

# CENSORSHIP RESISTANCE

---

**GRAD SEC**

NOV 14 2017



# TODAY'S PAPERS

## Examining How the Great Firewall Discovers Hidden Circumvention Servers

Roya Ensafi  
Princeton University

David Fifield  
UC Berkeley

Philipp Winter  
Karlstad & Princeton  
University

Nick Feamster  
Princeton University

Nicholas Weaver  
UC Berkeley & ICSI

Vern Paxson  
UC Berkeley & ICSI

### ABSTRACT

Recently, the operators of the national censorship infrastructure of China began to employ “active probing” to detect and block the use of privacy tools. This probing works by passively monitoring the network for suspicious traffic, then actively probing the corresponding servers, and blocking any that are determined to run circumvention servers such as Tor.

We draw upon multiple forms of measurements, some spanning years, to illuminate the nature of this probing. We identify the different types of probing, develop fingerprinting techniques to infer the physical structure of the system, localize the anomalies that trigger probing—showing that they differ from the “Good Firewall” infrastructure—and assess probing’s efficacy in blocking different versions of Tor. We conclude with a discussion of the implications for designing circumvention servers that resist such probing mechanisms.

### Categories and Subject Descriptors

C.2.0 [General]: Security and protection (e.g., firewalls);  
C.2.3 [Network Operations]: Network monitoring

### General Terms

Measurement

### Keywords

Active Probing, Deep Packet Inspection, Great Firewall of China, Censorship Circumvention, Tor

## 1. INTRODUCTION

Those in charge of the Chinese censorship apparatus spend considerable effort countering privacy tools. Among their most advanced techniques is what the Tor community terms *Probing* to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. ACM '15, October 26–30, 2015, Tokyo, Japan. Copyright is held by the owner(s)/author(s). Notification rights licensed to ACM. ACM 978-1-4503-2846-9/15/10...\$15.00. DOI: http://dx.doi.org/10.1145/2801973.2819692.

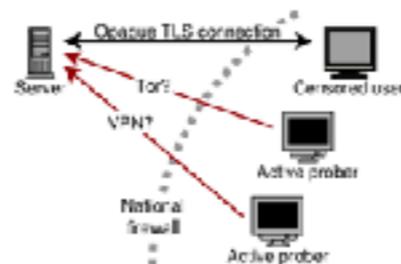


Figure 1: The firewall cannot determine, by mere inspection, whether the encrypted transaction carries a prohibited circumvention protocol. Therefore it issues its own probe and observes how the server responds.

“active probing”: passively monitoring the network for suspicious traffic, actively probing the corresponding servers, and blocking those determined to run circumvention services such as Tor.

The phenomenon of active probing arose presumably in response to enhanced circumvention systems that better resist traditional forms of blocking. For example, instead of employing a protocol recognizable by deep packet inspection (DPI), some of these systems unload their traffic inside TLS streams. Barring any subtle “tells” in the circumvention system’s communication, the censor cannot distinguish circumventing TLS from any other TLS, and thus cannot readily block the circumvention without incurring significant collateral damage. Active probing enables the censor to disambiguate the otherwise opaque traffic and once again obtain a measure of control over it.

Figure 1 illustrates the general scheme of active probing. The censor acts like a user and issues its own connections to a suspected circumvention server. If the server responds using a prohibited protocol, then the censor takes a blocking action, such as adding its IP address to a blacklist. If the circumvention server does not incorporate means control mechanisms or techniques to distinguish the censor’s probes from normal user connections, the censor can reliably identify and block it.

The effectiveness of active probing is reflected in its diverse uses. As of September 2015, researchers have documented

## Telex: Anticensorship in the Network Infrastructure

Eric Wustrow\*

Scott Wolchok\*

Ian Goldberg†

J. Alex Halderman\*

\*The University of Michigan

{erust, wolchok, jhalderm}@eecs.umich.edu

†University of Waterloo

iang@cs.uwaterloo.ca

### Abstract

In this paper, we present Telex, a new approach to resisting state-level Internet censorship. Rather than attempting to win the cat-and-mouse game of finding open proxies, we leverage censors’ unwillingness to completely block day-to-day Internet access. In effect, Telex converts innocuous, unblocked websites into proxies, without their explicit collaboration. We envision that friendly ISPs would deploy Telex stations on paths between censors’ networks and popular, uncensored Internet destinations. Telex stations would monitor seemingly innocuous flows for a special “tag” and transparently divert them to a forbidden website or service instead. We propose a new cryptographic scheme based on elliptic curves for tagging TLS handshakes such that the tag is visible to a Telex station but not to a censor. In addition, we use our tagging scheme to build a protocol that allows clients to connect to Telex stations while resisting both passive and active attacks. We also present a proof-of-concept implementation that demonstrates the feasibility of our system.

## 1 Introduction

The events of the Arab Spring have vividly demonstrated the Internet’s power to catalyze social change through the free exchange of ideas, news, and other information. The Internet poses such an existential threat to repressive regimes that some have completely disconnected from the global network during periods of intense political unrest, and many regimes are pursuing aggressive programs of Internet censorship using increasingly sophisticated techniques.

Today, the most widely-used tools for circumventing Internet censorship take the form of encrypted tunnels and proxies, such as Dynaweb [12], Icarusurf [30], and Tor [10]. While these designs can be quite effective at sneaking client connections past the censor, these systems inevitably lead to a cat-and-mouse game in which the

censor attempts to discover and block the services’ IP addresses. For example, Tor has recently observed the blocking of entry nodes and directory servers in China and Iran [28]. Though Tor is used to skirt Internet censors in these countries, it was not originally designed for that application. While it may certainly achieve its original goal of anonymity for its users, it appears that Tor and proxies like it are ultimately not enough to circumvent aggressive censorship.

To overcome this problem, we propose *Telex*: an “end-to-middle” proxy with no IP address, located within the network infrastructure. Clients invoke the proxy by using public-key steganography to “tag” otherwise ordinary TLS sessions destined for uncensored websites. Its design is unique in several respects:

**Architecture.** Previous designs have assumed that anti-censorship services would be provided by hosts at the edge of the network, as the end-to-end principle requires. We propose instead to provide these services in the core infrastructure of the Internet, along paths between the censor’s network and popular, nonblocked destinations. We argue that this will provide both lower latency and increased resistance to blocking.

**Deployment.** Many systems attempt to combat state-level censorship using resources provided primarily by volunteers. Instead, we investigate a government-scale response based on the view that state-level censorship needs to be countered by state-level anticensorship.

**Construction.** We show how a technique that the security and privacy literature most frequently associates with government surveillance—deep-packet inspection—can provide the foundation for a robust anticensorship system.

We expect that these design choices will be somewhat controversial, and we hope that they will lead to discussion about the future development of anticensorship systems.

# CENSORSHIP COMES IN MANY FORMS

---

## DROPPING PACKETS

**Network operators:** Block traffic in their own networks/countries

**Off-path attackers:** Inject TCP RST packets (next week)

**Routing-capable adversaries:** Can influence routes on the Internet

**Black-holing:** Announce a low-cost path, drop traffic

<https://www.youtube.com/watch?v=IzLPKuAOe50>

## MONITORING TRAFFIC

**Boomerang routing:** Source/destination close, but route goes through a country known to eavesdrop

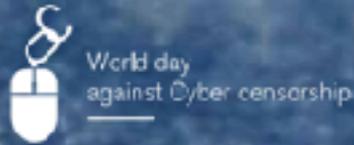
## DEANONYMIZATION

Identifying and going after **whistleblowers**

## MISDIRECTING TRAFFIC

**DNS injection:** Send back false DNS responses

# ENEMIES OF THE INTERNET



REPORTERS  
WITHOUT BORDERS  
FOR FREEDOM OF INFORMATION

~Annual report by  
Reporters without Borders

**2014**

- *Syria*
- *Russia*
- *Saudia Arabia*
- *UAE*
- *Cuba*
- *Belarus*
- *Pakistan*
- *Vietnam*
- *Turkmenistan*
- *Sudan*
- *Iran*
- *Bahrain*
- **USA**
- *UK*
- *Uzbekistan*
- *India*
- *China*
- *North Korea*
- *Ethiopia*
- *Surveillance dealers*

# ENEMIES OF THE INTERNET

Français  
Español



World day  
against Cyber censorship

## Enemies of the Internet

**REPORTERS  
WITHOUT BORDERS**  
FOR FREEDOM OF INFORMATION

[Home](#) [Enemies of the Internet](#) [The Map](#) [Recommendations](#) [Take Action!](#) [Archives](#)

## USA: NSA symbolises intelligence services' abuses

In June 2013, computer specialist Edward Snowden disclosed the extent of the surveillance practices of the U.S. and British intelligence services. Snowden, who worked for a government sub-contractor and had access to confidential documents, later exposed more targeted surveillance, focusing on the telecommunications of [world leaders and diplomats of allied countries](#). Activists, governments and international bodies have taken issue with the Obama administration, as the newspapers *The Guardian* and *The Washington Post* have revealed the extent of the surveillance. The main player in this vast surveillance operation is the highly secretive National Security Agency (NSA) which, in the light of Snowden's revelations, has come to symbolize the abuses by the world's intelligence agencies. Against this background, those involved in reporting on security issues have found their sources under increasing pressure.

The U.S. edition of *The Guardian* is still able to publish information from Edward Snowden, while the [British edition is not](#), but the country of the First Amendment has undermined confidence in the Internet and its own standards of security. U.S. surveillance practices and decryption activities are a direct threat to investigative journalists, especially those who work with sensitive sources for whom confidentiality is paramount and who are already under pressure.

### The NSA

Based in Fort Meade, Virginia, the NSA has always operated behind a wall of secrecy. According to legend, its acronym was jokingly said to mean "No Such Agency" because its work took place far from the eyes of U.S.

# ENEMIES OF THE INTERNET



World day  
against Cyber censorship

## Enemies of the Internet

Français  
Español

**REPORTERS  
WITHOUT BORDERS**  
FOR FREEDOM OF INFORMATION

[Home](#) [Enemies of the Internet](#) [The Map](#) [Recommendations](#) [Take Action!](#) [Archives](#)

### USA: NSA symbolises intelligence services' abuses

In June 2013, computer specialist Edward Snowden and British intelligence services. Snowden, who worked for the NSA, exposed more target [leaders and diplomats of allied countries](#). Activists criticised the Obama administration, as the newspapers *The Guardian* and *The New York Times* exposed the surveillance. The main player in this vast surveillance is the National Security Agency (NSA) which, in the light of Snowden's revelations, has become a symbol of intelligence agencies. Against this background, those involved in reporting on security issues have found their sources under increasing pressure.

The U.S. edition of *The Guardian* is still able to publish information from Edward Snowden, while the [British edition is not](#), but the country of the First Amendment has undermined confidence in the Internet and its own standards of security. U.S. surveillance practices and decryption activities are a direct threat to investigative journalists, especially those who work with sensitive sources for whom confidentiality is paramount and who are already under pressure.

#### The NSA

Based in Fort Meade, Virginia, the NSA has always operated behind a wall of secrecy. According to legend, its acronym was jokingly said to mean "No Such Agency" because its work took place far from the eyes of U.S.

### Pressure on journalists, sources and whistleblowers

The Obama administration has shown itself to be willing to interpret the protection of national security in a broad and abusive manner, [at the expense of freedom of information](#). A witch-hunt was launched against journalists' sources who disclosed confidential information about the powers of the state.

# ENEMIES OF THE INTERNET



World day  
against Cyber censorship

## Enemies of the Internet

Français  
Español

**REPORTERS  
WITHOUT BORDERS**  
FOR FREEDOM OF INFORMATION

[Home](#) [Enemies of the Internet](#) [The Map](#) [Recommendations](#) [Take Action!](#) [Archives](#)

### USA: NSA symbolises intelligence services' abuses

In June 2013, computer specialist Edward Snowden and British intelligence services. Snowden, who worked for the NSA, exposed more target [leaders and diplomats of allied countries](#). Activists criticised the Obama administration, as the newspapers *The Guardian* and *The New York Times* exposed the surveillance. The main player in this vast surveillance is the National Security Agency (NSA) which, in the light of Snowden's revelations, has become a symbol of intelligence agencies. Against this background, those involved in reporting on security issues have found their sources under increasing pressure.

The U.S. edition of *The Guardian* is still able to publish reports on NSA surveillance, but the country of the First Amendment has a long history of security. U.S. surveillance practices and decryption of communications, especially those who work with sensitive sources, are under increasing pressure.

#### The NSA

Based in Fort Meade, Virginia, the NSA has always operated behind a wall of secrecy. According to legend, its acronym was jokingly said to mean "No Such Agency" because its work took place far from the eyes of U.S.

### Pressure on journalists, sources and whistleblowers

The Obama administration has shown itself to be willing to interpret the protection of national security in a broad and abusive manner, [at the expense of freedom of information](#). A witch-hunt was launched against journalists' sources who disclosed confidential information about the powers of the state.

The NSA has been helped in its determined pursuit of WikiLeaks by GCHQ, since [all visitors to the website have been monitored by the British agency's TEMPORA surveillance system](#).

Their IP addresses and the terms entered in search engines to access the site are intercepted and recorded.

# COLLATERAL DAMAGE OF INTERNET CENSORSHIP

## The Collateral Damage of Internet Censorship by DNS Injection \*

Sparks  
Hovership Nebuchadnezzar  
Zion Virtual Labs  
zion.vlab@gmail.com

Neo<sup>1</sup>  
Hovership Nebuchadnezzar  
Zion Virtual Labs  
zion.vlab@gmail.com

Tank  
Hovership Nebuchadnezzar  
Zion Virtual Labs  
zion.vlab@gmail.com

Smith  
Hovership Nebuchadnezzar  
Zion Virtual Labs  
zion.vlab@gmail.com

Dozer  
Hovership Nebuchadnezzar  
Zion Virtual Labs  
zion.vlab@gmail.com

### ABSTRACT

Some ISPs and governments (most notably the Great Firewall of China) use DNS injection to block access to "unwanted" websites. The censorship tools inspect DNS queries near the ISP's boundary routers for sensitive domain keywords and injecting forged DNS responses, blocking the users from accessing censored sites, such as twitter.com and facebook.com. Unfortunately this causes large-scale collateral damage, affecting communication beyond what outside DNS traffic traverses a paper, we analyze the causes of the censorship and measure the Internet injecting activities and their effect. We inject forged replies even for transit of 43,000 measured open resolvers in US countries, may suffer some collateral from previous week, we find that age arises from resolvers querying TL transit passes through China rather than servers (S, I, J) located in China.

Categories and Subject Descriptors  
U.2.0 [Computer Communication

General Terms  
Measurement, Security

### Keywords

DNS, packet injection, Internet censorship, Great Firewall of China, coll

### 1. INTRODUCTION

Since DNS is essential for effectively is a common target for censorship system approach involves packet injection observes DNS requests and injects falsification. Yet censorship systems just the censored network.

\*We use pseudonyms to protect the authors.  
<sup>1</sup>Corresponding author.

As a concrete example, consider a query for www.epc.com from a US user, using a US-based DNS resolver. The US resolver will need to contact one of the DNS TLD authorities for .com, located in Germany. If the path to the selected TLD authority passes through China, then the Chinese Great Firewall will see this query and inject a reply which the US resolver will accept, cache, and return to the user, preventing the user from contacting the proper web

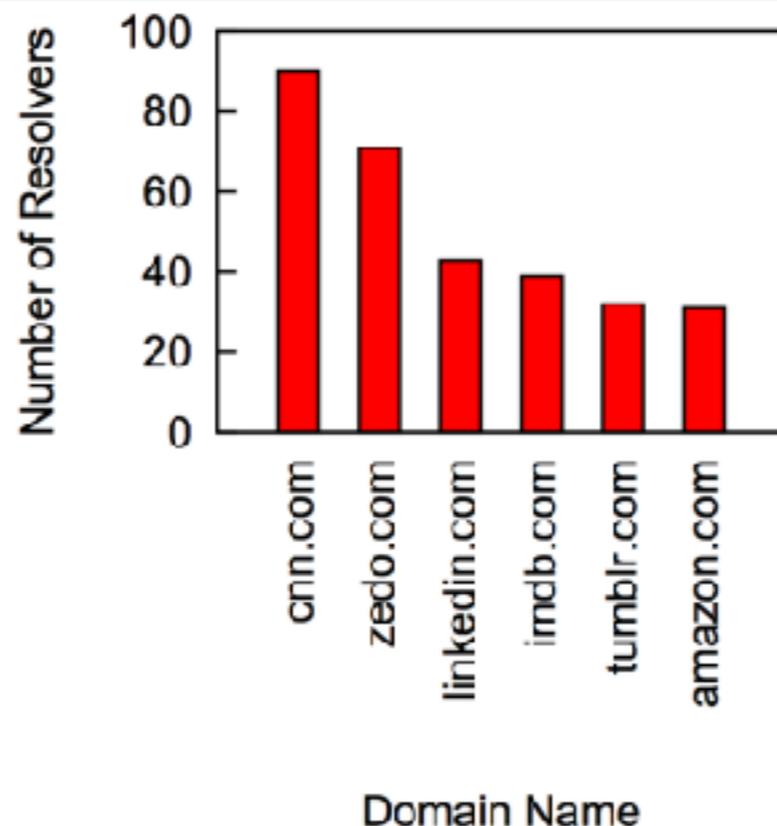


Figure 4: Affected domain names.

China censors the traffic to or from those within its borders *Known*

They do this via DNS injection

*Known / expected*

They do this to *any traffic* that traverses its borders *Not known*

More traffic traverses China's borders than we realized *Oh geez..*

# CIRCUMVENTING THE CONSTITUTION

## LOOPHOLES FOR CIRCUMVENTING THE CONSTITUTION: UNRESTRAINED BULK SURVEILLANCE ON AMERICANS BY COLLECTING NETWORK TRAFFIC ABROAD

Axel Arnbak and Sharon Goldberg\*

Cite as: Axel Arnbak and Sharon Goldberg, *Loopholes for Circumventing the Constitution: Unrestrained Bulk Surveillance on Americans by Collecting Network Traffic Abroad*, 21 MICH. TELECOMM. & TECH. L. REV. 317 (2015).

This manuscript may be accessed online at [repository.law.umich.edu](http://repository.law.umich.edu).

### ABSTRACT

This Article reveals interdependent legal and technical loopholes that the US intelligence community could use to circumvent constitutional and statutory safeguards for Americans. These loopholes involve the collection of Internet traffic on foreign territory, and leave Americans as unprotected as foreigners by current United States (US) surveillance laws. This Article will also describe how modern Internet protocols can be manipulated to deliberately divert American's traffic abroad, where traffic can then be collected under a more permissive legal regime (Executive Order 12333) that is overseen solely by the executive branch of the US government. Although the media has reported on some of the techniques we describe, we cannot establish the extent to which these loopholes are exploited in practice.

An actionable short-term remedy to these loopholes involves updating the antiquated legal definition of "electronic surveillance" in the Foreign Intelligence Surveillance Act (FISA), that has remained largely intact since 1978. In the long term, however, a fundamental reconsideration of established principles in US surveillance law is required, since

\* Axel Arnbak is a Faculty Researcher at the Institute for Information Law, University of Amsterdam and a Research Affiliate at the Berkman Center for Internet & Society, Harvard University. Sharon Goldberg is Associate Professor of Computer Science, Boston University and a Research Fellow, Sloan Foundation. She gratefully acknowledges the support of the Sloan Foundation. Both authors thank Timothy H. Edgar, Ethan Heilman, Susan Landau, Alex Matthews, Bruce Schneier, Hayt Shulman, Marry Wheeler and various attendees of the PETS'14 and TPRC'14 conferences for discussions and advice that have greatly aided this work. Alexander Abdo, David Choffnes, Nico van Eijk, Edward Felten, Daniel K. Gillmore, Jennifer Redford, Julian Sanchez and the anonymous reviewers for HotPETS'14 each provided insightful comments on drafts of this Article. Views and errors expressed in this Article remain the sole responsibility of the authors. This Article was submitted on September 1, 2014 and a brief update was concluded on December 26, 2014. All URLs have been checked on this date. An earlier version of this Article was first posted online on June 27, 2014.

## LEGAL REGIMES

Patriot Act

Foreign Intelligence Surveillance Act (FISA)

EO 12333

## WHAT CAN BE MONITORED?

Communication with foreign entities

## DO ROUTERS COUNT?

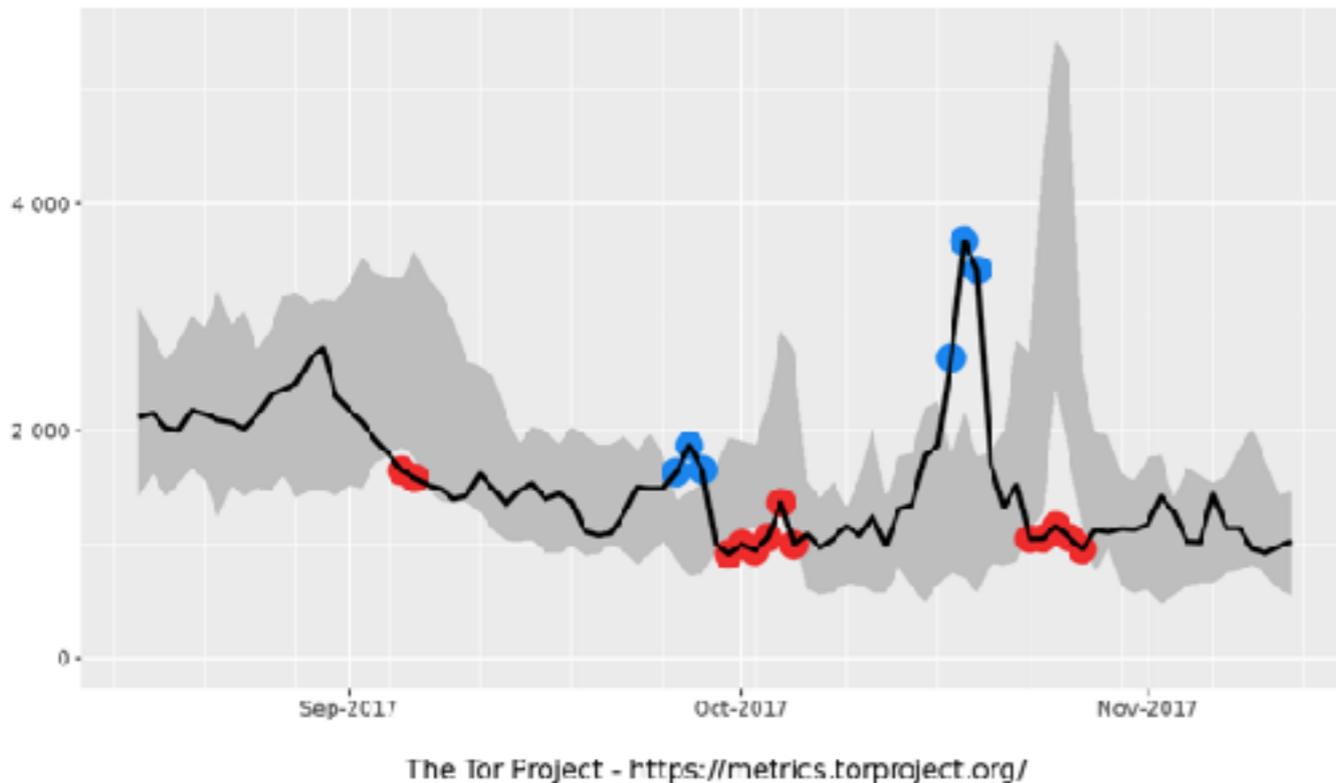
What if the US routed traffic out of its borders, then back in — would this count as communication with a foreign entity?

## THIS PAPER: YES, PROBABLY

So any traffic could be easily monitored

# BLOCKING TOR

Directly connecting users from China



Estimate the number of users on day  $i$  based on previous days' users

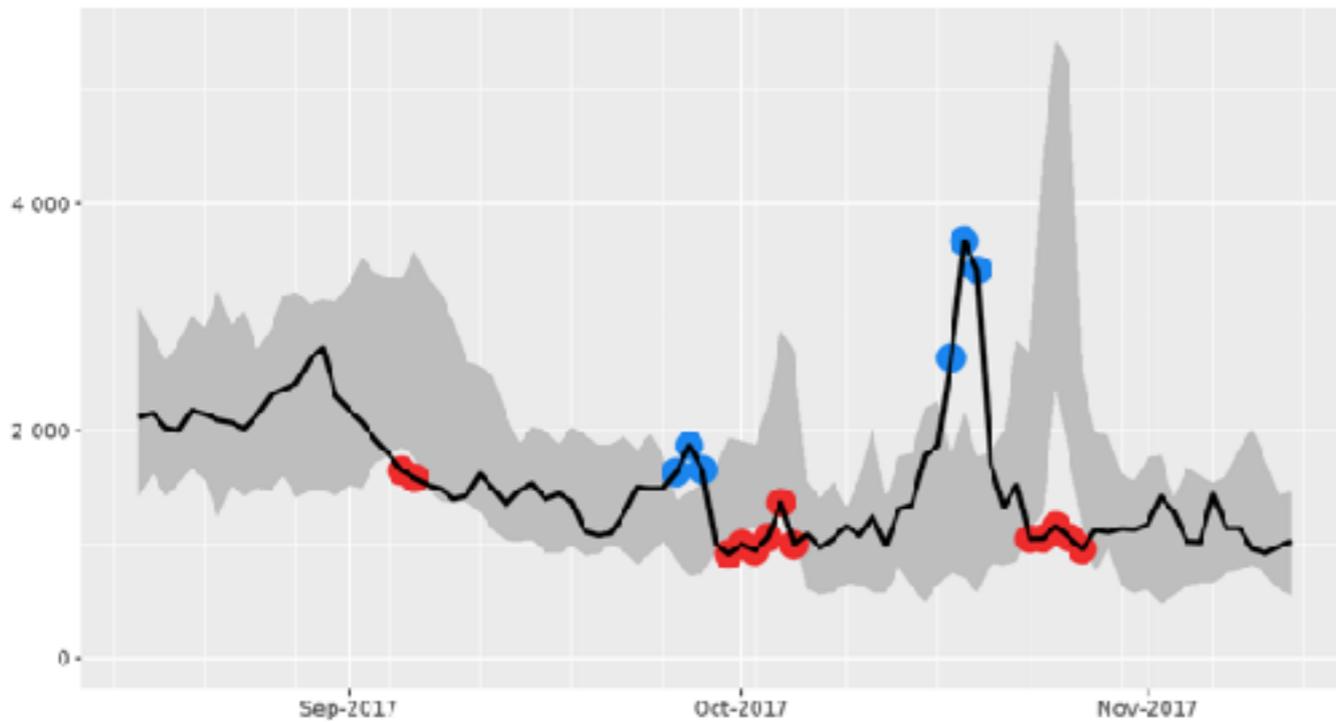
**Gray area:** Range of estimated users;  
Usage naturally fluctuates

**Downturn event:** Drops below  
Possibly indicates censorship

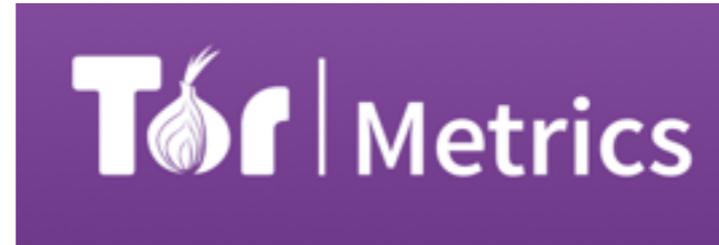
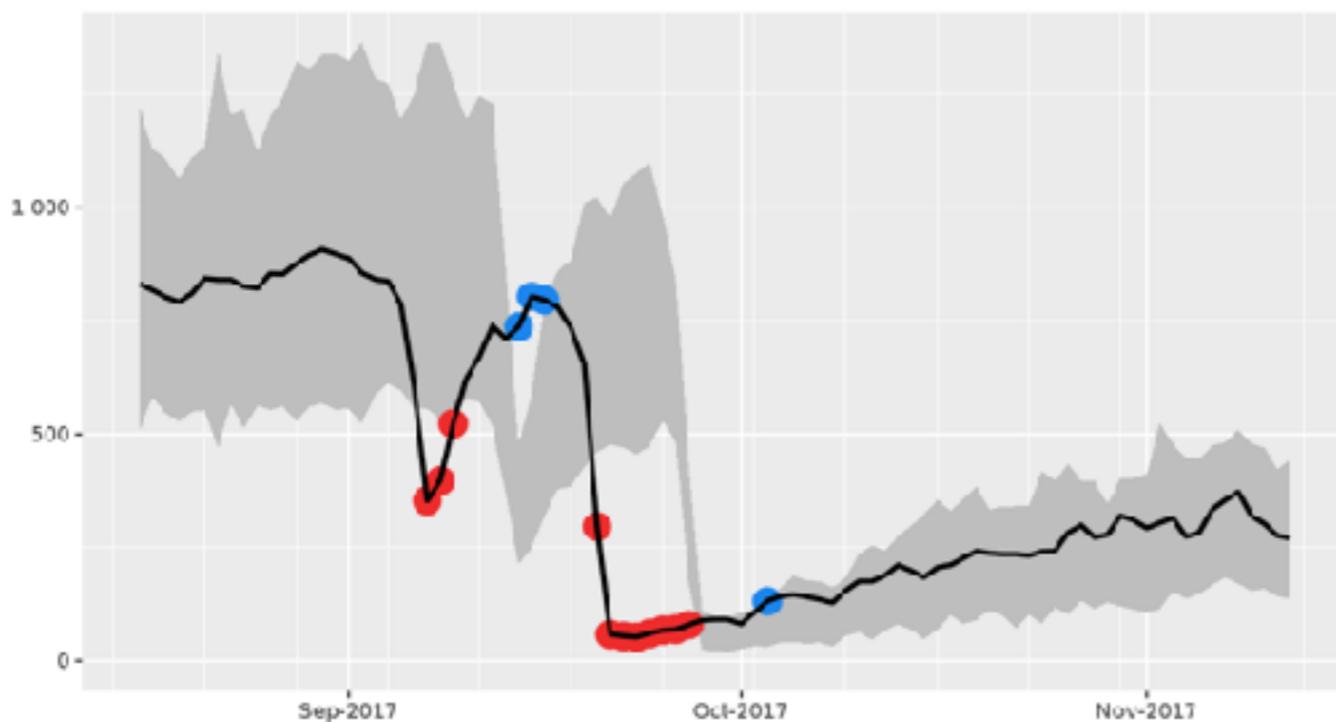
**Upturn event:** Rises above "normal"  
Possibly indicates circumvention

# BLOCKING TOR

Directly connecting users from China



Directly connecting users from Puerto Rico



Estimate the number of users on day  $i$  based on previous days' users

**Gray area:** Range of estimated users;  
Usage naturally fluctuates

**Downturn event:** Drops below  
Possibly indicates censorship

**Upturn event:** Rises above "normal"  
Possibly indicates circumvention

# HOW TO BLOCK TOR

48715 KB/s	no	-Bandwidth (KB/s)	uptime	-Hostname
		74288	6 d	105.170.42.18 [105.170.42.18]
	IPredator	66557	17 h	exit1.speedstorm [187.231.221.211]
	BasimulChien	61478	30 h	torserver1.dionix.org [20.125.04.101]
	Private/Apple/001	52367	33 d	tor-exit-node1.privatevpn.pub [173.52.161.96]
	public	48776	2 d	ns000920.p-509-04.eu [5.39.64.7]
	subell	44765	2 d	tor-exit.kuhells.net [178.217.167.39]
	torservernode	41370	13 d	torservernode.somsector.nl [70.109.23.1]
	gonggong	40765	42 h	dm [178.63.28.116]
	0s3d05	39360	13 d	snowdon.gop-security.net [62.158.7.171]
	trichid	38100	43 d	tor-exit1.hacktrickintermediat[62.118.42.118.16]
	0s3d04	37795	13 d	snowdon.gop-security.net [62.158.7.171]
	apt2	36300	19 d	tor-exit-2.apt.pub [165.96.14.171]
	ForBotMIBGB	36273	109 d	tor-exit.mibcloud [195.163.1.11]
	TheDarkLord	36250	4 d	tor-exit-1.p-79-137-105.eu [79.137.105.154]
	apt1	34976	19 d	tor-exit-1.apt.pub [165.96.14.171]
	ory	34530	6 d	ory.jp-end.nl [192.42.113.101]
	KyveloRoukia	34477	13 d	216.218.222.14 [216.218.222.14]
	DisplayTheTor	33068	97 d	mail.mesures.fr [192.210.213.17]
	jobUNCU	32530	11 h	tor01.tolnet.unc.edu [204.85.191.30]
	Oryx	32099	7 d	oryx.jp-end.nl [192.42.113.102]
	speckner1	31547	13 d	chill.kowlimann.net [138.201.189.12]
	TheSilence	31509	0 d	pekkow.fr [192.210.93.164]
	torfa	31393	5 d	torserver.worwick.de [79.172.163.32]
	apt3	31251	13 d	wogyo.10g.cherunet.com [37.220.35.202]
	iney	30900	40 h	dynamic-82-220-89-53.fls.asn.net [82.220.89.53]
	lodant	30896	131 d	62.210.88.142.rov.ponytelecom.eu [62.210.88.142]
	quadrad	30354	2 d	tor3.qadrad.de [148.251.180.229]
	fluxent	30739	2 d	antifluxent.de [5.9.102.198]
	regard3	30568	3 d	regard3.fr [192.210.244.164]
	CriticalMass	30129	3 d	77.247.181.166 [77.247.181.166]
	wow	30024	24 d	ns0005051.p-37-137-94.eu [37.137.94.56]
	McConnickReceps	29954	15 d	wholesomeaservermedia.mt.edu [18.85.22.204]
	01-poli-0118	29844	4 d	tor-exit-1.p-107-74-73.eu [107.74.73.179]
	torservernode	29749	5 h	151.80.238.152 [151.80.238.152]
	Torix BE2	29700	0 d	tor-exit-5-39-33.eu [5.39.33.178]
	HaveBlam	29204	3 d	rainbow.worwick.de [77.247.181.164]
	StanMash	28871	13 d	216.210.222.12 [216.210.222.12]
	Unnamed	28705	10 d	217.73.179.177
	Torix BE1	28377	4 d	tor-exit-6-30-33.eu [6.30.33.176]
	glau	27705	20 h	154.18.149.74 [154.18.149.74]
	marylou2	27586	7 d	marylou.noc.ogronet.net [89.234.167.264]
	0s3d01	27402	6 d	0s3d.nl [81.121.25.100]
	metnet	26985	40 d	metnet.oc.worwick.de [77.205.124.26]
	0s3d07	26354	6 d	0s3d.nl [81.121.25.100]
	ParEpisomerDissis	26338	6 d	de.mon.fr [162.172.101.187]
	0s3d06	26198	11 d	chilik.ann.li [173.193.252.11]
	FR2250E	26089	14 d	hostby.westpo.eu [5.188.11.186]
	torservernode	26043	5 d	84-244-27-203.de.nwbelium.nl [84.244.27.203]
	TCMra	25938	209 d	tor0385.serverprof24.com [188.138.75.101]
	jobUNCI	26246	24 d	tor01.tolnet.unc.edu [204.85.191.31]
	Access@16at2S	25783	18 d	31-173-145-85.fth.glasoperator.nl [85.145.173.31]
	artku	24964	90 d	static.234.211.201.130.clients.your-server.de [130.201.211.204]
	deppor	24771	7 d	freedom.jp-end.nl [192.42.113.102]
	MilesFarwa	24528	13 d	relay1.toropeninternet.o [62.210.123.240]
	icidbk	23920	25 d	185.107.81.233 [185.107.81.233]
	DEFI4	23797	63 d	tor-exit1-readme.cdn.se [171.25.183.73]
	crifa	23702	5 d	chicmsky.kaservers.net [77.247.181.166]
	marylou1	23599	7 d	marylou.noc.ogronet.net [89.234.167.264]
	PhantomTrin7	23354	13 d	85.19.167.150 [85.19.167.150]
	redpibal	23289	87 d	62.210.82.11.rov.ponytelecom.eu [62.210.82.11]
	tor	23160	21 h	85.240.227.165 [85.240.227.165]
	IVPS	23149	3 d	192.36.27.6 [192.36.27.6]
	TwoBogrov	23067	13 d	polkovnik.kaservers.net [77.247.181.166]
	BrainStone	22921	35 d	[no world [188.165.222.39]
	Gorosaofus7	22839	25 d	84.203.204.175 [84.203.204.175]
	GrayZone	22791	42 h	static.85.21.130.94.clients.your-server.de [84.130.21.85]
	DEFI0	22720	84 d	tor-exit0-readme.cdn.se [171.25.183.20]
	tor	22759	11 d	185.100.87.207 [185.100.87.207]
	Lskov0	22239	13 d	relay0.lskov.tor-relays.net [149.50.220.290]
	drumstotham	22158	21 h	89.31.57.58 [89.31.57.58]
	DarWin1210	21909	3 d	tor-relay-b.darwin1210.ma [48.4.77.210]
	PhantomTrin3	21429	13 d	85.19.167.152 [85.19.167.152]
	PhantomTrin4	21407	13 d	85.19.167.151 [85.19.167.151]
	watcrae	21252	2 d	183-172-213-115.eu.kaservers.net [183.172.213.115]
	aurora	21208	2 d	aurora.ann.li [171.126.262.12]
	tor	21133	0 d	tor-exit-1.p-107-74-73.eu [107.74.73.179]

**Option 1:** Get a list of all Tor nodes  
Insert them as firewall rules

**Bridge nodes:** Tor does not list some nodes;  
Users must learn them out of band

**This week's paper:** Censors discover them  
by actively probing

Scan IP addresses, sending protocol-specific  
messages: handshake (TLS, obfs), Versions (Tor),  
HTTPS Post (SoftEther), HTTP GET (AppSpot)

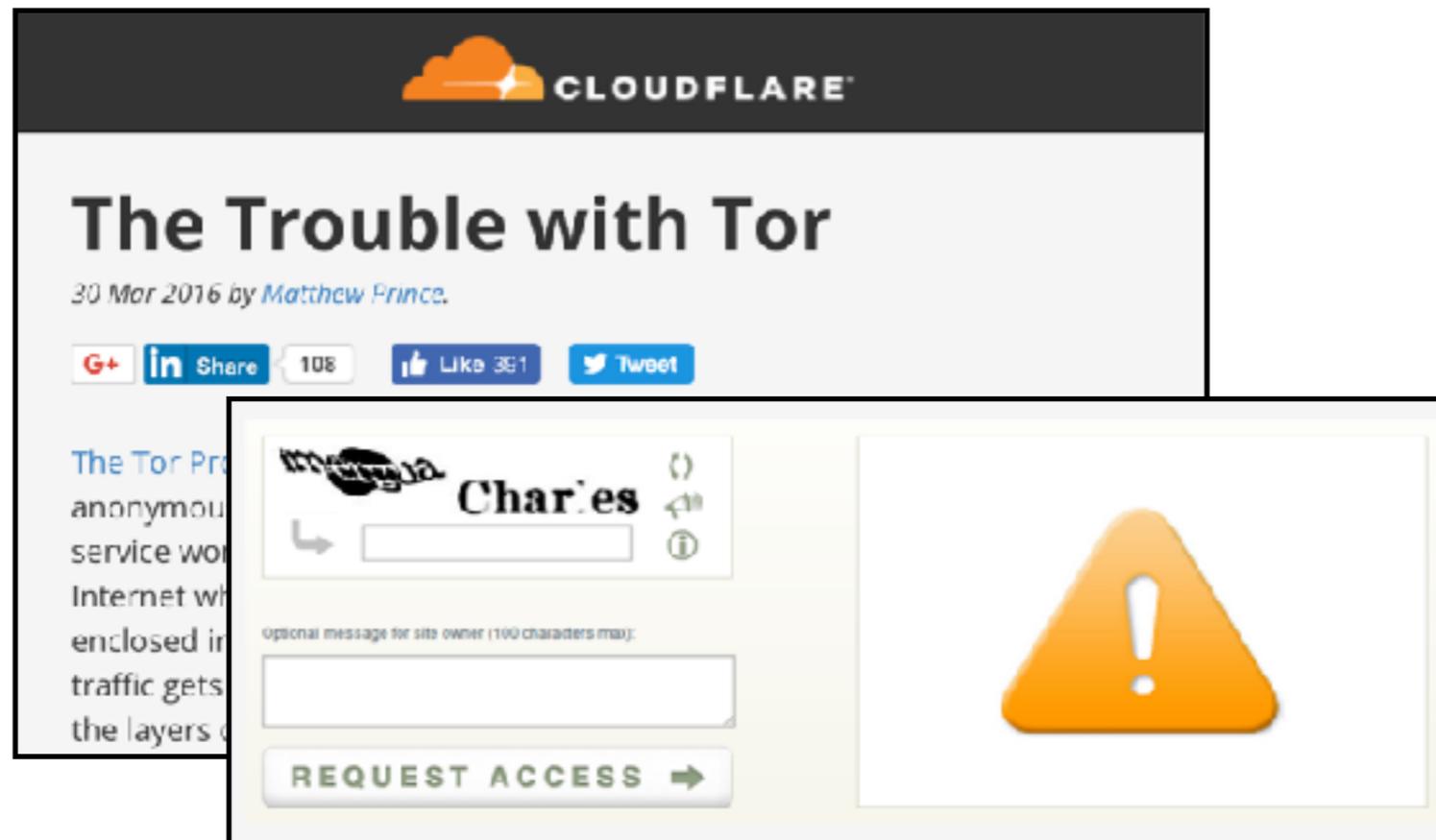
# HOW TO BLOCK TOR

---

# HOW TO BLOCK TOR

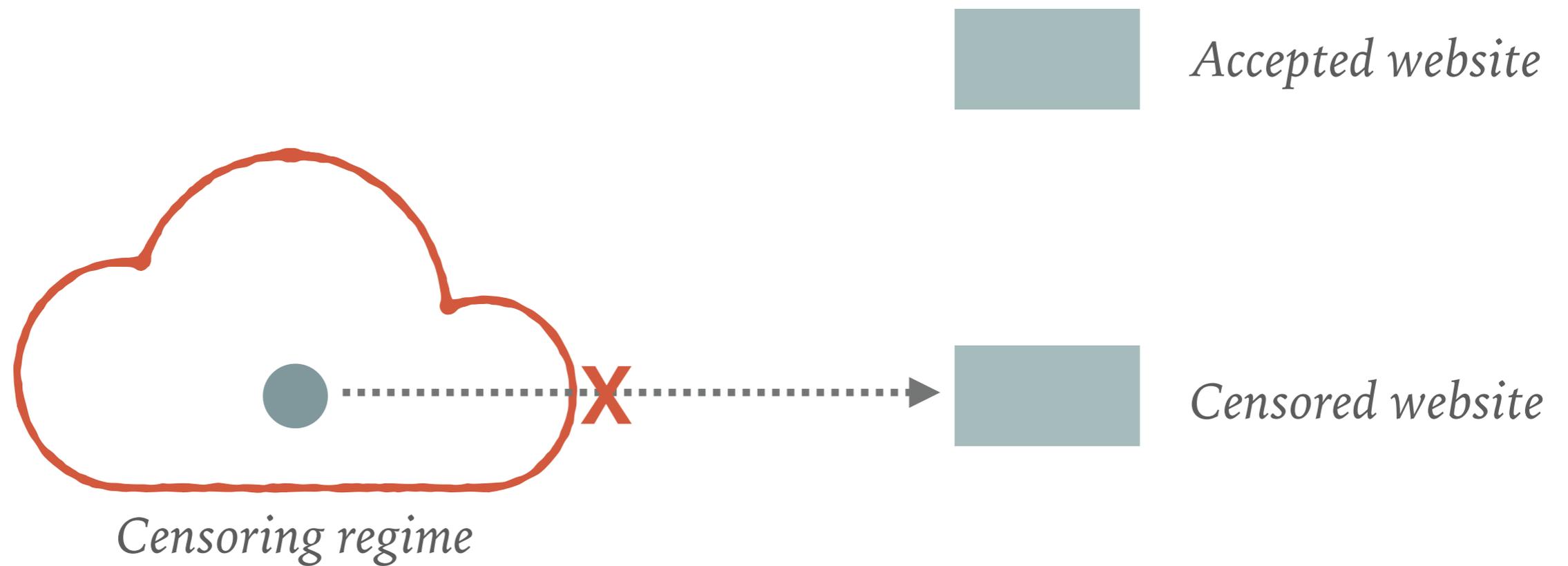
---

**Option 2:** IP-based reputation schemes;  
Will eventually block exit nodes because  
attackers **launder** their attack traffic thru Tor



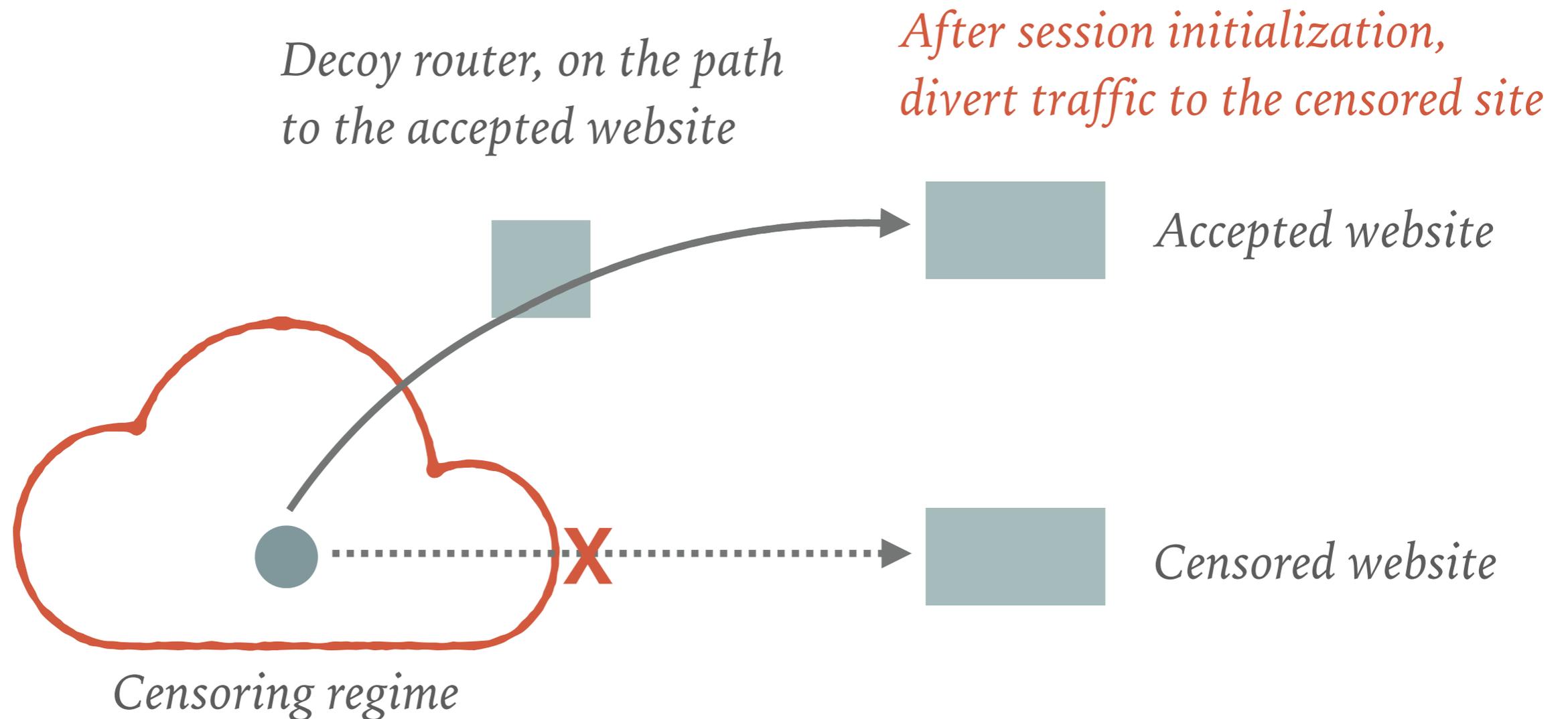
# DECOY ROUTING

---



# DECOY ROUTING

---

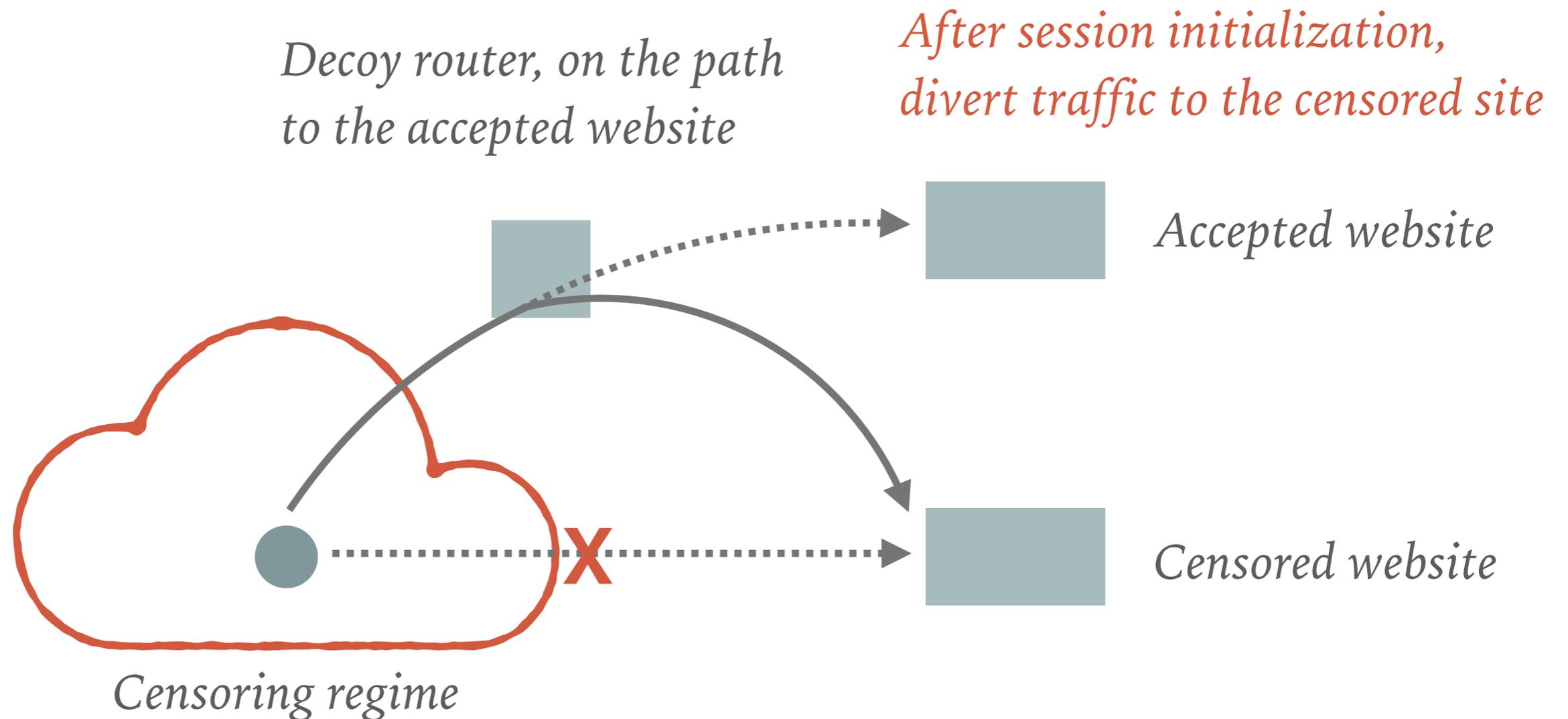


*How does the decoy router know the true destination but the censor doesn't?*

*Client includes "tags" in TLS handshakes that only the decoy router can identify*

# DECOY ROUTING

---



*How does the decoy router know the true destination but the censor doesn't?*

*Client includes "tags" in TLS handshakes that only the decoy router can identify*

# DECOY ROUTING TAGS

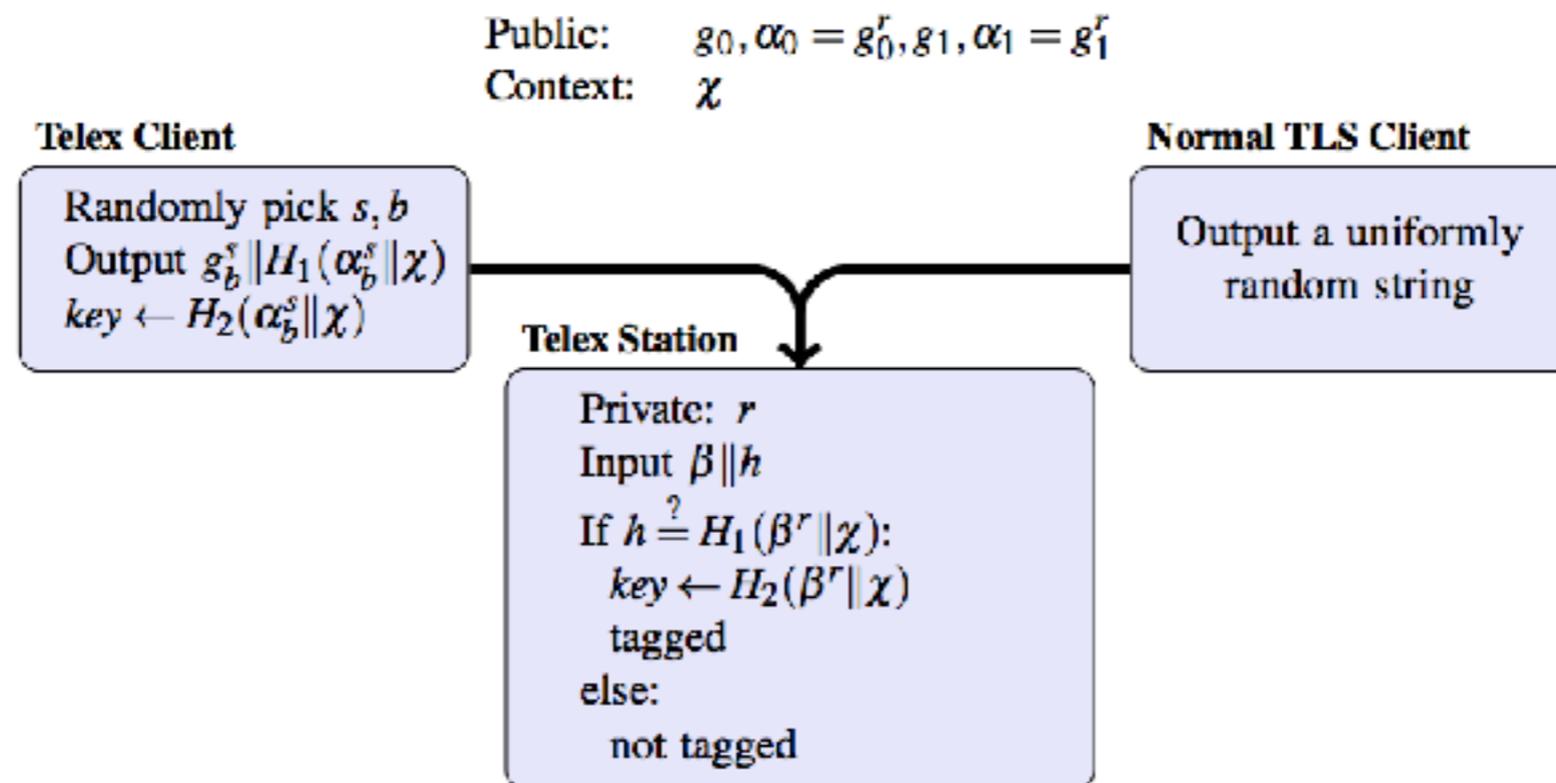


Figure 2: **Tag creation and detection** — Telex intercepts TLS connections that contain a steganographic tag in the ClientHello message's nonce field (normally a uniformly random string). The Telex client generates the tag using public parameters (shown above), but it can only be recognized by using the private key  $r$  embedded in the Telex station.

