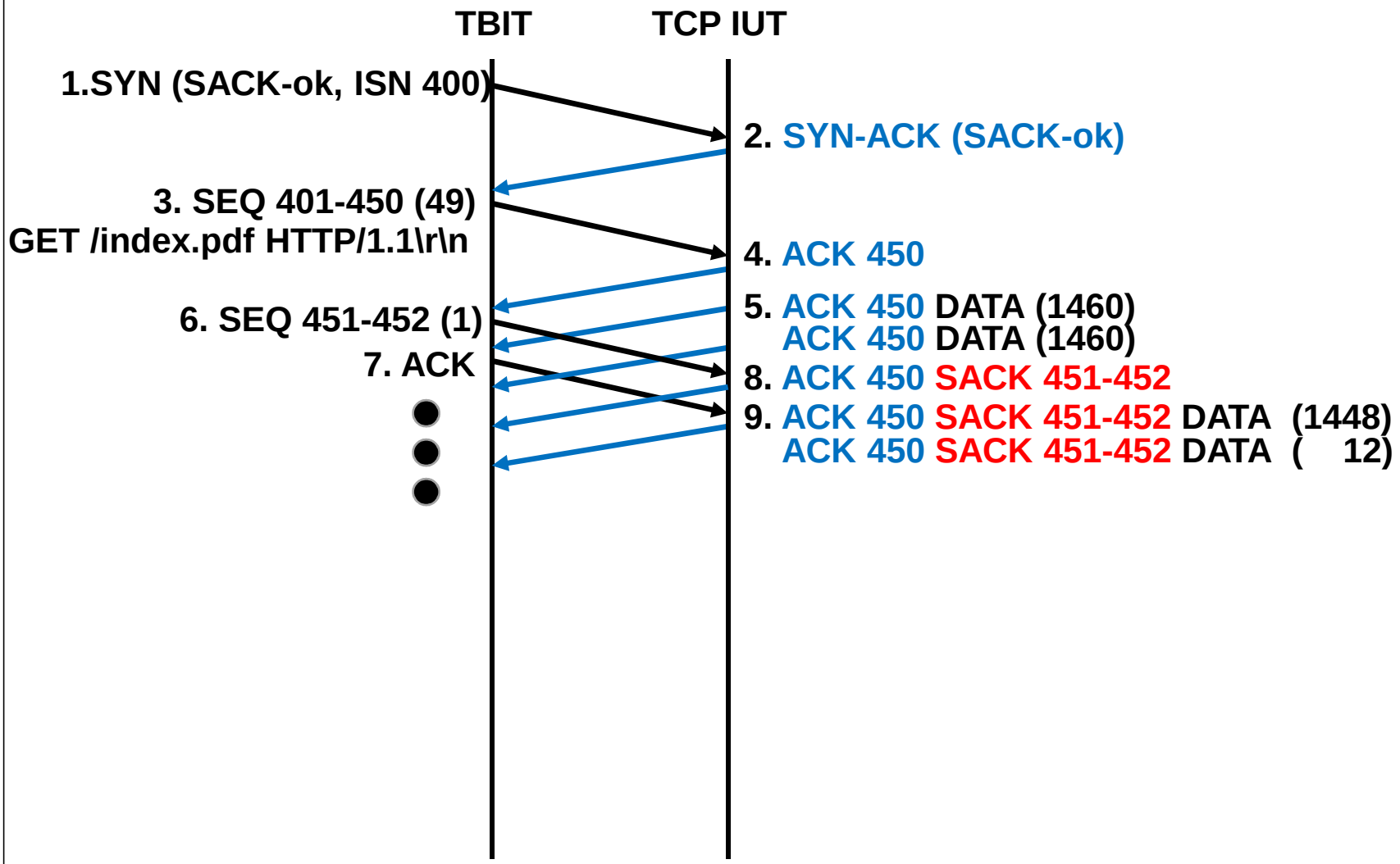
Misbehavior F:

mishandling of data due to SACK processing



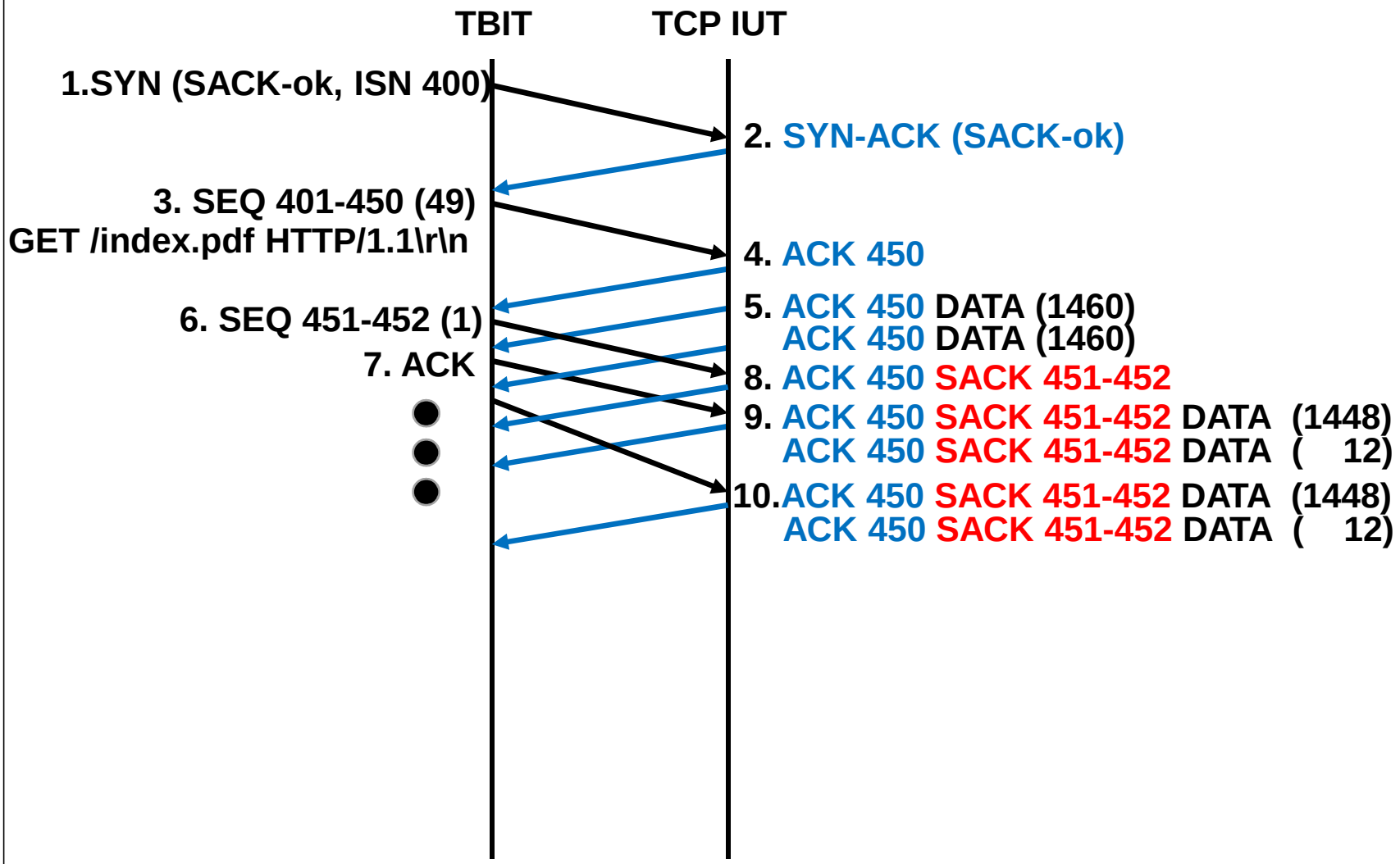
Misbehavior F:

mishandling of data due to SACK processing



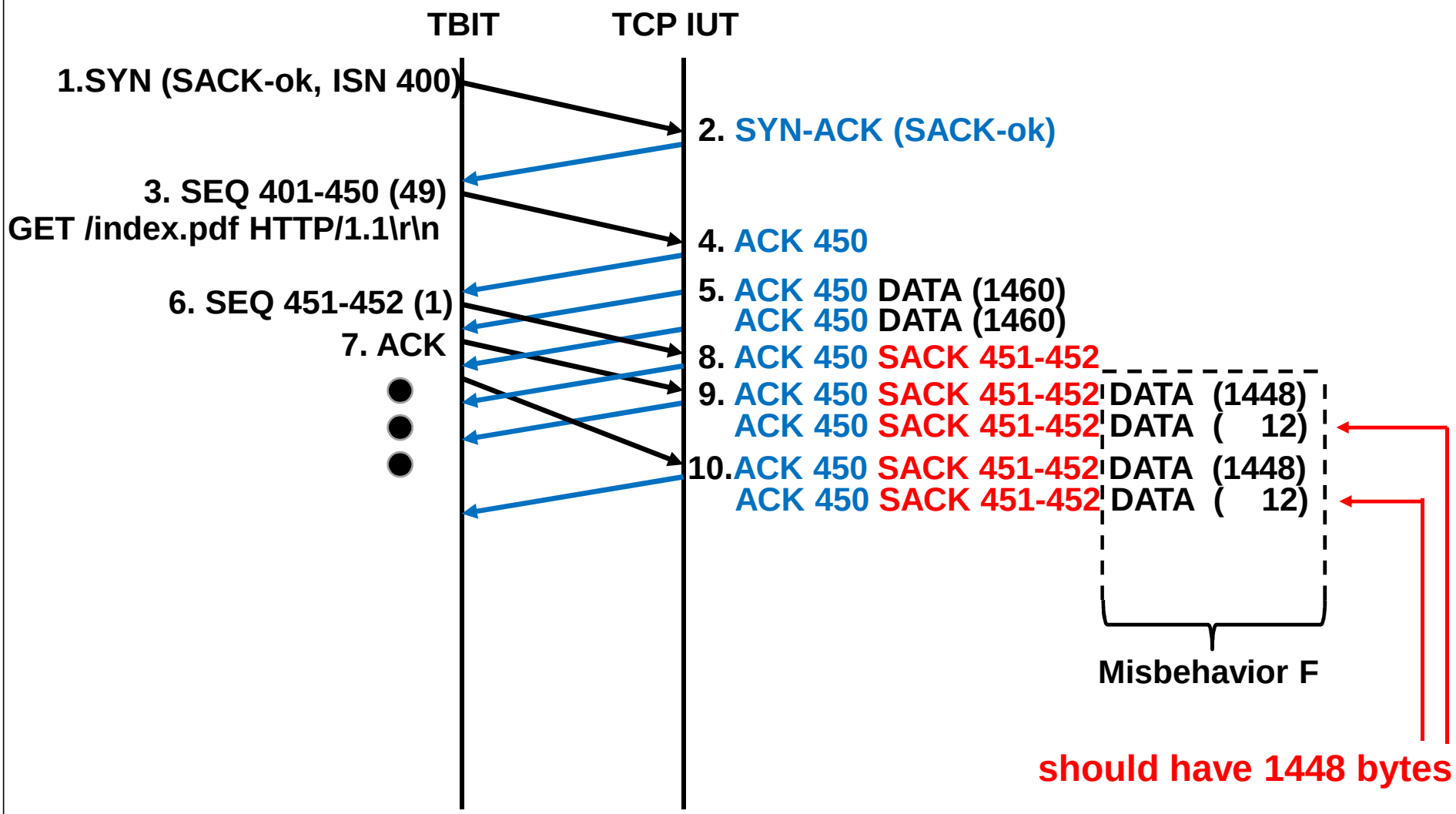
Misbehavior F:

mishandling of data due to SACK processing



Misbehavior F:

mishandling of data due to SACK processing

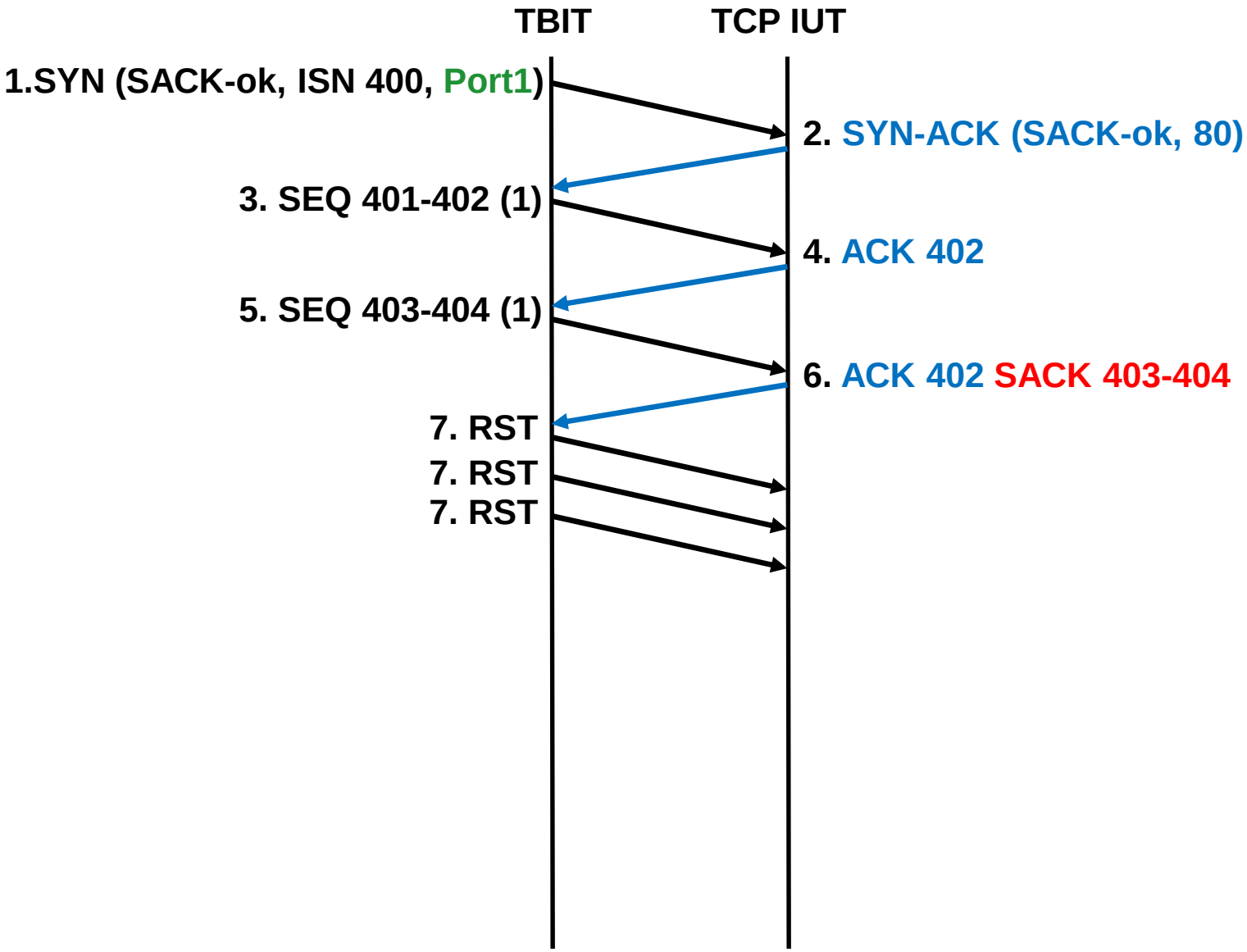