# AVOIDING CENSORS

---

## One approach

1. Map the Internet
2. Choose paths that do not go through the attackers' countries

# AVOIDING CENSORS

---

## One approach

1. Map the Internet  *Incredibly difficult research problem unto itself!*
2. Choose paths that do not go through the attackers' countries

# AVOIDING CENSORS

---

## One approach

1. Map the Internet  *Incredibly difficult research problem unto itself!*
2. Choose paths that do not go through the attackers' countries

Is it possible to get *provable avoidance*?

# ALIBI ROUTING

## Alibi Routing

Dave Levin<sup>\*</sup> Youndo Lee<sup>\*</sup> Luke Valenta<sup>†</sup> Zhihao Li<sup>\*</sup> Victoria Lal<sup>\*</sup>  
Cristian Lumezanu<sup>‡</sup> Neil Spring<sup>\*</sup> Bobby Bhattacharjee<sup>\*</sup>

<sup>\*</sup> University of Maryland <sup>†</sup> University of Pennsylvania <sup>‡</sup> NEC Labs

### ABSTRACT

There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did not traverse certain parts of the world while on their way to or from another host. This paper introduces the problem of finding “proofs of avoidance”—evidence that the paths taken by a packet and its response avoided a user-specified set of “forbidden” geographic regions. Proving that something did not happen is often intractable, but we demonstrate a low-overhead proof structure built around the idea of what we call “alibis”: relays with particular timing constraints that, when upheld, would make it impossible to traverse both the relay and the forbidden regions.

We present *Alibi Routing*, a peer-to-peer overlay routing system for finding alibis securely and efficiently. One of the primary distinguishing characteristics of Alibi Routing is that it does not require knowledge of—or modifications to—the Internet’s routing hardware or policies. Rather, Alibi Routing is able to derive its proofs of avoidance from user-provided GPS coordinates and speed-of-light propagation delays. Using a PlanetLab deployment and larger-scale simulations, we evaluate Alibi Routing to demonstrate that many source-destination pairs can avoid countries of their choosing with little latency inflation. We also identify when Alibi Routing does not work: it has difficulty avoiding regions that users are very close to (or, of course, inside of).

### Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols; C.2.0 [Computer-Communication Networks]: General—Security and protection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGCOMM '15, August 17–21, 2015, London, United Kingdom  
© 2015 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-2842-0/15/08...\$15.00  
DOI: <http://dx.doi.org/10.1145/2786056.2787600>

### Keywords

Alibi Routing; Provable route avoidance; Censorship avoidance; Peer-to-peer; Overlay routing

### 1. INTRODUCTION

Users have little control over where in the world their packets travel en route to their destinations. Some mechanisms exist to provide insight into where packets traveled, such as the record-route IP option, overlay routing systems [87], or to a lesser extent source-routing. While these approaches expose a subset of the path the user’s packets took, they do not allow a user to determine or provably influence where their packets do *not* go.

This paper introduces a new primitive we call *provable avoidance routing*. With provable avoidance routing, a user specifies arbitrary geographic regions—such as countries or UN voting blocs—to be avoided while communicating with a destination. If successful, the primitive returns *proof* that the user’s packets did not traverse the forbidden regions. If it is unsuccessful, it concludes only that the packets *may* have traversed them.

The goal of provable avoidance routing is *detection*, as opposed to *prevention*. In other words, alone, it is unable to ensure a user’s packets *will not* traverse a region of the world—we do not require modifications to the underlying routing protocols or hardware, and so we are subject to all of today’s uncertainties as to where packets will travel. Rather, what we are able to provide is assurance that the user’s packets and their respective responses took paths that *did not* traverse regions of the world. Our proofs of avoidance are provided on a per-packet basis, and are *a posteriori*: only after sending the packet and getting a reply can we ascertain whether or not the round-trip communication avoided the forbidden region.

While outright prevention would be ideal, detection can be a powerful tool, as well. For example, consider one of the greatest threats to open communication on the Internet: censorship. Beyond just dropping [34] or logging [29] users’ traffic, censorship can take many forms, including injecting packets with false information [4]. Recent results indicate that many users may be censored not by their (or their destination’s) countries, but by regimes through which their packets transit; a group of anonymous researchers demonstrated that DNS queries that merely traverse China’s borders are

## QUESTION

Can we provably avoid countries known to censor/attack?

# ALIBI ROUTING

## Alibi Routing

Dave Levin<sup>\*</sup> Youndo Lee<sup>\*</sup> Luke Valenta<sup>†</sup> Zhihao Li<sup>\*</sup> Victoria Lai<sup>\*</sup>  
Cristian Lumezanu<sup>‡</sup> Neil Spring<sup>\*</sup> Bobby Bhattacharjee<sup>\*</sup>

<sup>\*</sup> University of Maryland <sup>†</sup> University of Pennsylvania <sup>‡</sup> NEC Labs

### ABSTRACT

There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did not traverse certain parts of the world while on their way to or from another host. This paper introduces the problem of finding “proofs of avoidance”—evidence that the paths taken by a packet and its response avoided a user-specified set of “forbidden” geographic regions. Proving that something did not happen is often intractable, but we demonstrate a low-overhead proof structure built around the idea of what we call “alibis”: relays with particular timing constraints that, when upheld, would make it impossible to traverse both the relay and the forbidden regions.

We present *Alibi Routing*, a peer-to-peer overlay routing system for finding alibis securely and efficiently. One of the primary distinguishing characteristics of Alibi Routing is that it does not require knowledge of—or modifications to—the Internet’s routing hardware or policies. Rather, Alibi Routing is able to derive its proofs of avoidance from user-provided GPS coordinates and speed-of-light propagation delays. Using a PlanetLab deployment and larger-scale simulations, we evaluate Alibi Routing to demonstrate that many source-destination pairs can avoid countries of their choosing with little latency inflation. We also identify when Alibi Routing does not work: it has difficulty avoiding regions that users are very close to (or, of course, inside of).

### Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols; C.2.0 [Computer-Communication Networks]: General—Security and protection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGCOMM '15, August 17–21, 2015, London, United Kingdom  
© 2015 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-2542-0/15/08...\$15.00  
DOI: <http://dx.doi.org/10.1145/2786056.2787600>

### Keywords

Alibi Routing; Provable route avoidance; Censorship avoidance; Peer-to-peer; Overlay routing

### 1. INTRODUCTION

Users have little control over where in the world their packets travel en route to their destinations. Some mechanisms exist to provide insight into where packets traveled, such as the record-route IP option, overlay routing systems [87], or to a lesser extent source-routing. While these approaches expose a subset of the path the user’s packets took, they do not allow a user to determine or provably influence where their packets do *not* go.

This paper introduces a new primitive we call *provable avoidance routing*. With provable avoidance routing, a user specifies arbitrary geographic regions—such as countries or UN voting blocs—to be avoided while communicating with a destination. If successful, the primitive returns *proof* that the user’s packets did not traverse the forbidden regions. If it is unsuccessful, it concludes only that the packets *may* have traversed them.

The goal of provable avoidance routing is *detection*, as opposed to *prevention*. In other words, alone, it is unable to ensure a user’s packets *will not* traverse a region of the world—we do not require modifications to the underlying routing protocols or hardware, and so we are subject to all of today’s uncertainties as to where packets will travel. Rather, what we are able to provide is assurance that the user’s packets and their respective responses took paths that *did not* traverse regions of the world. Our proofs of avoidance are provided on a per-packet basis, and are *a posteriori*: only after sending the packet and getting a reply can we ascertain whether or not the round-trip communication avoided the forbidden region.

While outright prevention would be ideal, detection can be a powerful tool, as well. For example, consider one of the greatest threats to open communication on the Internet: censorship. Beyond just dropping [34] or logging [29] users’ traffic, censorship can take many forms, including injecting packets with false information [4]. Recent results indicate that many users may be censored not by their (or their destination’s) countries, but by regimes through which their packets transit; a group of anonymous researchers demonstrated that DNS queries that merely traverse China’s borders are

## QUESTION

Can we provably avoid countries known to censor/attack?

## LESSONS LEARNED

Give your papers a more descriptive title; nobody knows what it’s about!

### Alibi routing

[D Levin, Y Lee, L Valenta, Z Li, V Lai...](#) - ACM SIGCOMM ..., 2015 - dl.acm.org

Abstract There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did not traverse certain parts of the world while on their way to or from another host. This paper

☆ ⓘ Cited by 11 Related articles All 16 versions

# ALIBI ROUTING

## Alibi Routing

Dave Levin<sup>\*</sup> Younsoo Lee<sup>\*</sup> Luke Valenta<sup>†</sup> Zhihao Li<sup>†</sup> Victoria Lai<sup>†</sup>  
Cristian Lumezanu<sup>‡</sup> Neil Spring<sup>\*</sup> Bobby Bhattacharjee<sup>\*</sup>

<sup>\*</sup> University of Maryland <sup>†</sup> University of Pennsylvania <sup>‡</sup> NEC Labs

### ABSTRACT

There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did not traverse certain parts of the world while on their way to or from another host. This paper introduces the problem of finding “proofs of avoidance”—evidence that the paths taken by a packet and its response avoided a user-specified set of “forbidden” geographic regions. Proving that something did not happen is often intractable, but we demonstrate a low-overhead proof structure built around the idea of what we call “alibis”: relays with particular timing constraints that, when upheld, would make it impossible to traverse both the relay and the forbidden regions.

We present *Alibi Routing*, a peer-to-peer overlay routing system for finding alibis securely and efficiently. One of the primary distinguishing characteristics of Alibi Routing is that it does not require knowledge of—or modifications to—the Internet’s routing hardware or policies. Rather, Alibi Routing is able to derive its proofs of avoidance from user-provided GPS coordinates and speed-of-light propagation delays. Using a PlanetLab deployment and larger-scale simulations, we evaluate Alibi Routing to demonstrate that many source-destination pairs can avoid countries of their choosing with little latency inflation. We also identify when Alibi Routing does not work: it has difficulty avoiding regions that users are very close to (or, of course, inside of).

### Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols; C.2.0 [Computer-Communication Networks]: General—Security and protection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGCOMM '15, August 17–21, 2015, London, United Kingdom  
© 2015 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-2542-3/15/08...\$15.00  
DOI: <https://dx.doi.org/10.1145/2786056.2787600>

### Keywords

Alibi Routing; Provable route avoidance; Censorship avoidance; Peer-to-peer; Overlay routing

### 1. INTRODUCTION

Users have little control over where in the world their packets travel en route to their destinations. Some mechanisms exist to provide insight into where packets traveled, such as the record-route IP option, overlay routing systems [87], or to a lesser extent source-routing. While these approaches expose a subset of the path the user’s packets took, they do not allow a user to determine or provably influence where their packets do *not* go.

This paper introduces a new primitive we call *provable avoidance routing*. With provable avoidance routing, a user specifies arbitrary geographic regions—such as countries or UN voting blocs—to be avoided while communicating with a destination. If successful, the primitive returns *proof* that the user’s packets did not traverse the forbidden regions. If it is unsuccessful, it concludes only that the packets *may* have traversed them.

The goal of provable avoidance routing is *detection*, as opposed to *prevention*. In other words, alone, it is unable to ensure a user’s packets *will not* traverse a region of the world—we do not require modifications to the underlying routing protocols or hardware, and so we are subject to all of today’s uncertainties as to where packets will travel. Rather, what we are able to provide is assurance that the user’s packets and their respective responses took paths that *did not* traverse regions of the world. Our proofs of avoidance are provided on a per-packet basis, and are *a posteriori*: only after sending the packet and getting a reply can we ascertain whether or not the round-trip communication avoided the forbidden region.

While outright prevention would be ideal, detection can be a powerful tool, as well. For example, consider one of the greatest threats to open communication on the Internet: censorship. Beyond just dropping [34] or logging [29] users’ traffic, censorship can take many forms, including injecting packets with false information [4]. Recent results indicate that many users may be censored not by their (or their destination’s) countries, but by regimes through which their packets transit; a group of anonymous researchers demonstrated that DNS queries that merely traverse China’s borders are

## QUESTION

Can we provably avoid countries known to censor/attack?

## LESSONS LEARNED

Give your papers a more descriptive title; nobody knows what it’s about!

### Alibi routing

D Levin, Y Lee, L Valenta, Z Li, V Lai... - ACM SIGCOMM ..., 2015 - dl.acm.org

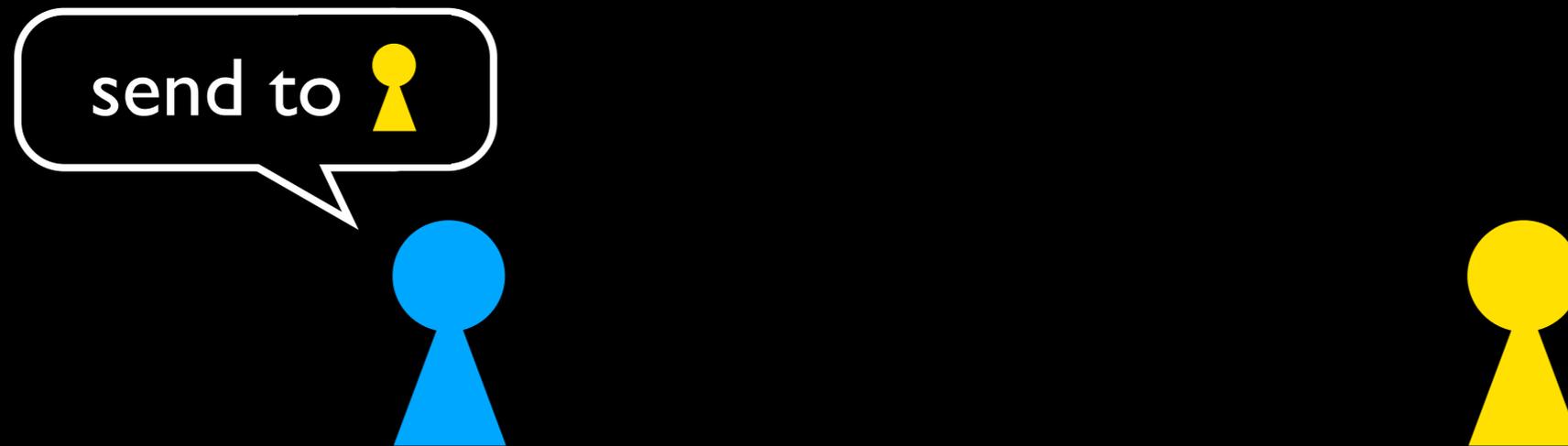
Abstract There are several mechanisms by which users can gain insight into where their packets have gone, but no mechanisms allow users undeniable proof that their packets did not traverse certain parts of the world while on their way to or from another host. This paper

☆ ⓘ Cited by 11 Related articles All 16 versions

Also, yes, it’s possible to get provable avoidance without even knowing where exactly packets went

# Users lack control over routing

Mostly relegated to destination-based routing



# Users lack control over routing

Mostly relegated to destination-based routing



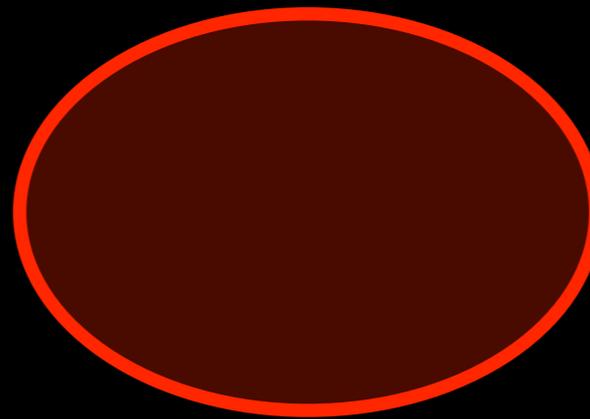
# Users lack control over routing

Collateral damage of censorship

send to



Censor-free



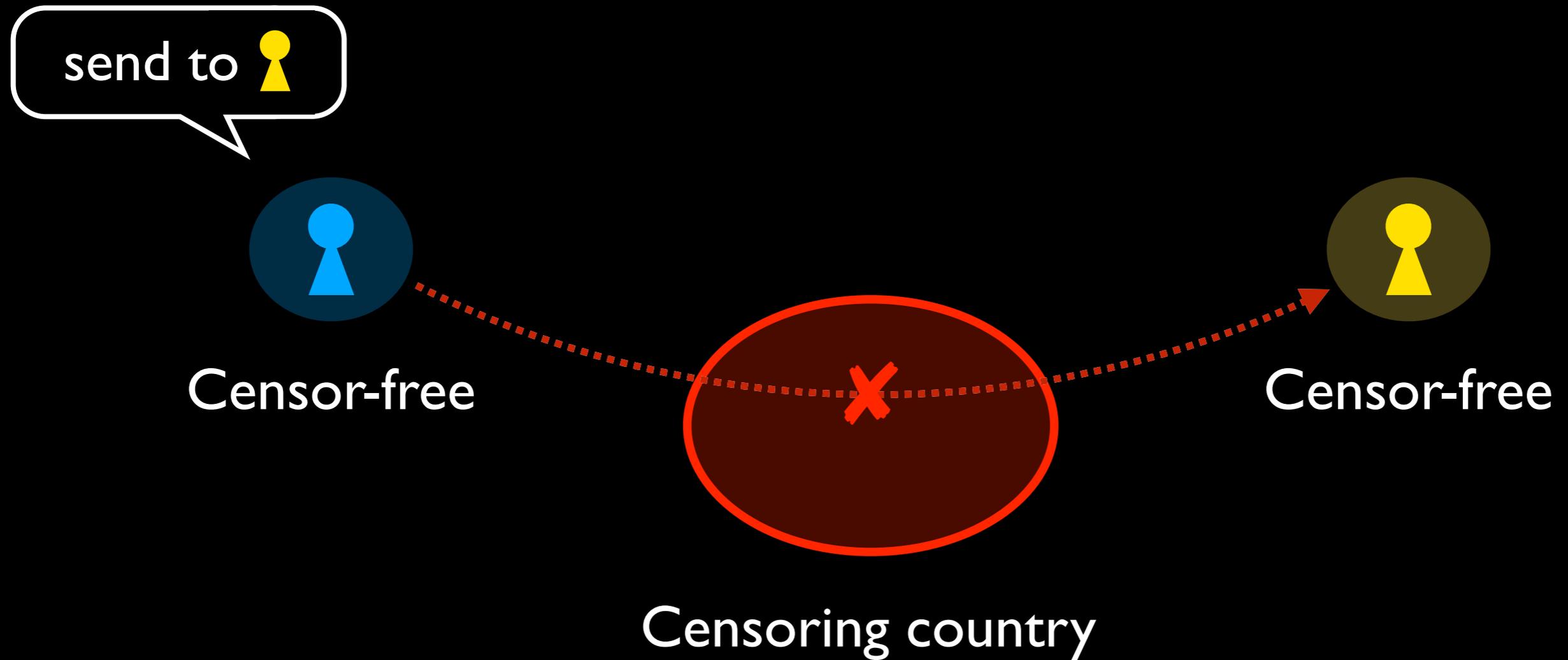
Censoring country



Censor-free

# Users lack control over routing

Collateral damage of censorship



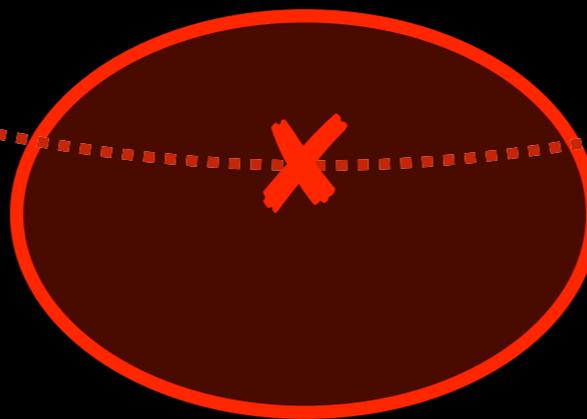
# Users lack control over routing

Collateral damage of censorship

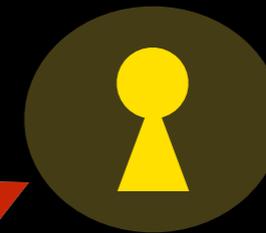
send to 



Censor-free



Censoring country



Censor-free

Encryption  
(HTTPS)

Anonymity  
(Tor)

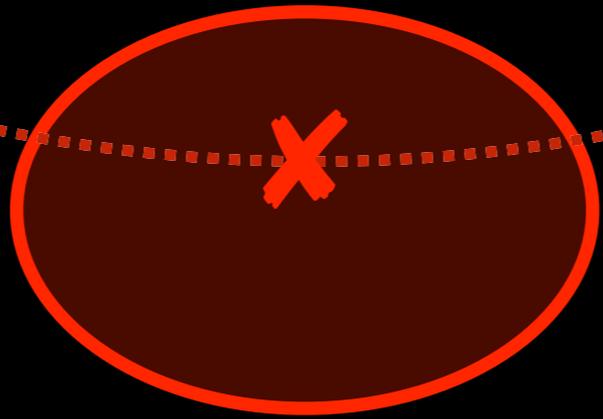
Hide info, but are still  
subject to censorship

# This work

send to 



Censor-free



Censoring country



Censor-free

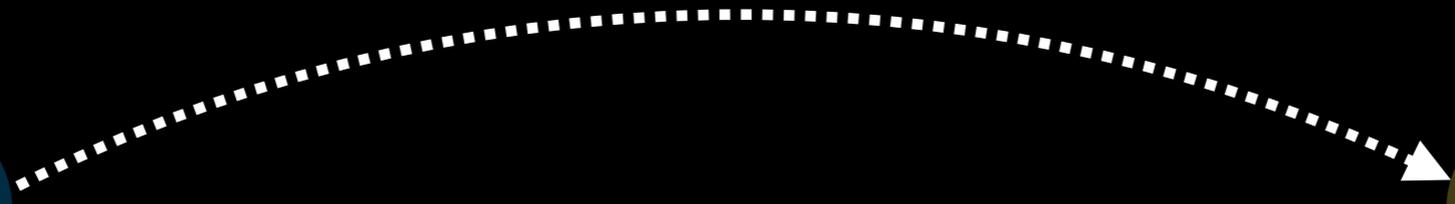


# This work

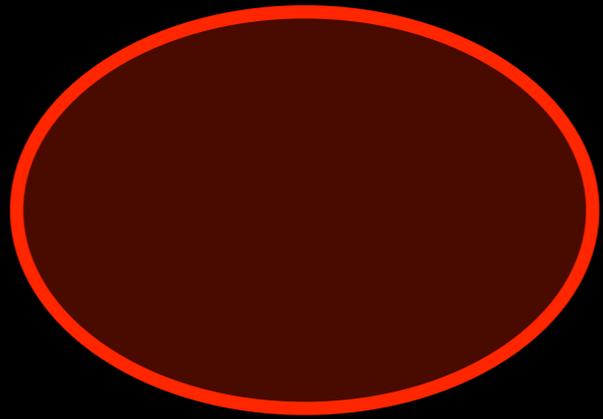
send to 



Censor-free



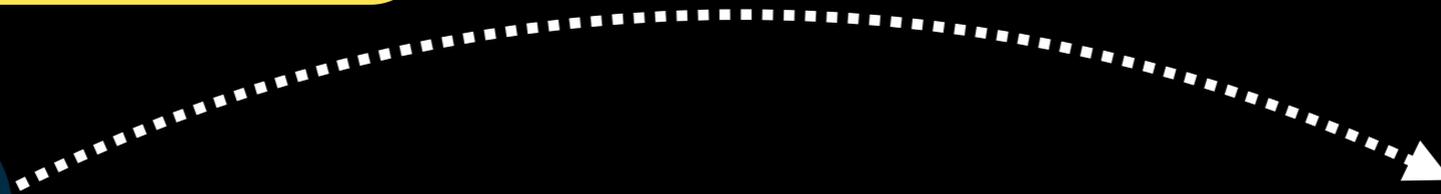
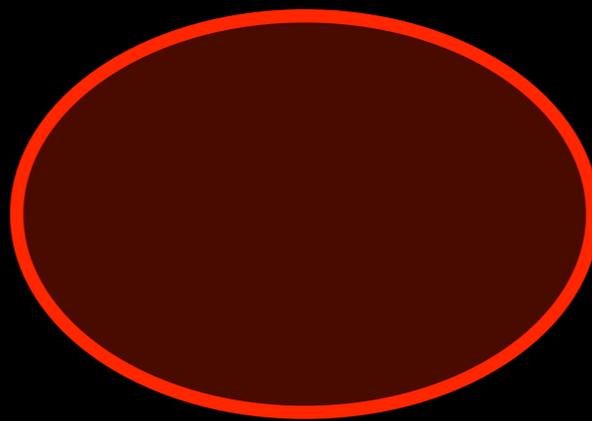
Censor-free



Censoring country

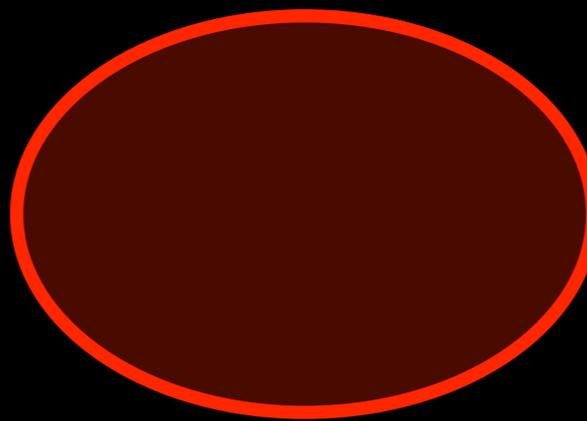
# Provable avoidance routing

send to  but avoid 



# Provable avoidance routing

send to  but avoid 



Provably disjoint paths

Diffie-Hellman

Avoiding boomerangs

Distinct vantage points

A broadly applicable primitive

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary geographic regions**

Proof

Provide **proofs** of avoidance

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary geographic regions**

Proof

Provide **proofs** of avoidance

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary geographic regions**

Without having to know  
underlying routes

Proof

Provide **proofs** of avoidance

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary** geographic regions

Proof

Provide **proofs** of avoidance

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary** geographic regions

Proof

Provide **proofs** of avoidance

Goal: proof that it **did not** traverse

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary** geographic regions

Proof

Provide **proofs** of avoidance

Goal: proof that it **did not** traverse

Non-goal: proof that it **cannot** traverse

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary** geographic regions

Proof

Provide **proofs** of avoidance

Goal: proof that it *did not* traverse.....

**Unadulterated roundtrip of communication**

Non-goal: proof that it *cannot* traverse.....

# Provable route avoidance goals

Flexibility

Users request their traffic to **avoid** transiting **arbitrary geographic regions**

Proof

Provide **proofs** of avoidance

How do you prove that something *did not* happen?

# Proving the impossible

How do you prove  $\textcircled{X}$  did *not* happen  
without enumerating everything that *could have*?

# Proving the impossible

How do you prove  $\textcircled{X}$  did *not* happen  
without enumerating everything that *could have*?

A

# Proving the impossible

How do you prove  $\textcircled{X}$  did *not* happen without enumerating everything that *could have*?

$\textcircled{A}$

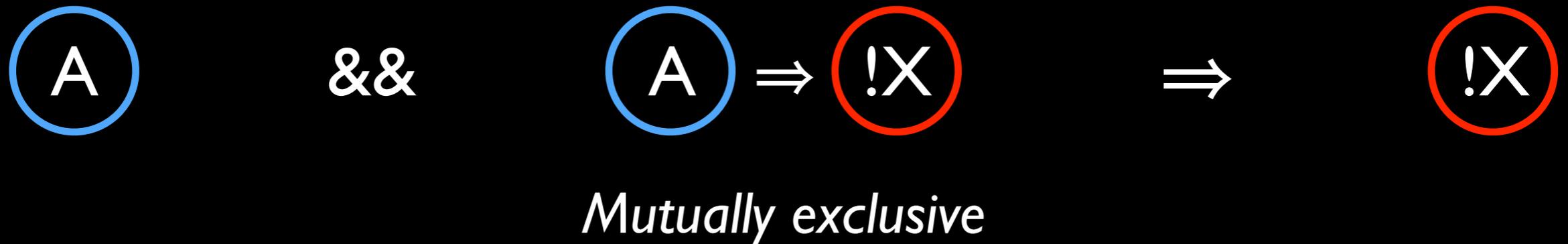
&&

$\textcircled{A} \Rightarrow \textcircled{!X}$

*Mutually exclusive*

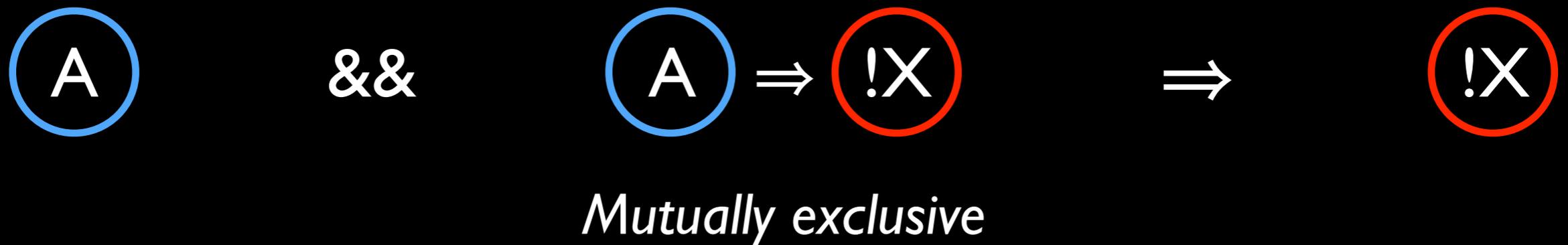
# Proving the impossible

How do you prove  $\textcircled{X}$  did *not* happen without enumerating everything that *could have*?



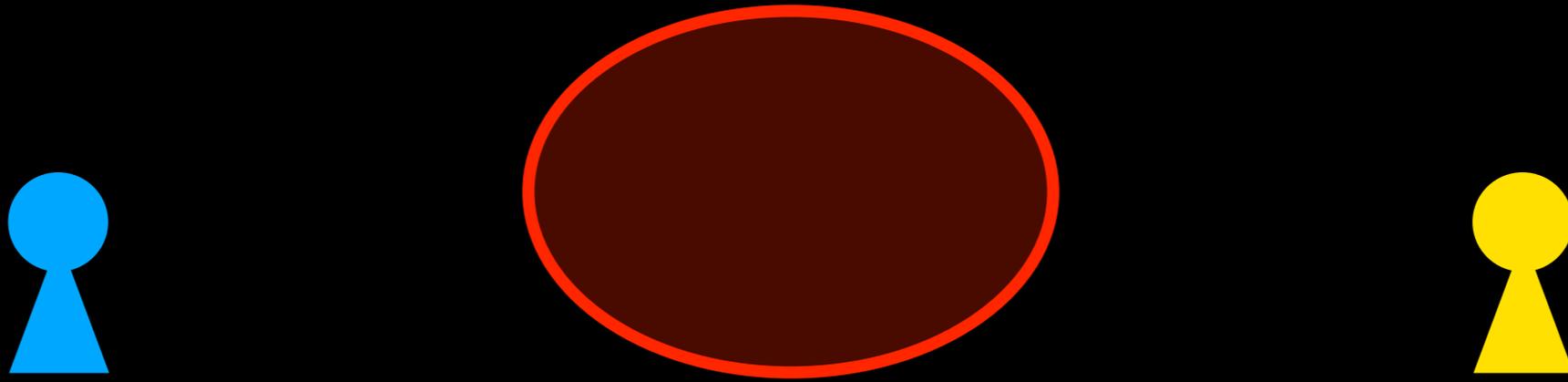
# Proving the impossible

How do you prove  $\textcircled{X}$  did *not* happen without enumerating everything that *could have*?