MISBEHAVIOR

G

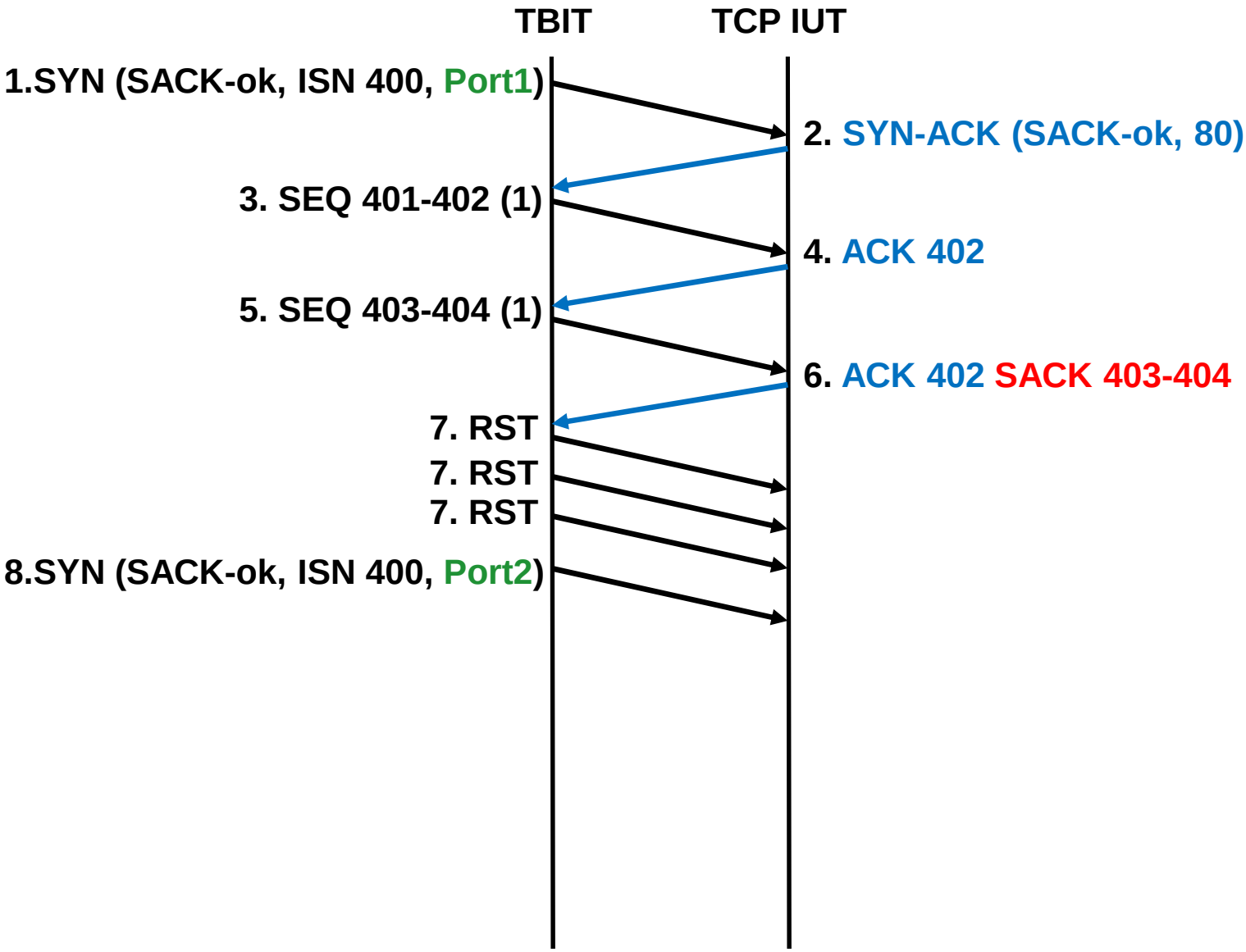
Misbehavior G:

SACK reappearance in consecutive connections



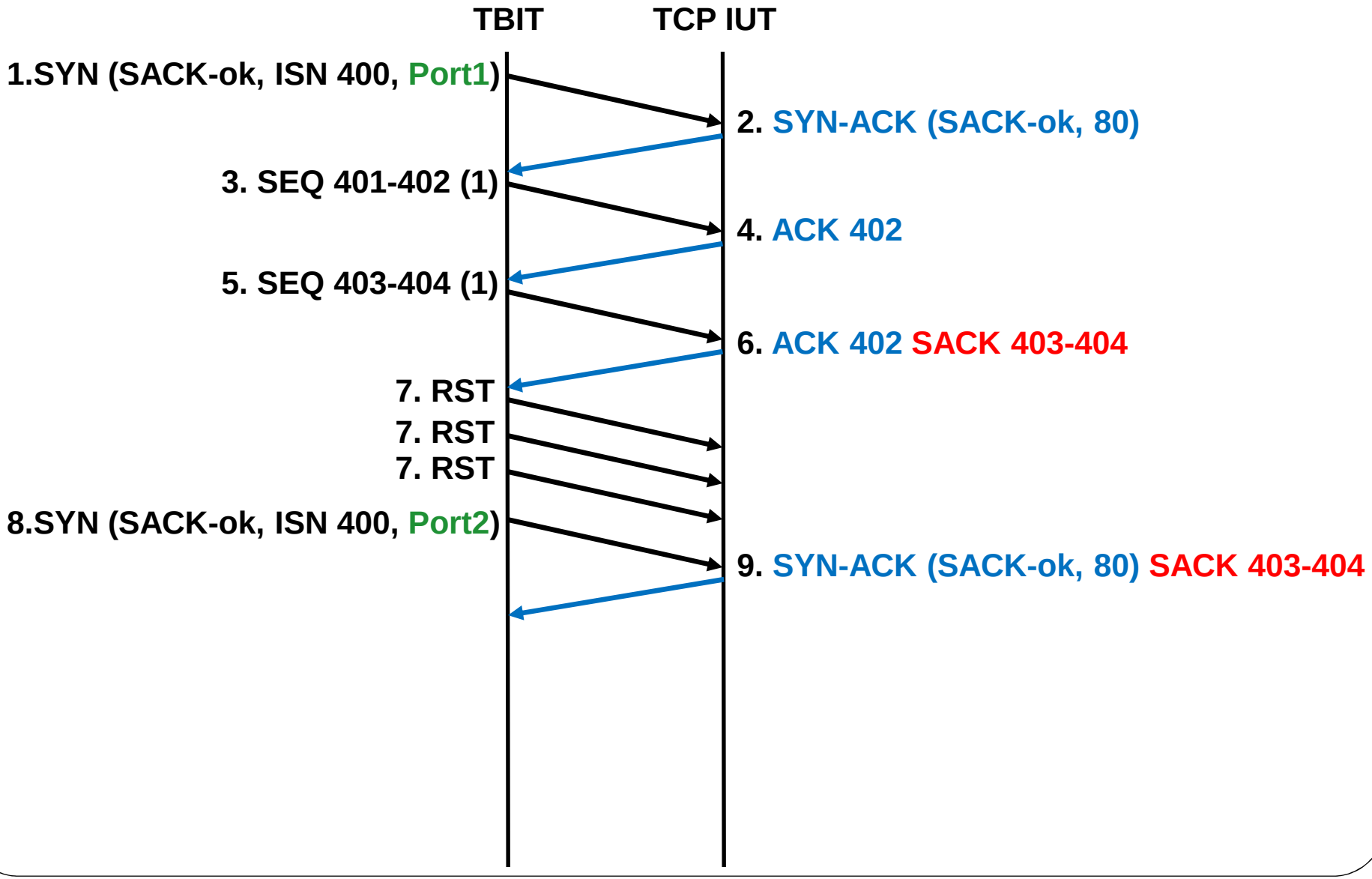
Misbehavior G:

SACK reappearance in consecutive connections



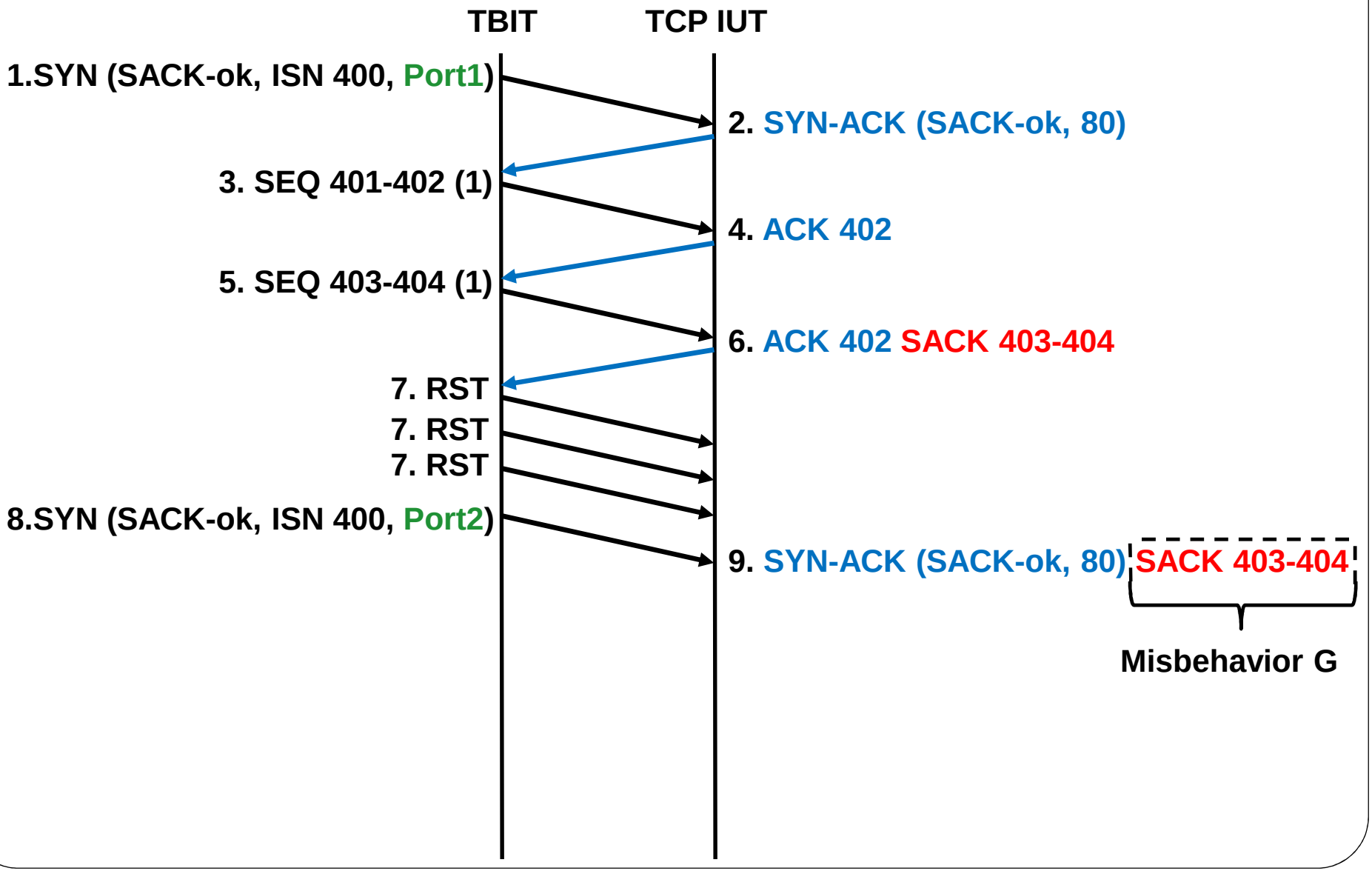
Misbehavior G:

SACK reappearance in consecutive connections



Misbehavior G:

SACK reappearance in consecutive connections



OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

Results

Operating System	Misbehavior						
	A	*B*	*C*	*D*	*E*	*F*	*G*
FreeBSD 5.3, 5.4	Y			Y			
Linux 2.2.20 (Debian 3)						Y	
Linux 2.4.18 (Red Hat 8)						Y	
Linux 2.4.22 (Fedora 1)						Y	
Linux 2.6.12 (Ubuntu 5.10)						Y	
Linux 2.6.15 (Ubuntu 6.06)						Y	
Linux 2.6.18 (Debian 4)						Y	
OpenBSD 4.2, 4.5, 4.6, 4.7	Y			Y			
OpenSolaris 2008.05						Y	Y
OpenSolaris 2009.06						Y	Y
Solaris 10							Y
Solaris 11						Y	
Windows 2000	Y	Y	Y	Y	Y		
Windows XP	Y	Y	Y	Y	Y		
Windows Server 2003	Y	Y	Y	Y	Y		
Windows Vista				Y	Y		
Windows Server 2008				Y	Y		
Windows 7				Y	Y		

Results

Operating System	Misbehavior						
	A	***B***	***C***	***D***	***E***	***F***	***G***
FreeBSD 6.0							
FreeBSD 7.3							
FreeBSD 8.0							
Linux 2.6.31 (Ubuntu 9.10)							
Mac OS X 10.5							
Mac OS X 10.6							

OUTLINE

1. Introduction
2. Experimental design
3. 7 SACK misbehaviors
4. Results
5. Summary

Summary

- we accidentally discovered 7 unexpected SACK behaviors while researching reneging
- we designed a methodology using TBIT to identify OSes exhibiting these misbehaviors
- we applied the methodology on 29 OSes
  - found at least one OS for each misbehavior
- Conclusion: SACKs - simple in concept; complex to implement

Nasif Ekiz, Abuthahir Rahman, Paul Amer
"***Misbehaviors in TCP SACK Generation***"
ACM Computer Communications Review, 41(2), 4/2011

Questions?