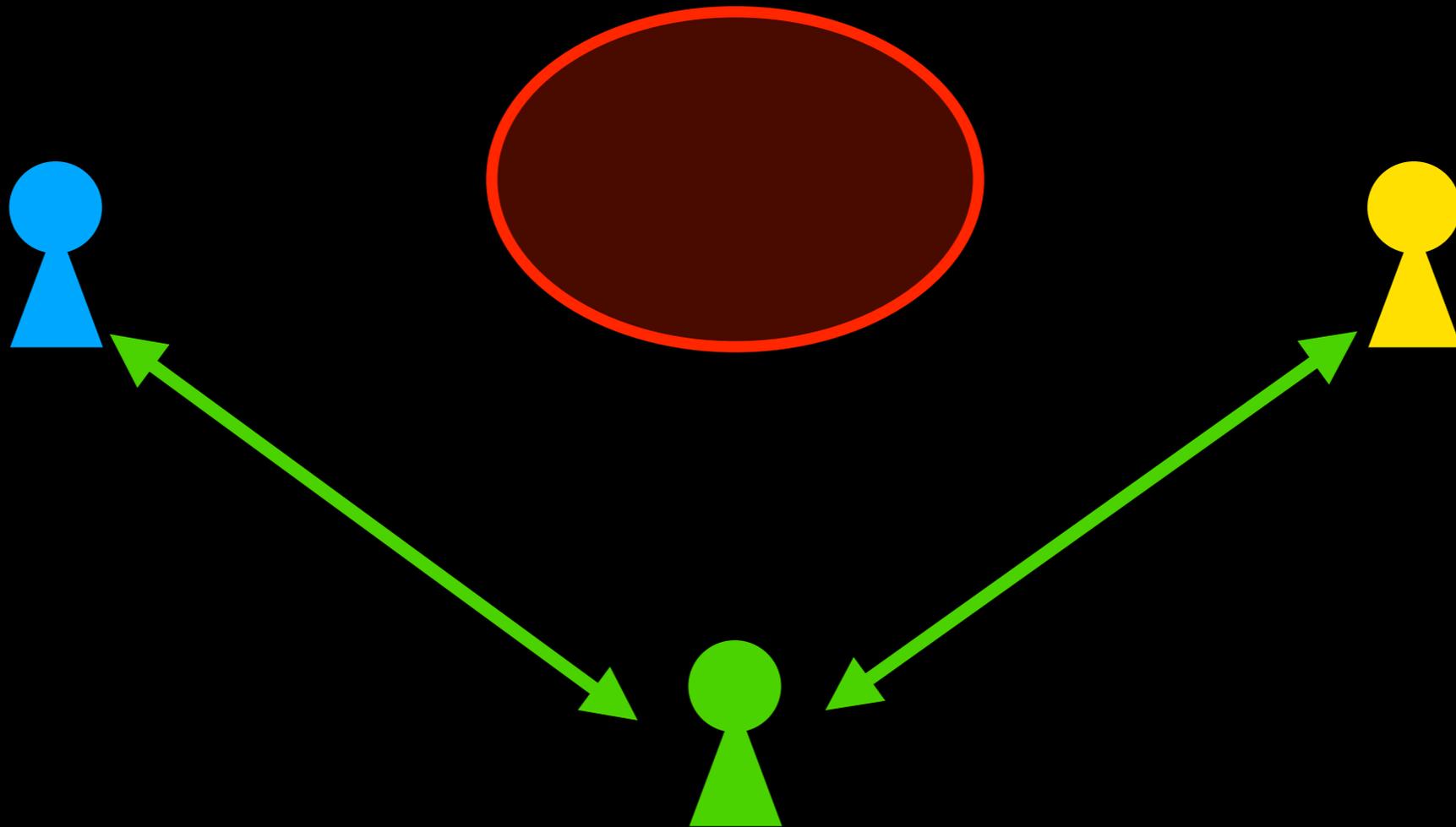


$\textcircled{A}$  is an **alibi**

# Achieving provable avoidance

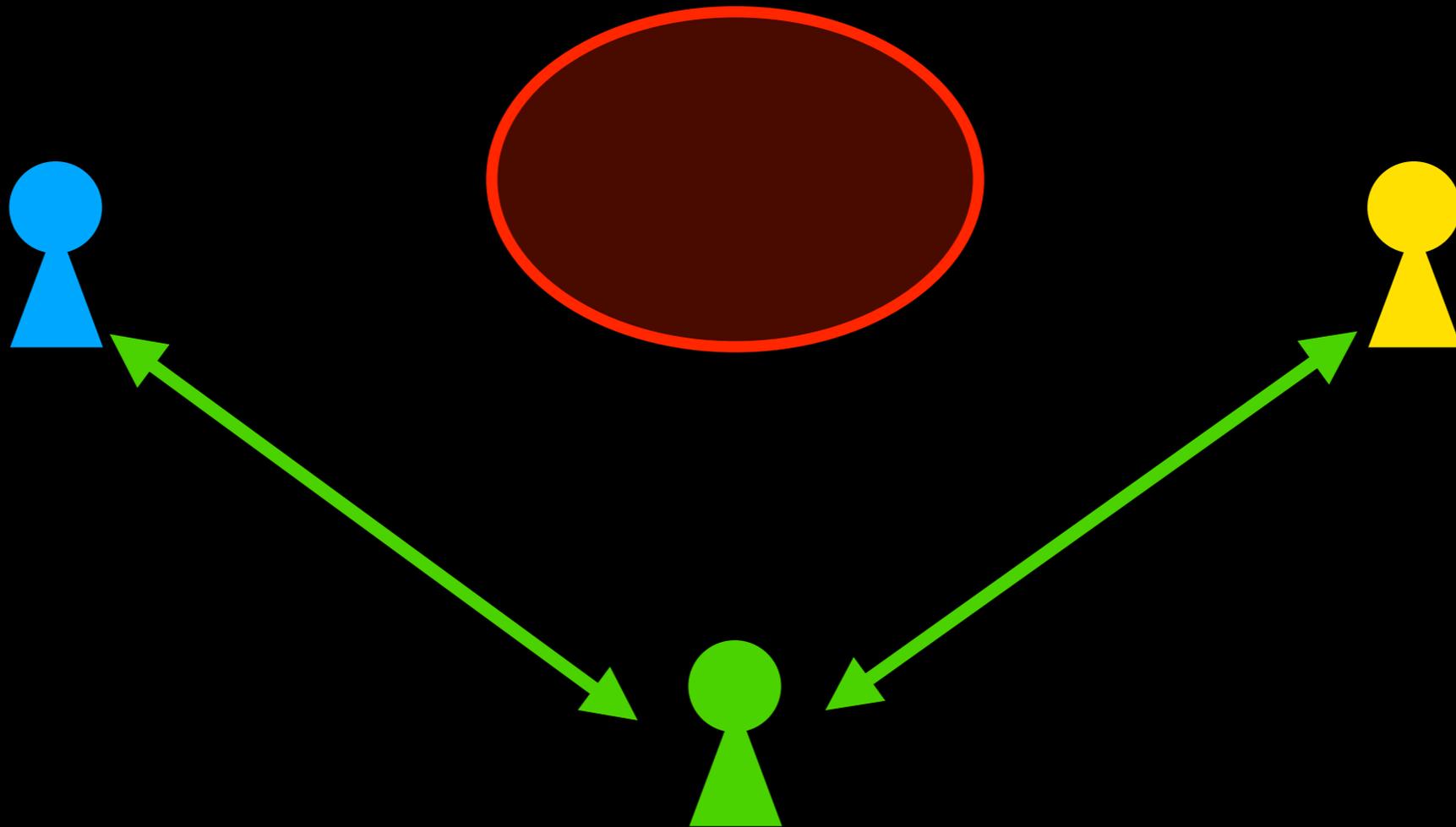


# Achieving provable avoidance



Solicit participation from a **relay**

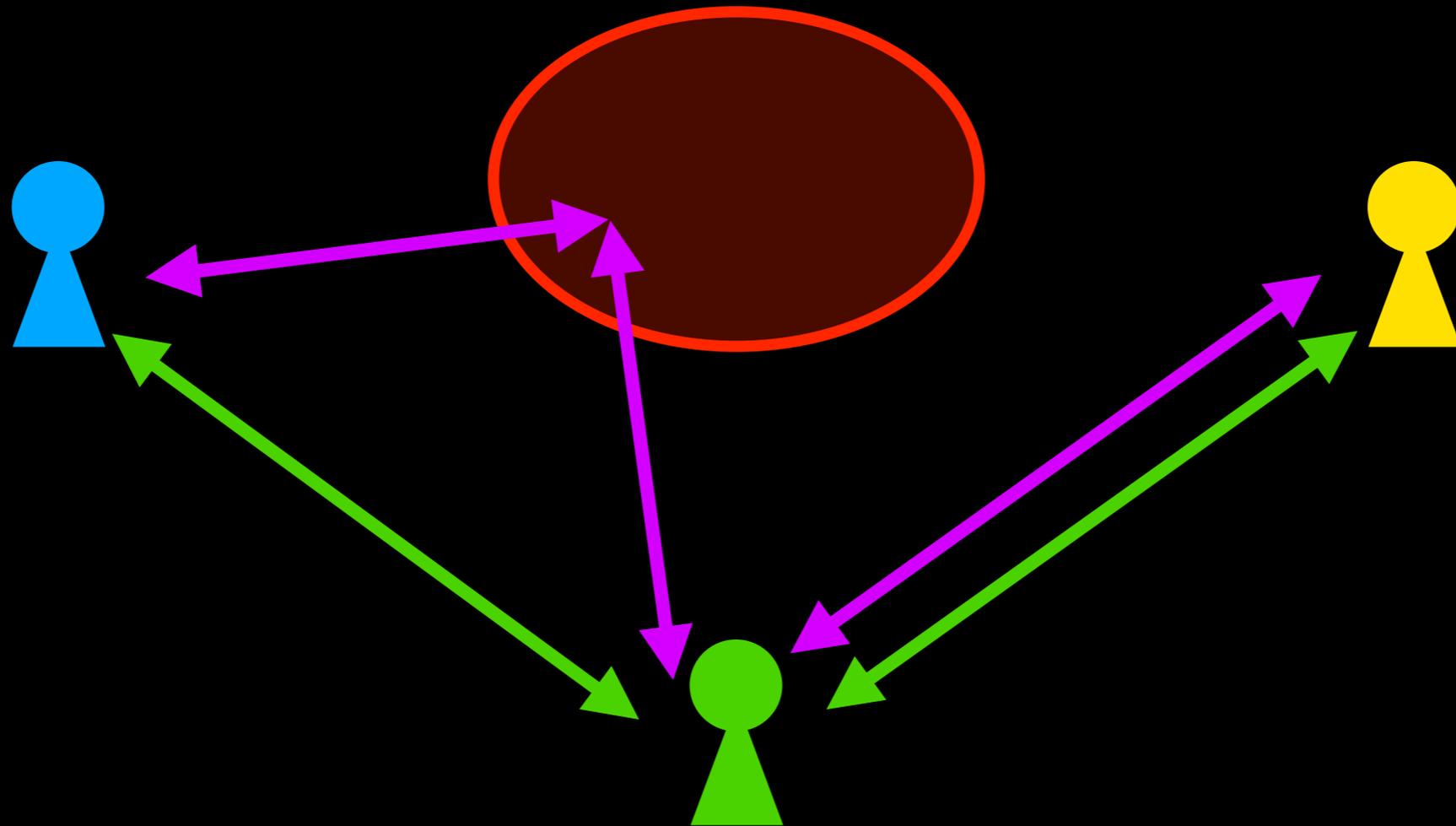
# Achieving provable avoidance



Reply contains a  
MAC from 

⇒ The packet traversed 

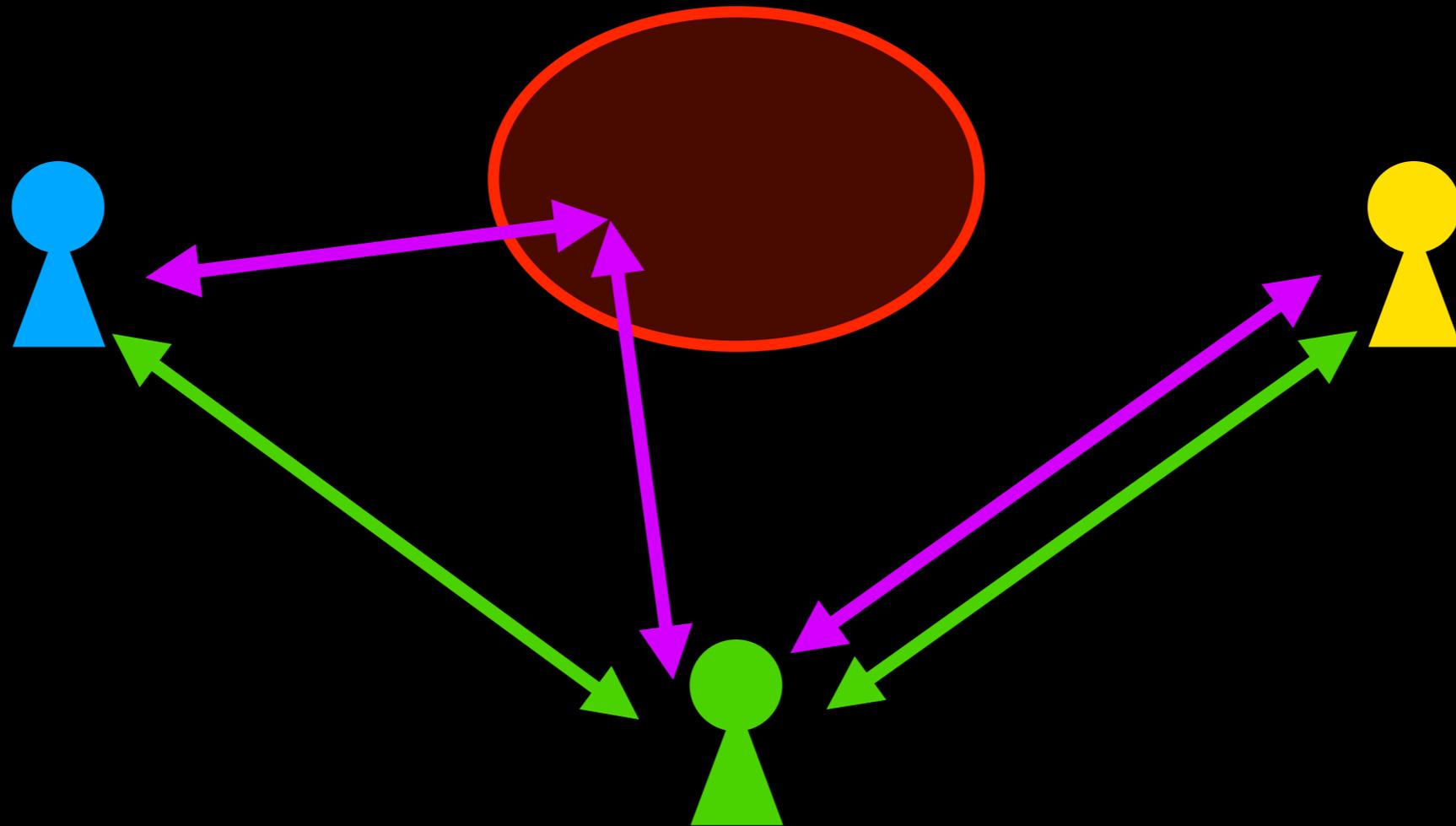
# Achieving provable avoidance



Reply contains a  
MAC from 

⇒ The packet traversed 

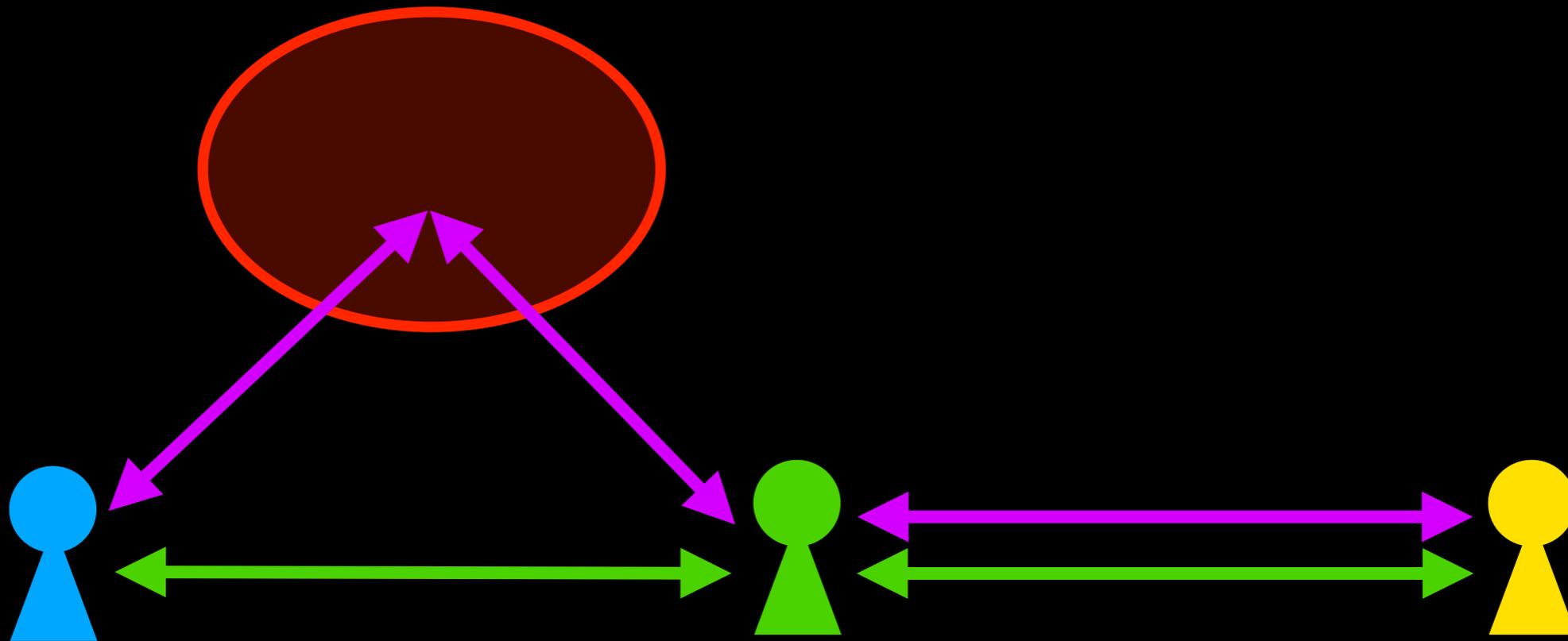
# Achieving provable avoidance



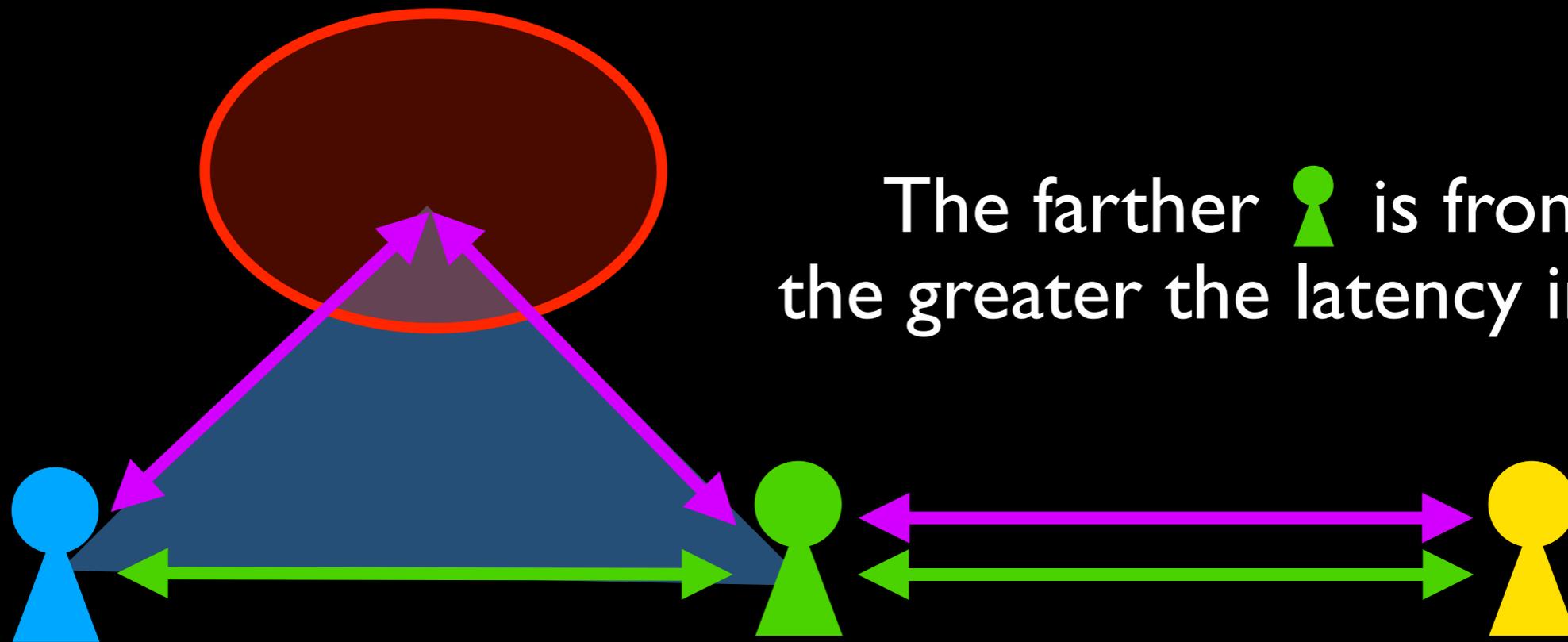
Reply contains a  
MAC from 

⇒ The packet traversed 

# Achieving provable avoidance

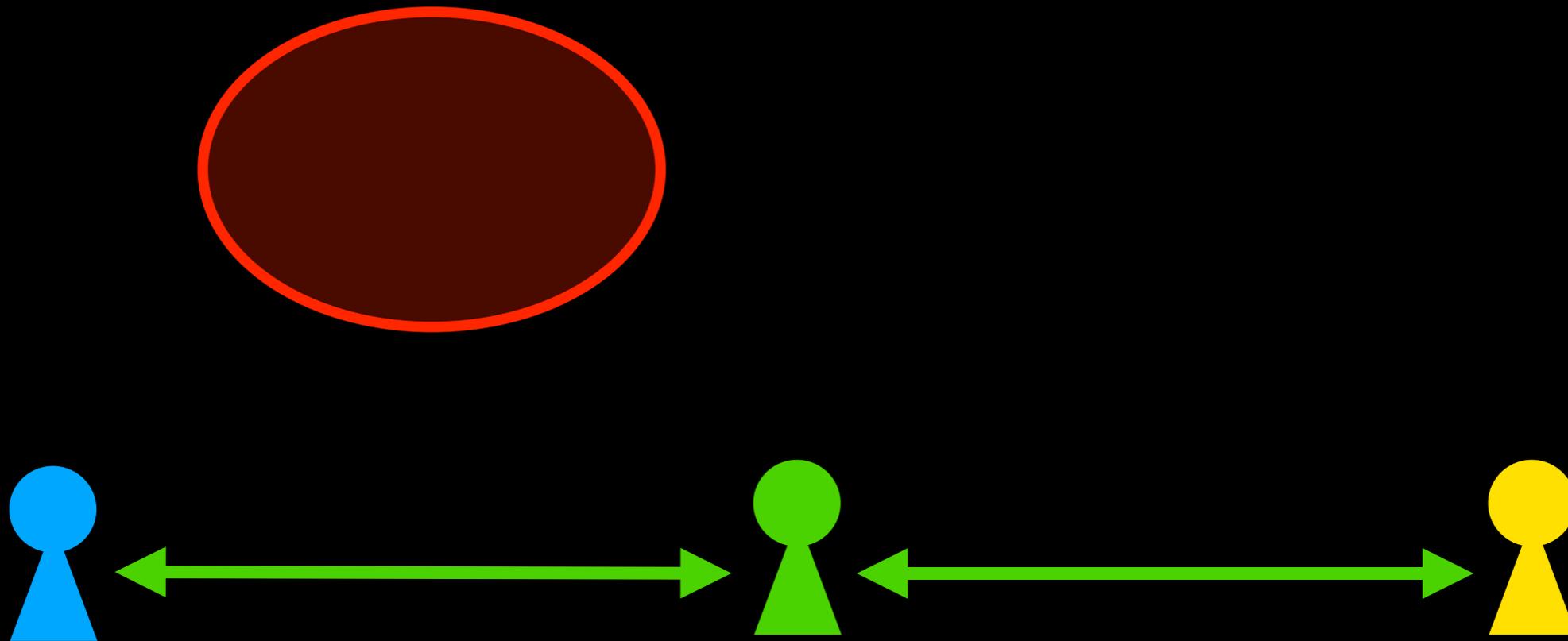


# Achieving provable avoidance

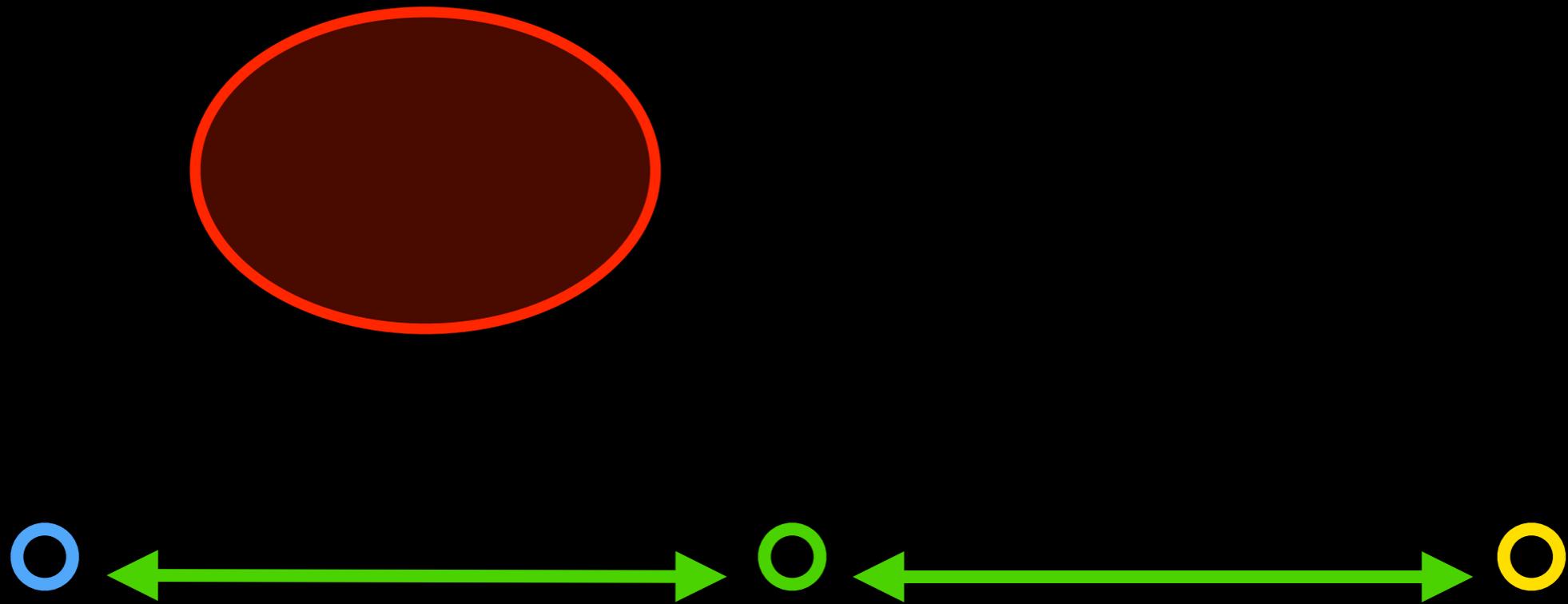


The farther  is from   
the greater the latency increase

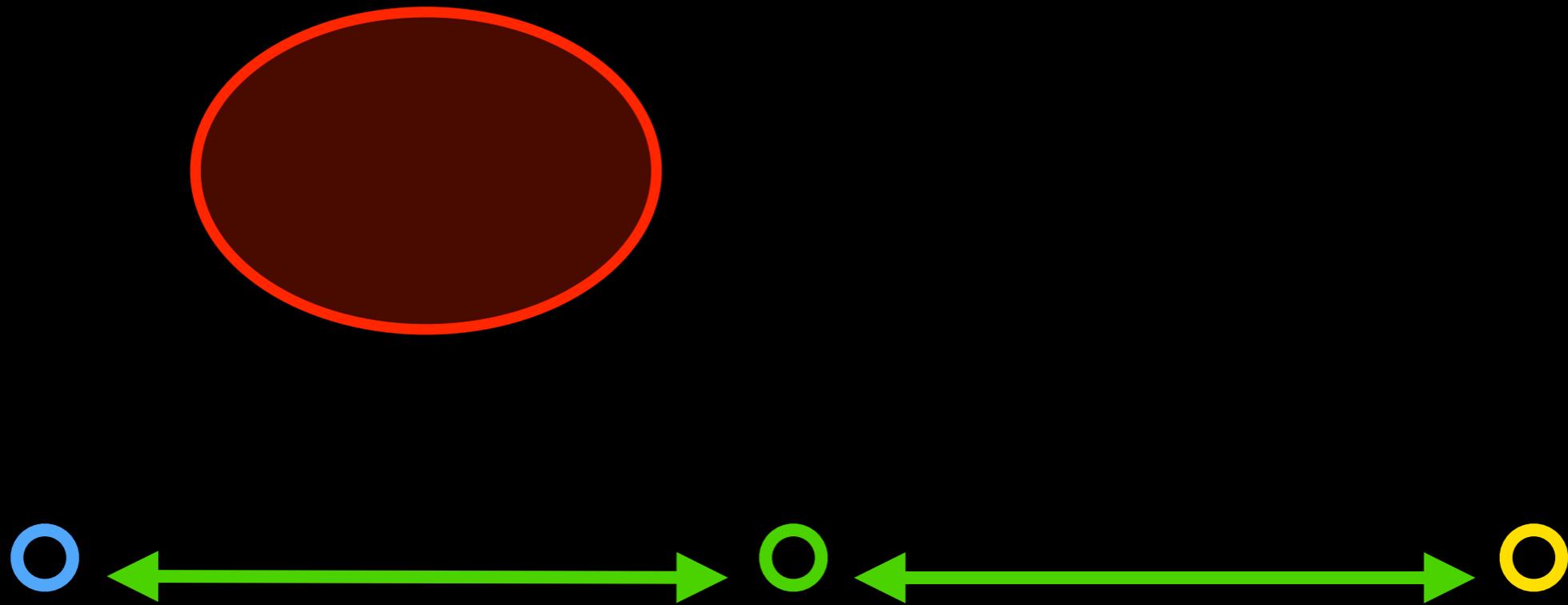
# Achieving provable avoidance



# Achieving provable avoidance

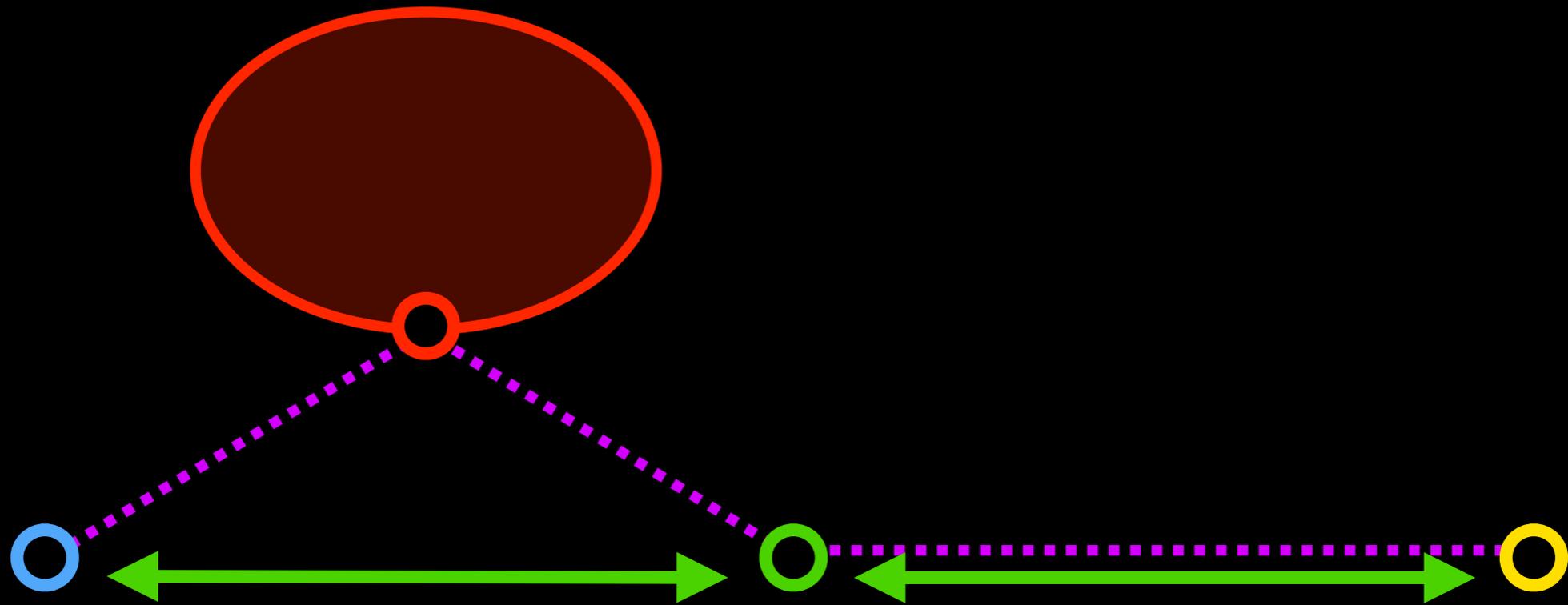


# Achieving provable avoidance



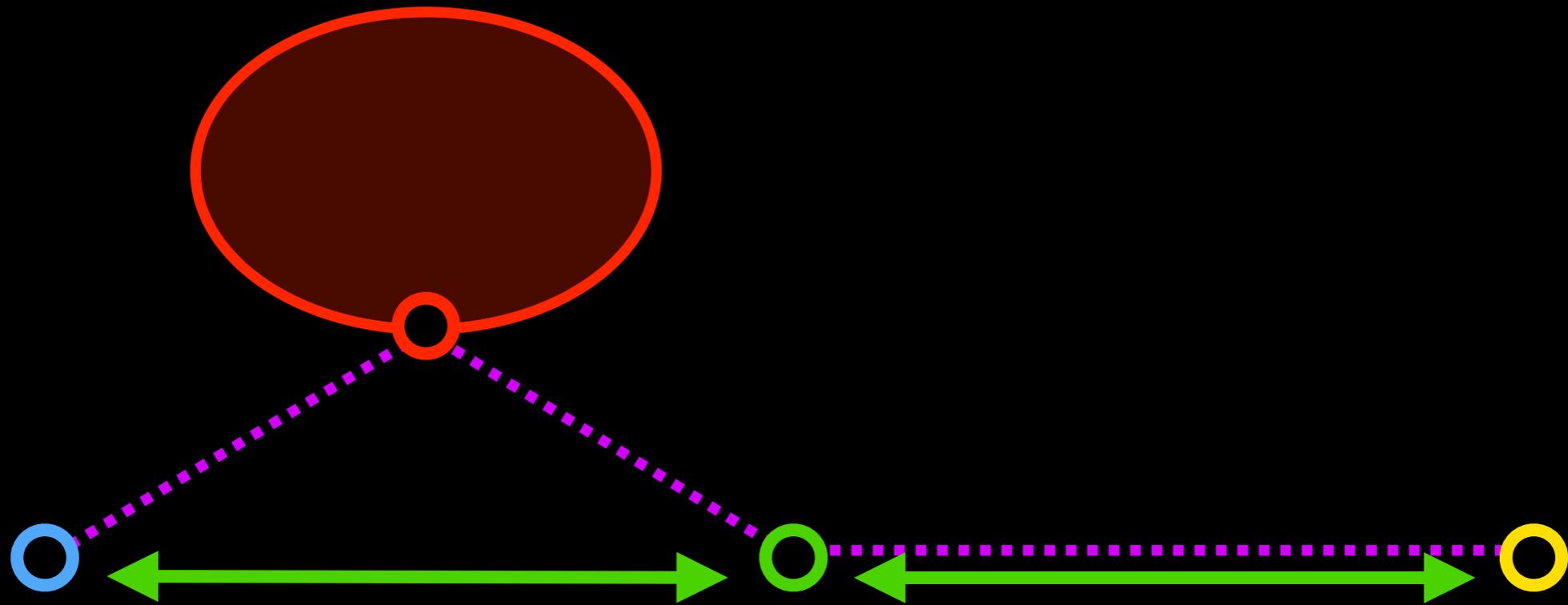
The shortest *possible* distance  
thru  and 

# Achieving provable avoidance



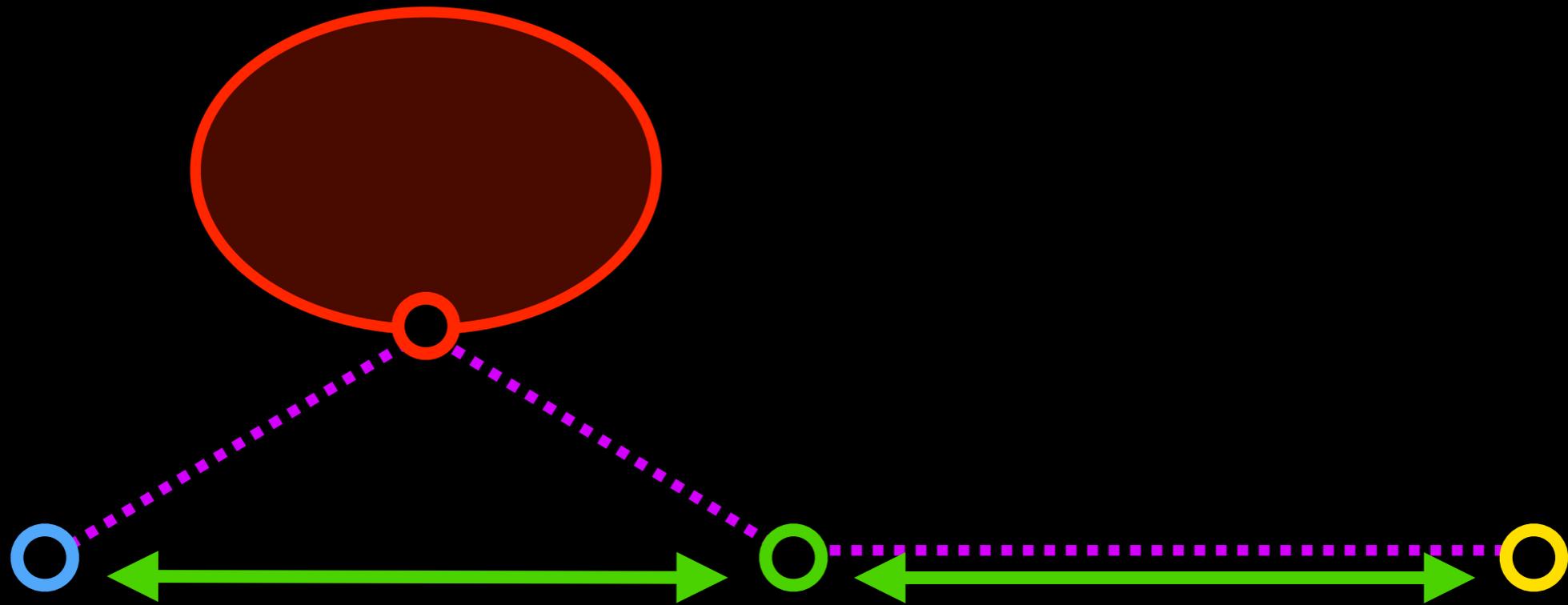
The shortest *possible* distance  
thru  and 

# Achieving provable avoidance



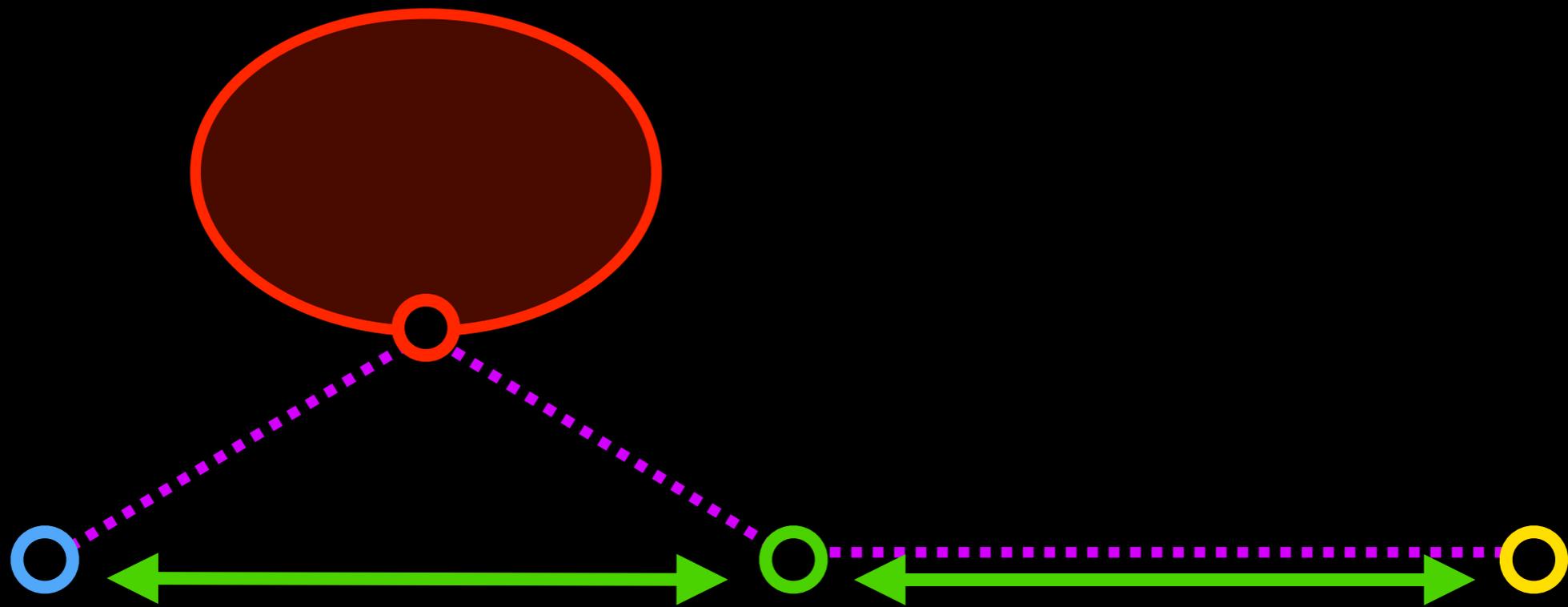
The shortest *possible* distance =  $d$   
thru  and 

# Achieving provable avoidance



The shortest *possible* RTT  
thru  and  =  $2d/c$

# Achieving provable avoidance



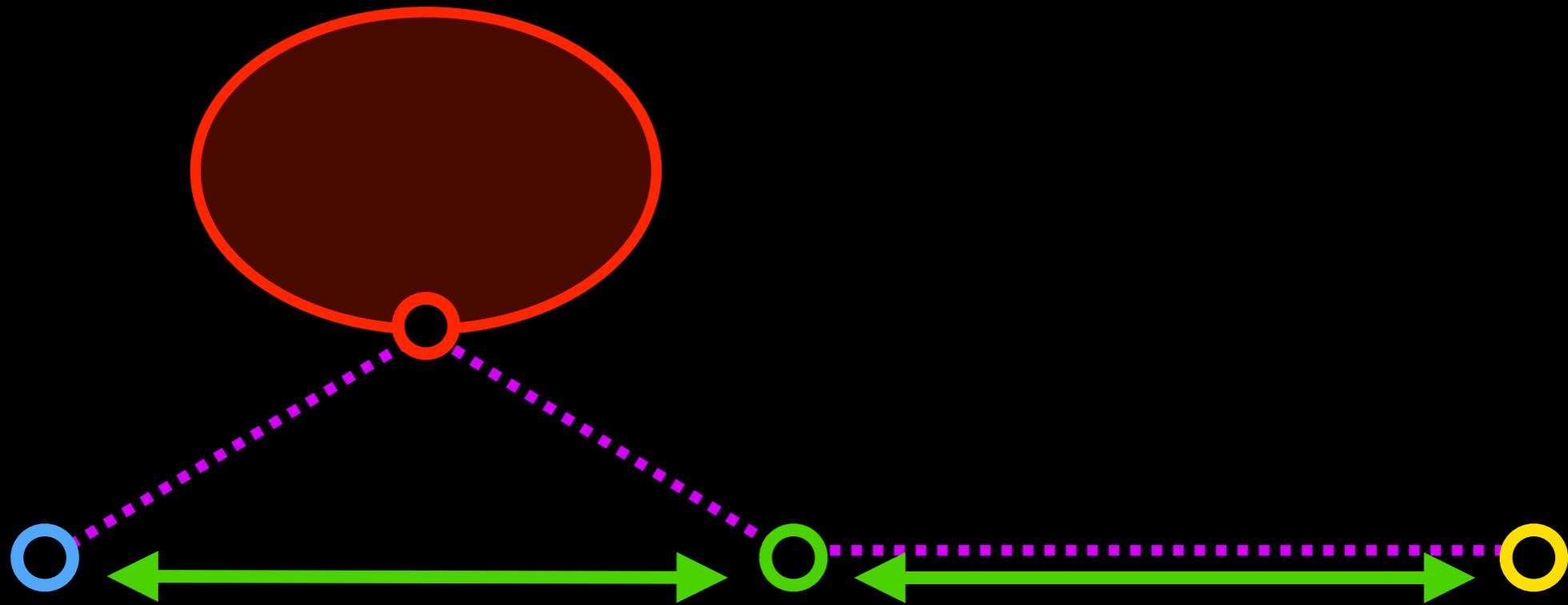
Measured  
RTT

$\ll$

The shortest *possible* RTT  
thru  and 

=  $2d/c$

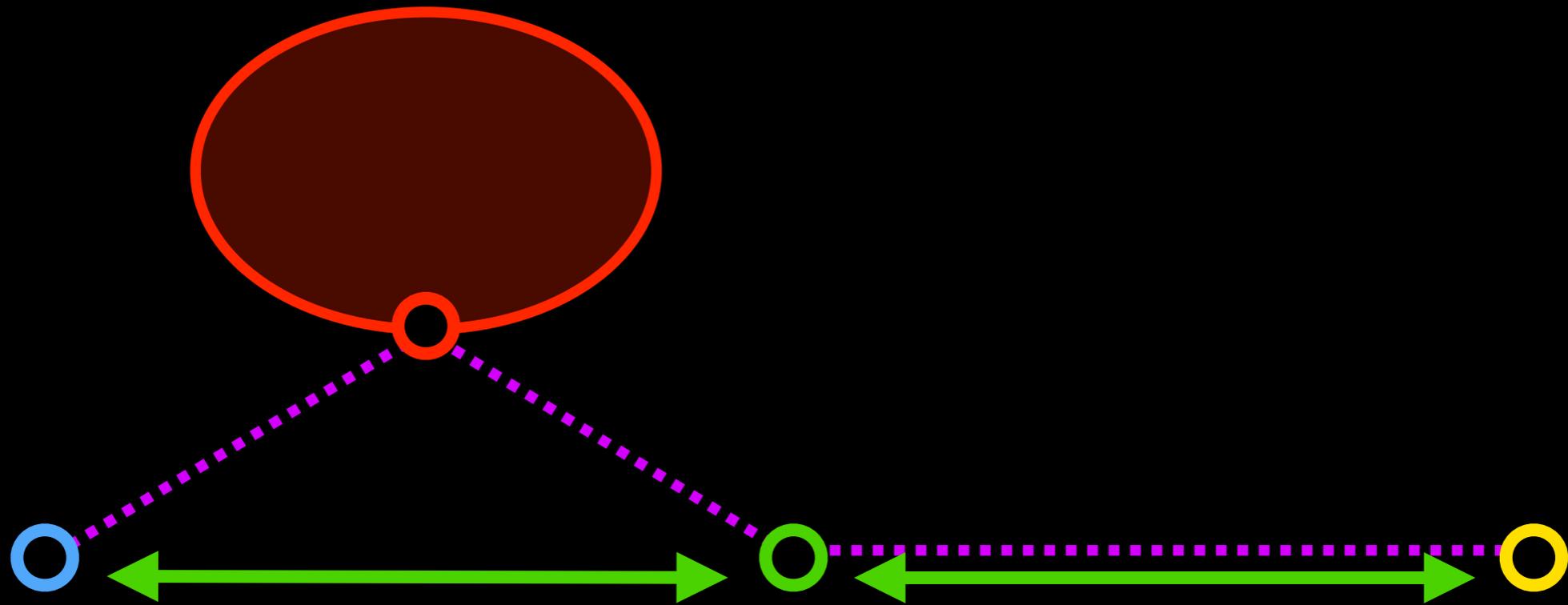
# Achieving provable avoidance



Measured **RTT**  $\ll$  The shortest *possible* **RTT** thru  and  =  $2d/c$

$\Rightarrow$  It could not have traversed  and 

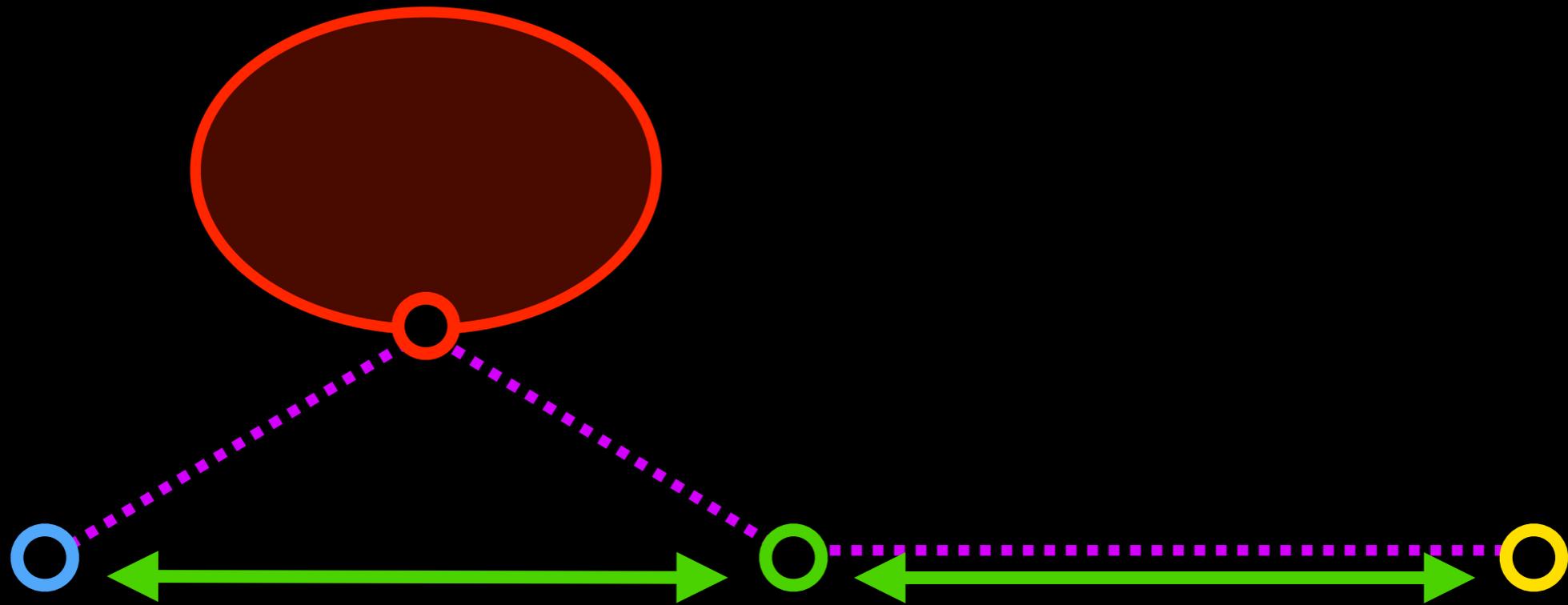
# Achieving provable avoidance



$$\text{Measured RTT} \ll 2d/c$$

⇒ It could not have traversed  and 

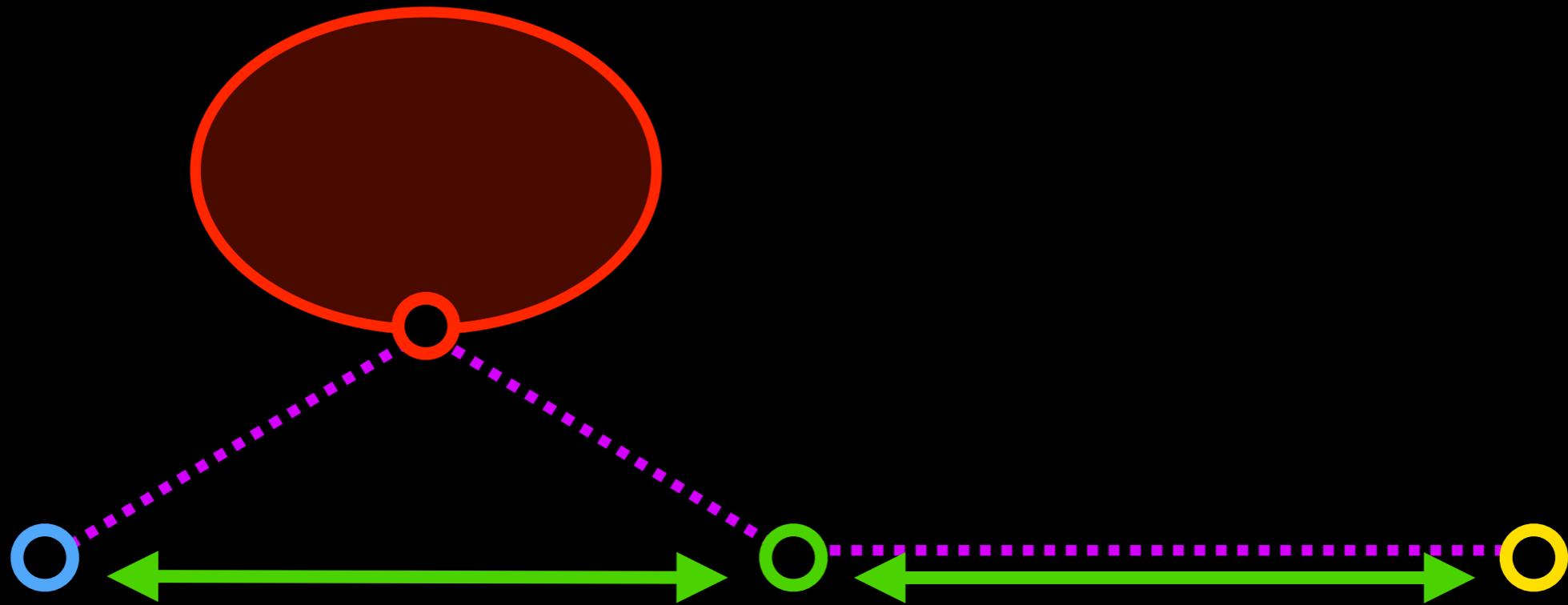
# Achieving provable avoidance



Measured  
RTT  $\ll 3d/c$

$\Rightarrow$  It could not have traversed  and 

# Achieving provable avoidance



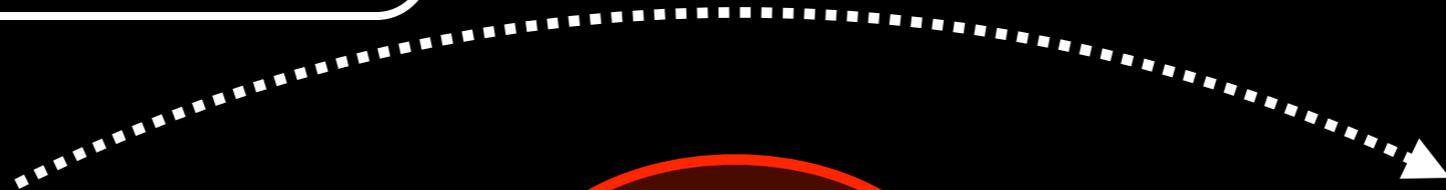
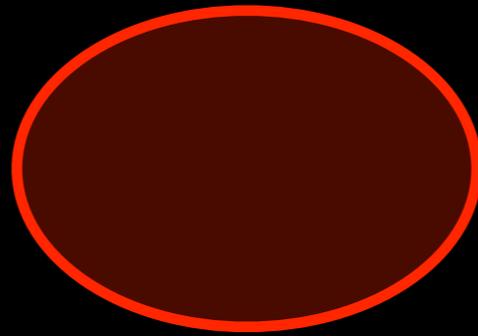
Safety factor

$$(1 + \delta) * \text{Measured RTT} \ll 3 d / c$$

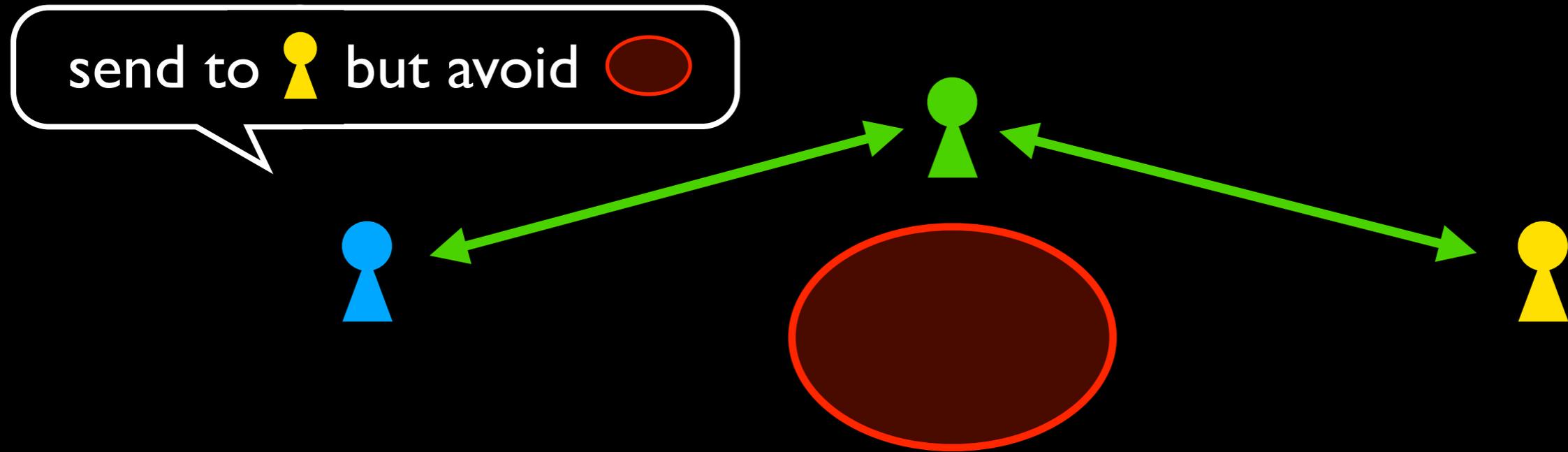
⇒ It could not have traversed  and 

# Achieving provable avoidance

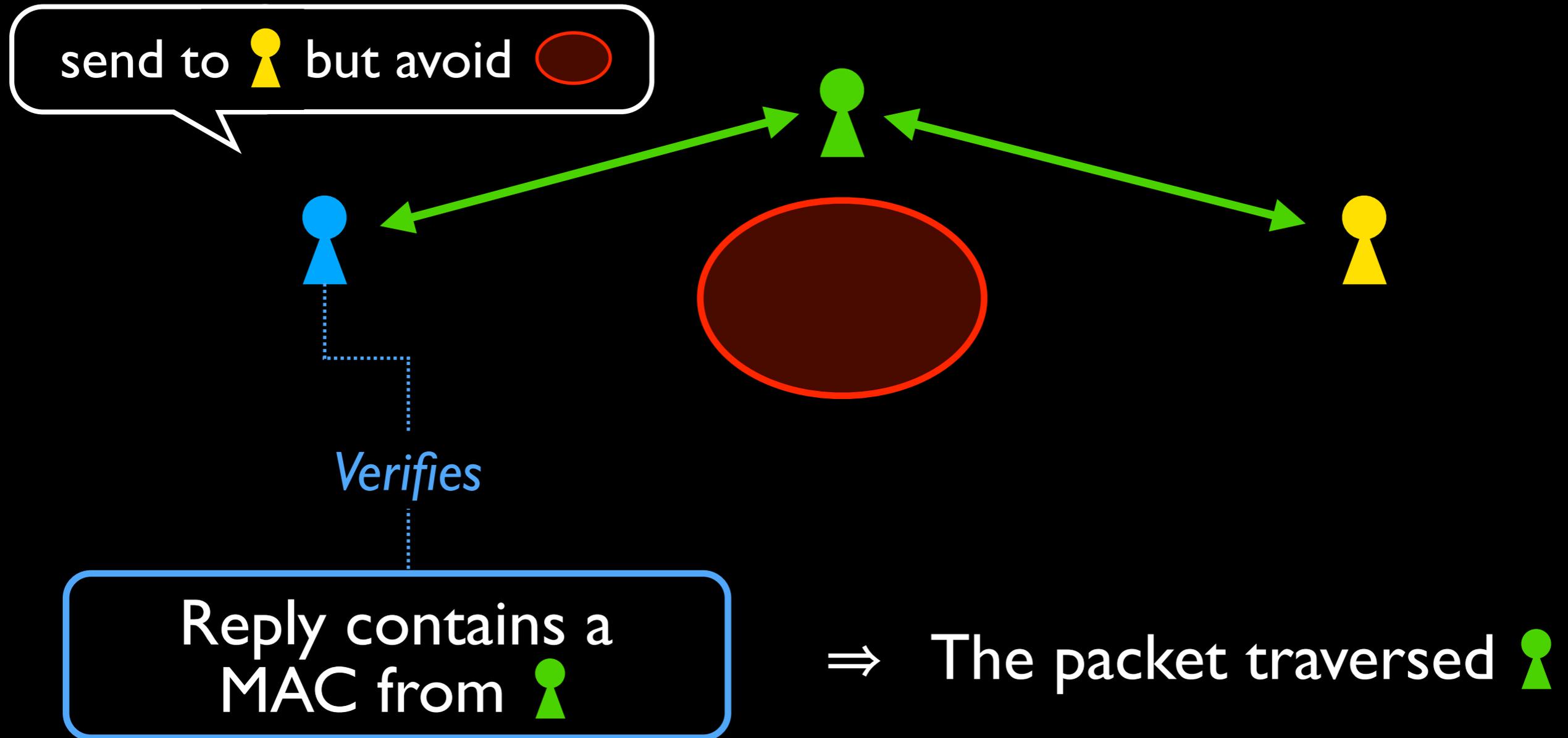
send to  but avoid 



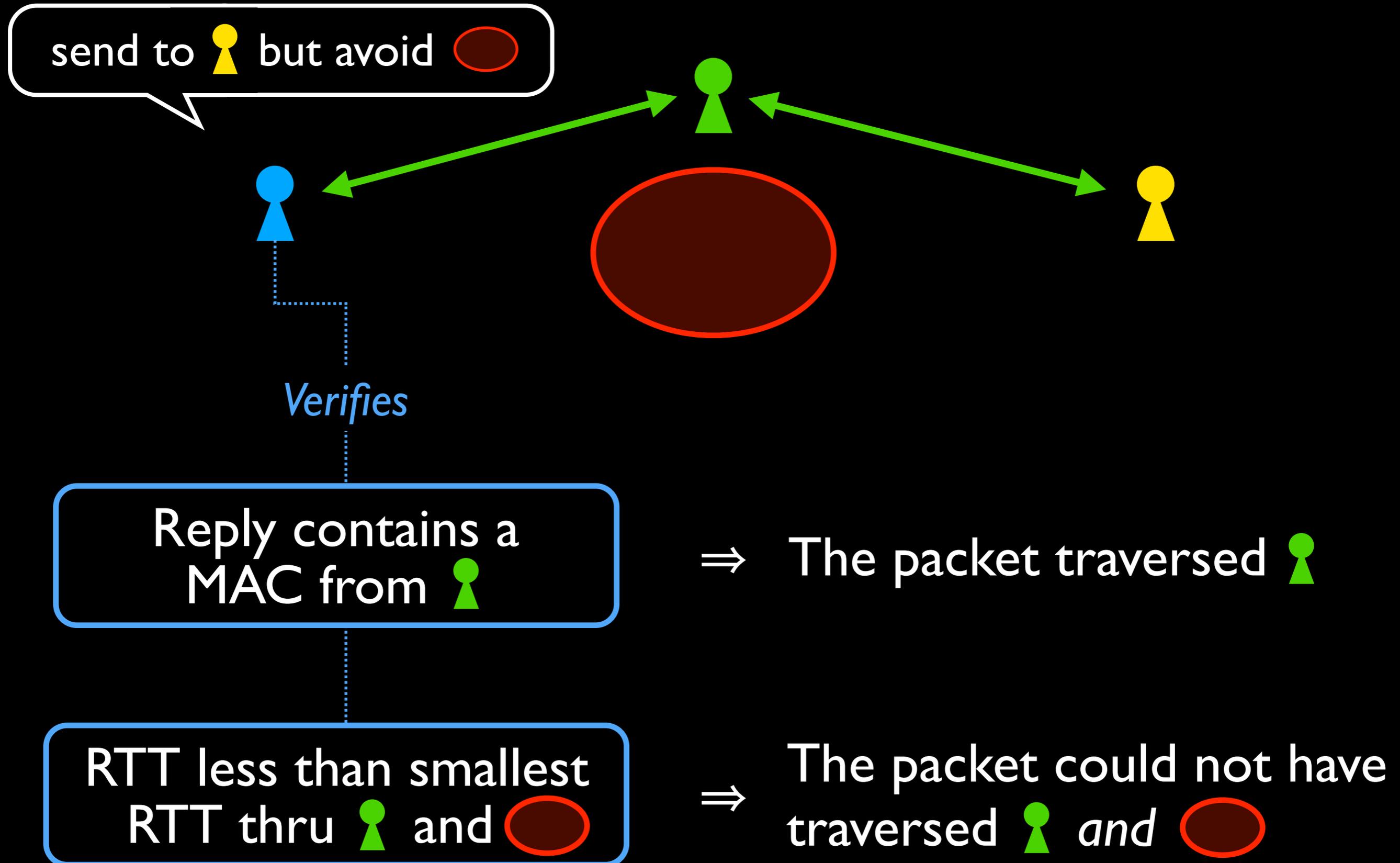
# Achieving provable avoidance



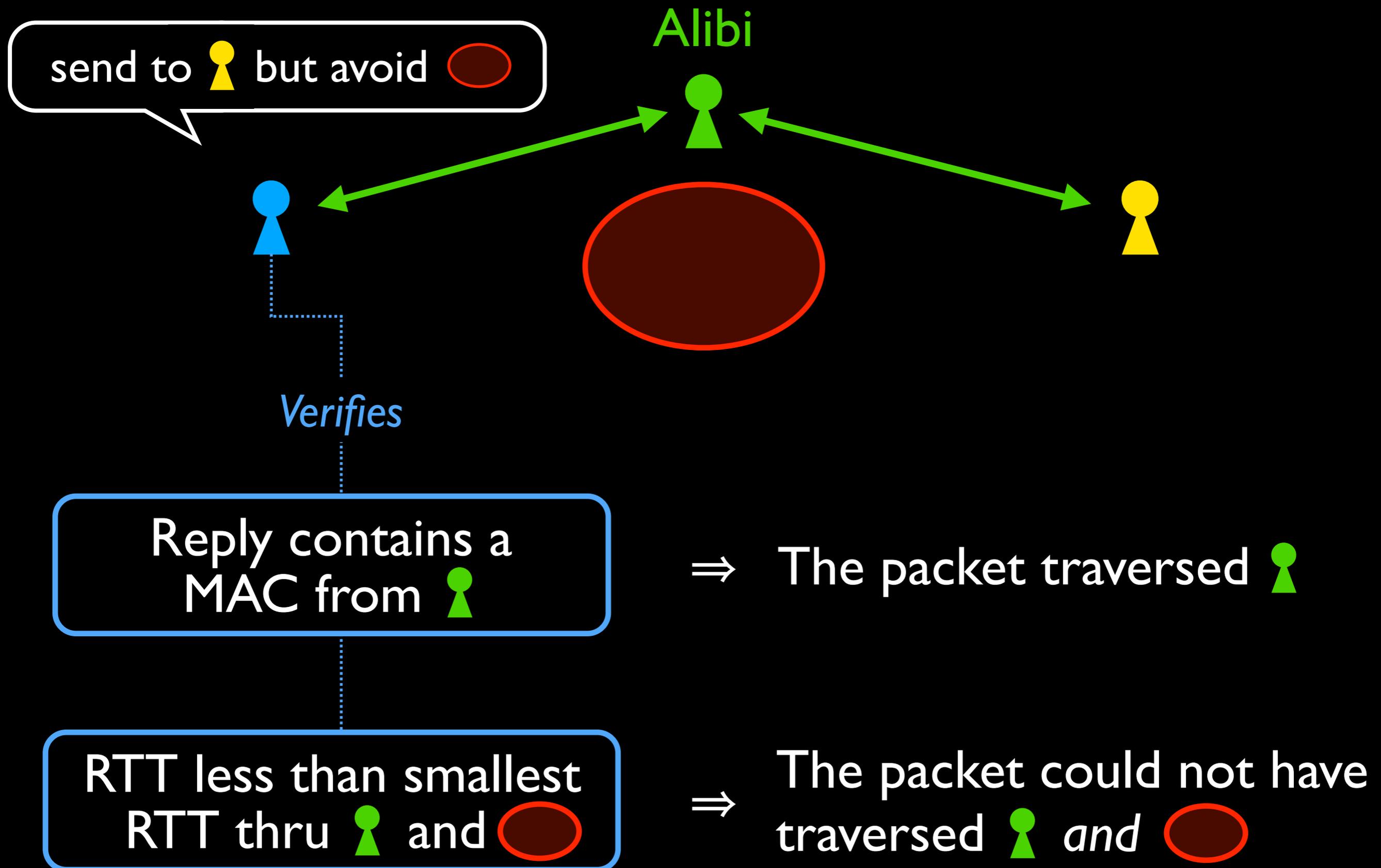
# Achieving provable avoidance



# Achieving provable avoidance



# Achieving provable avoidance



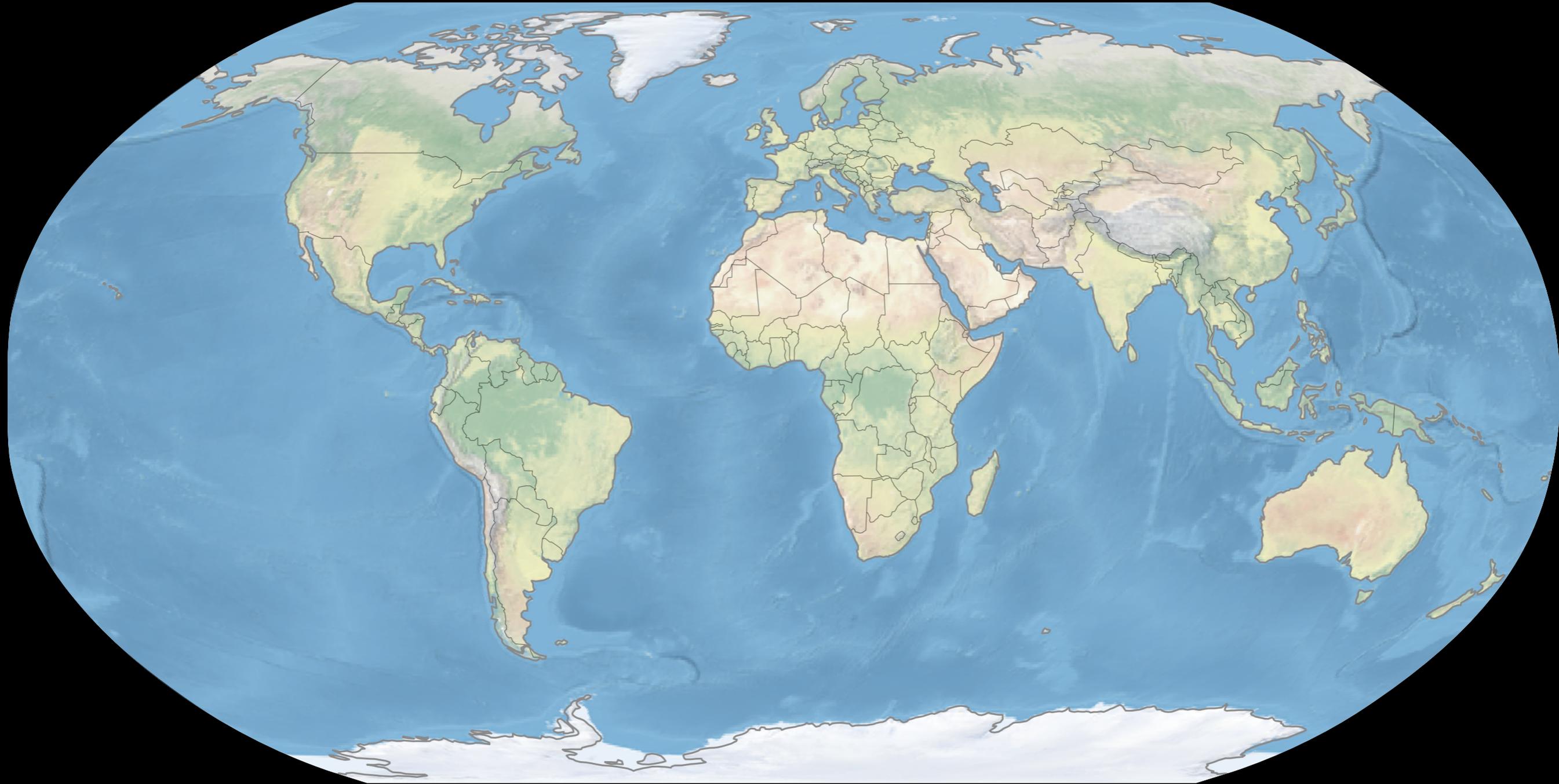
# Alibi Routing

Peer-to-peer protocol for  
finding potential alibis

- Users choose **forbidden regions**
- Users compute **target regions**
  - Where alibis *might* be
- Alibi Routing **recursively searches** for peers within the target regions

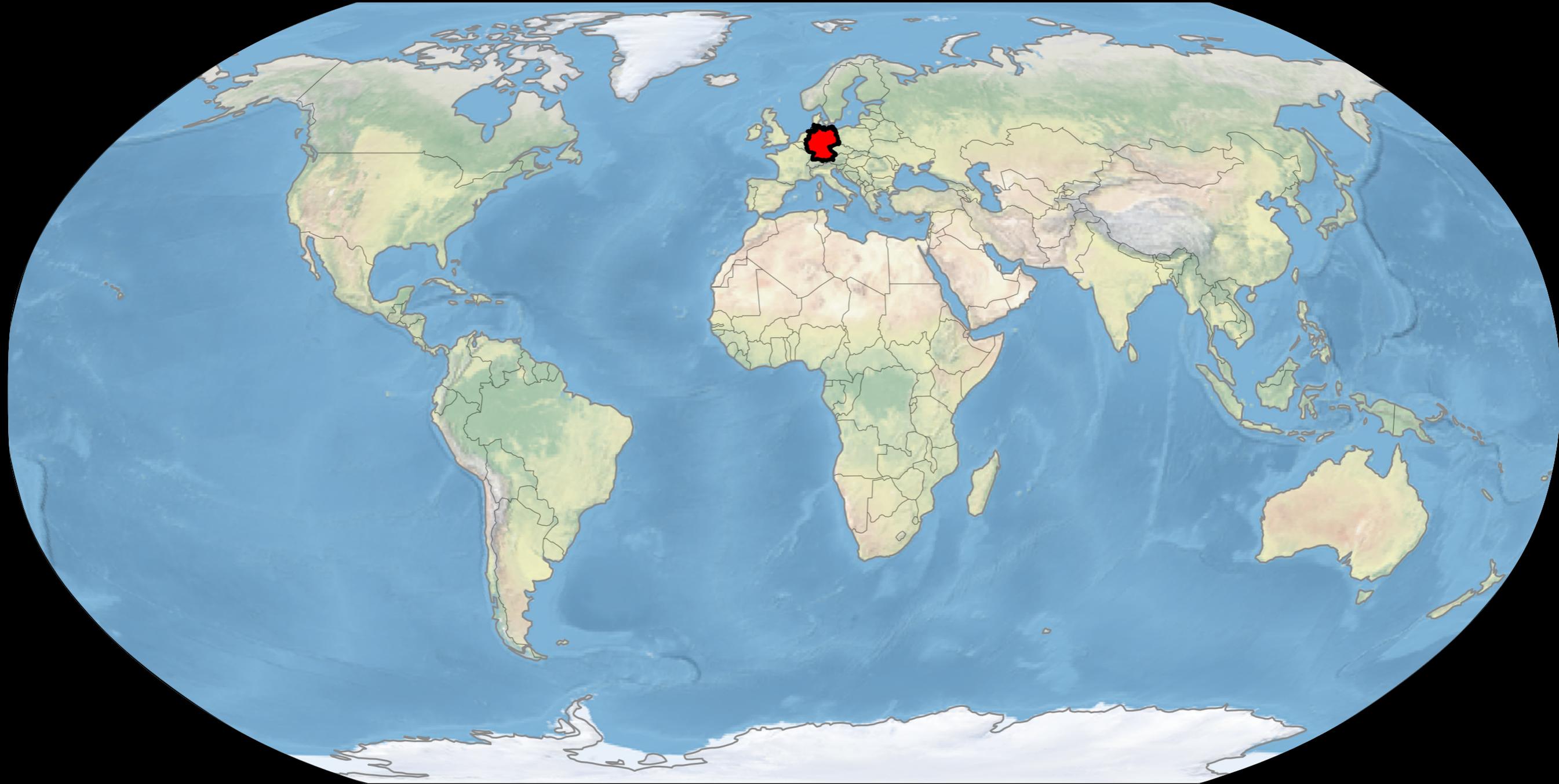
# Choose forbidden regions

User-specified regions to avoid



# Choose forbidden regions

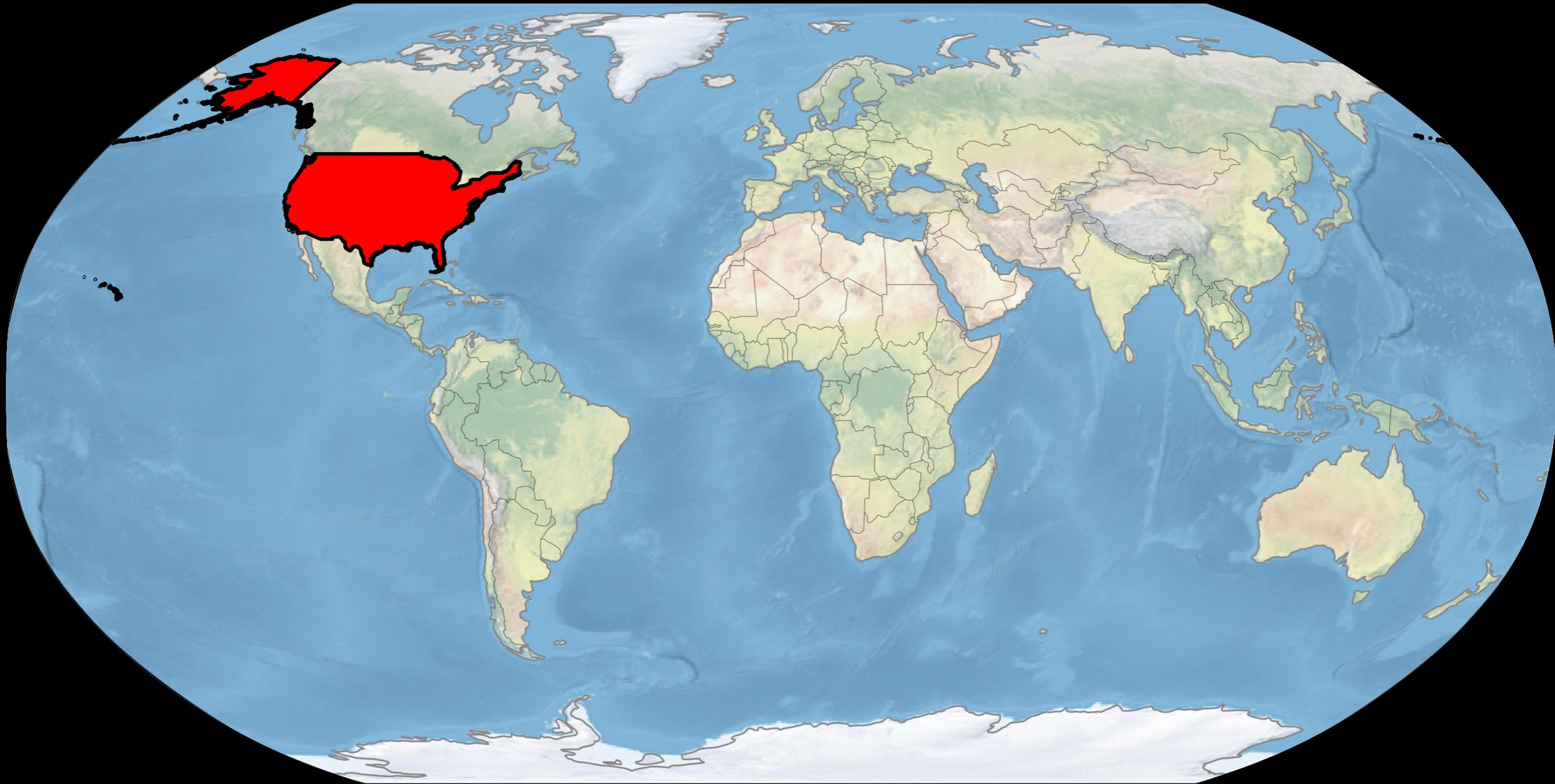
User-specified regions to avoid



Arbitrary sets of polygons, defined over lat/lon

# Choose forbidden regions

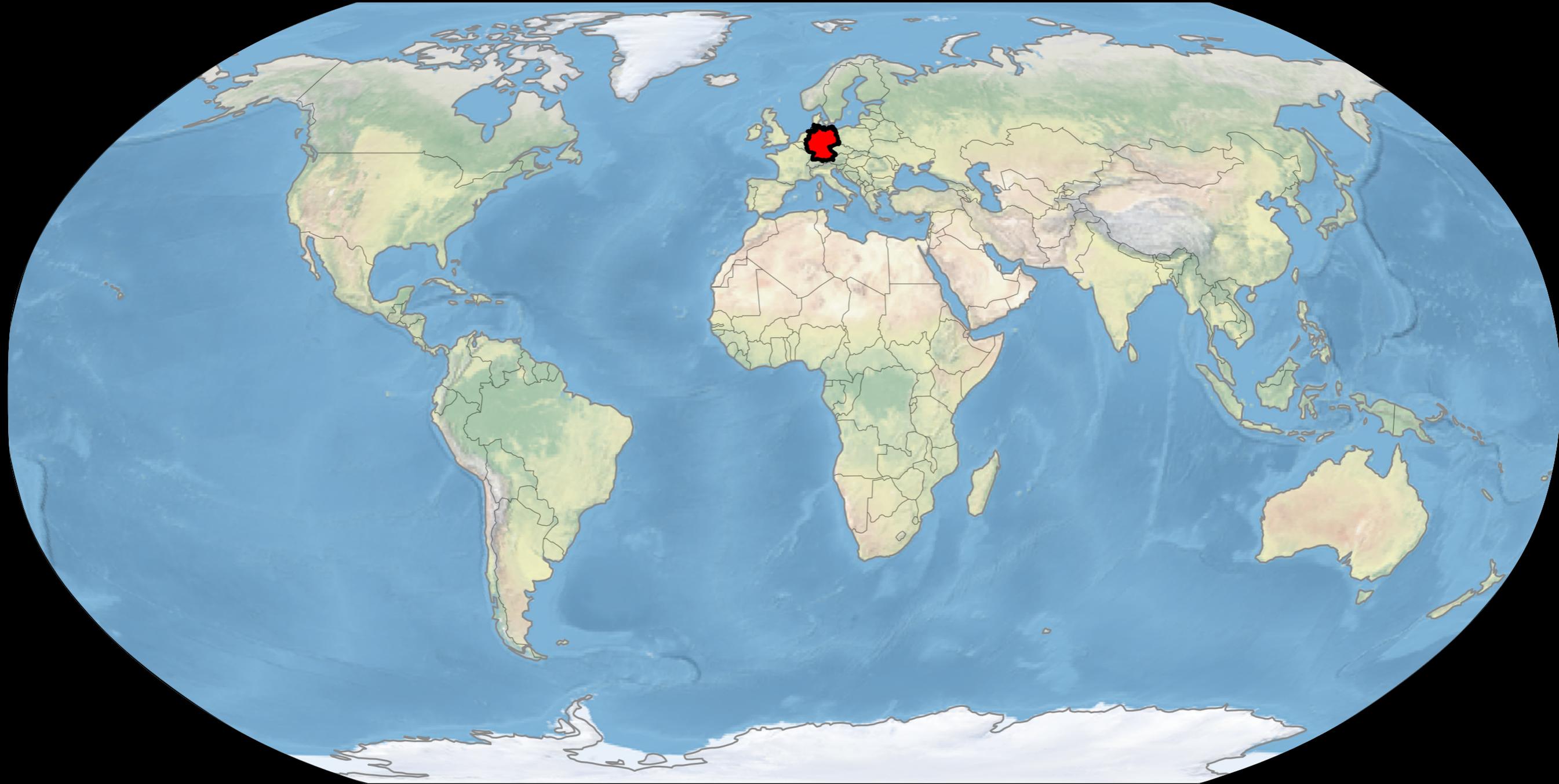
User-specified regions to avoid



Arbitrary sets of polygons, defined over lat/lon

# Choose forbidden regions

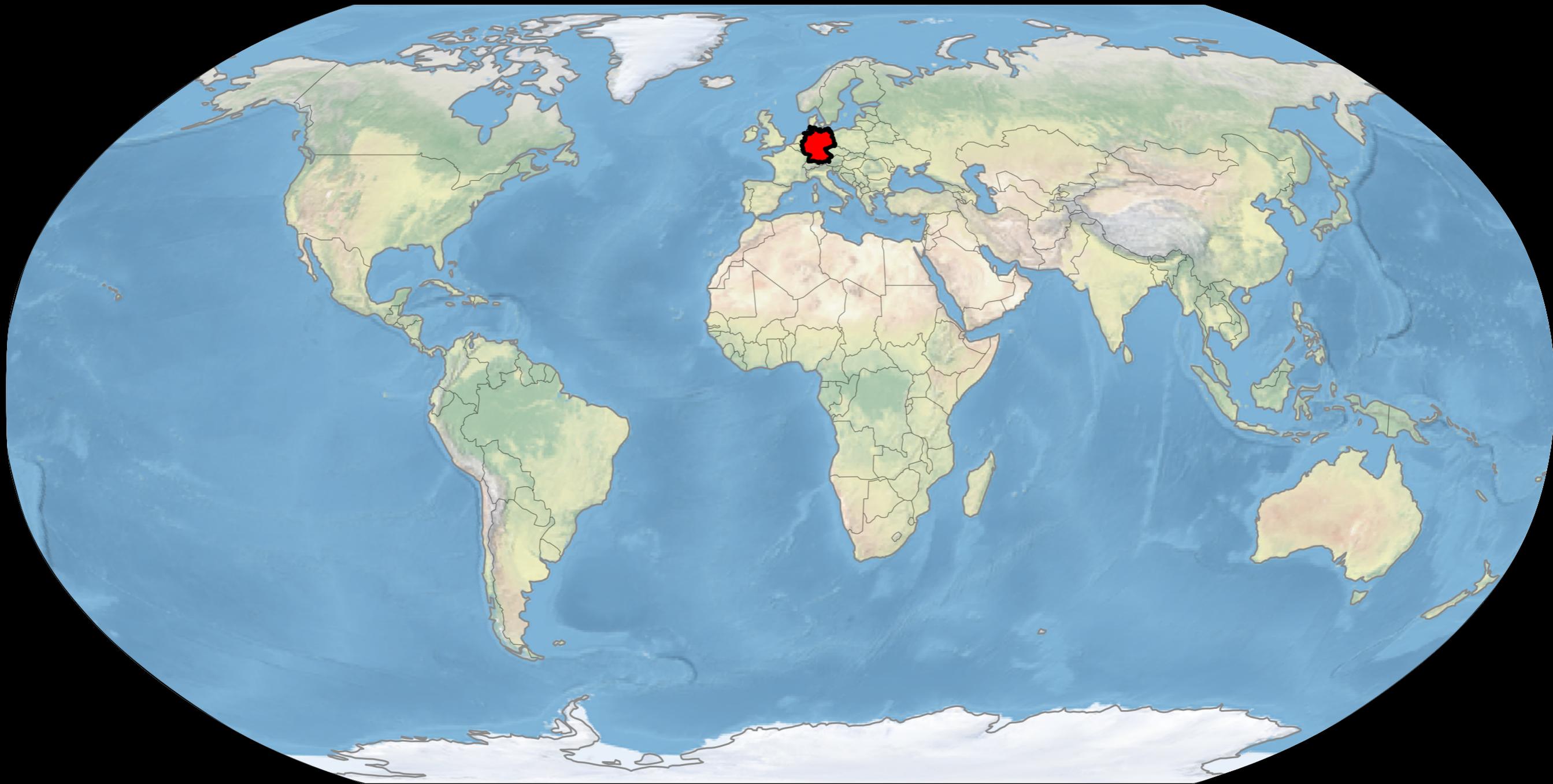
User-specified regions to avoid



Arbitrary sets of polygons, defined over lat/lon

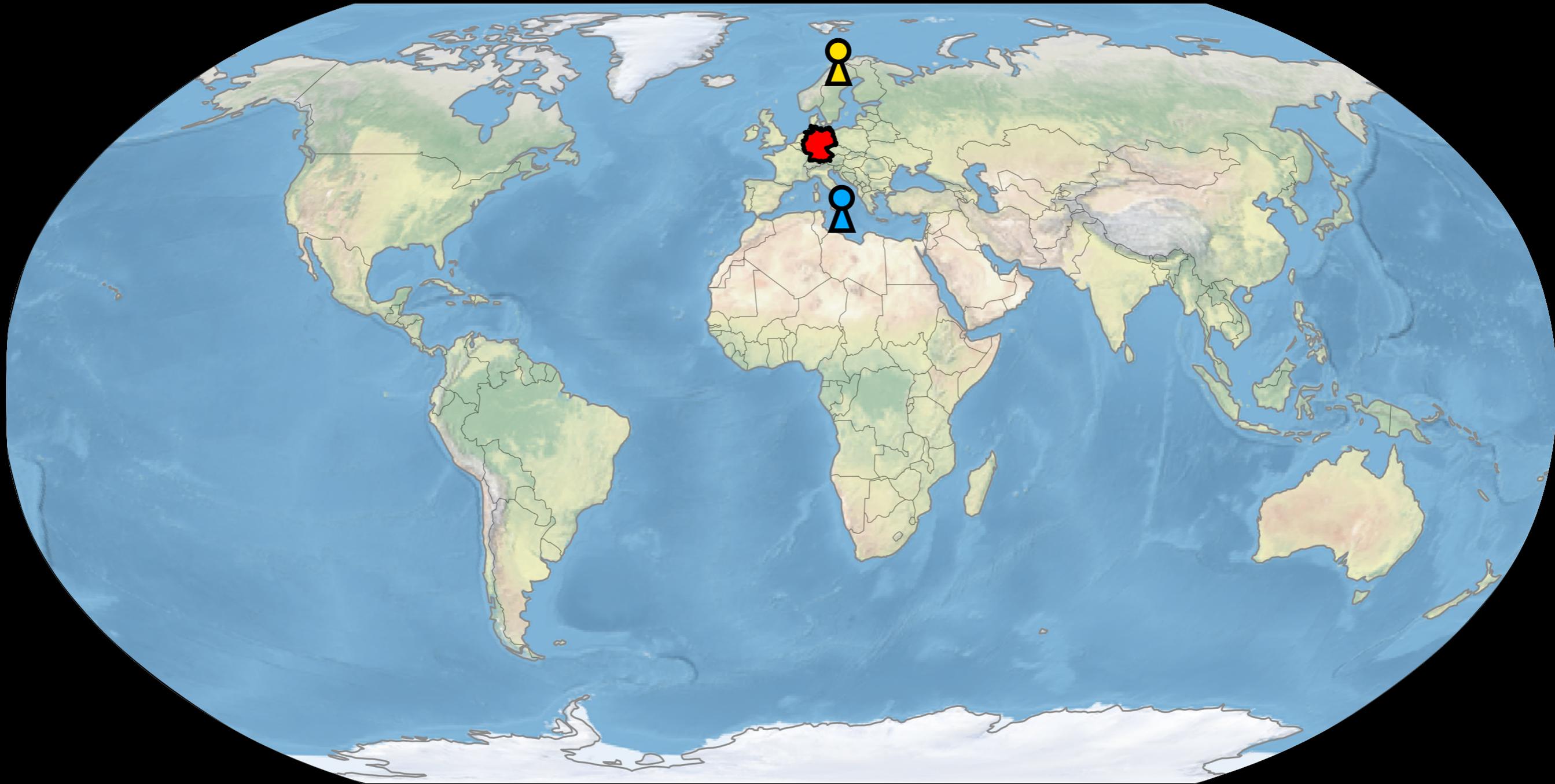
# Compute target regions

Where alibis *might* be



# Compute target regions

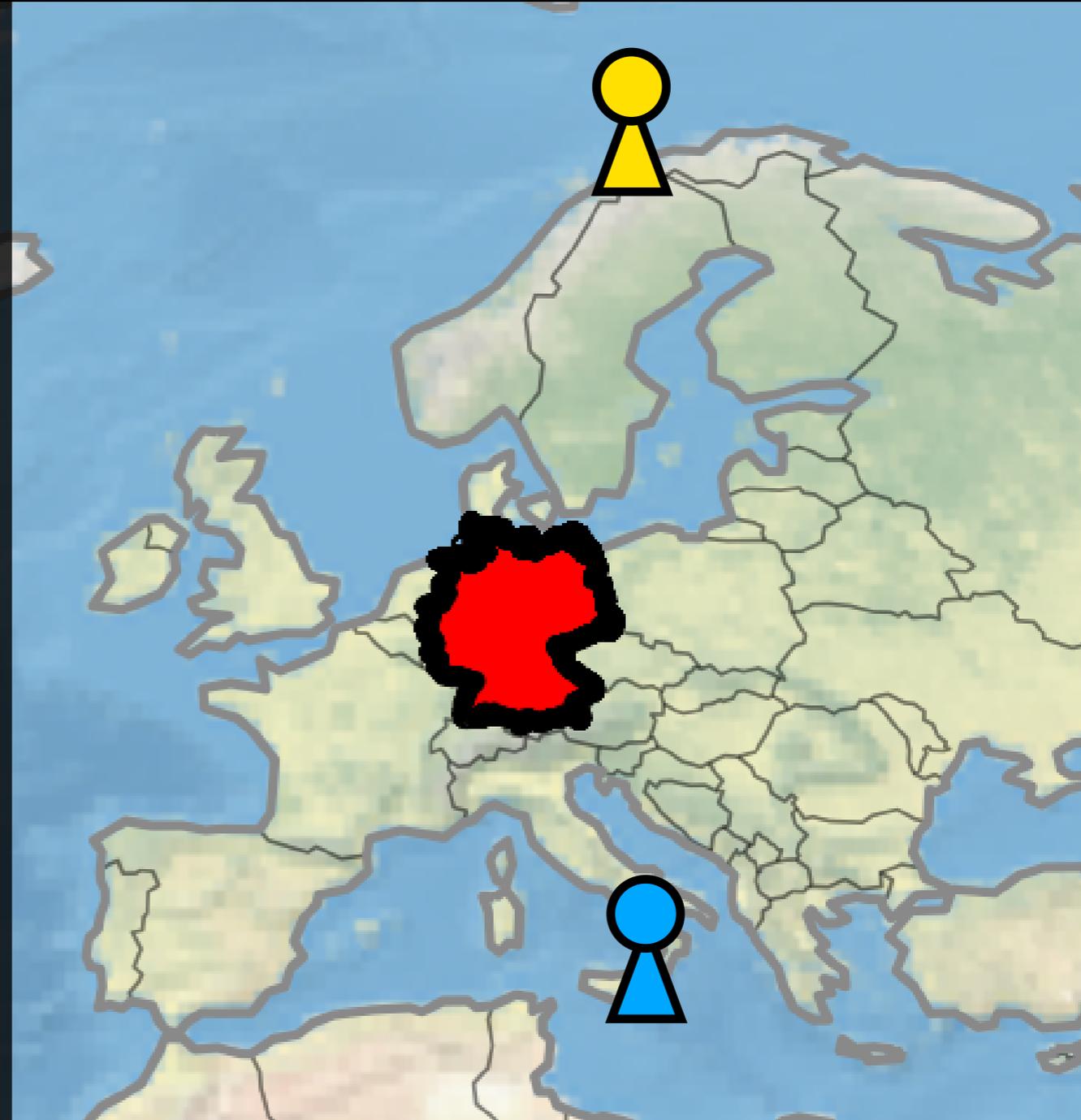
Where alibis *might* be



# Compute target regions

Where alibis *might* be

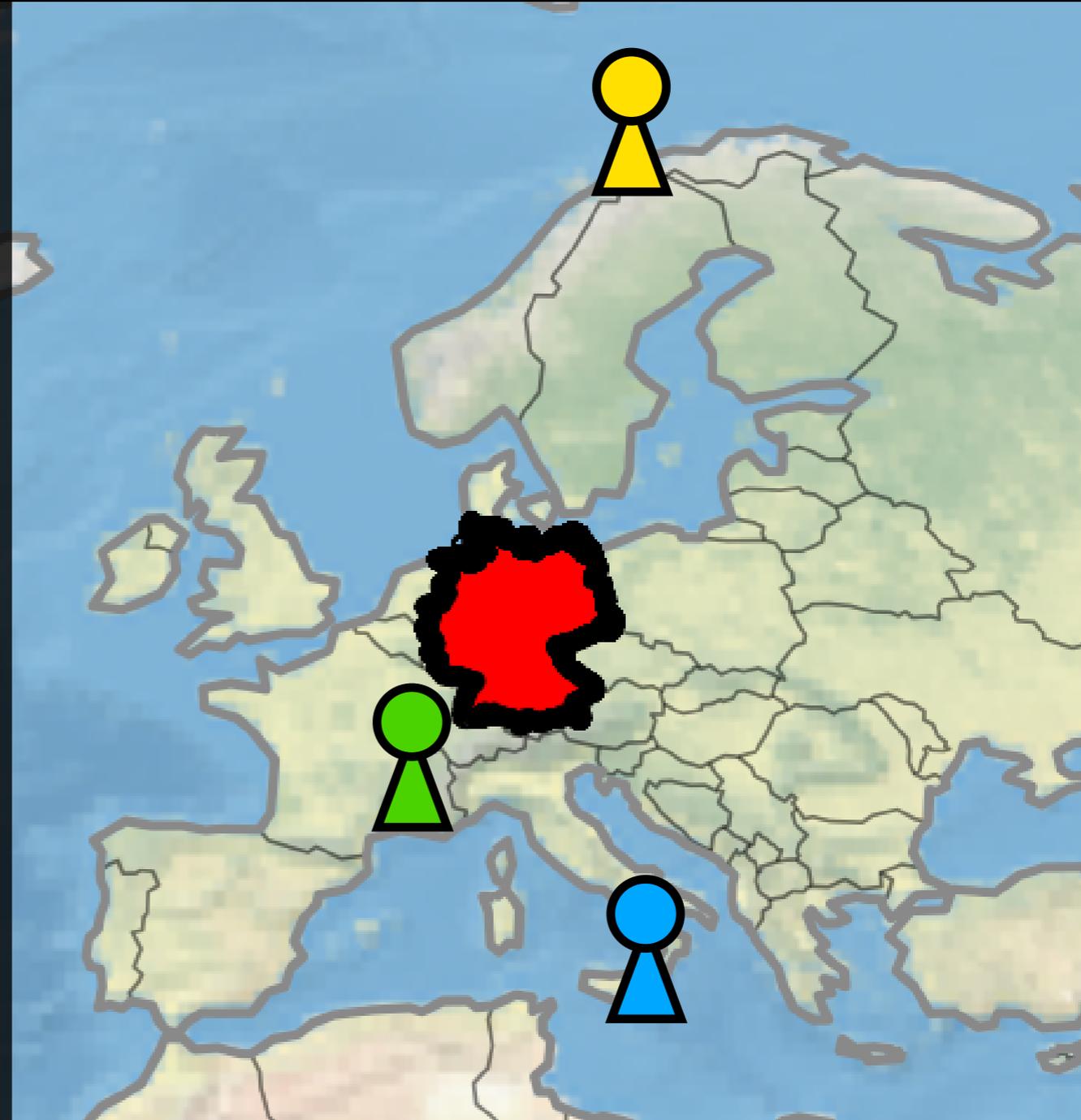
Exclude locations where  
alibis cannot exist



# Compute target regions

Where alibis *might* be

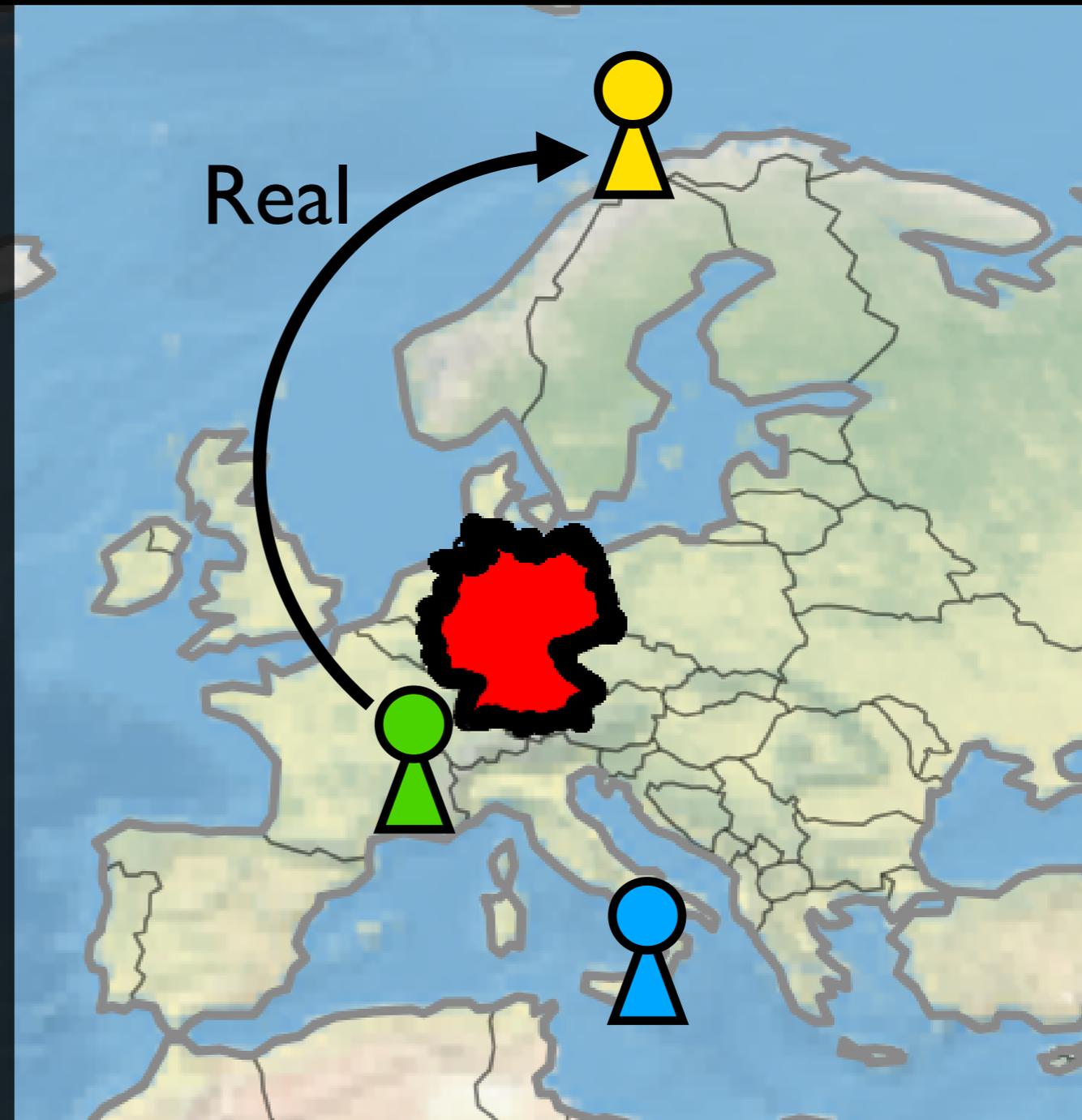
Exclude locations where  
alibis cannot exist



# Compute target regions

Where alibis *might* be

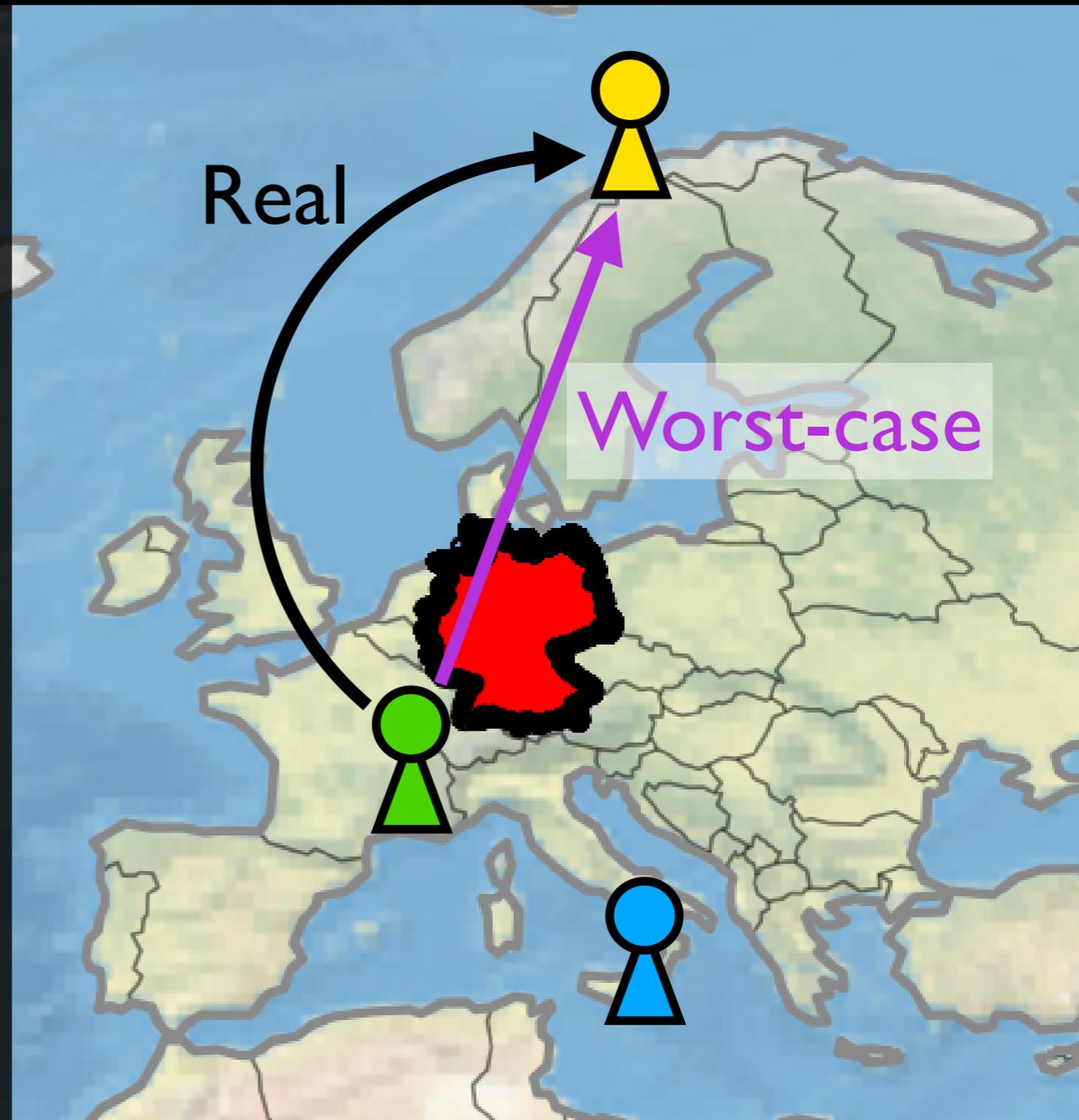
Exclude locations where  
alibis cannot exist



# Compute target regions

Where alibis *might* be

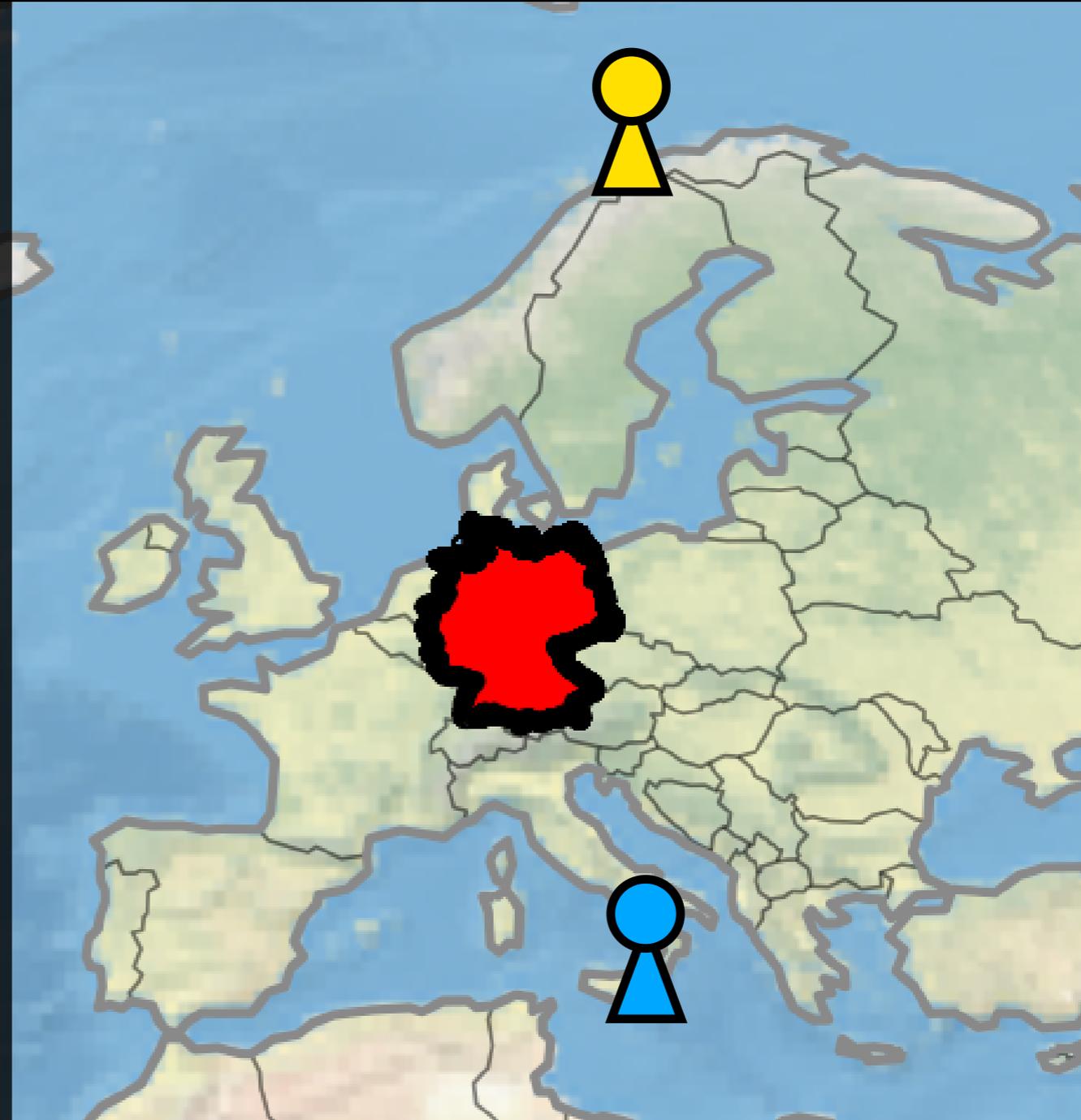
Exclude locations where  
alibis cannot exist



# Compute target regions

Where alibis *might* be

Exclude locations where  
alibis cannot exist

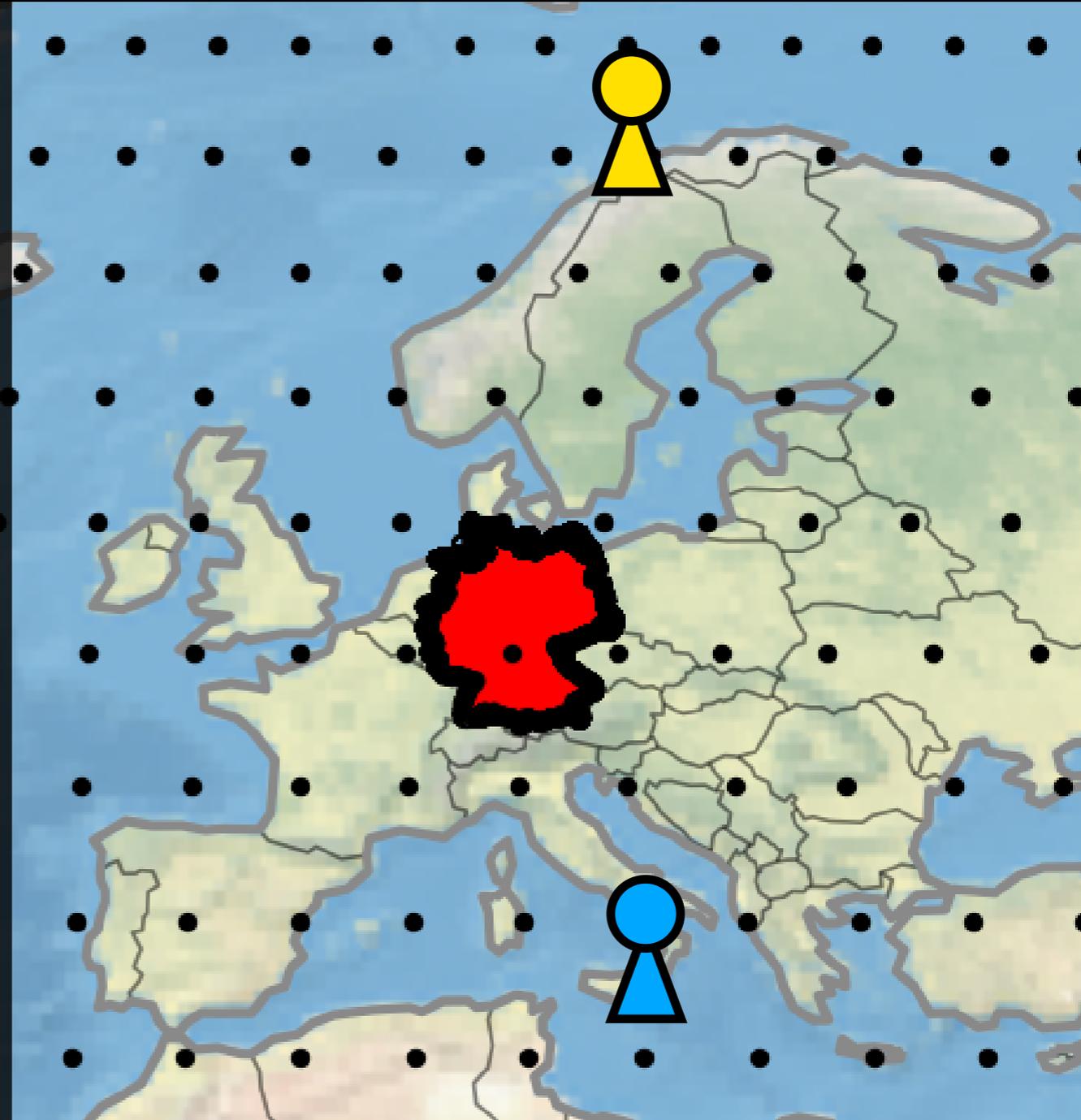


# Compute target regions

Where alibis *might* be

Exclude locations where  
alibis cannot exist

Segment the world  
into a grid



# Compute target regions

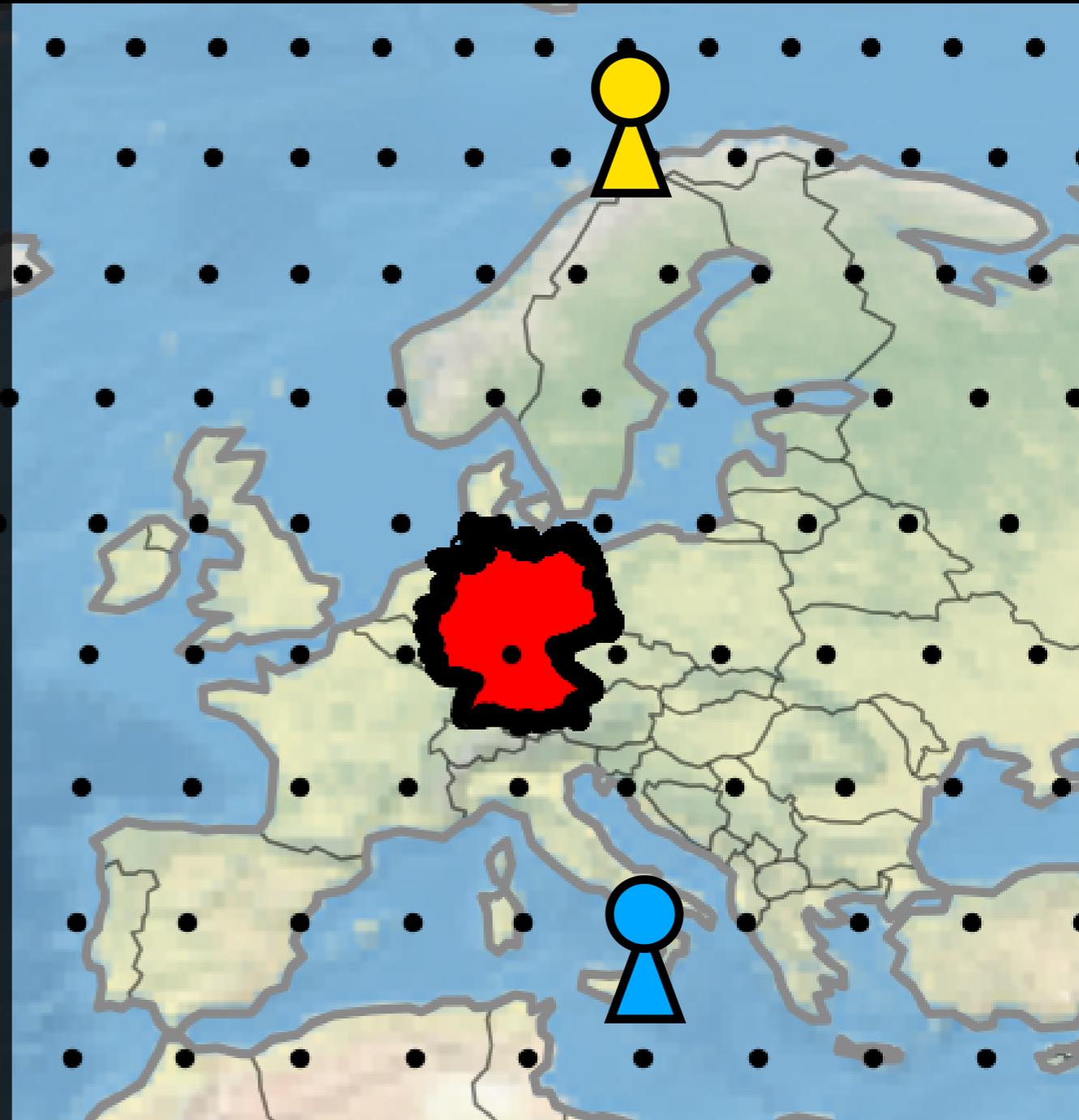
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Measured RTT} \leq 3 d / c$$



# Compute target regions

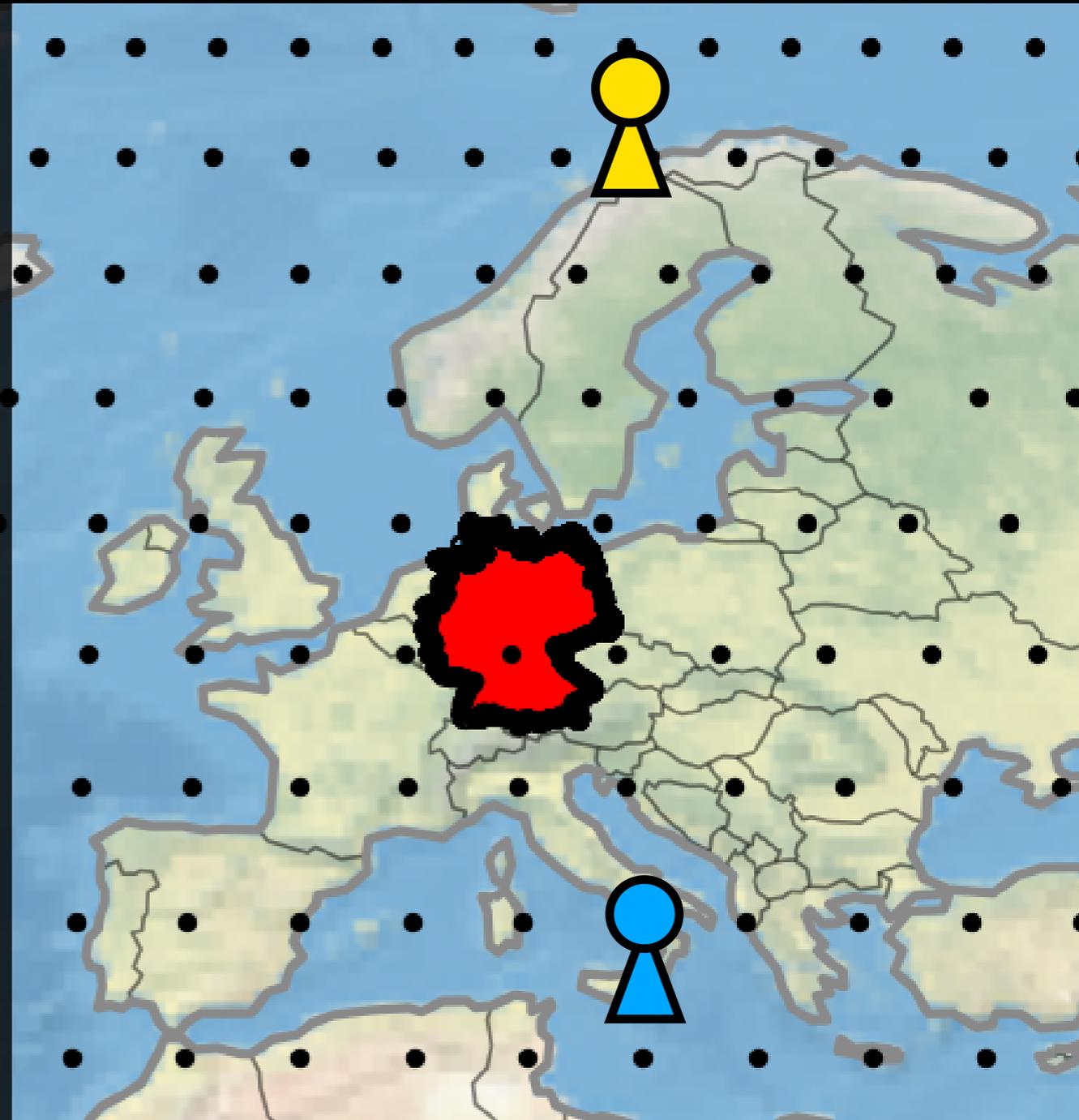
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Measured RTT} \leq 3 d / c$$



# Compute target regions

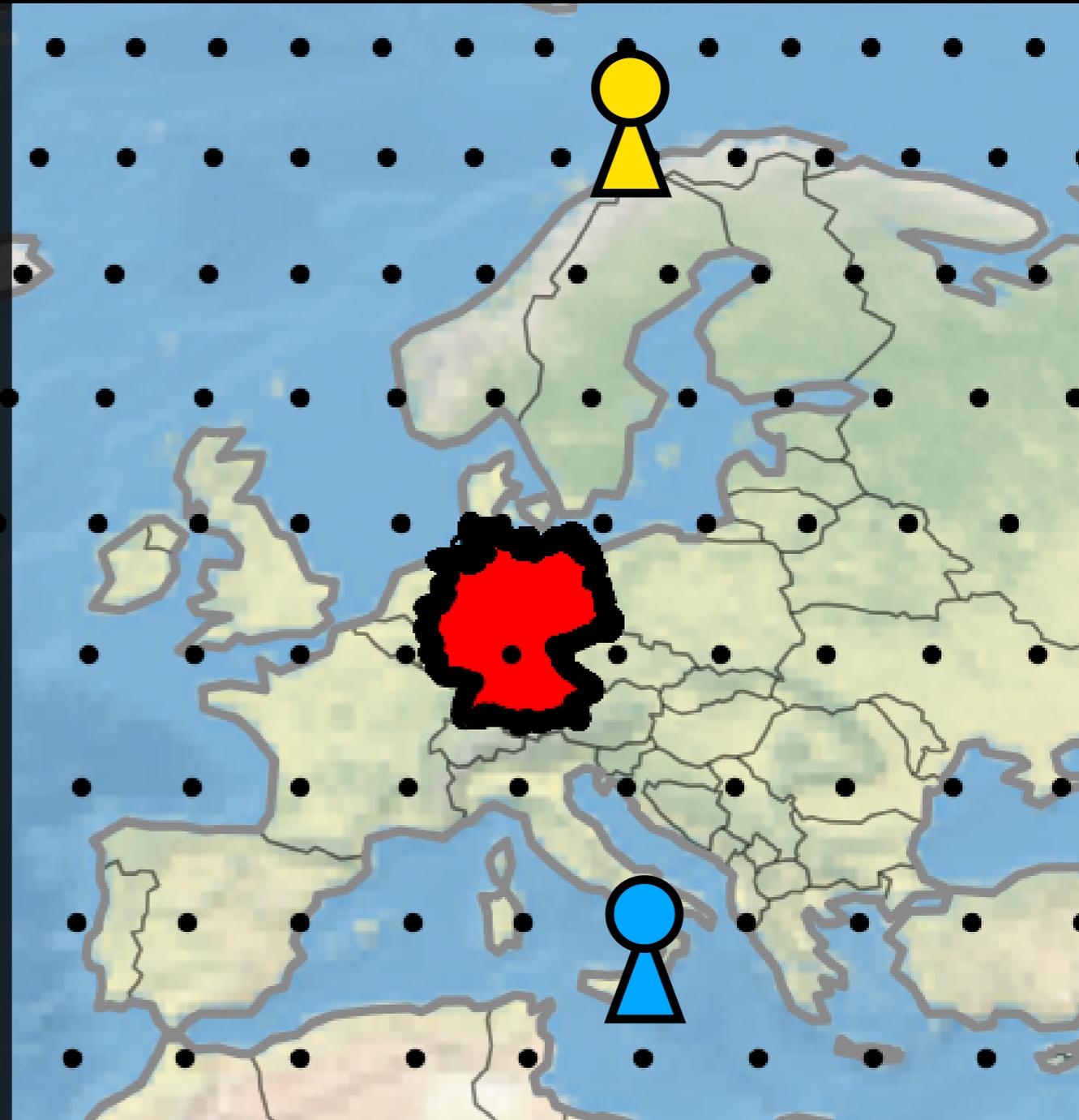
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

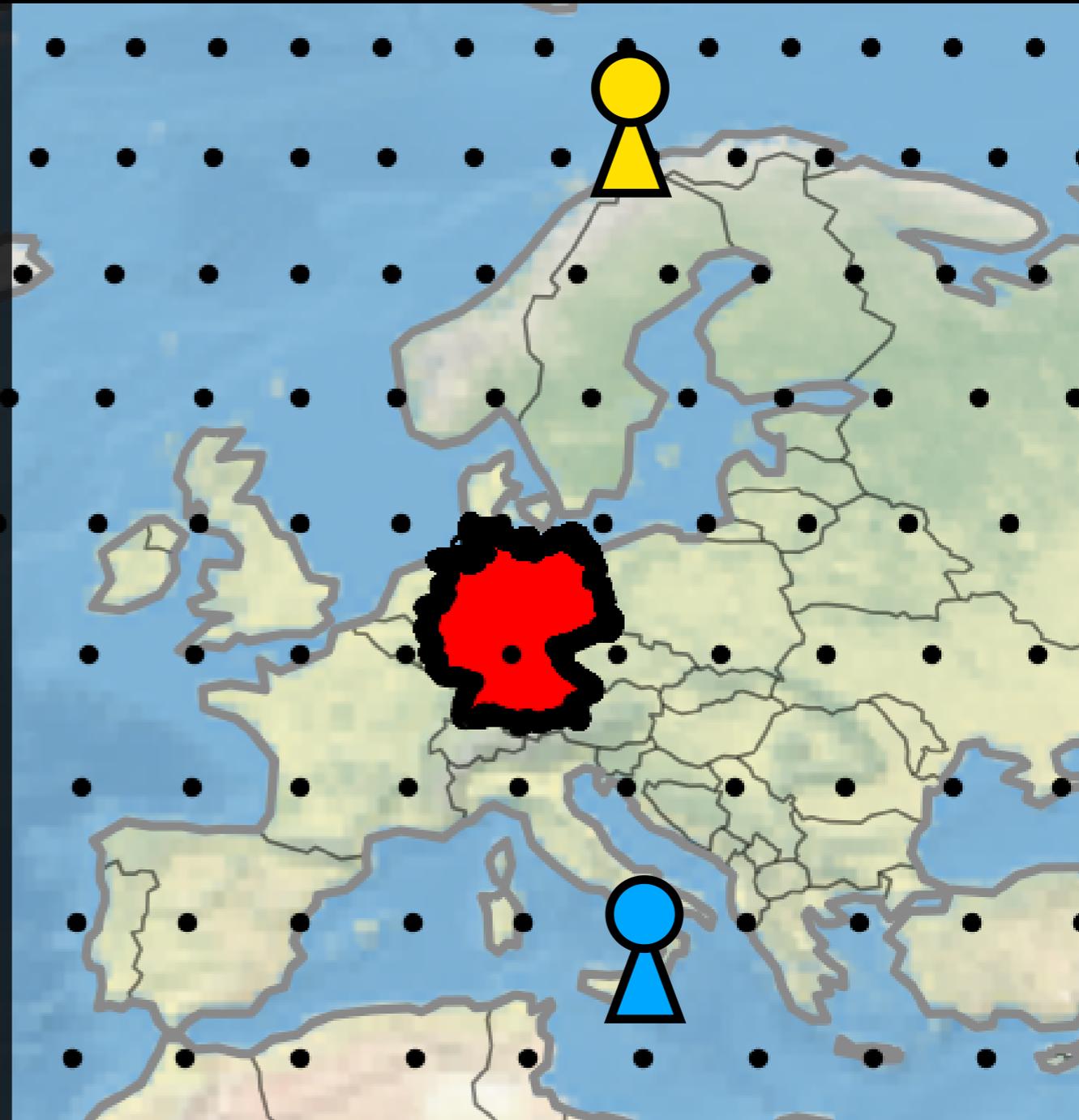
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

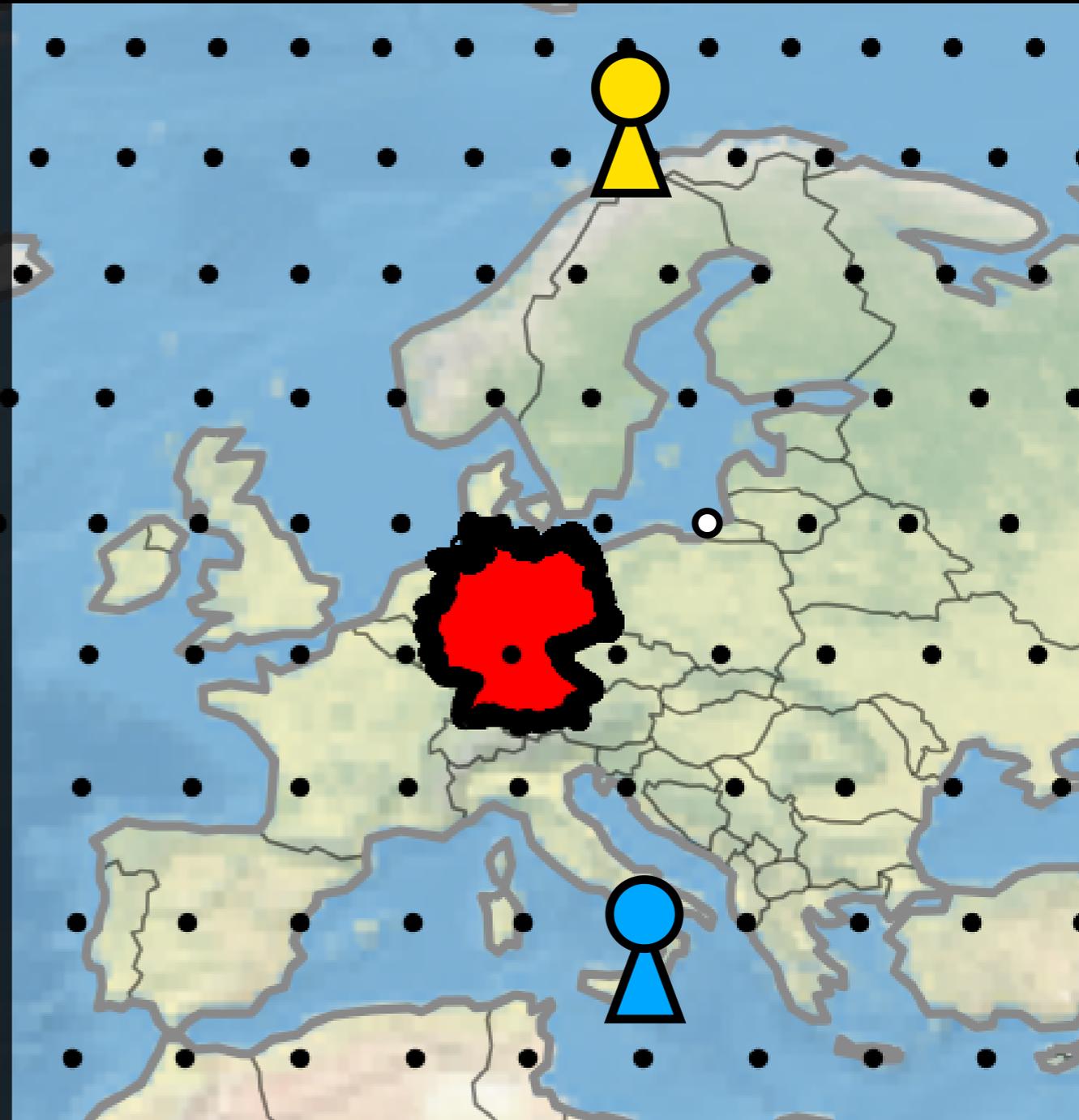
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

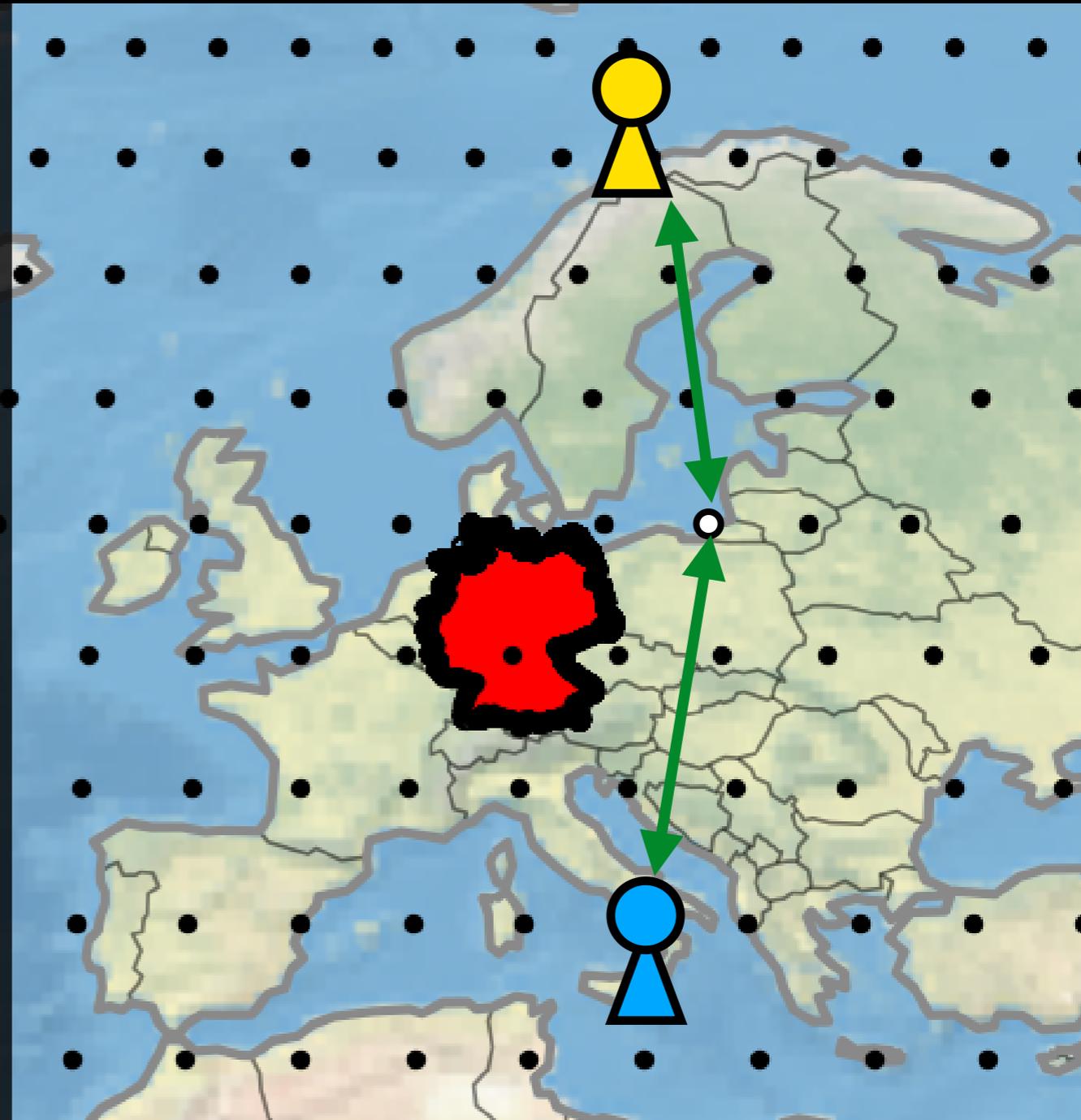
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

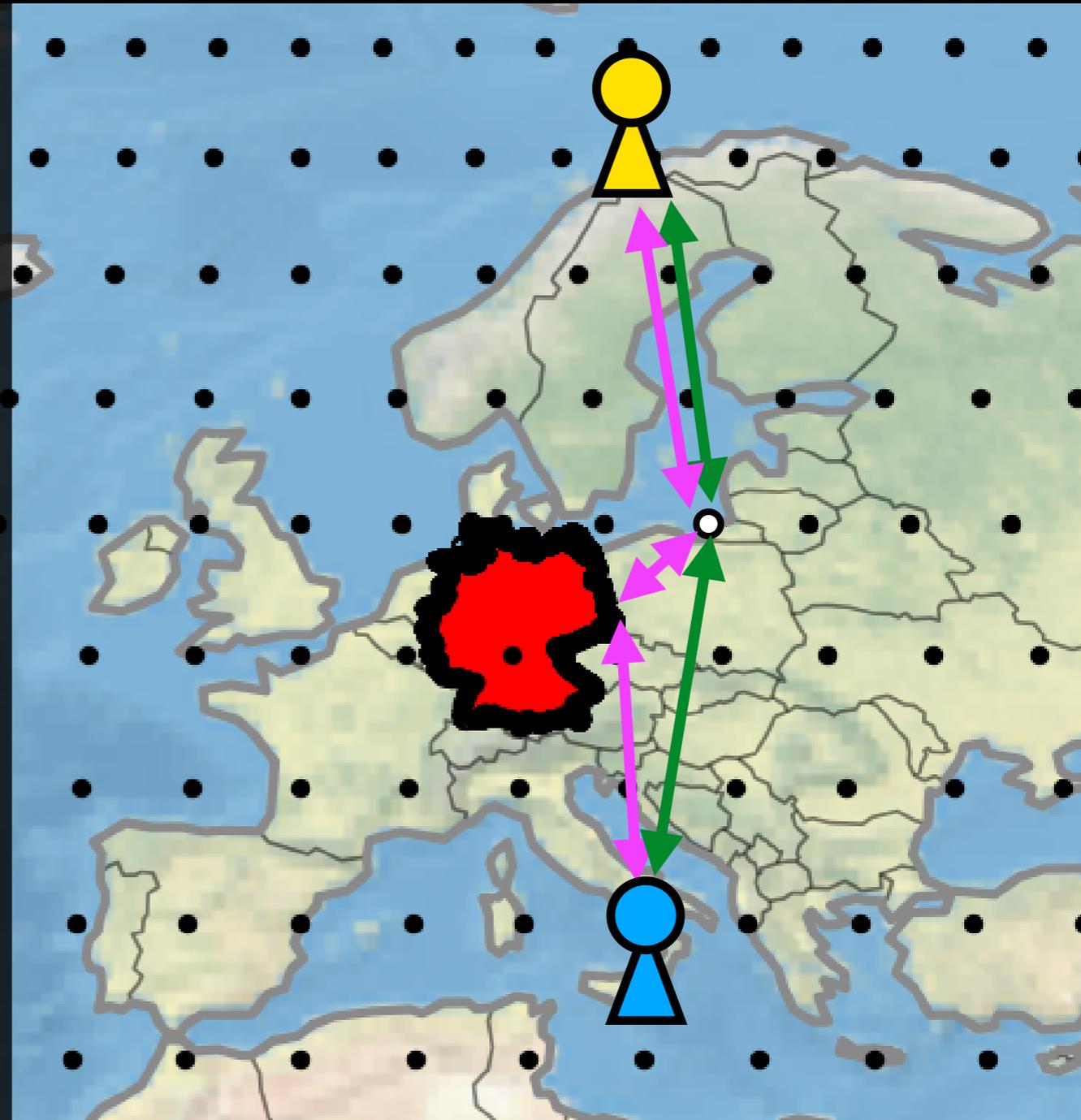
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

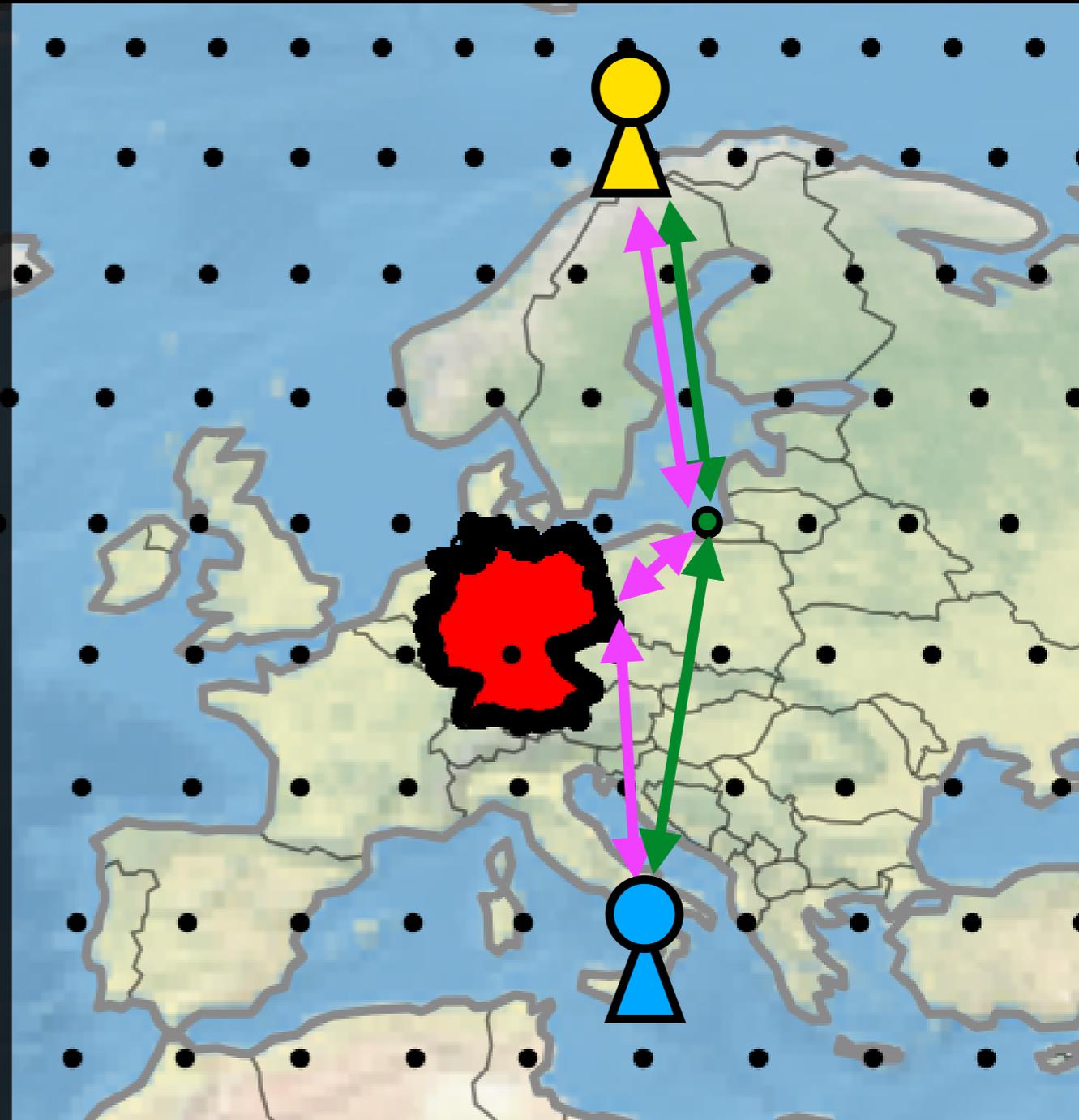
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

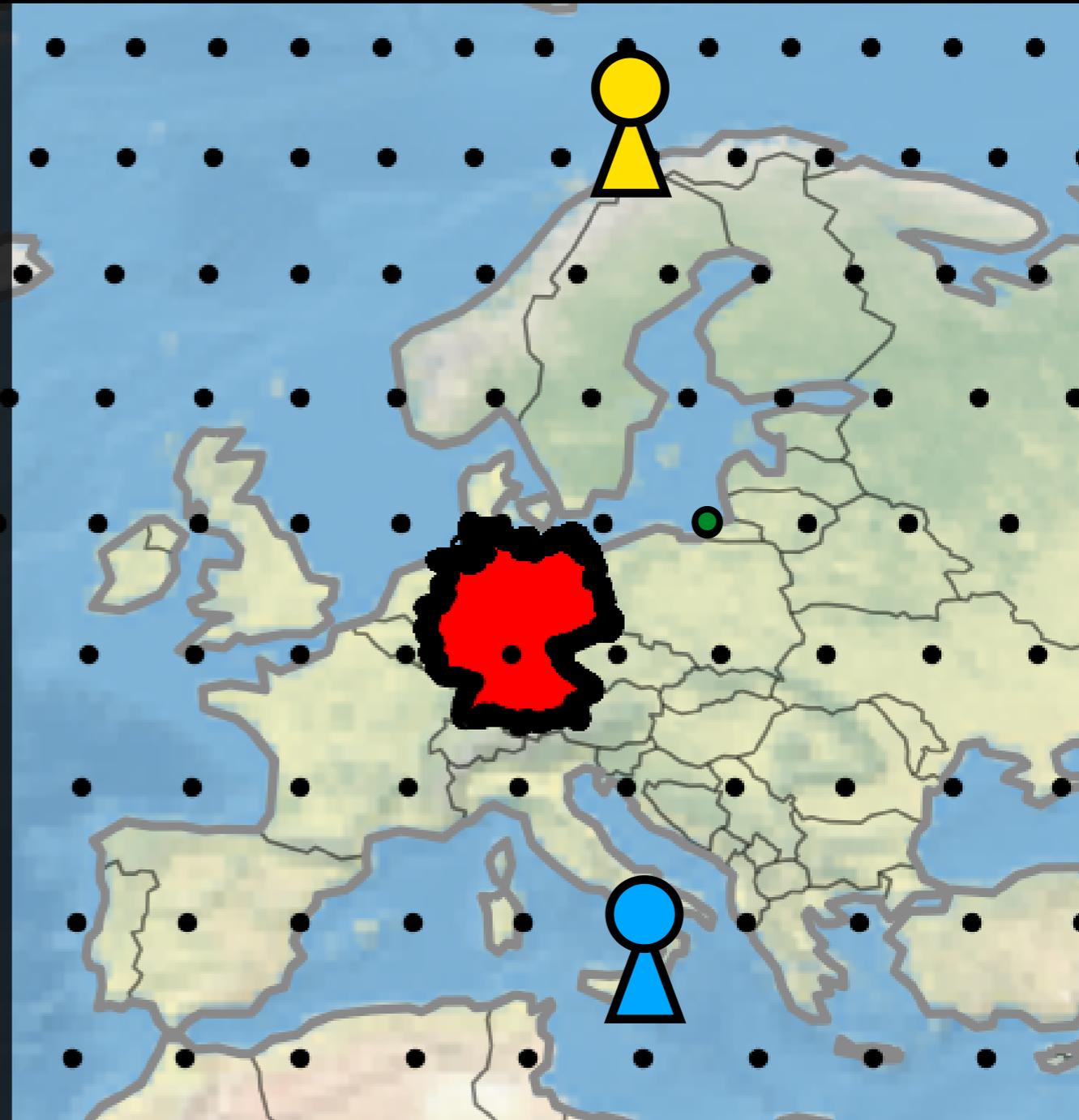
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

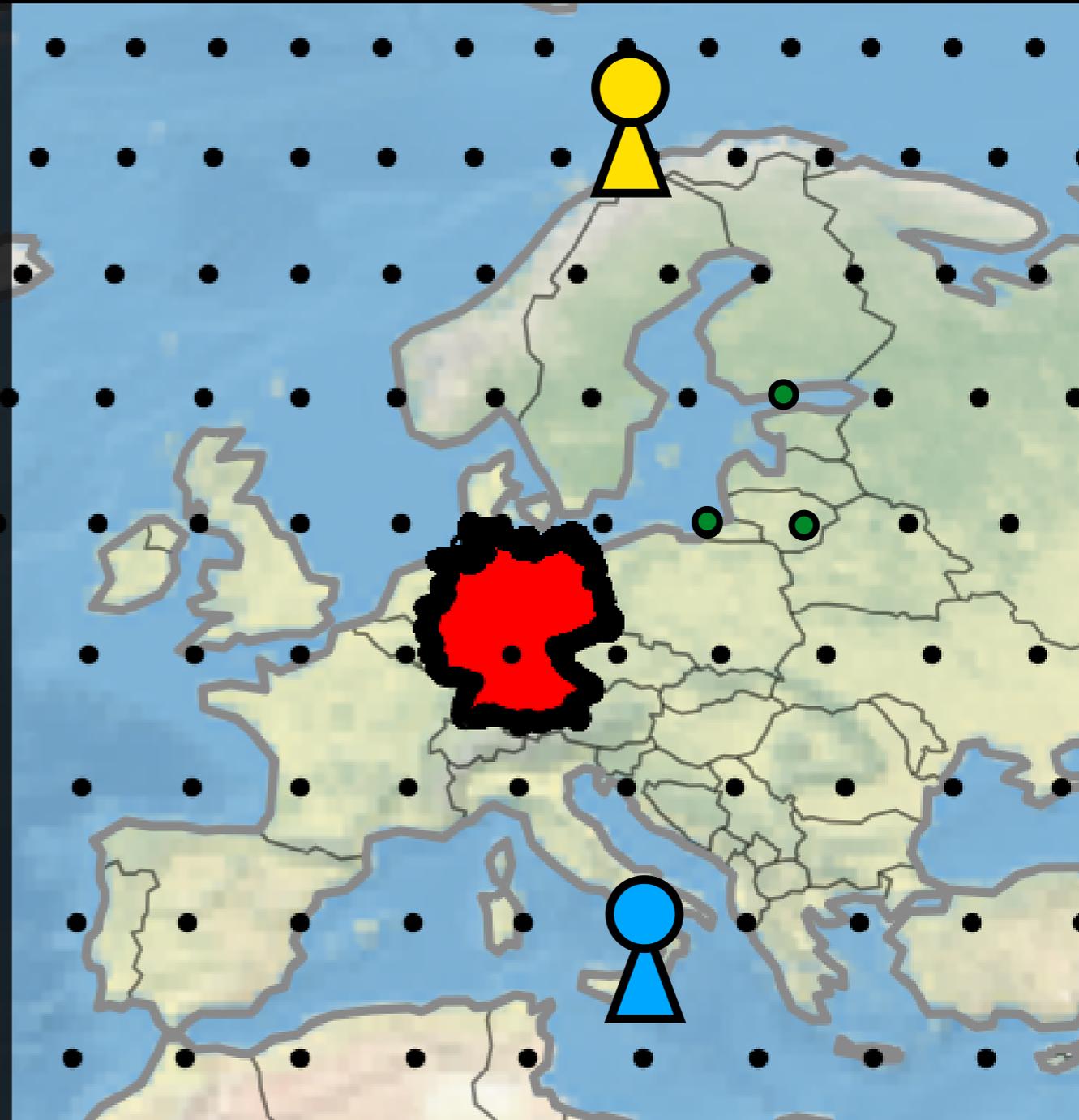
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

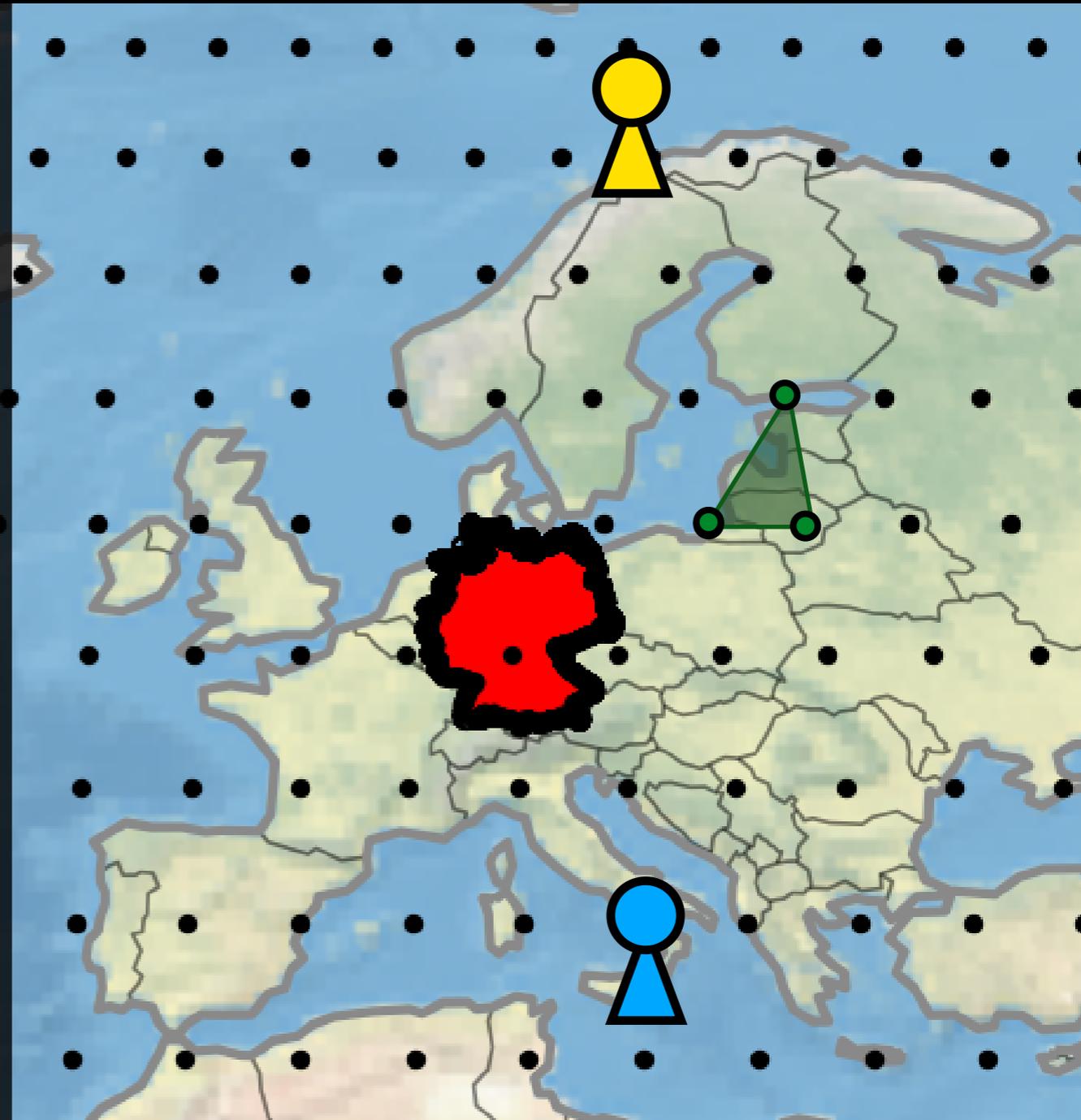
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

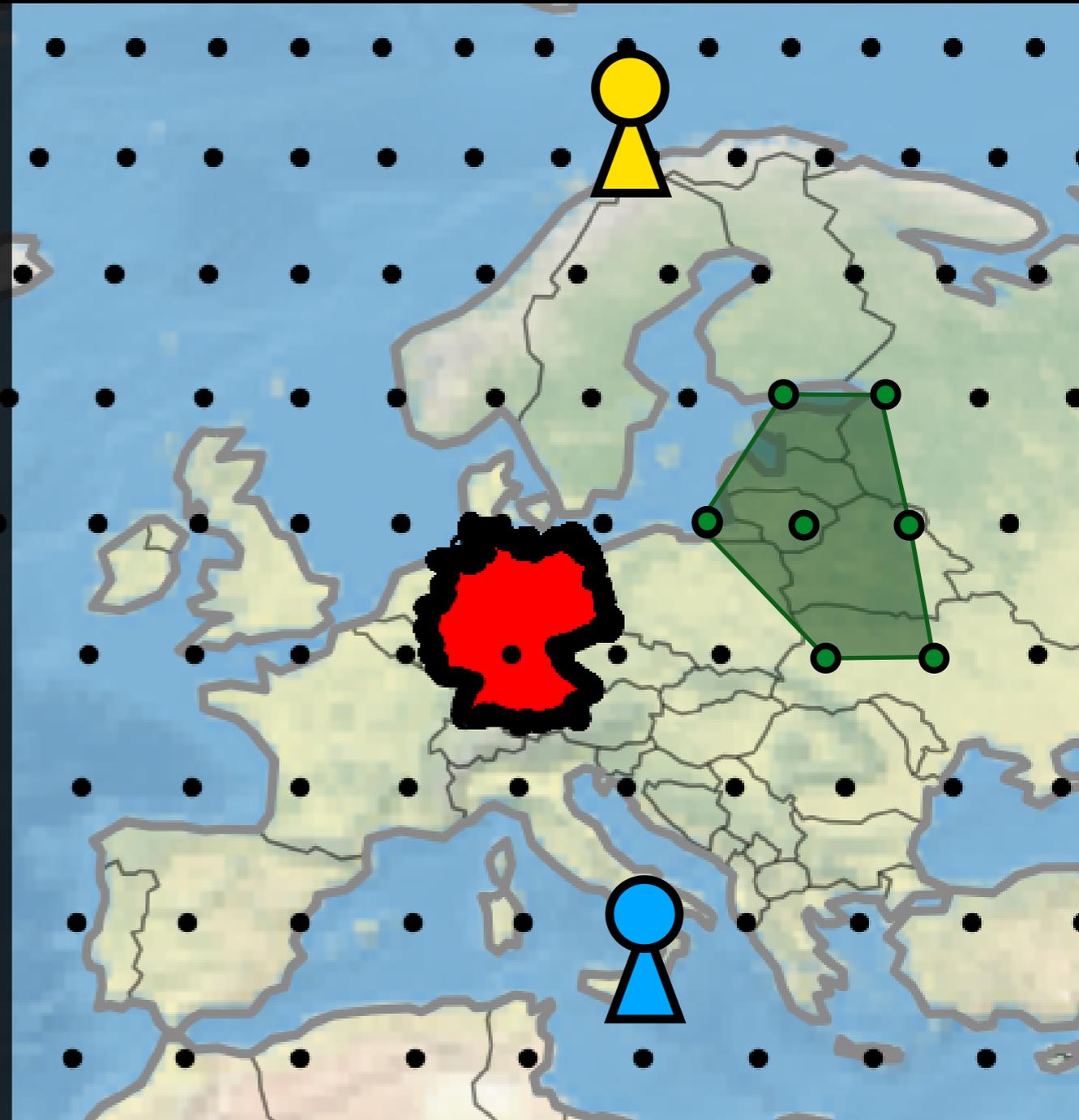
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



# Compute target regions

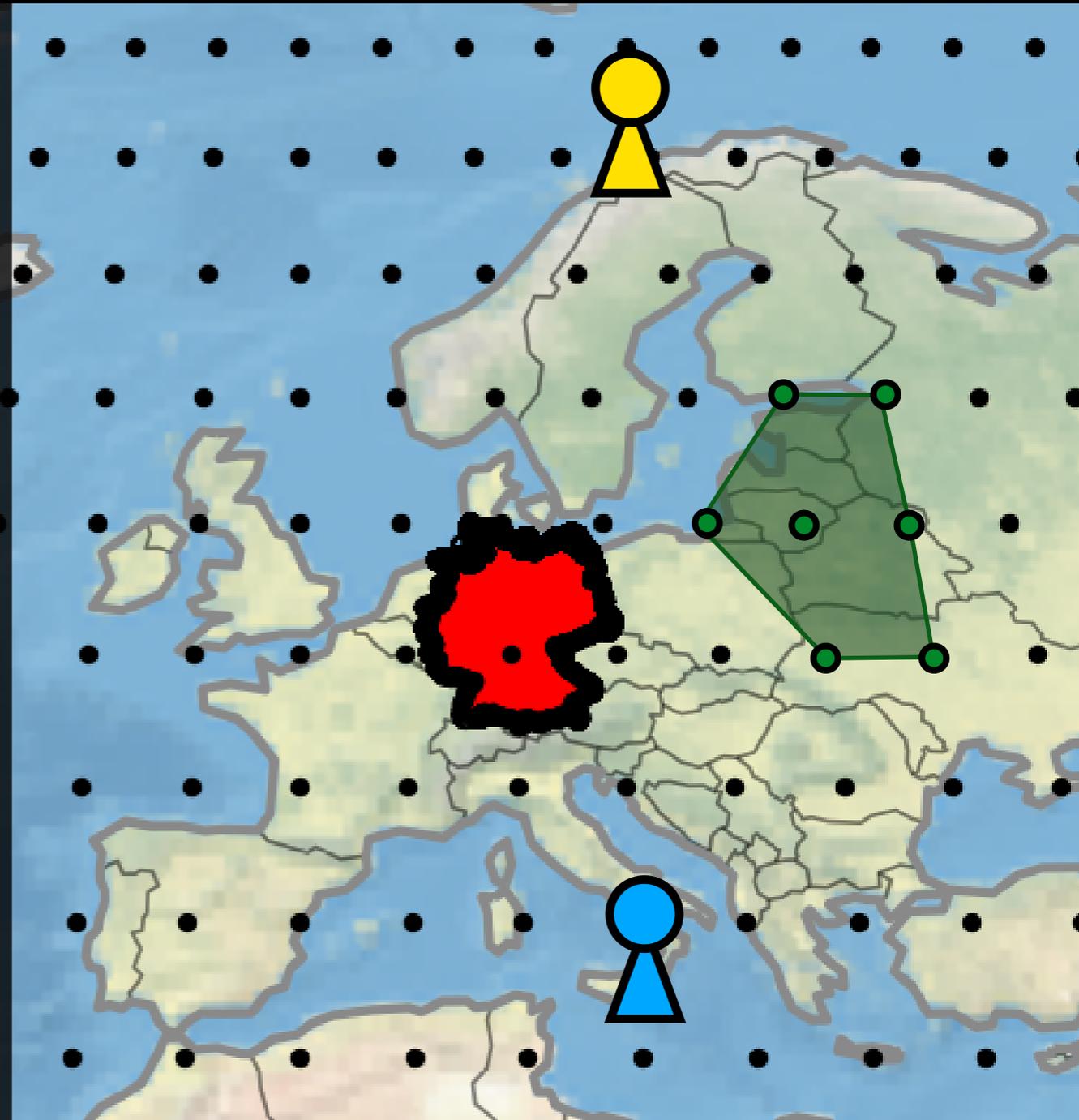
Where alibis *might* be

Exclude locations where alibis cannot exist

Segment the world into a grid

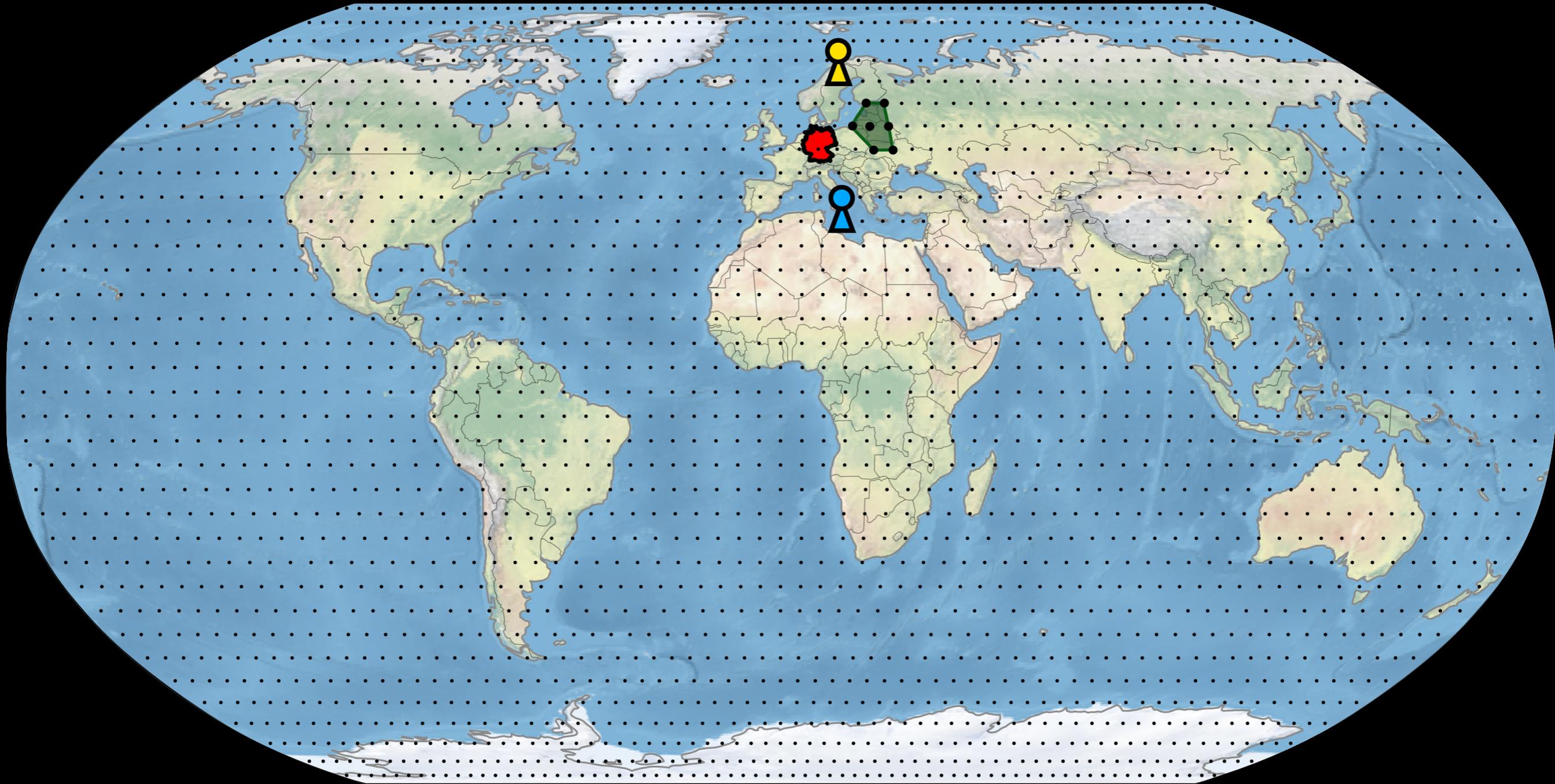
Include a grid point if:

$$(1 + \delta) * \text{Min possible RTT} \leq 3 d / c$$



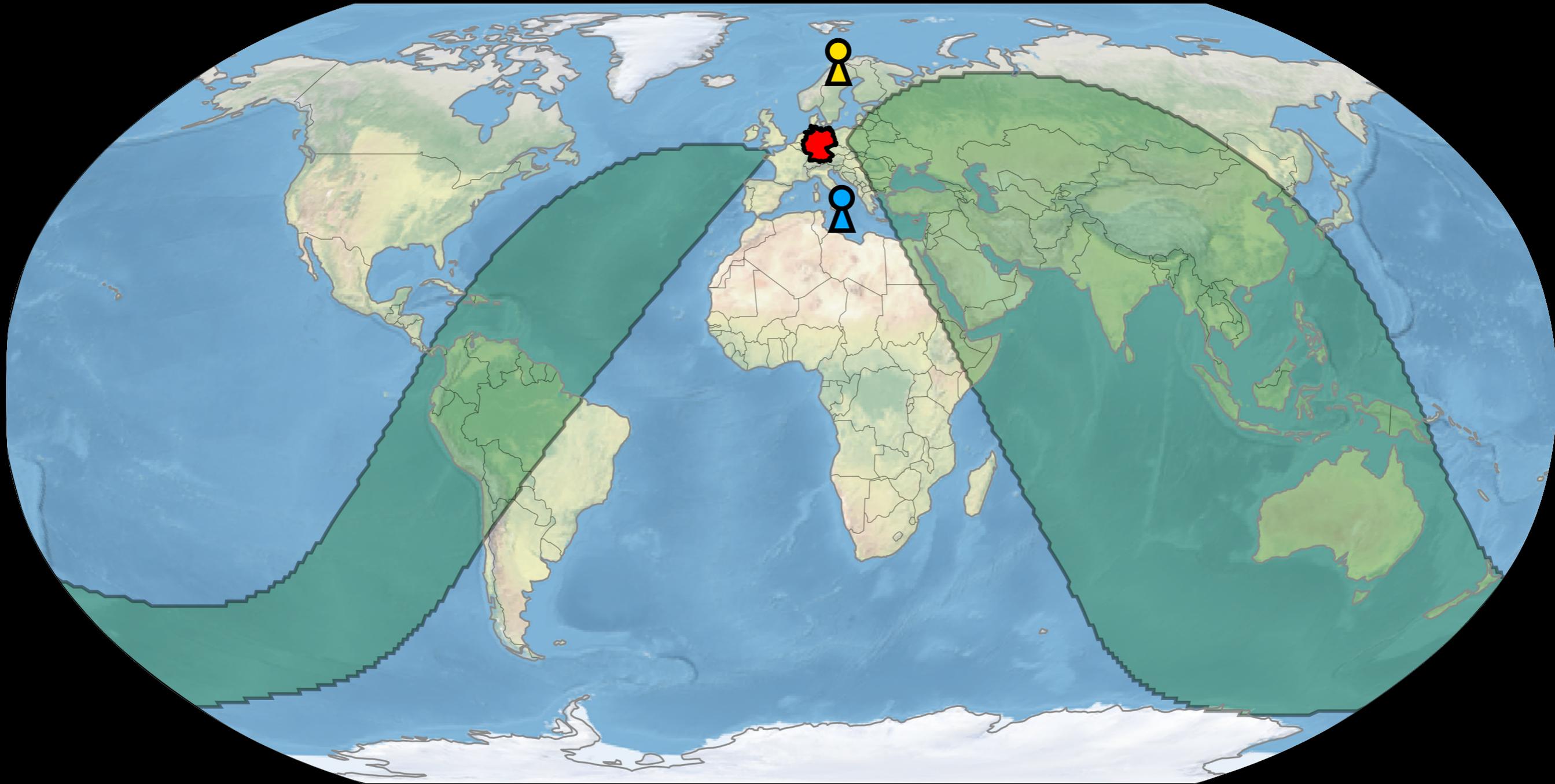
# Compute target regions

Where alibis *might* be



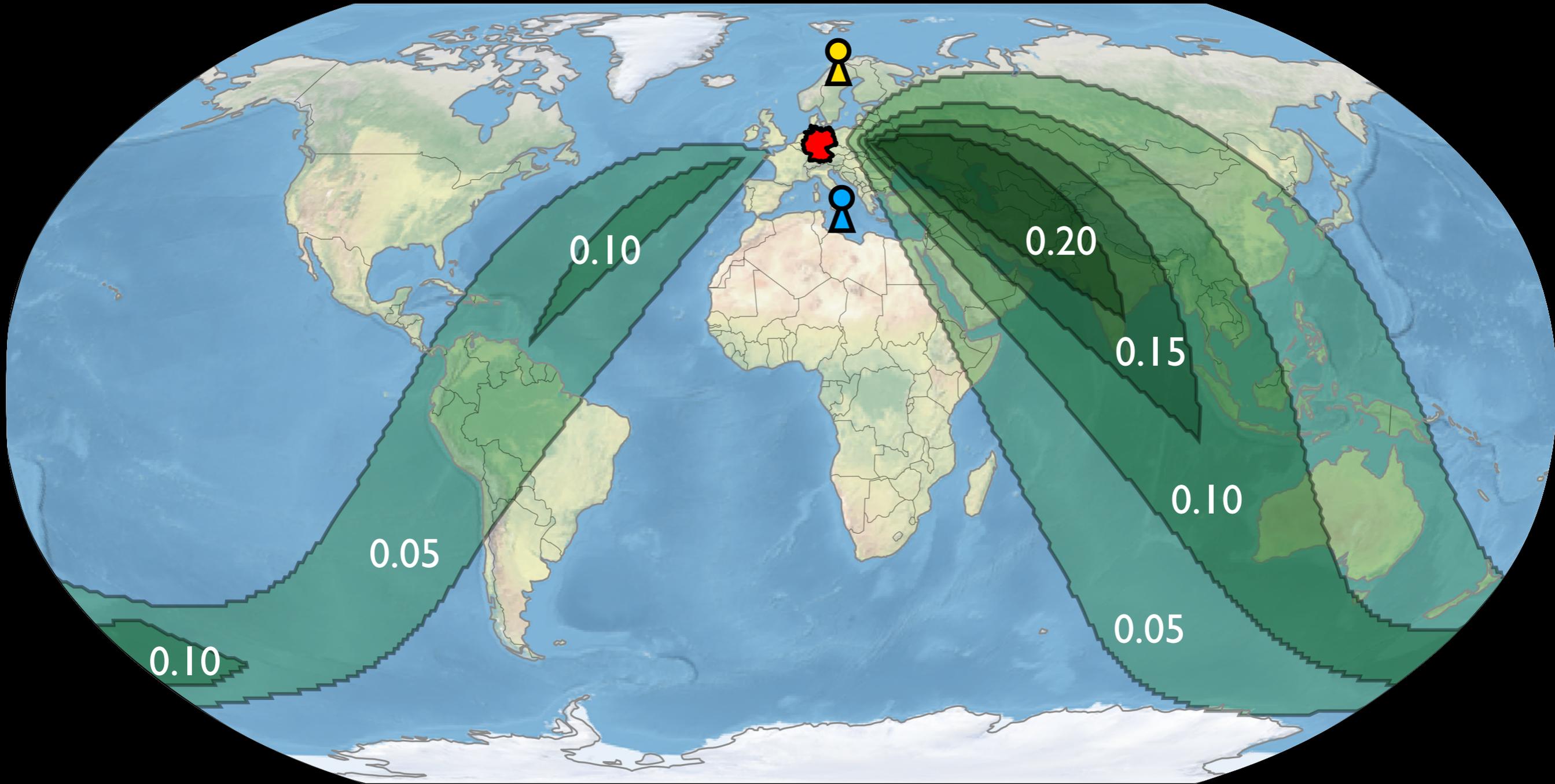
# Compute target regions

Where alibis *might* be



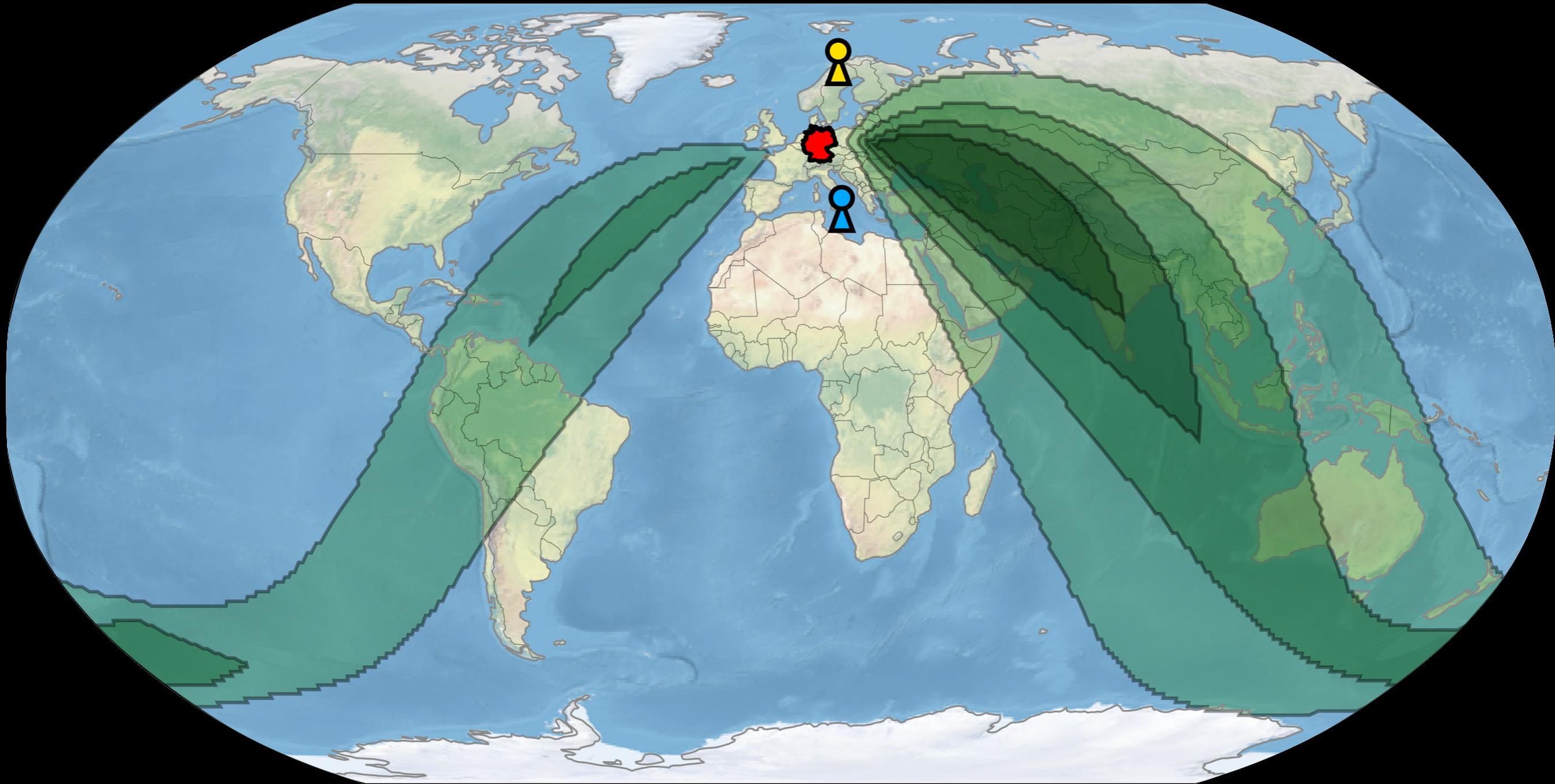
# Compute target regions

Where alibis *might* be



# Compute target regions

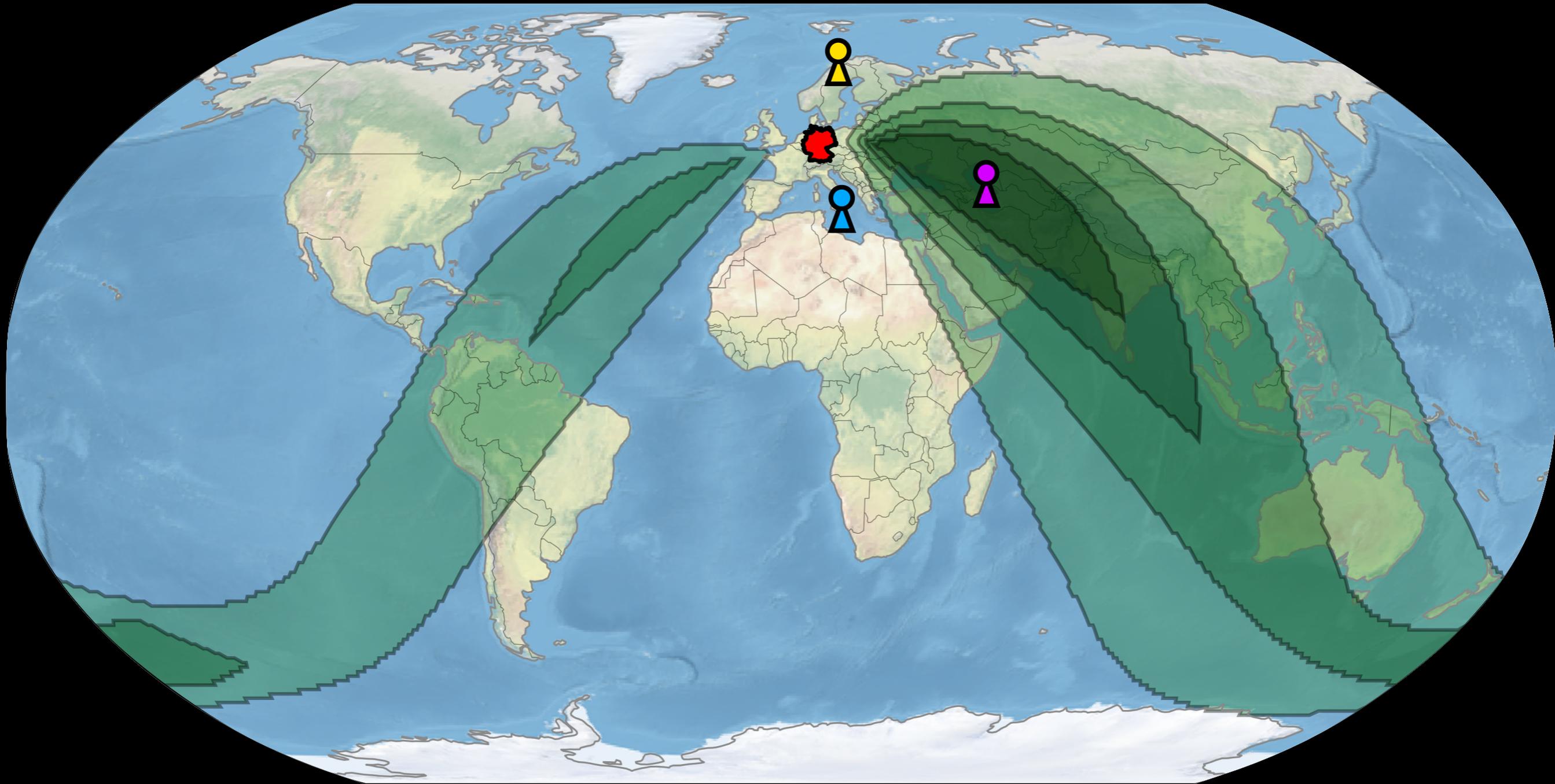
Where alibis *might* be



Being in a target region is a **necessary but not sufficient** condition of an alibi

# Compute target regions

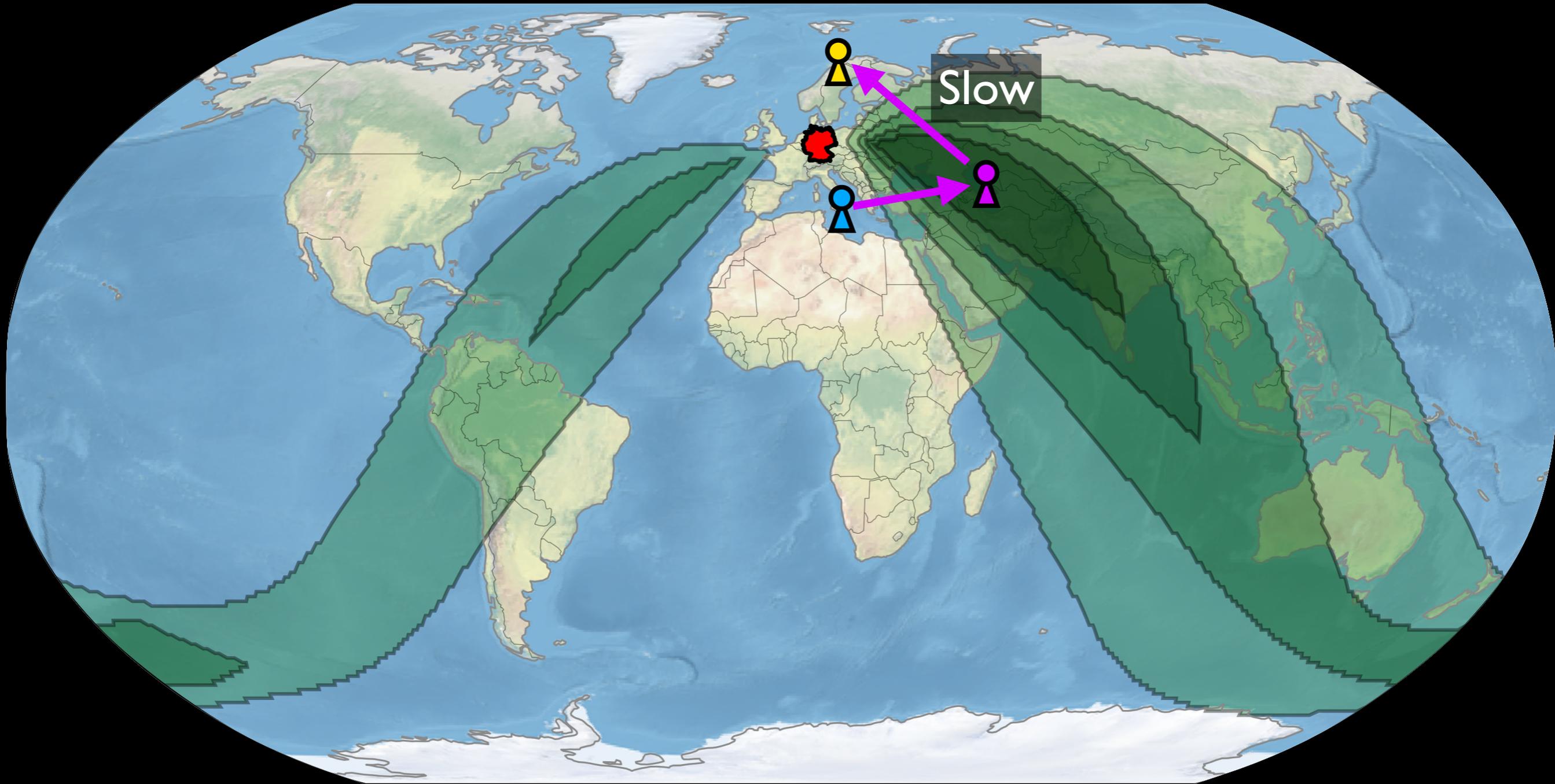
Where alibis *might* be



Being in a target region is a **necessary but not sufficient** condition of an alibi

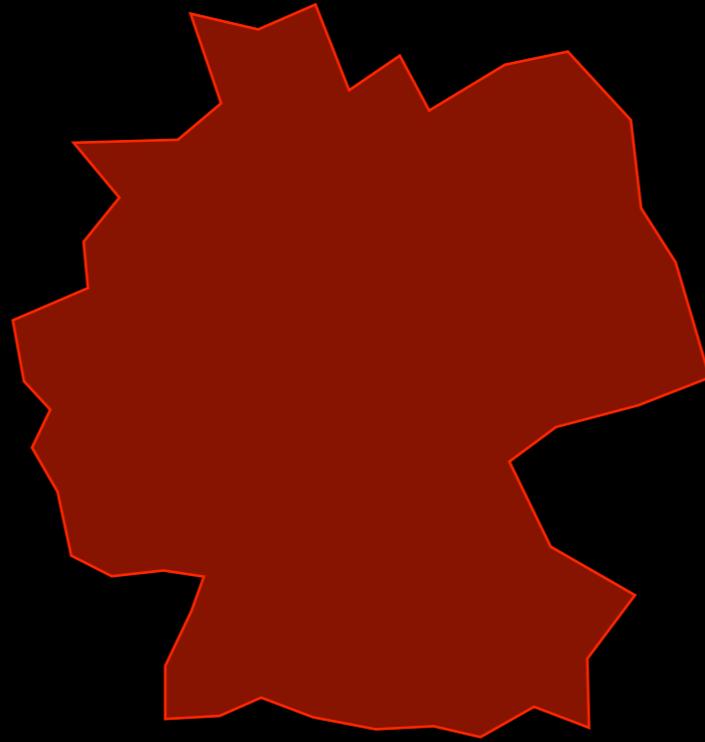
# Compute target regions

Where alibis *might* be



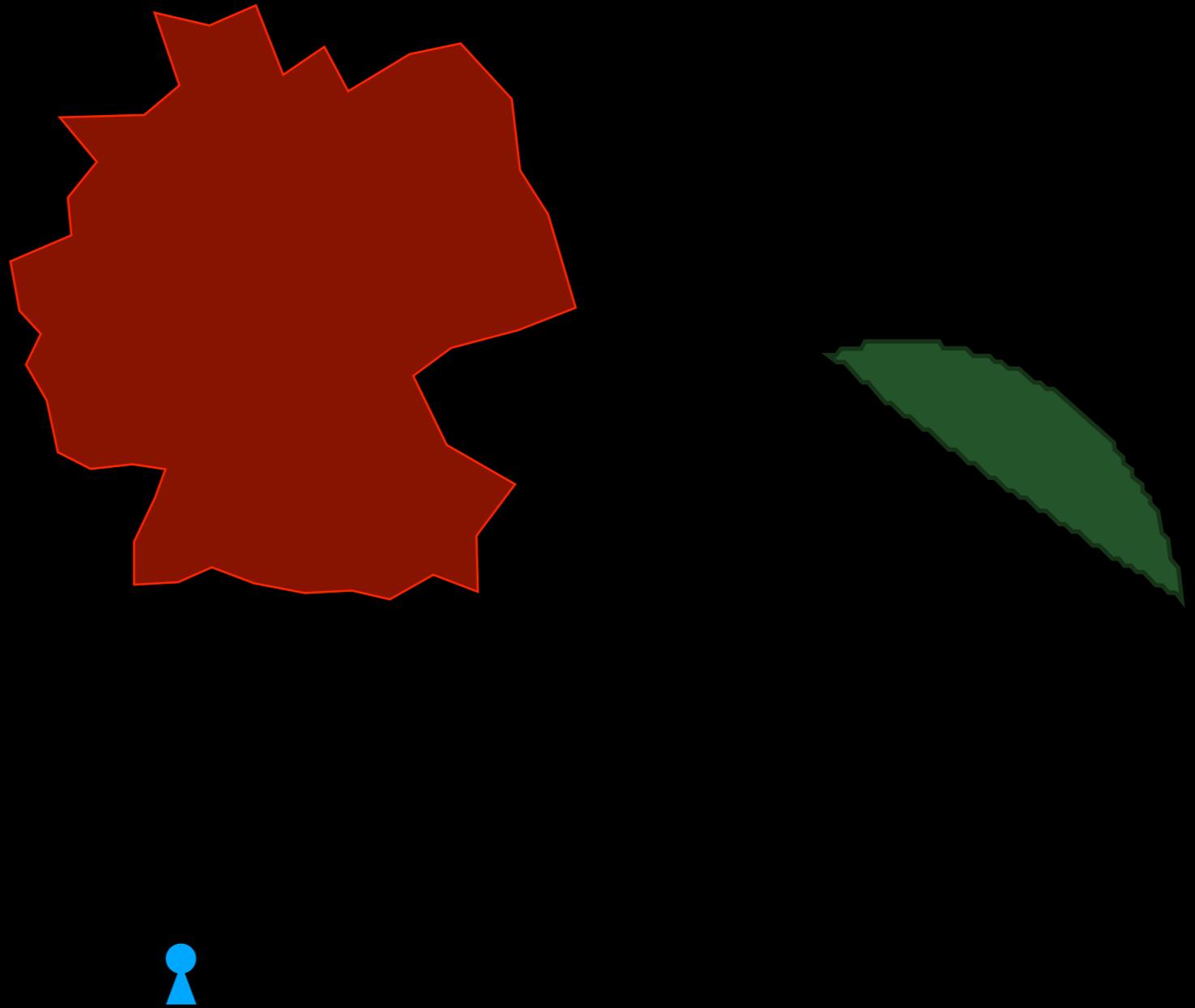
Being in a target region is a **necessary but not sufficient** condition of an alibi





# Peer-to-peer:

Every participant has a set of “neighbor” peers



# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



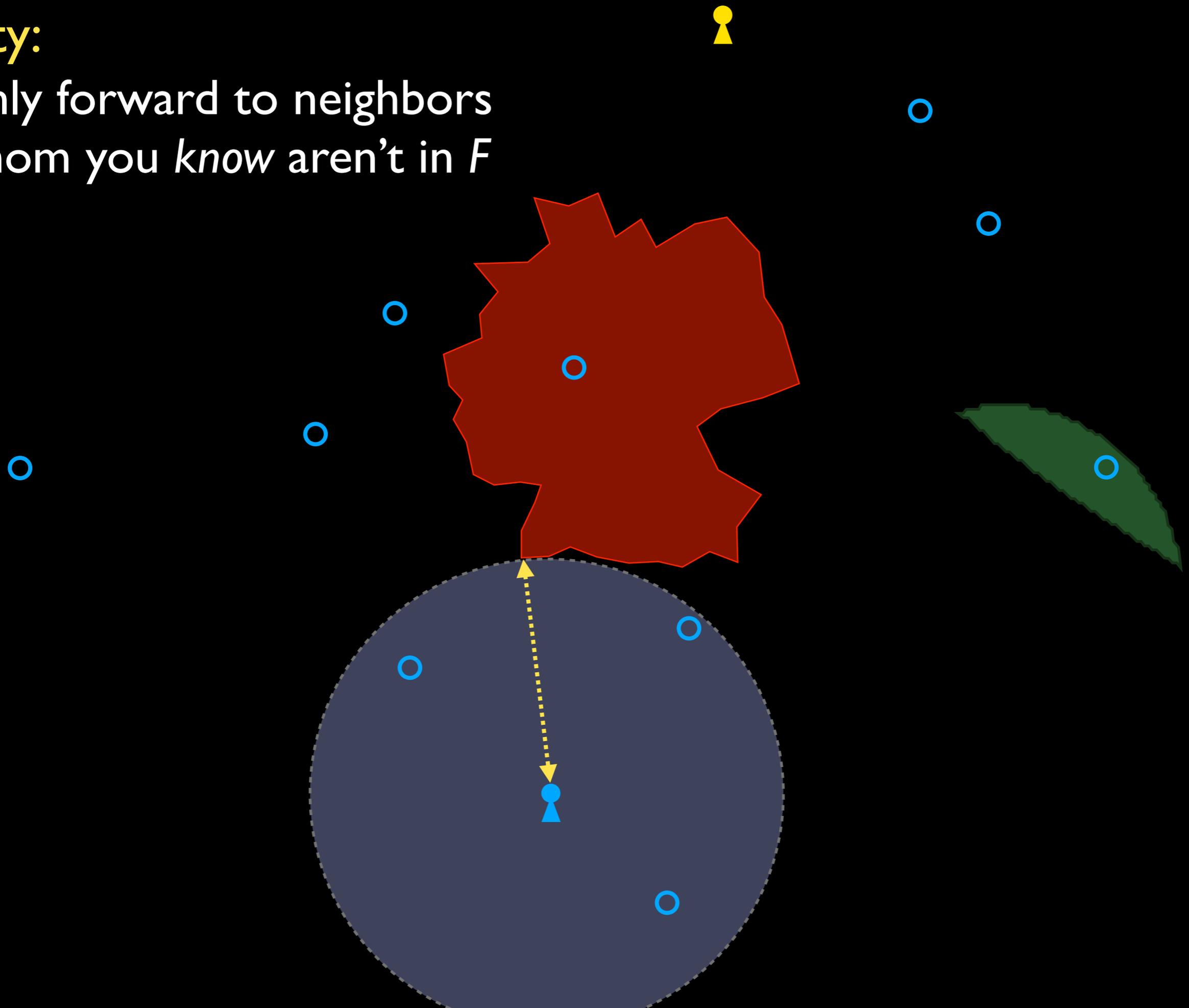
# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



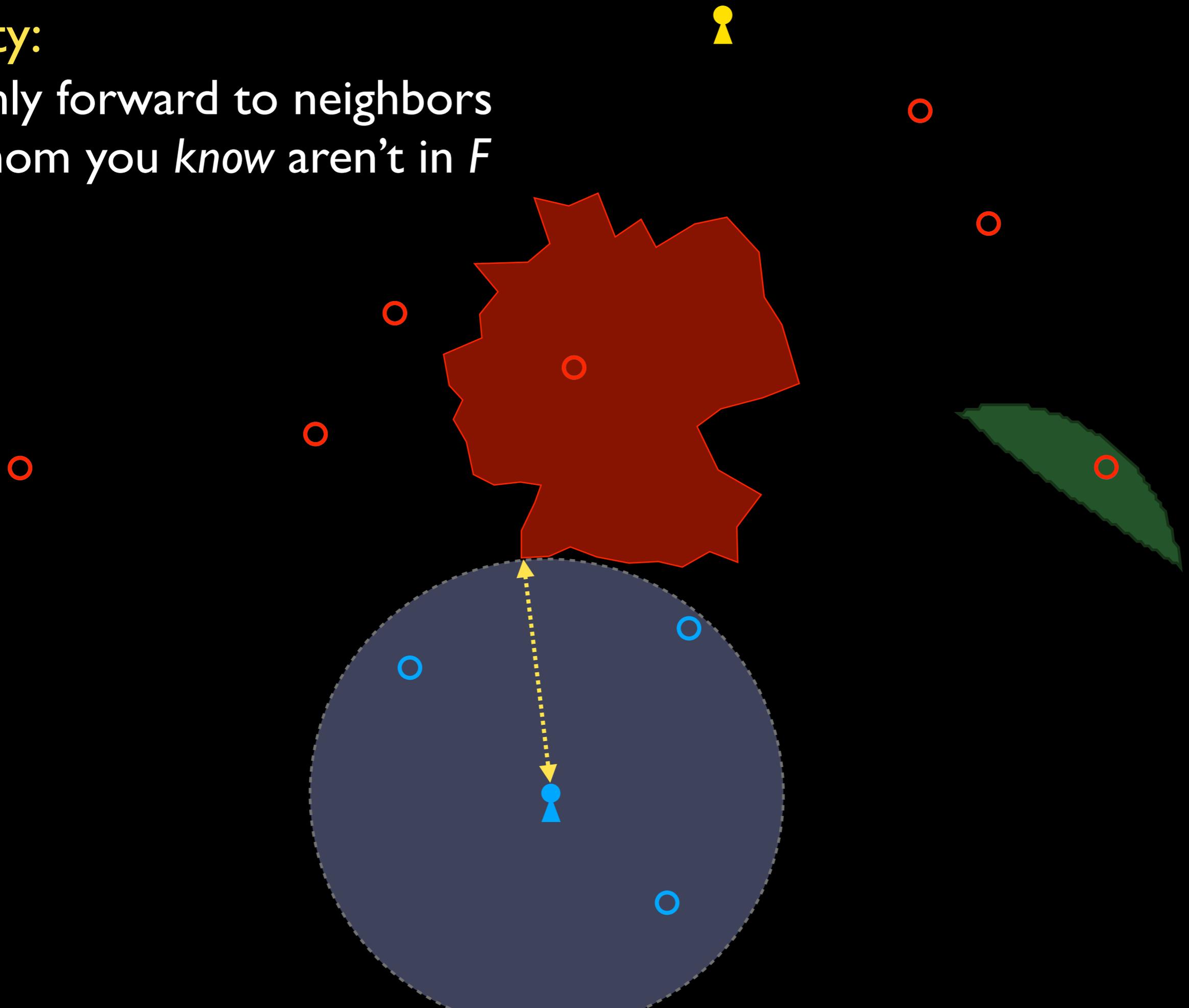
# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



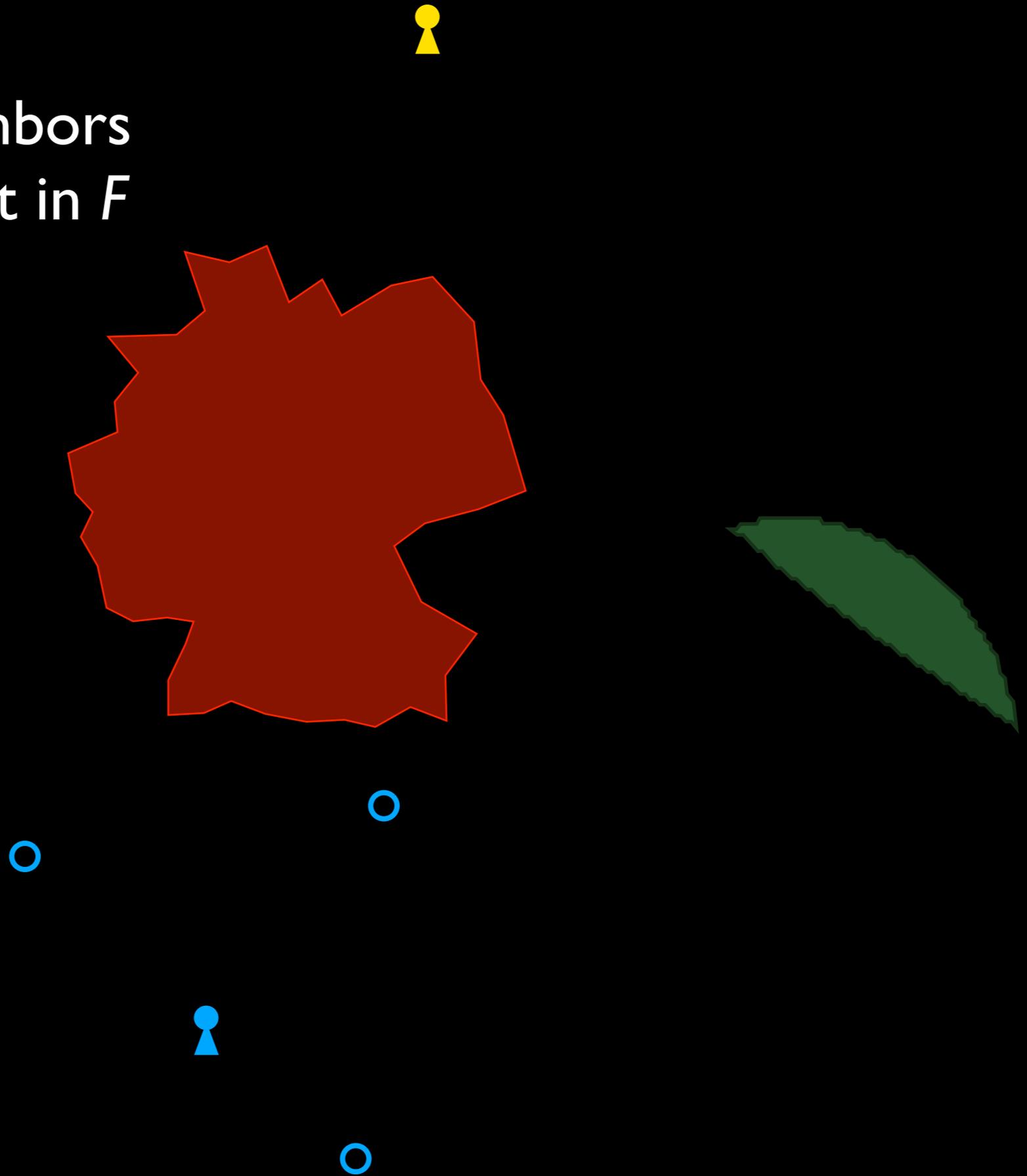
# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



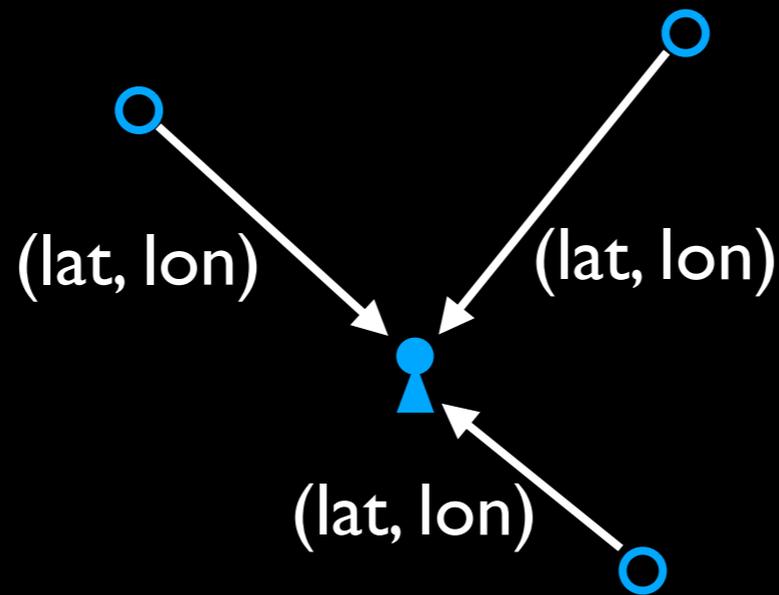
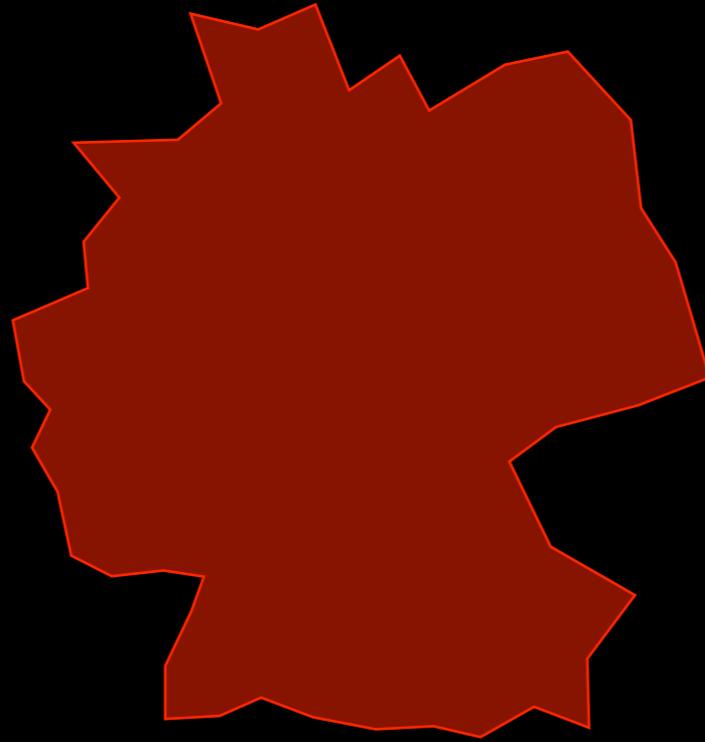
# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



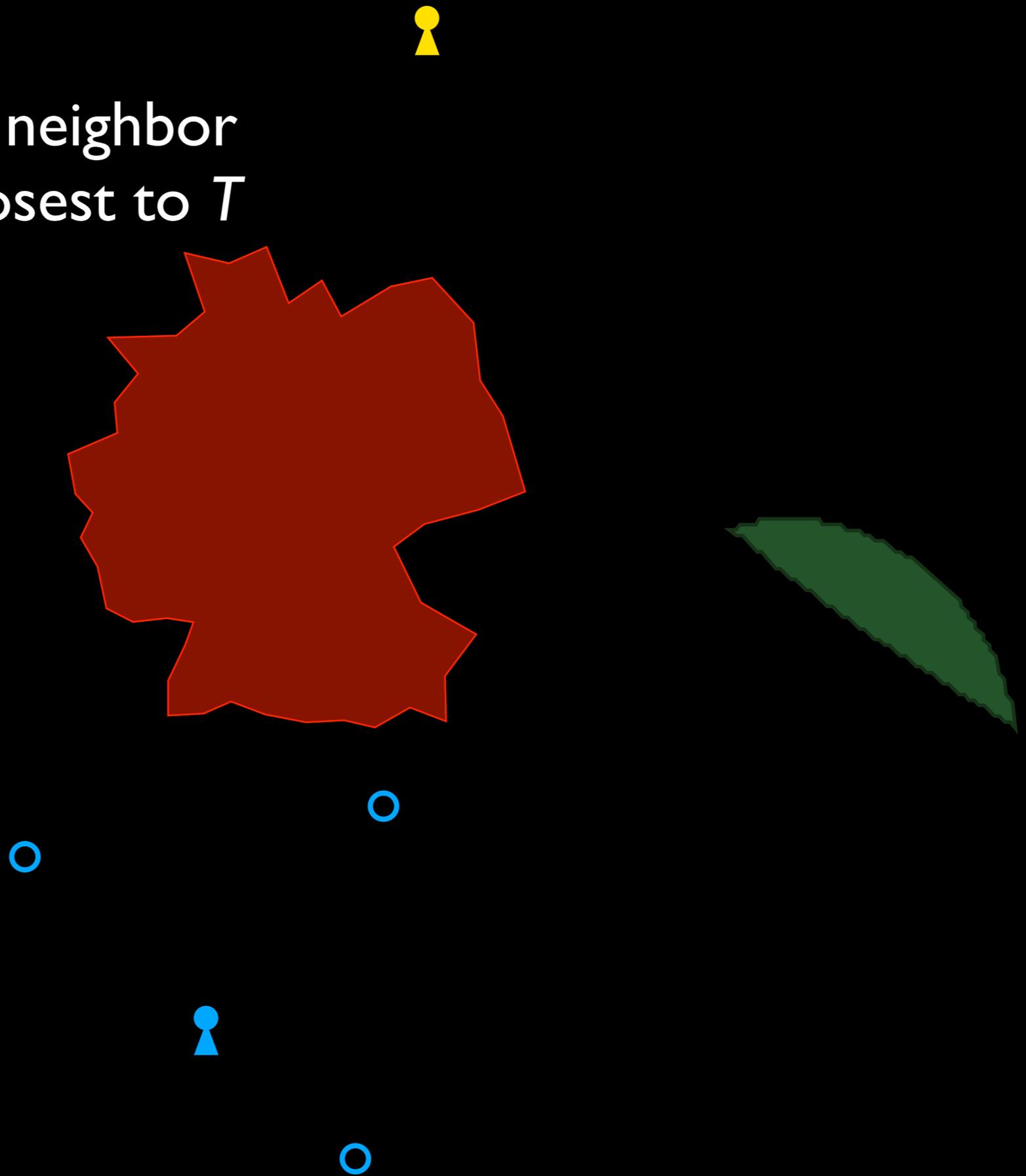
# Safety:

Only forward to neighbors  
whom you *know* aren't in  $F$



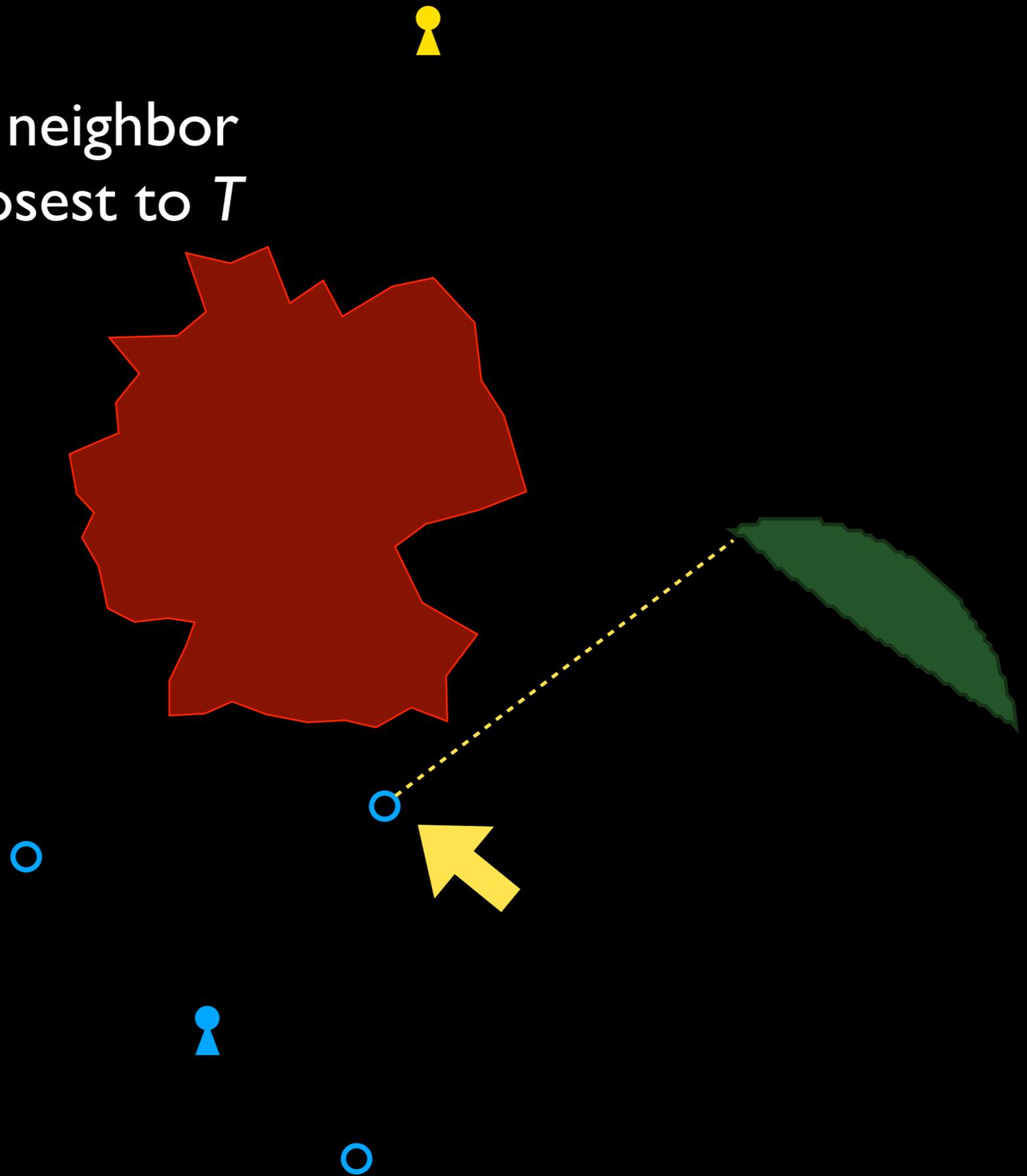
# Progress:

Forward to the (safe) neighbor  
whose safe zone is closest to  $T$



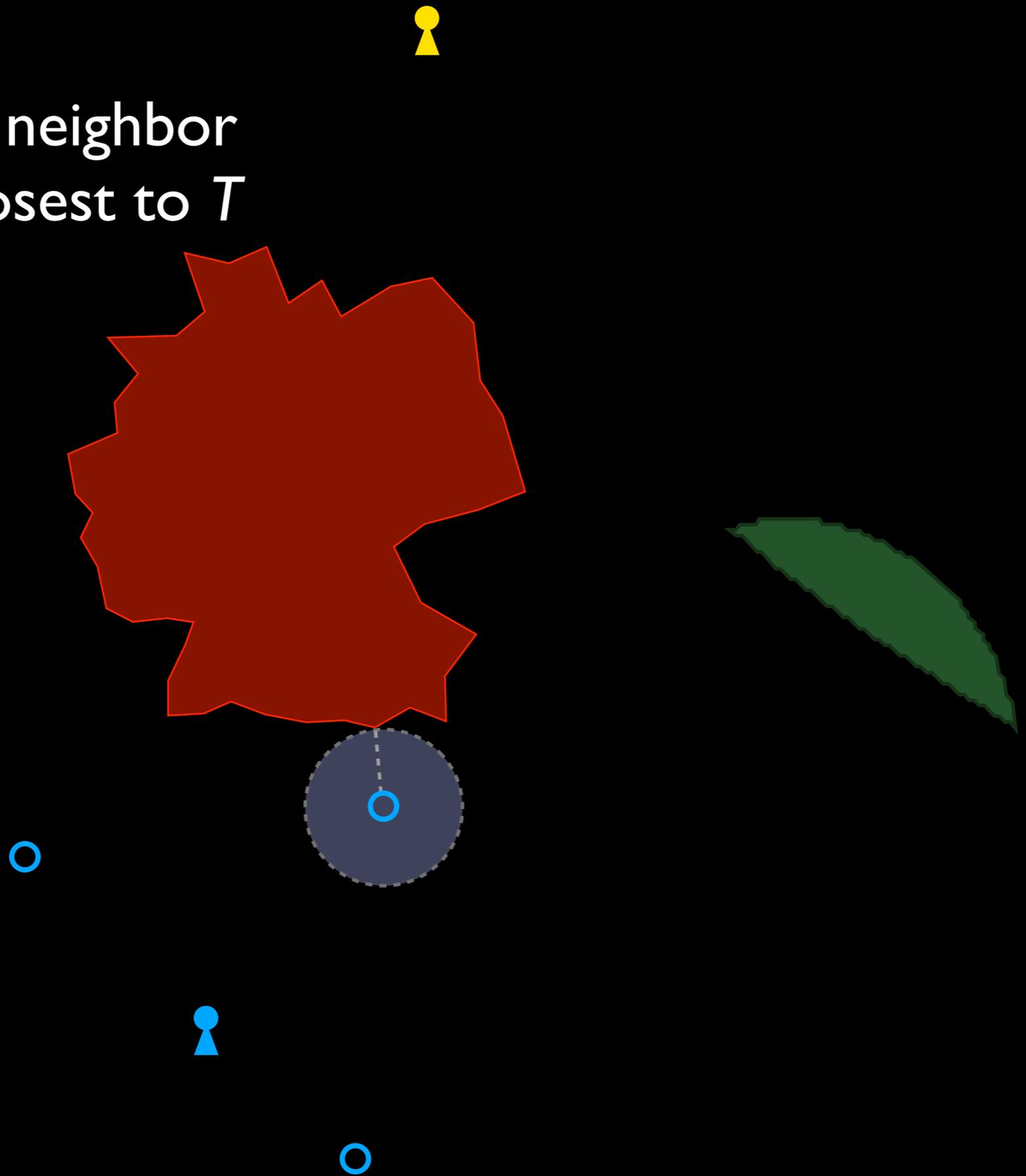
# Progress:

Forward to the (safe) neighbor  
whose safe zone is closest to  $T$



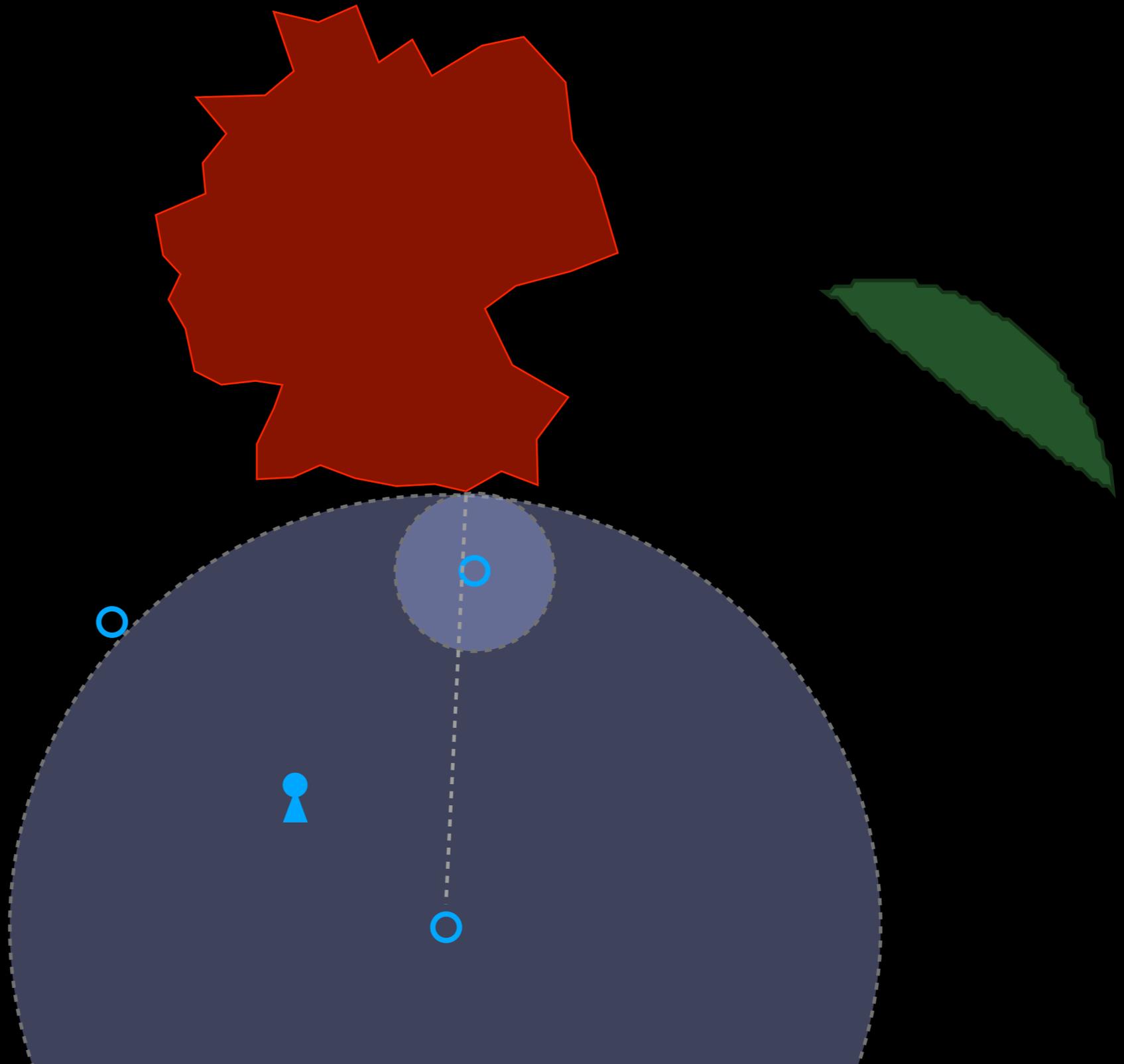
# Progress:

Forward to the (safe) neighbor  
whose safe zone is closest to  $T$



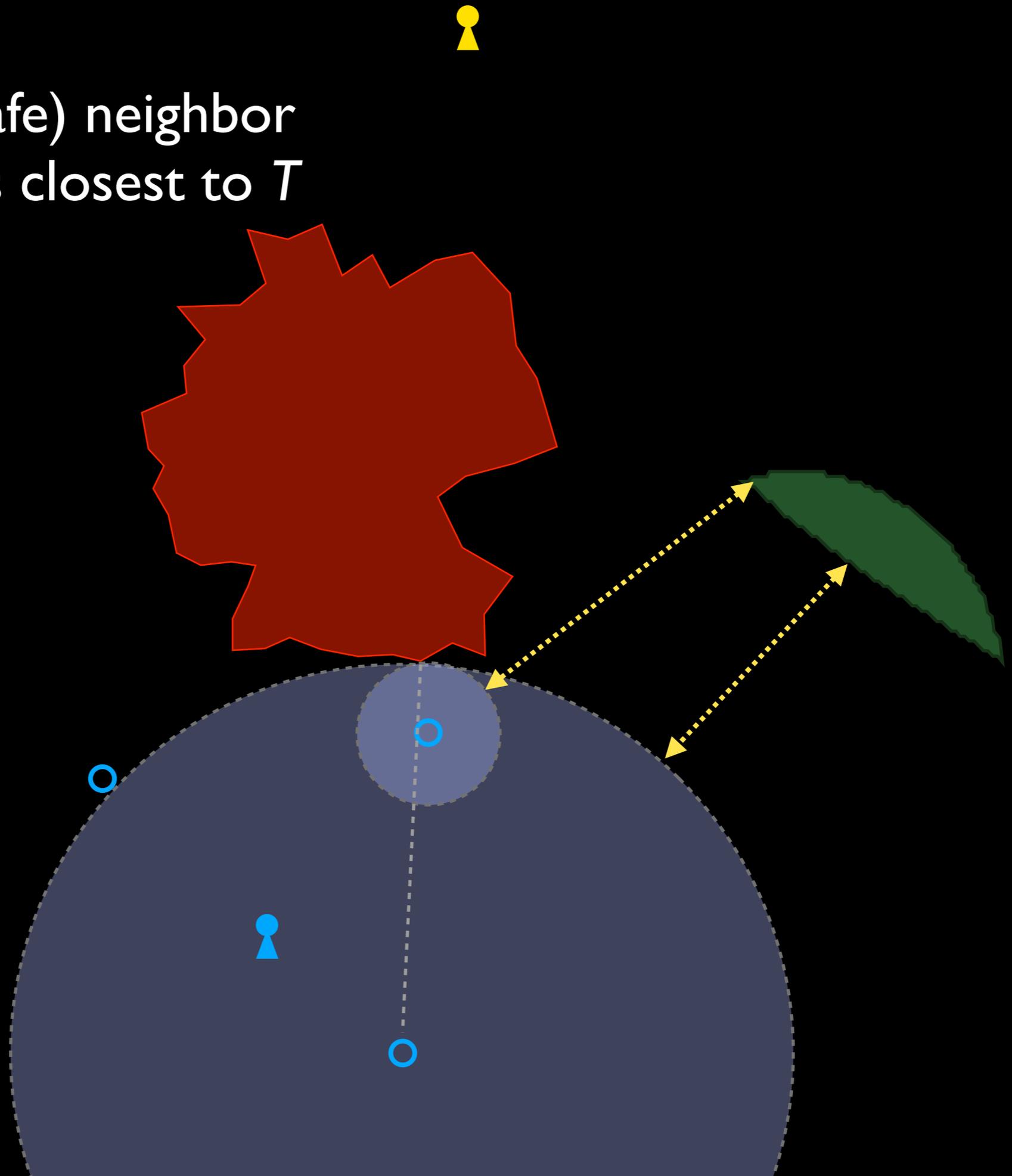
# Progress:

Forward to the (safe) neighbor  
whose safe zone is closest to  $T$



# Progress:

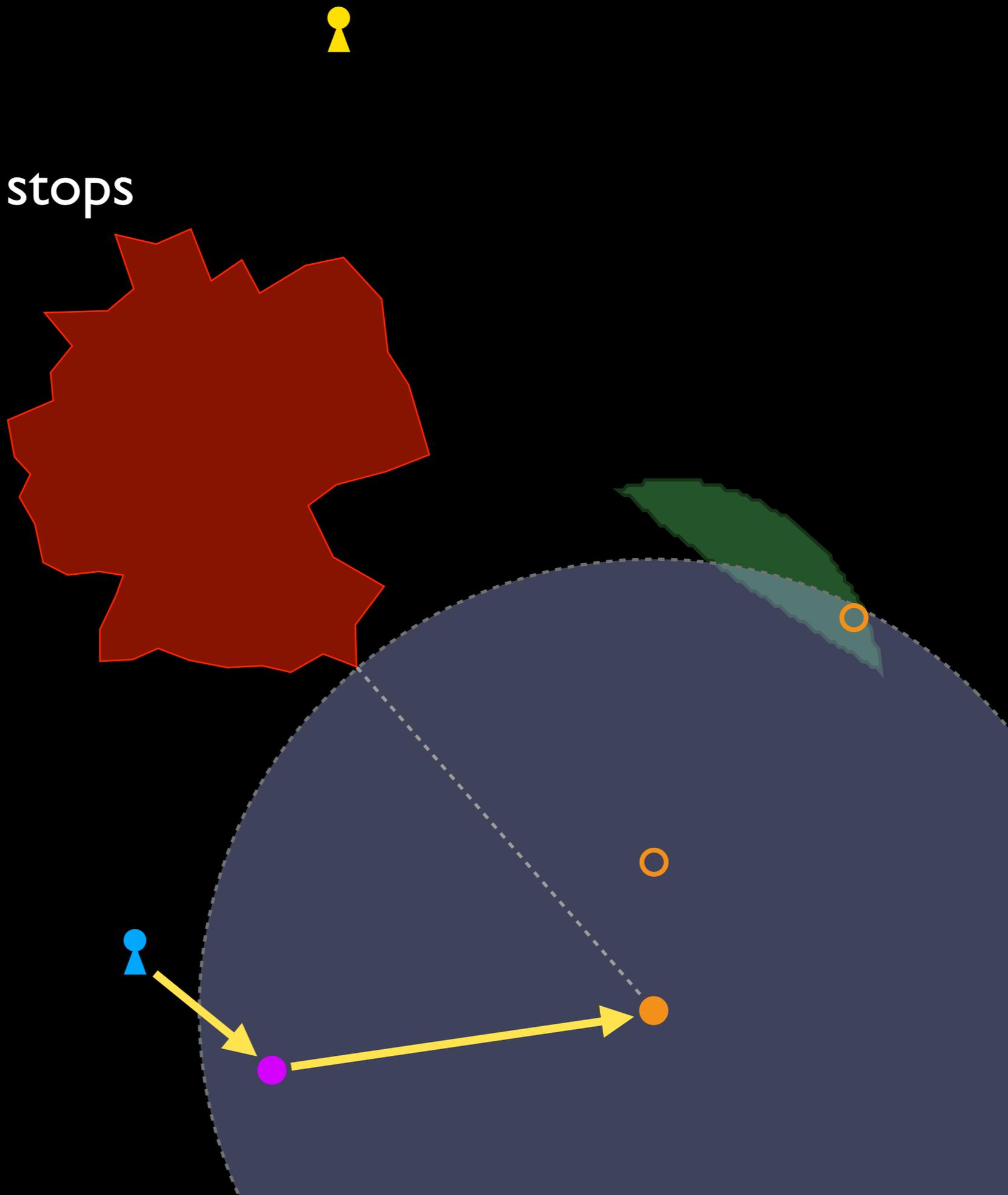
Forward to the (safe) neighbor  
whose safe zone is closest to  $T$



# Recursive forwarding

Forward  $F$  and  $T$

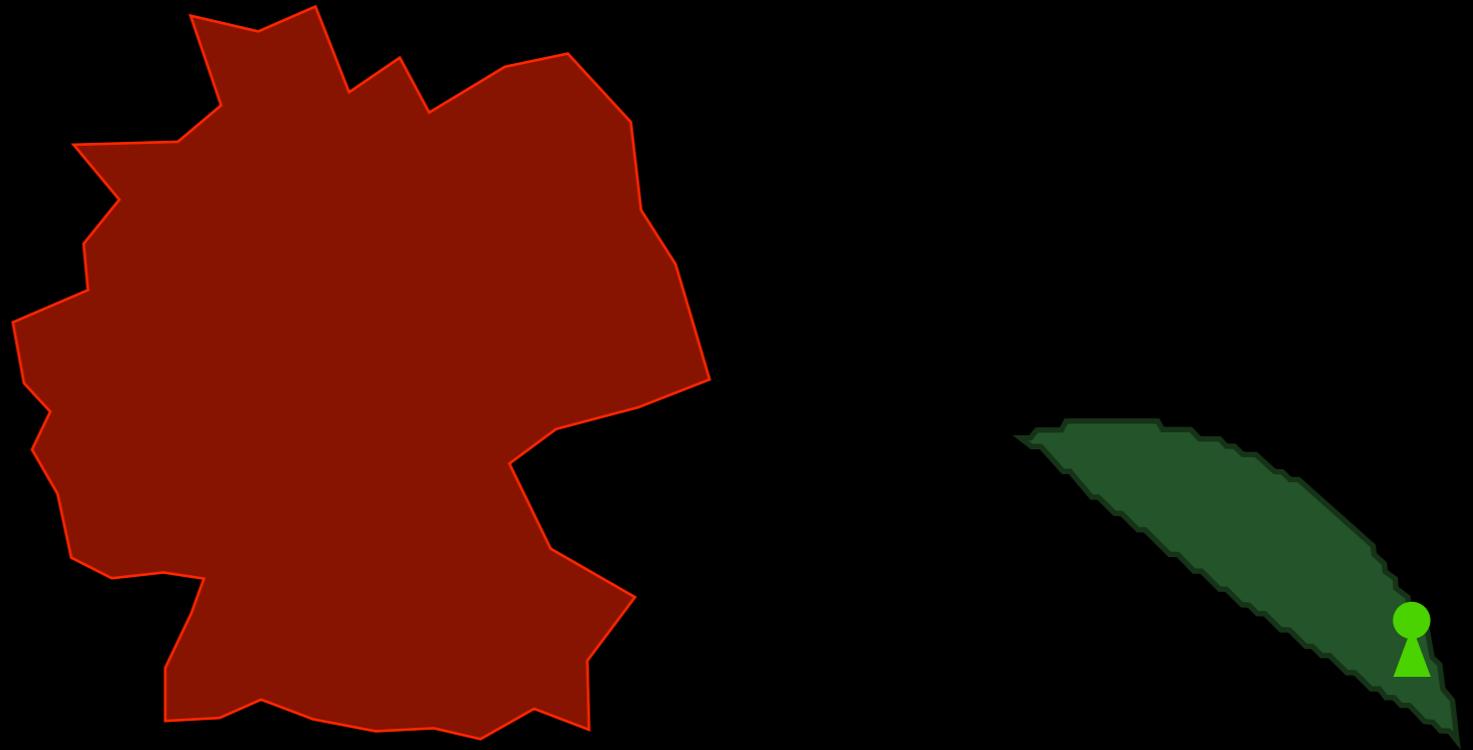
Continue until progress stops



# Recursive forwarding

Forward  $F$  and  $T$

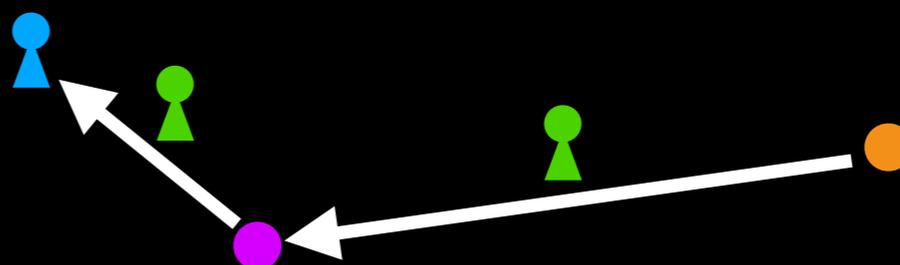
Continue until progress stops



# Recursive forwarding

Forward  $F$  and  $T$

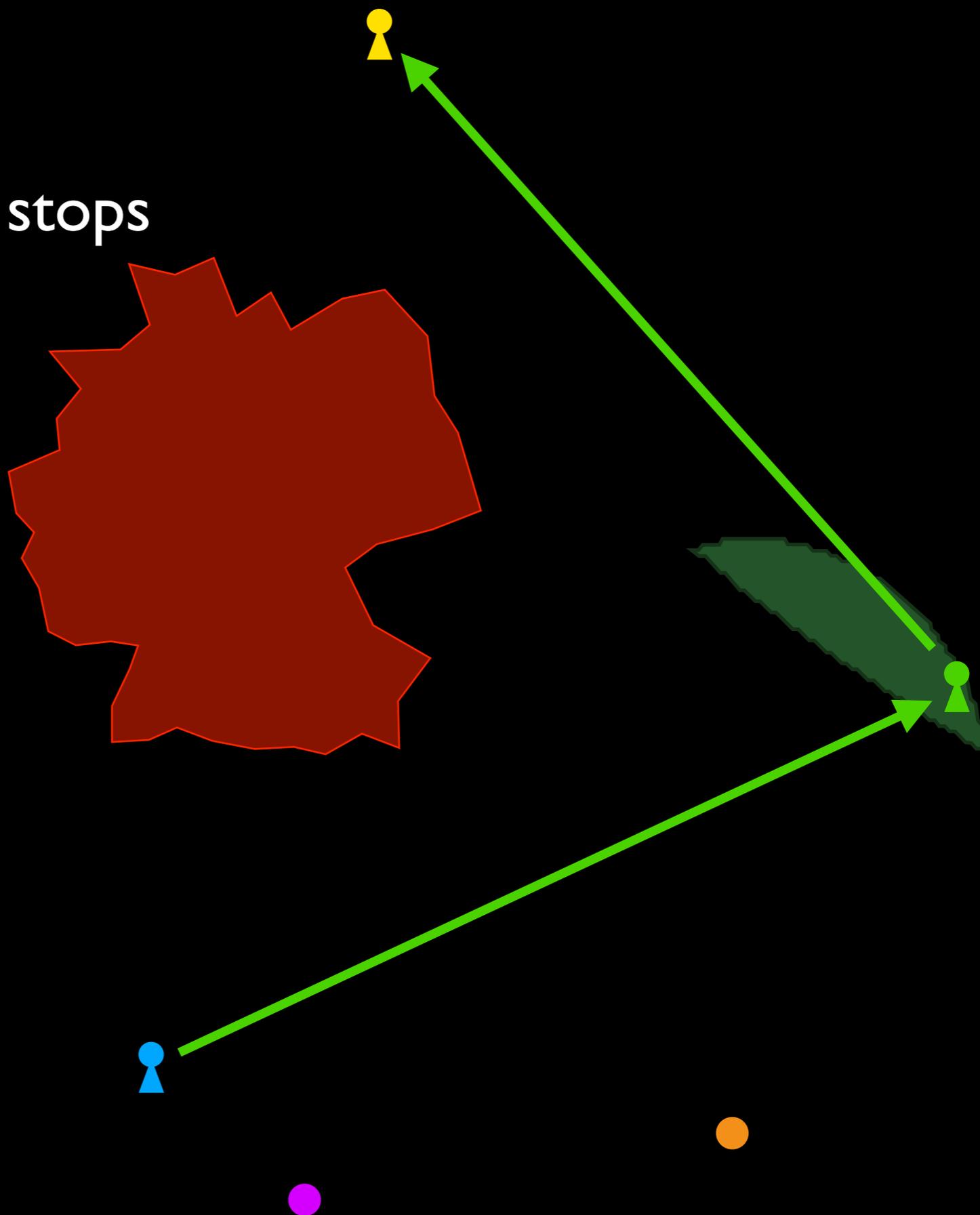
Continue until progress stops



# Recursive forwarding

Forward  $F$  and  $T$

Continue until progress stops



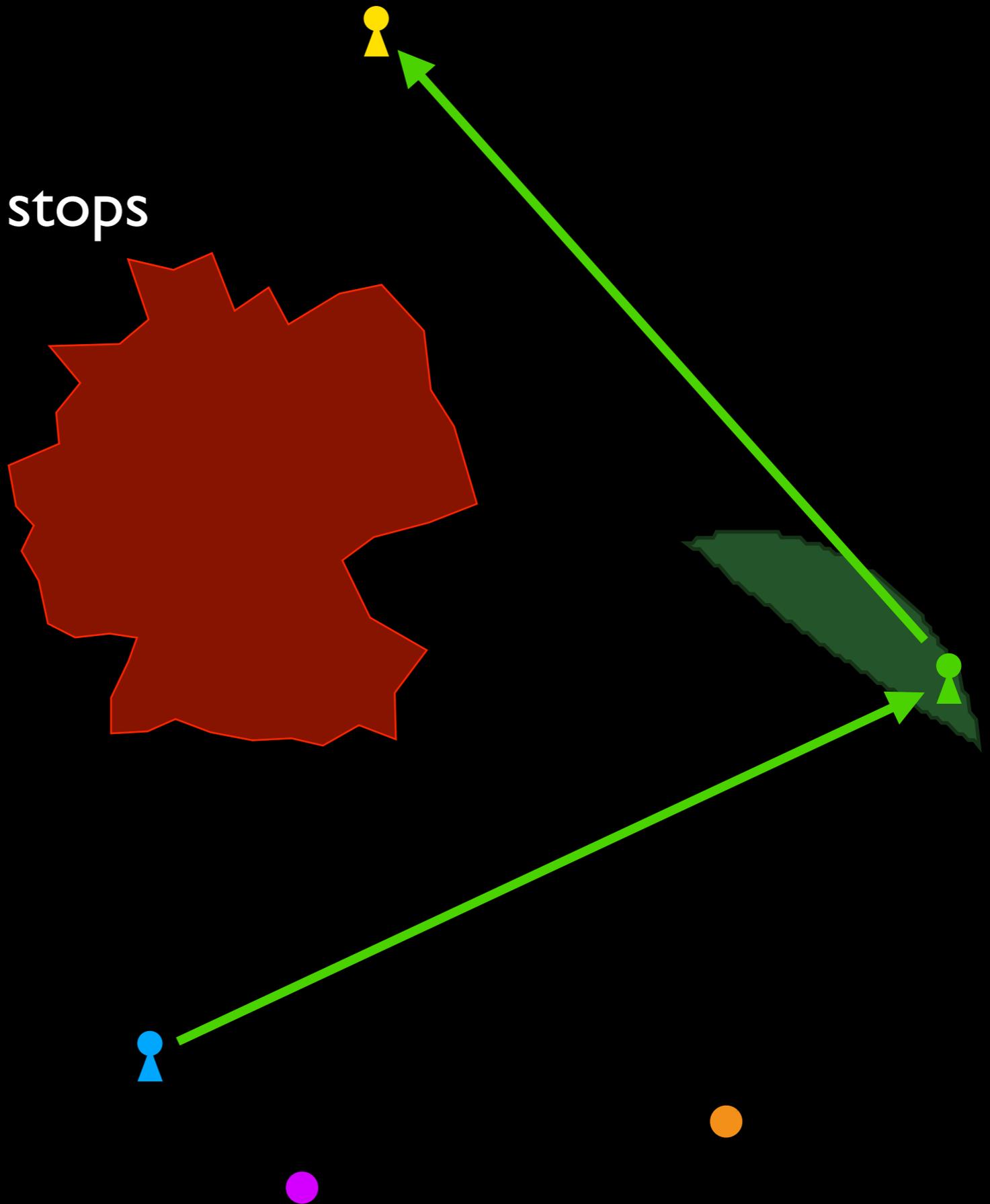
# Recursive forwarding

Forward  $F$  and  $T$

Continue until progress stops

Alibi routing finds  
*potential* alibis

Proofs of avoidance  
allow verification



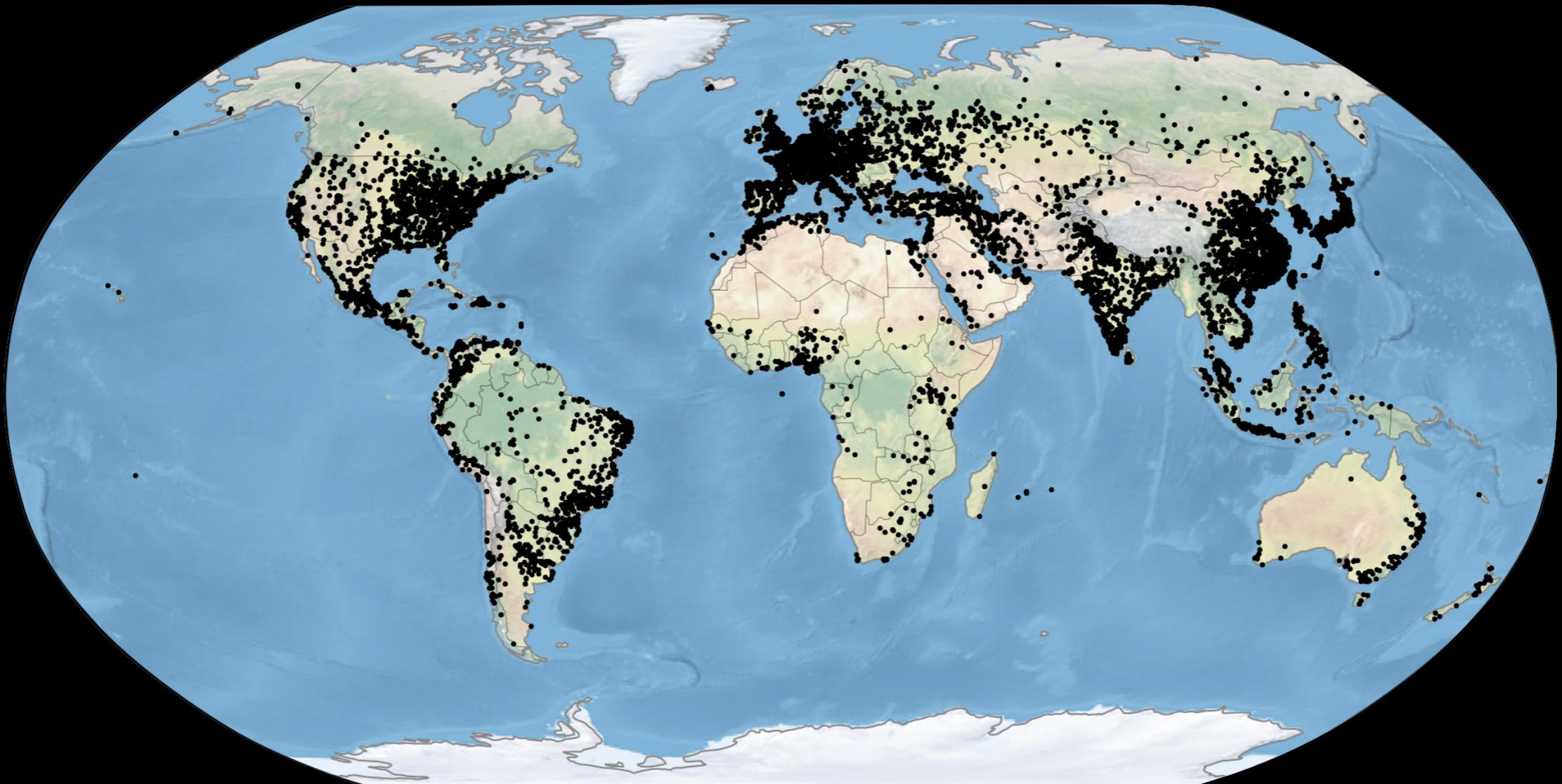
# Implementation and Evaluation

Implementation  
on PlanetLab

425 nodes

Simulation  
(for scale)

20k nodes



# Implementation and Evaluation

Implementation  
on PlanetLab

425 nodes

Simulation  
(for scale)

20k nodes

China Iran PR Korea Syria Saudi Arabia

Known censors (Reporters without Borders)

India Japan USA

Most Internet users

# Implementation and Evaluation

Implementation  
on PlanetLab

425 nodes

Simulation  
(for scale)

20k nodes

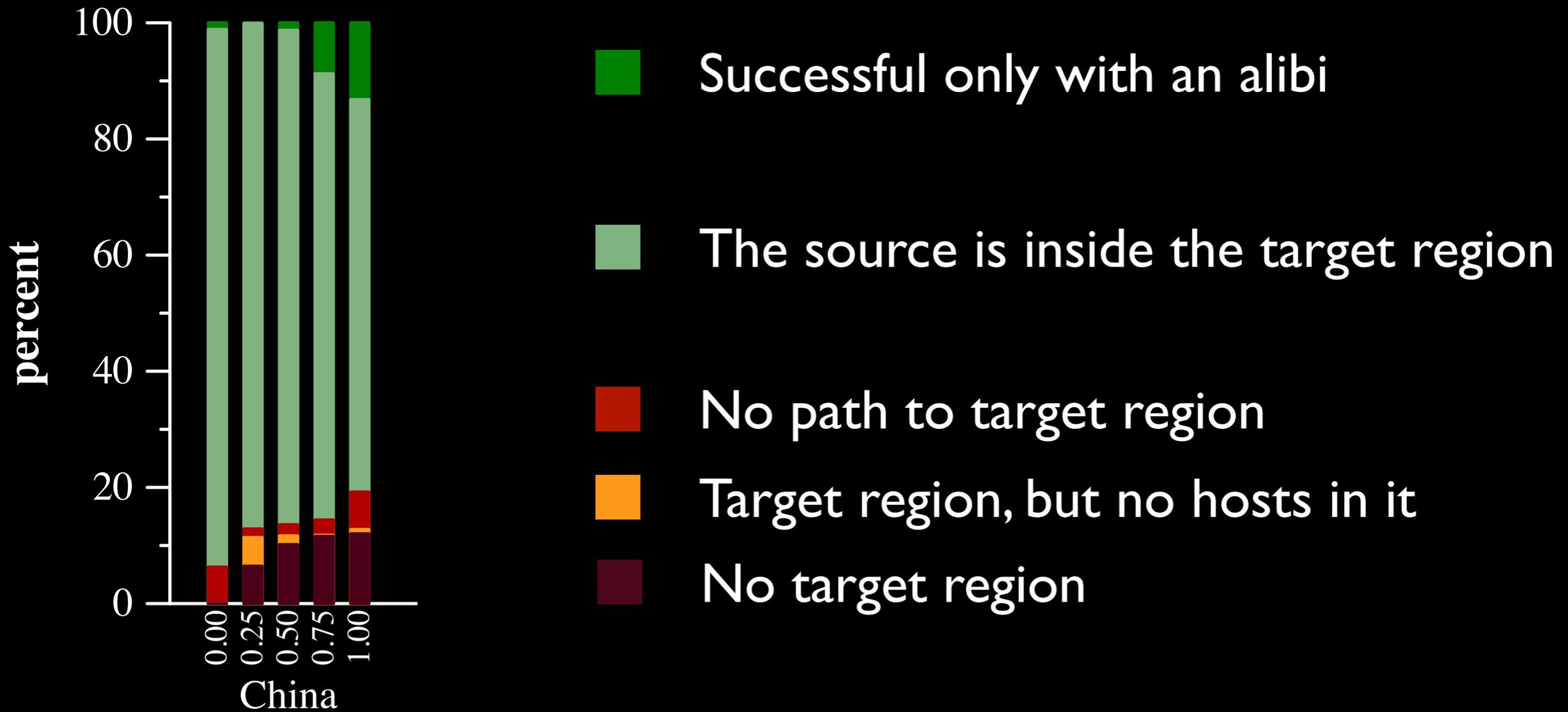
China Iran PR Korea Syria Saudi Arabia

Known censors (Reporters without Borders)

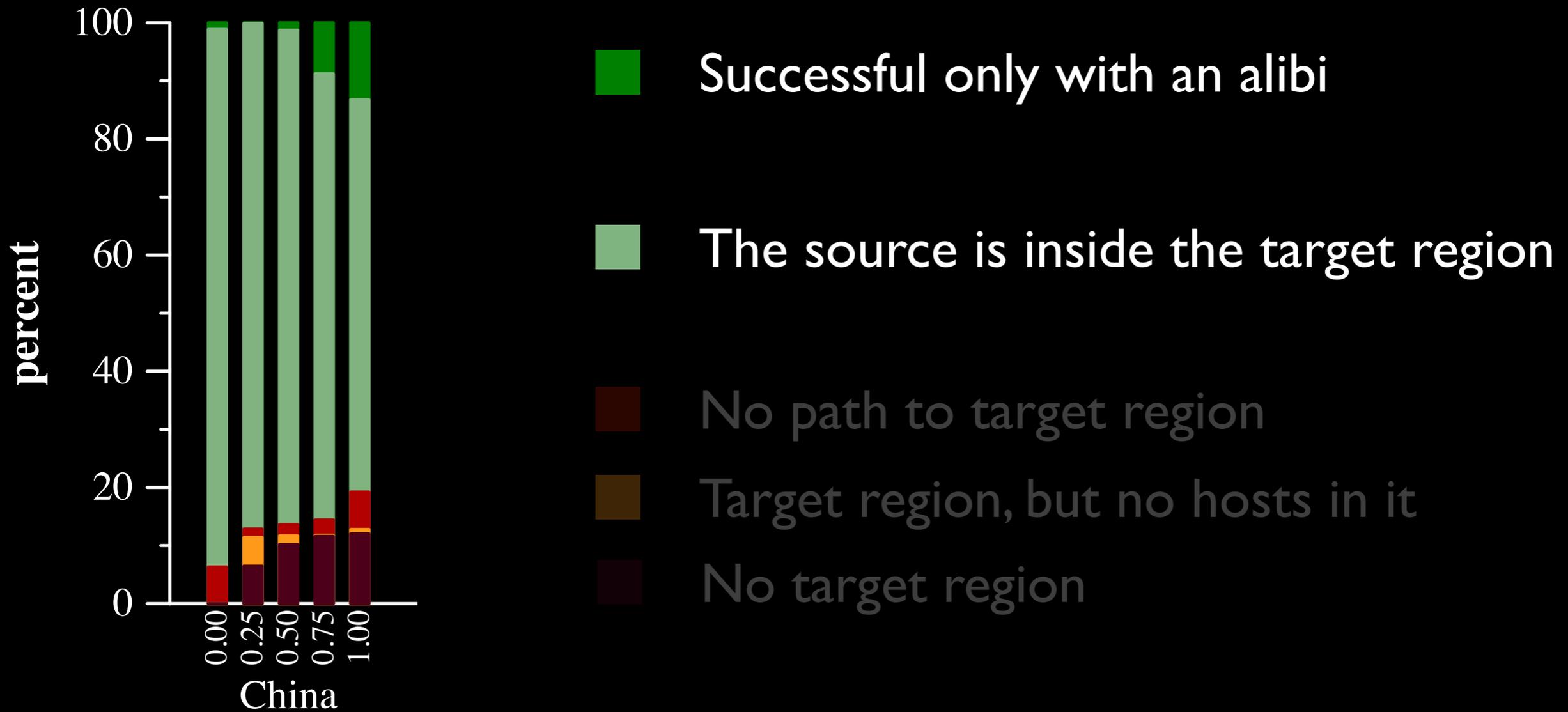
India Japan USA

Most Internet users

# Alibi Routing success rates

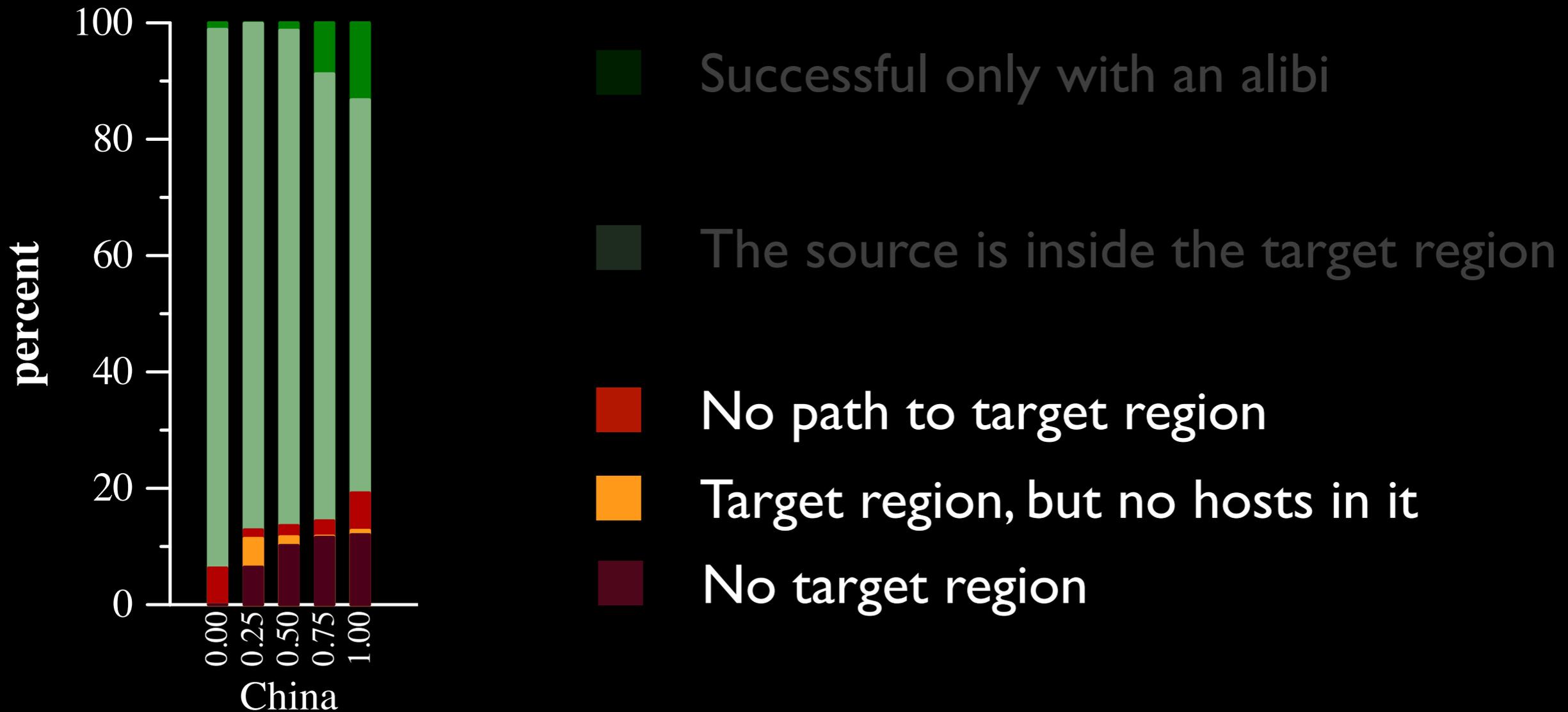


# Alibi Routing success rates



Most src-dst pairs can provably avoid

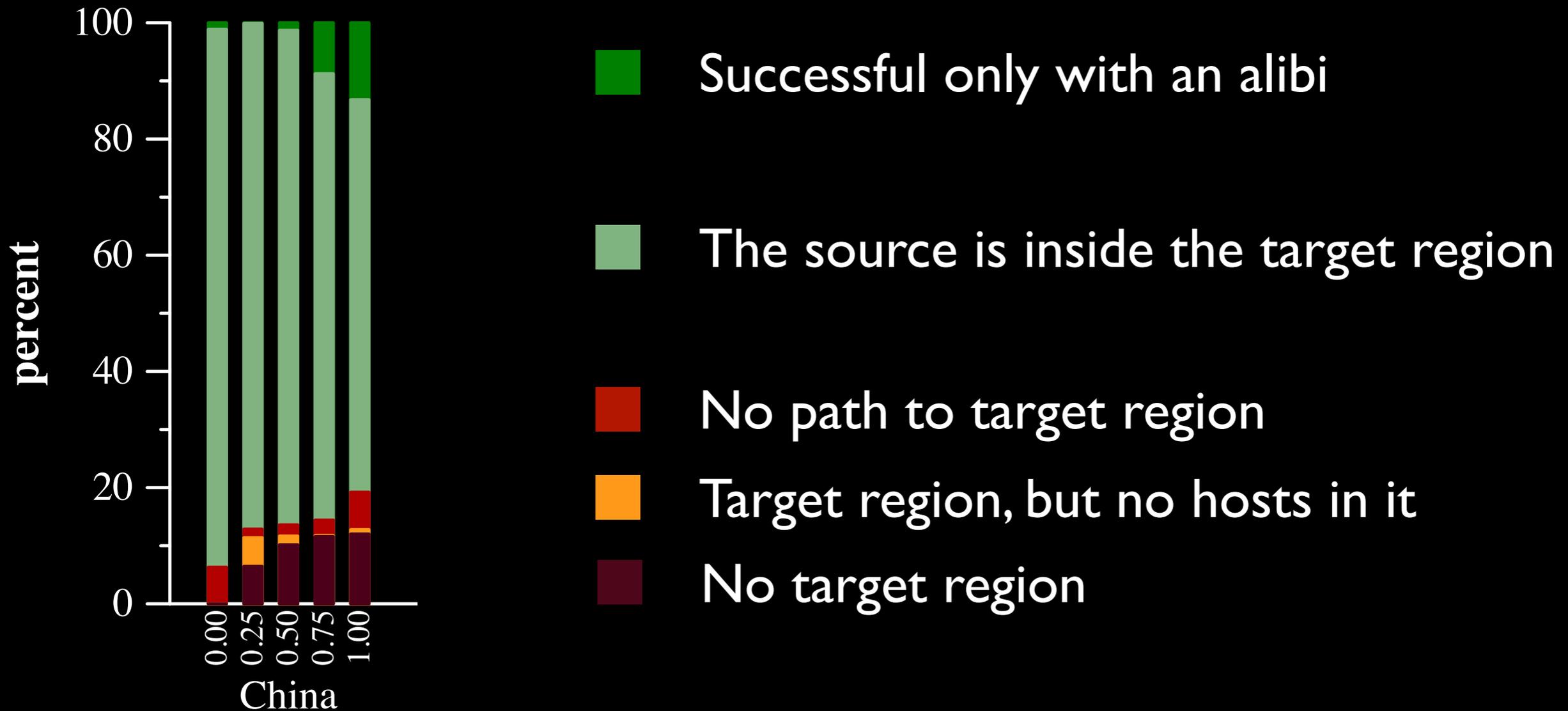
# Alibi Routing success rates



Most src-dst pairs can provably avoid

Failure typically arises when the target region is too small or non-existent

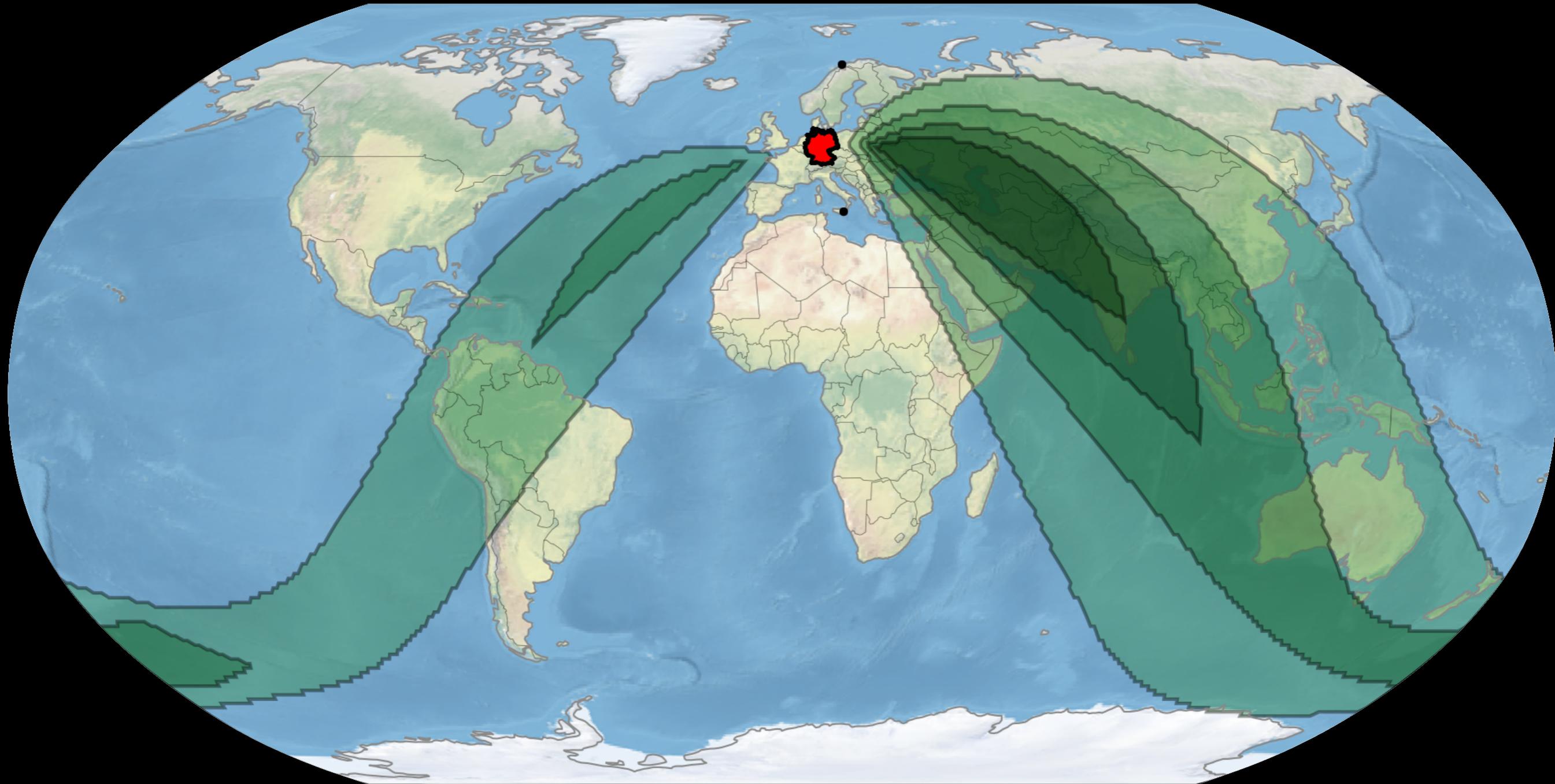
# Alibi Routing success rates



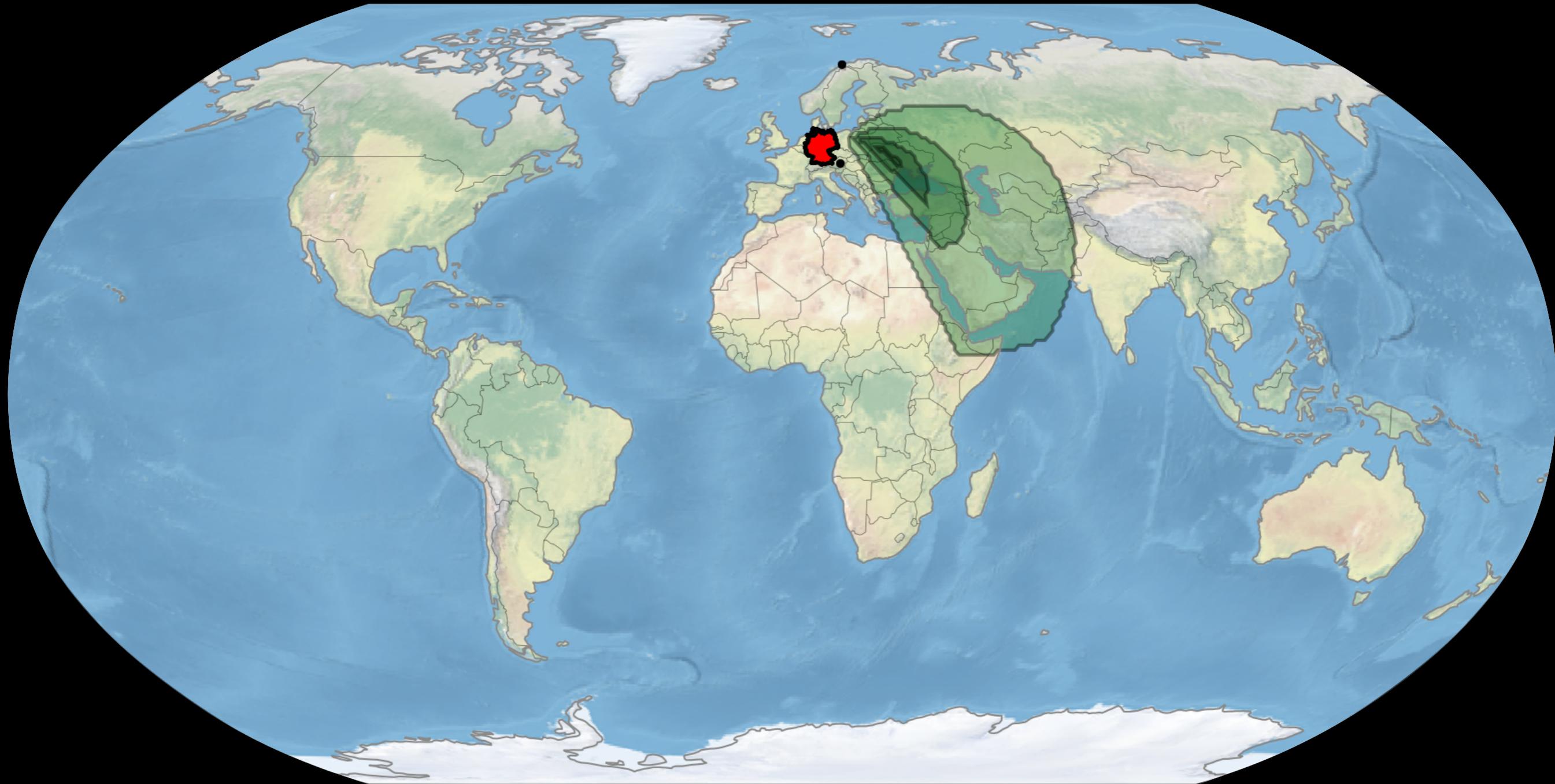
Most src-dst pairs can provably avoid

Failure typically arises when the target region is too small or non-existent

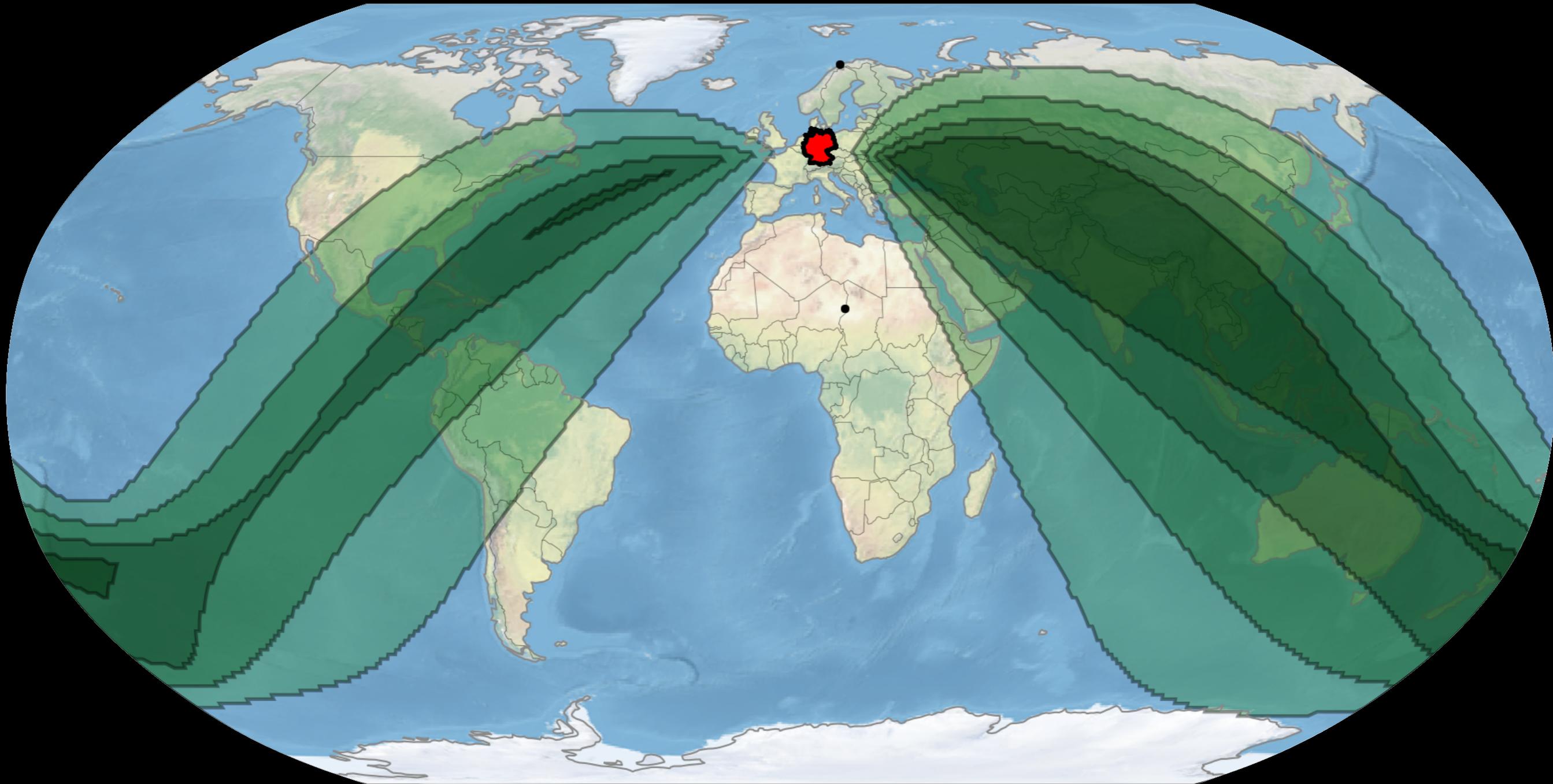
# Proximity's effect on target regions



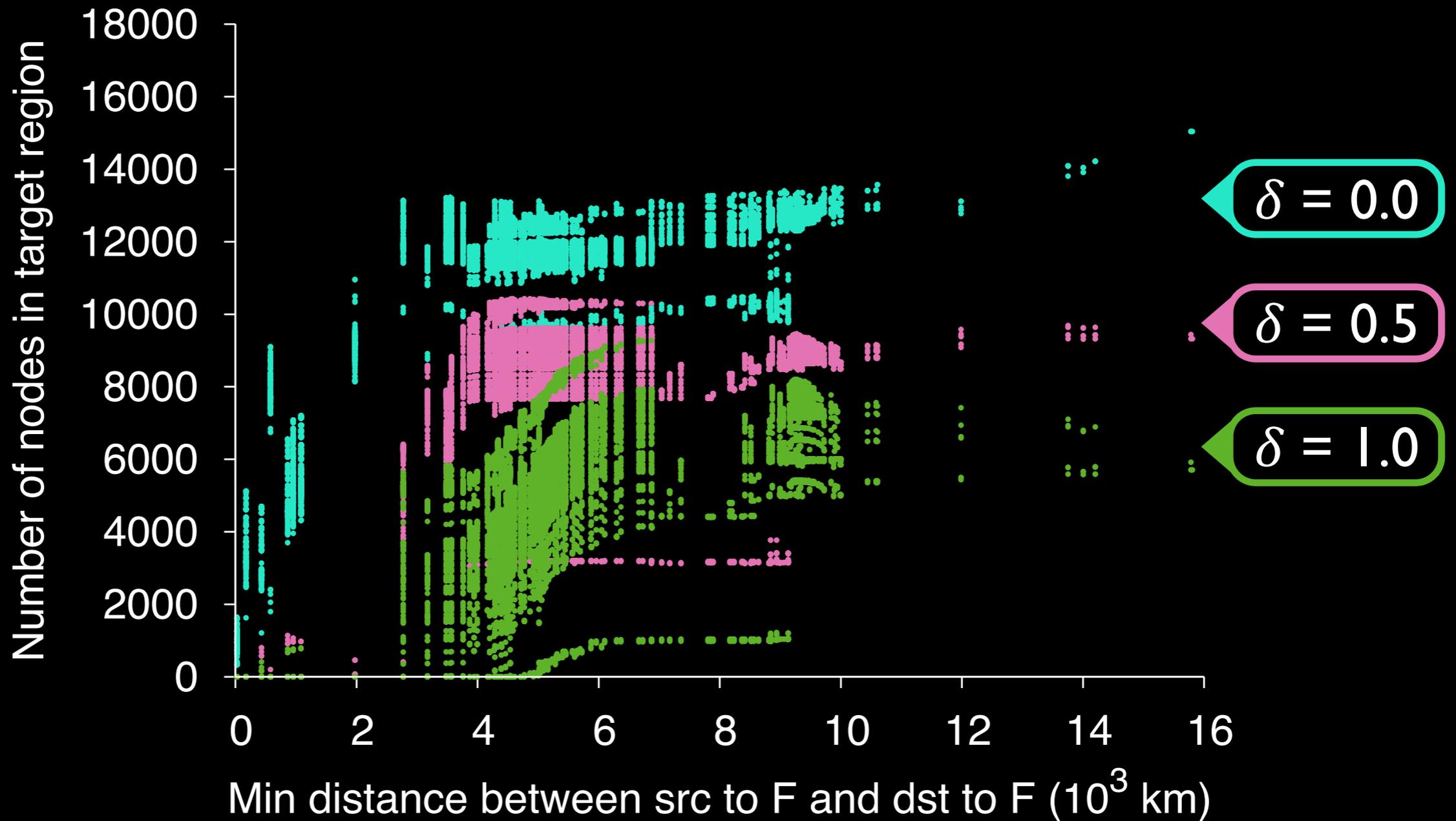
# Proximity's effect on target regions



# Proximity's effect on target regions



# Proximity's effect on target regions



Failure is likely when source or destination are very close to the forbidden region

# Other results

- Routes through alibis incur **little increase in latency**
  - Sometimes even **lower** latencies
- Alibi Routing incurs little communication overhead
- Countries with higher **routing centrality** are harder, but not impossible, to avoid

**Provable avoidance is possible  
safely and efficiently**

# Summary

- **Provable avoidance routing**
  - Users to specify where they want their packets *not* to go
- “**Proof by alibi**” makes it **possible to provably avoid arbitrary geographic regions** without ISP/BGP support
- **Alibi Routing** finds potential alibis
  - **Successfully**, so long as src/dst not *too* close
  - **At low cost** in terms of latency inflation

Code and data available at:

[alibi.cs.umd.edu](http://alibi.cs.umd.edu)

# Tor Network

## Vulnerable to censorship

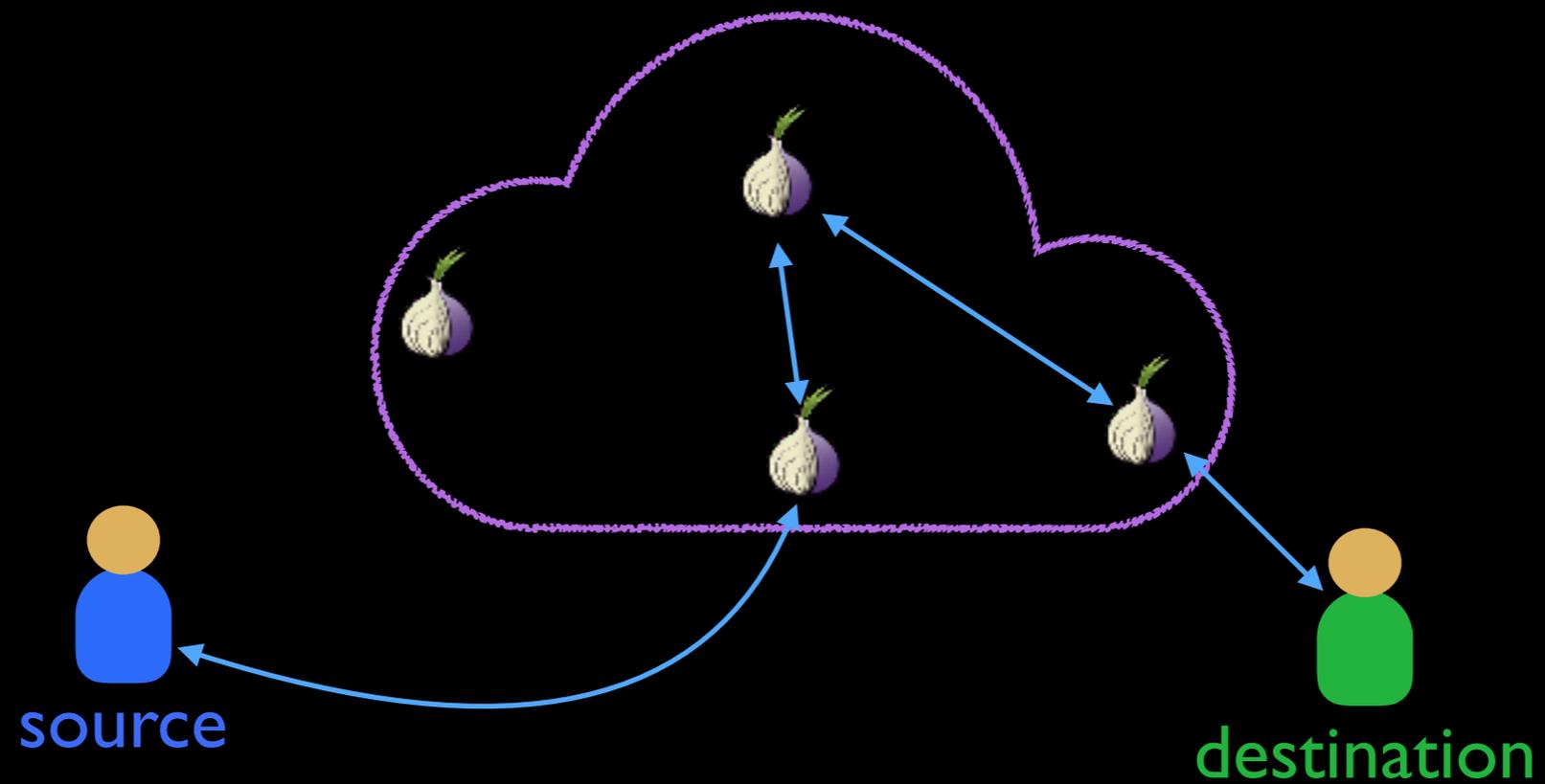
[Anon. CCR'12]



# Tor Network

## Vulnerable to censorship

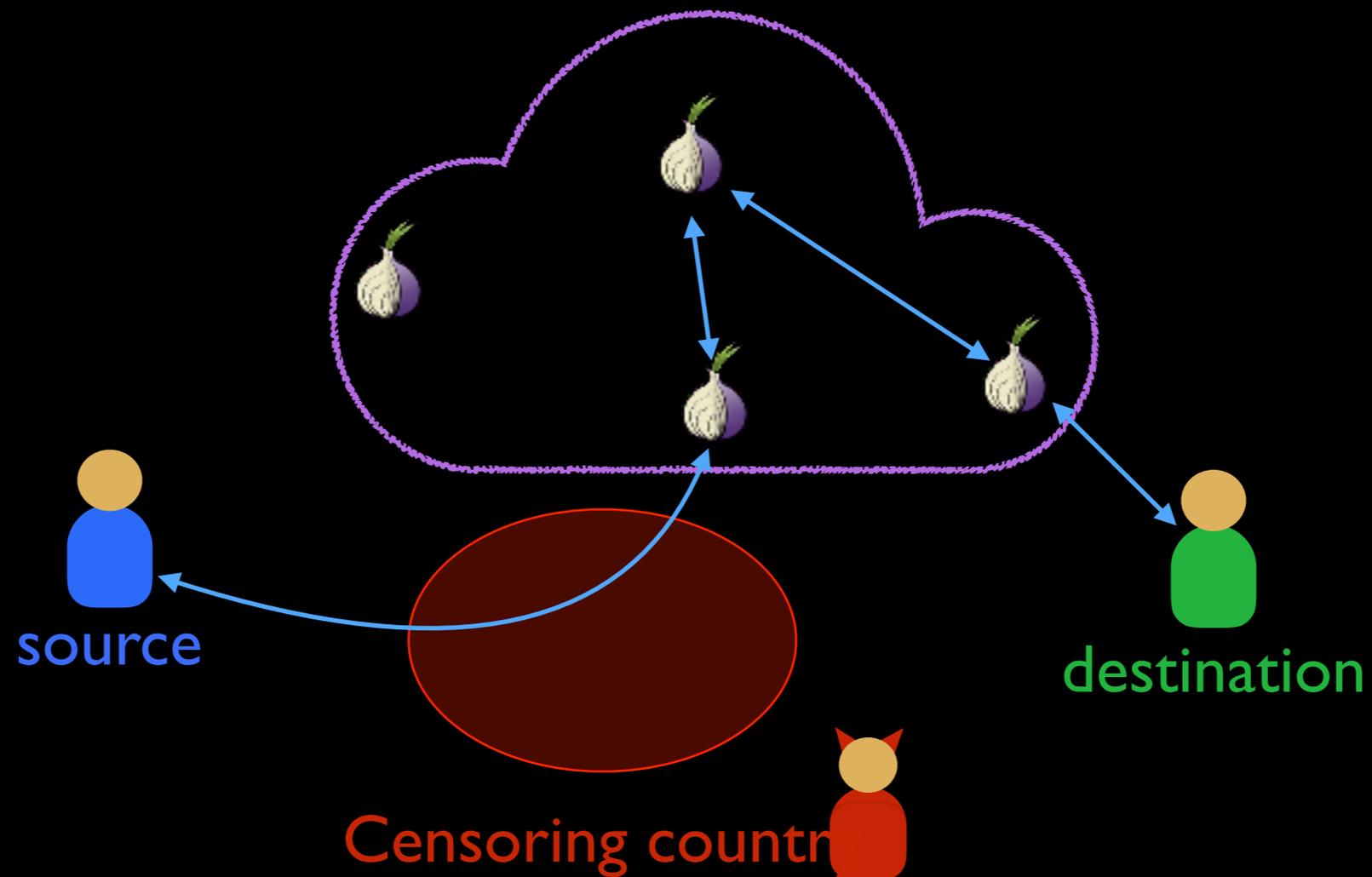
[Anon. CCR'12]



# Tor Network

## Vulnerable to censorship

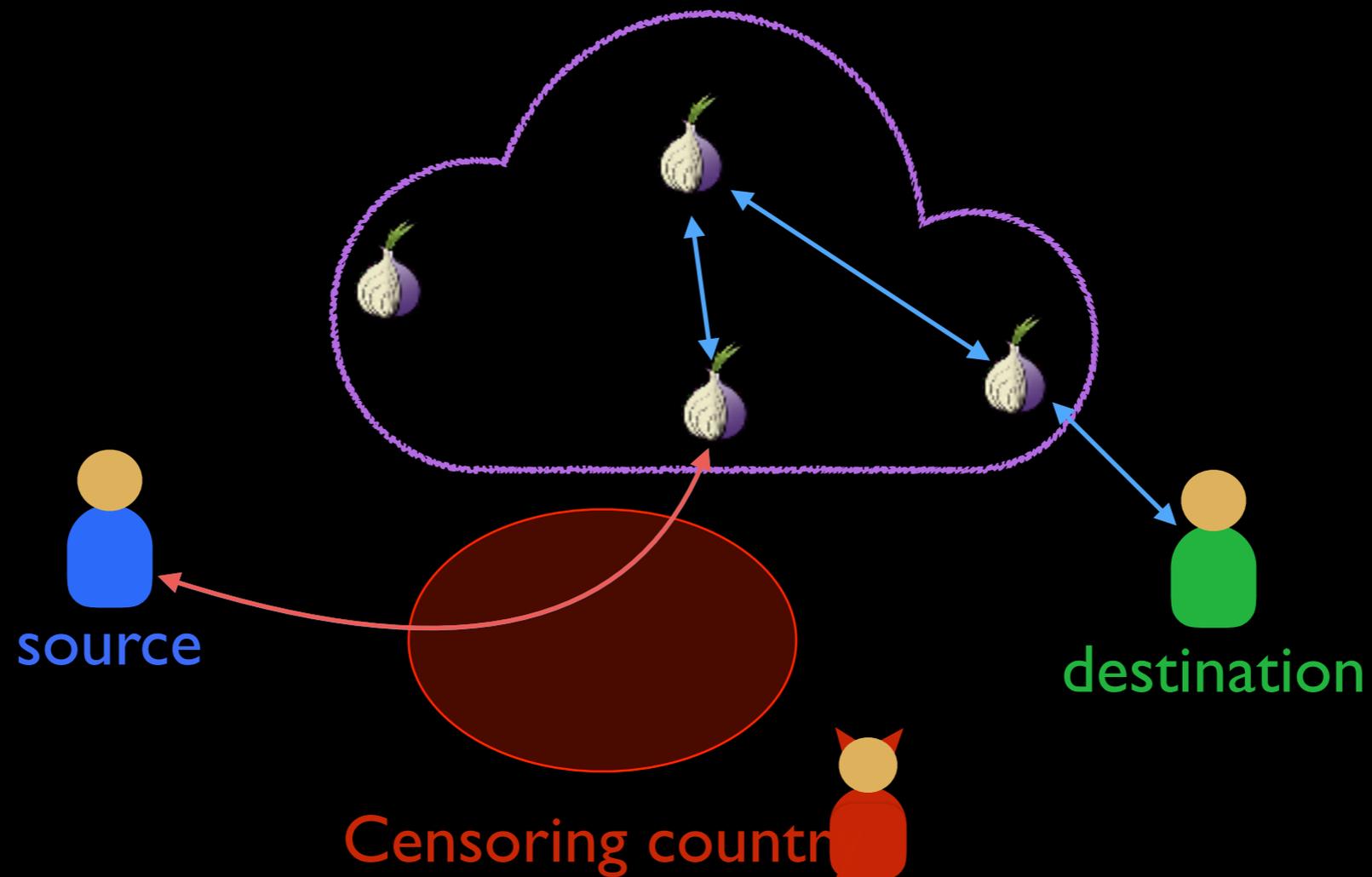
[Anon. CCR'12]



# Tor Network

## Vulnerable to censorship

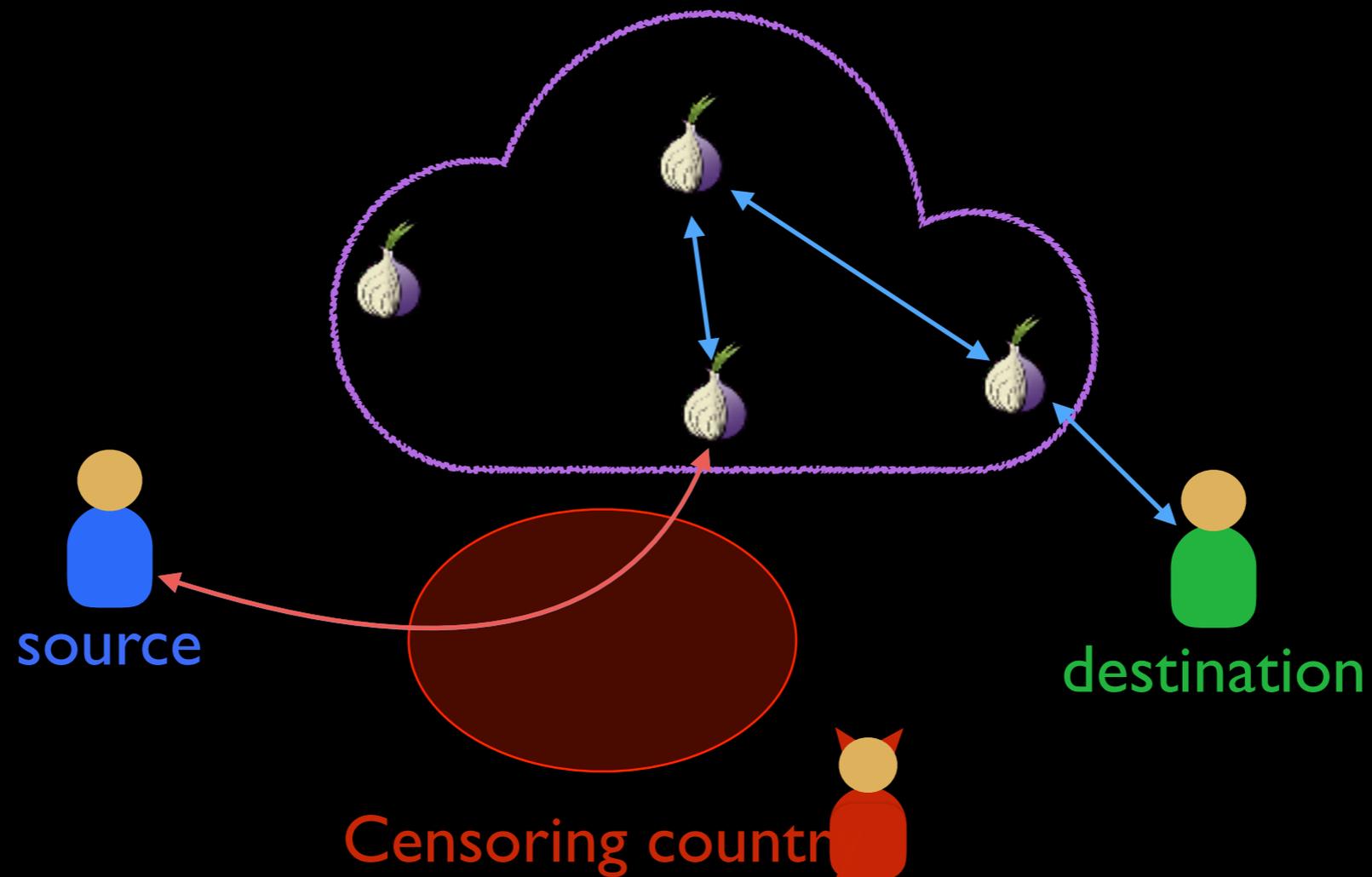
[Anon. CCR'12]



# Tor Network

## Vulnerable to censorship

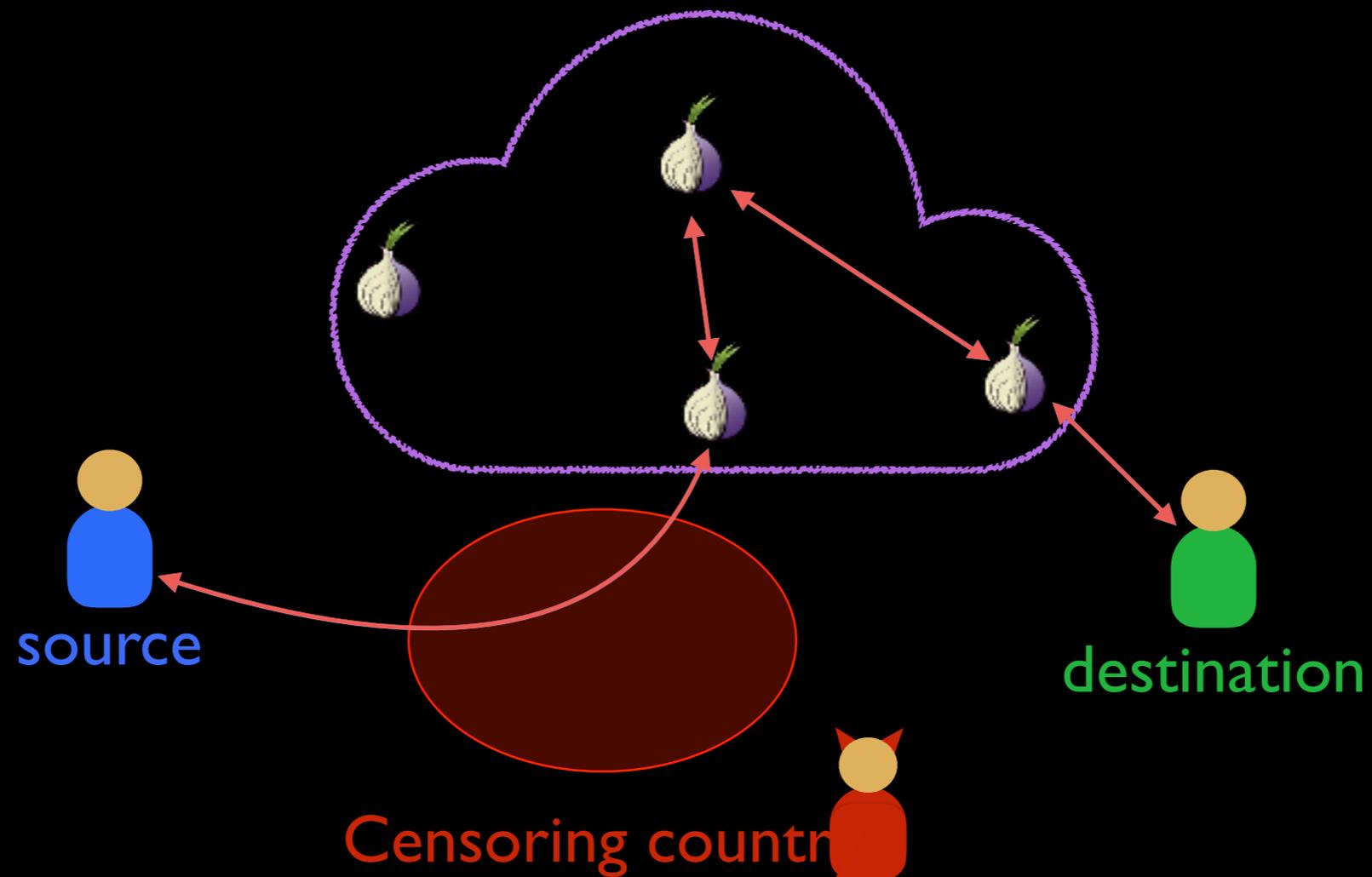
[Anon. CCR'12]



# Tor Network

## Vulnerable to censorship

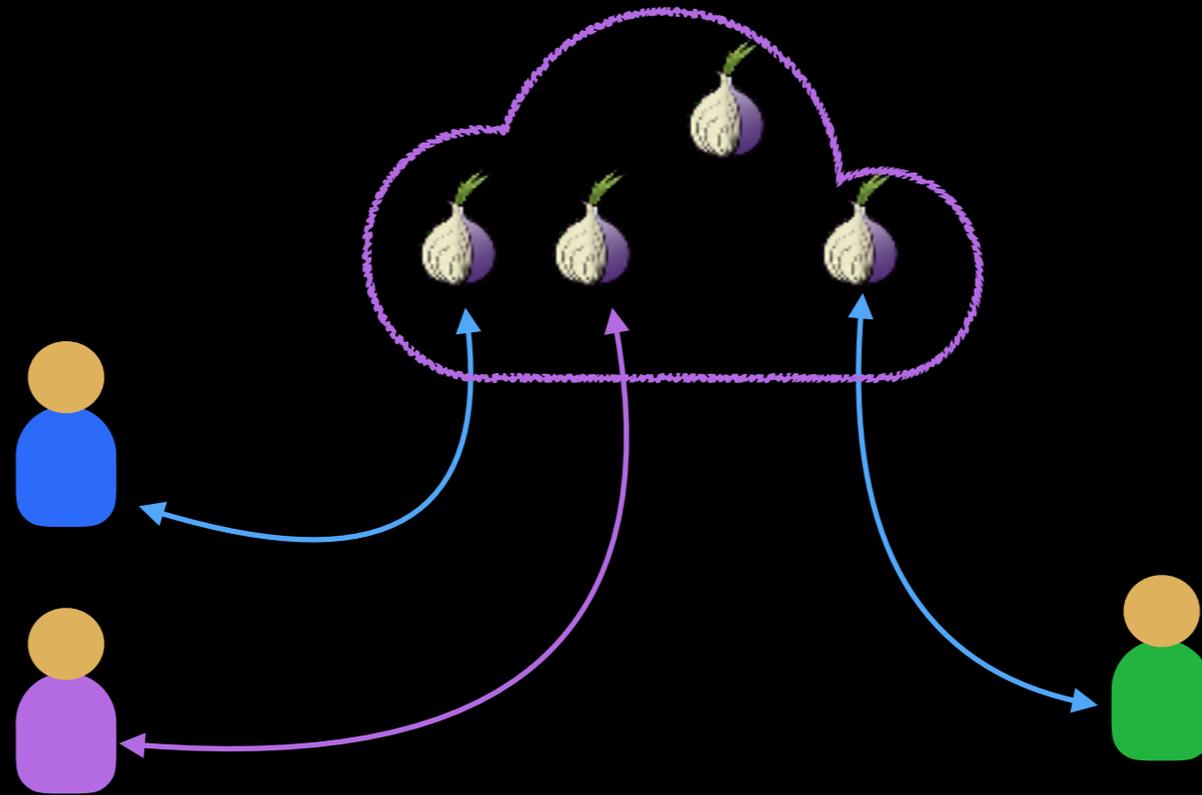
[Anon. CCR'12]



# Tor Network

## Vulnerable to traffic correlation attacks

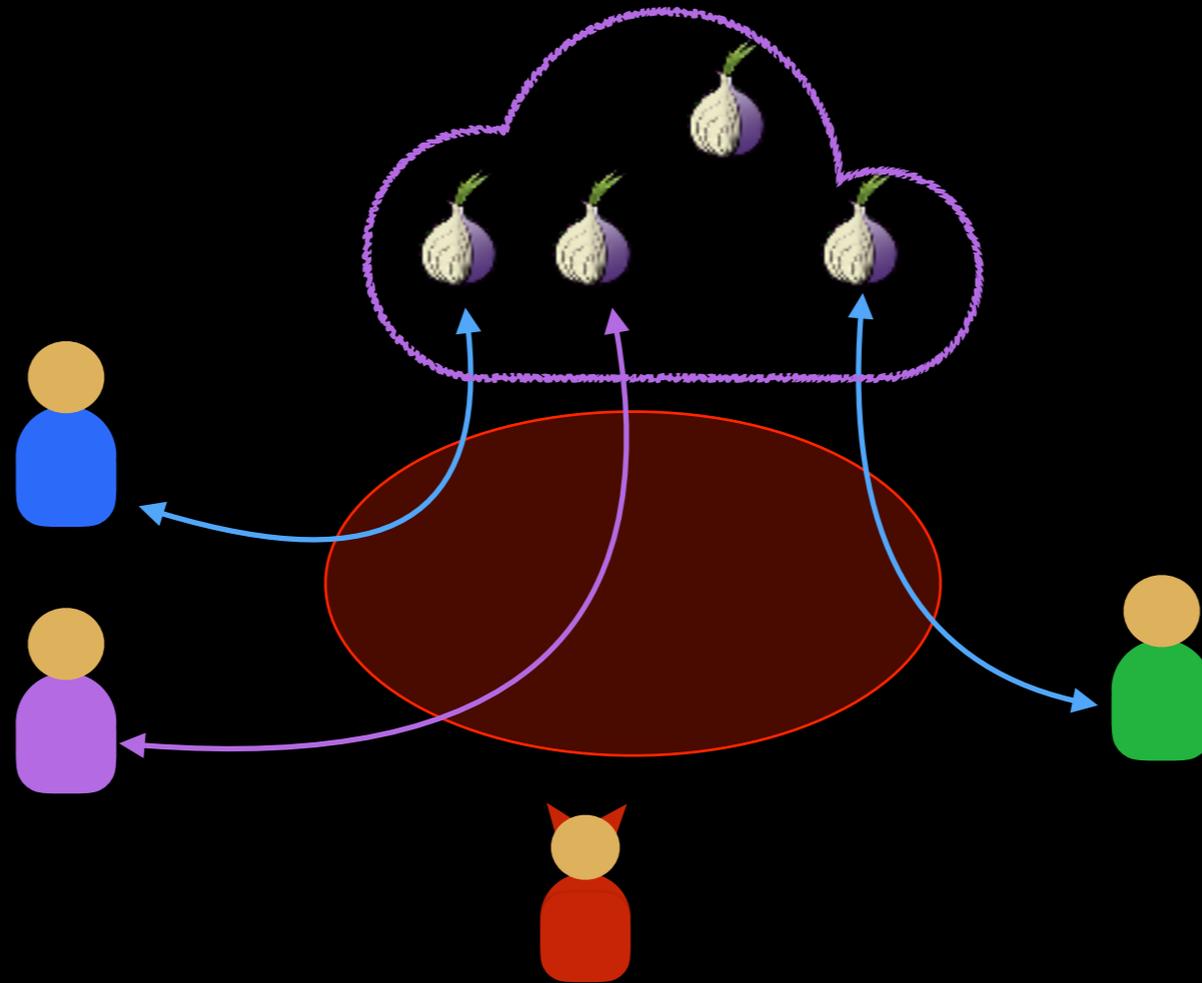
[Hopper et al., ACM TISSEC. 2010; Gilad et al., PETS 2012]



# Tor Network

## Vulnerable to traffic correlation attacks

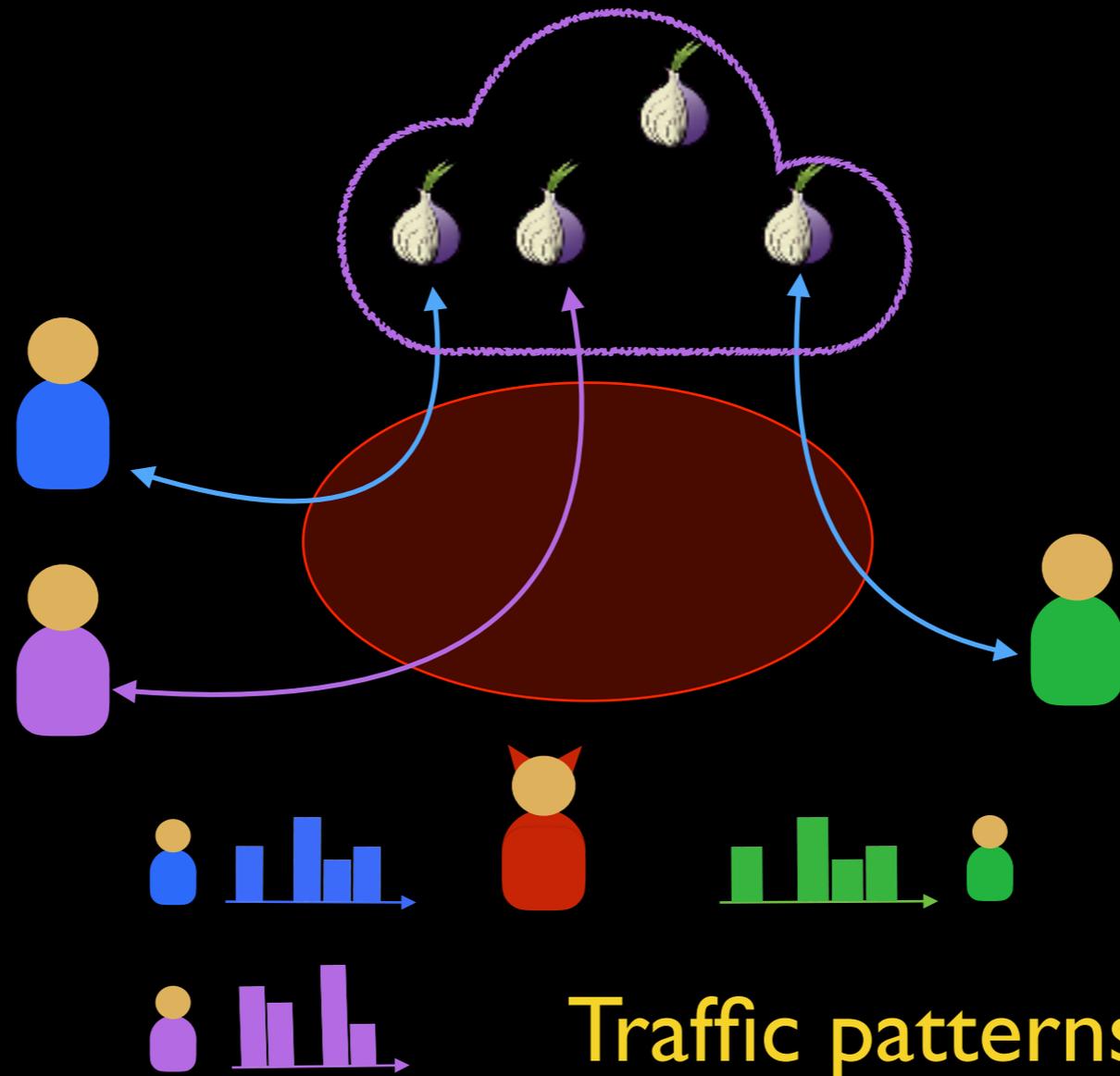
[Hopper et al., ACM TISSEC. 2010; Gilad et al., PETS 2012]



# Tor Network

## Vulnerable to traffic correlation attacks

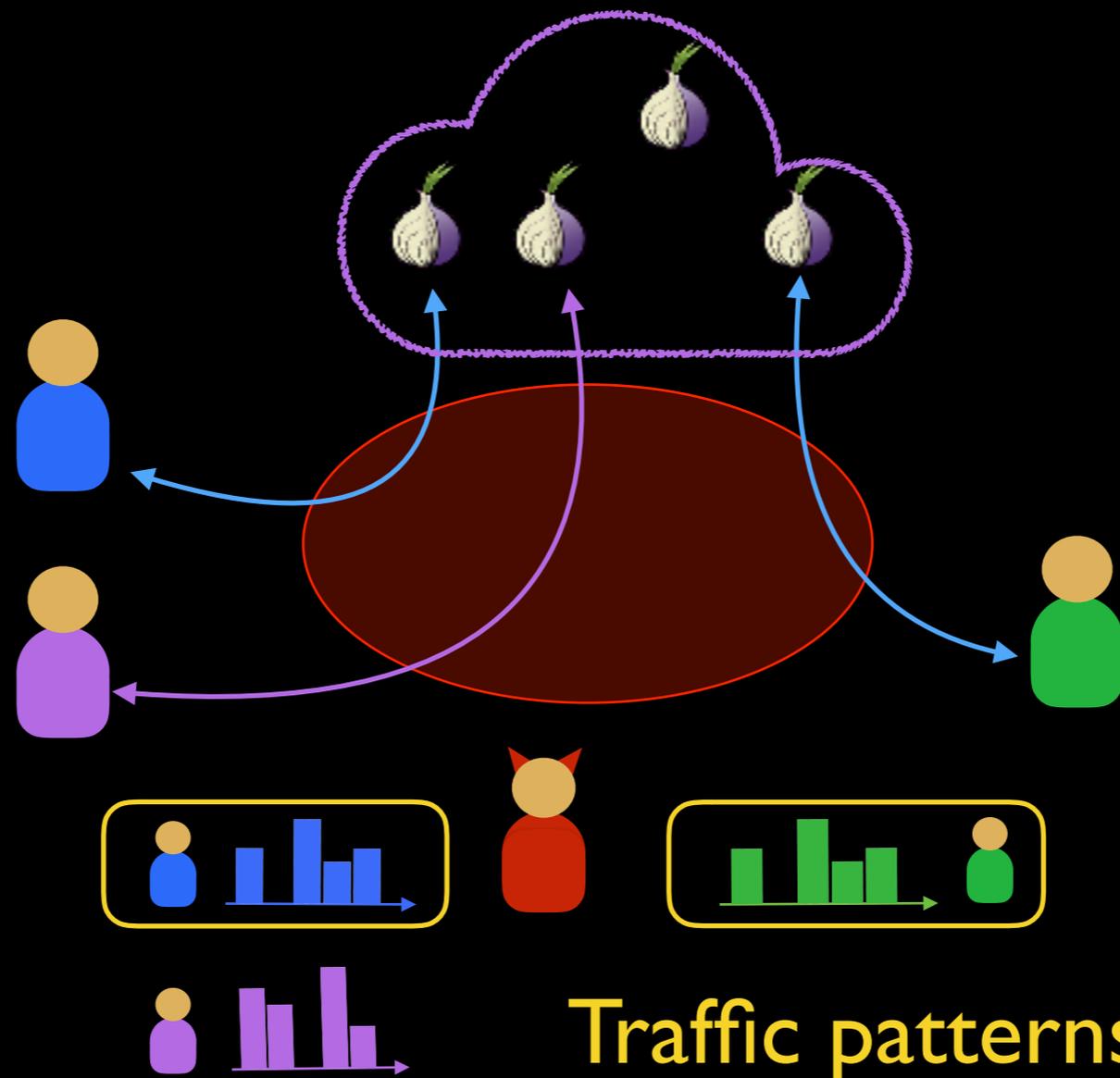
[Hopper et al., ACM TISSEC. 2010; Gilad et al., PETS 2012]



# Tor Network

## Vulnerable to traffic correlation attacks

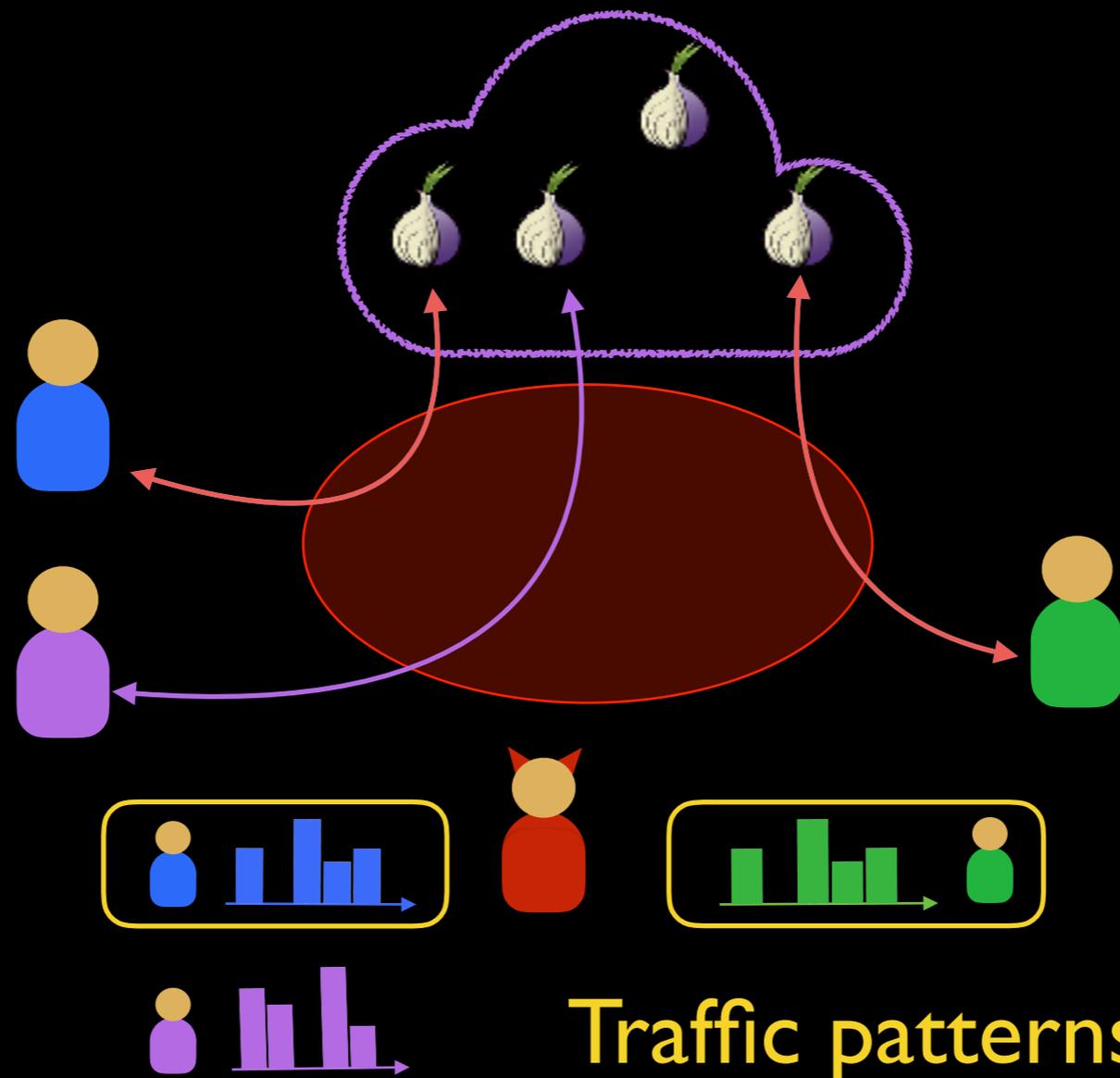
[Hopper et al., ACM TISSEC. 2010; Gilad et al., PETS 2012]



# Tor Network

## Vulnerable to traffic correlation attacks

[Hopper et al., ACM TISSEC. 2010; Gilad et al., PETS 2012]



# Threat model

## Nation-state adversary

# Threat model

## Nation-state adversary

- Adversaries can:
  - launch various attacks when on the path
  - hide from network topology measurement (e.g. *traceroute*)
  - attract routes to their administrative domains

# Threat model

## Nation-state adversary

- Adversaries can:
  - launch various attacks when on the path
  - hide from network topology measurement (e.g. *traceroute*)
  - attract routes to their administrative domains

# Threat model

## Nation-state adversary

- Adversaries can:
  - launch various attacks when on the path
  - hide from network topology measurement (e.g. *traceroute*)
  - attract routes to their administrative domains
- Adversaries cannot:

# Threat model

## Nation-state adversary

- Adversaries can:
  - launch various attacks when on the path
  - hide from network topology measurement (e.g. *traceroute*)
  - attract routes to their administrative domains
- Adversaries cannot:
  - **Fundamental assumption:**  
We know the geographic boundaries wherein the attackers reside

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

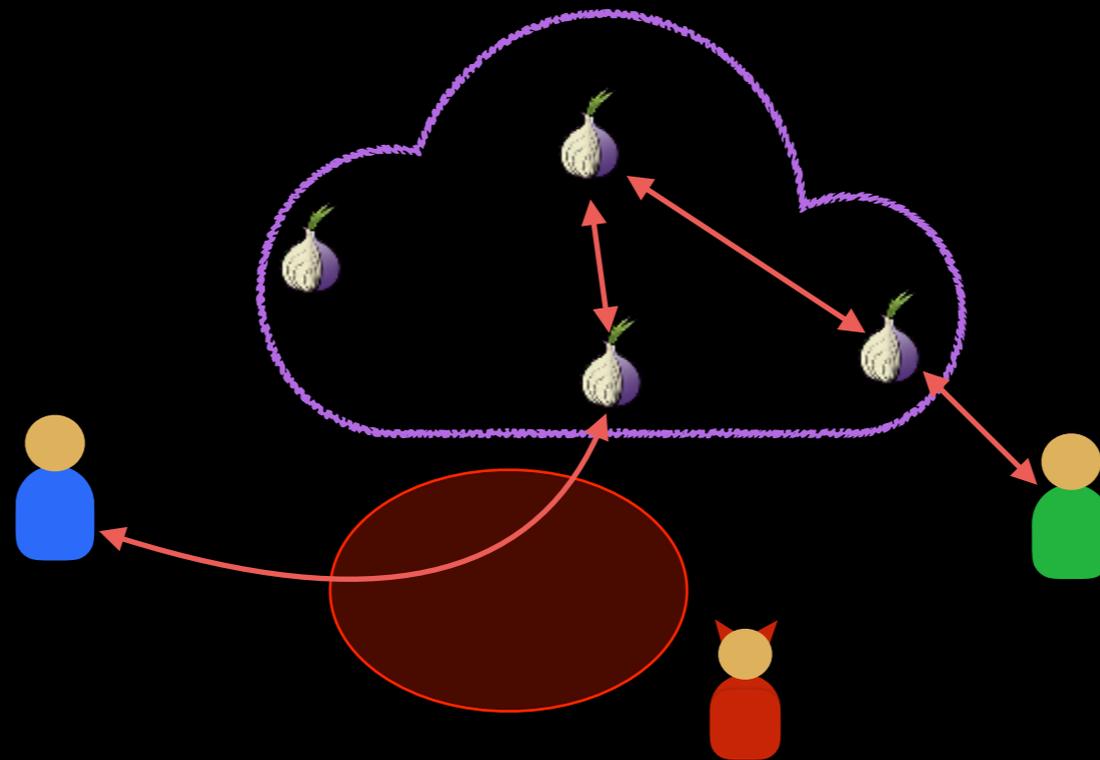
Never-twice

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

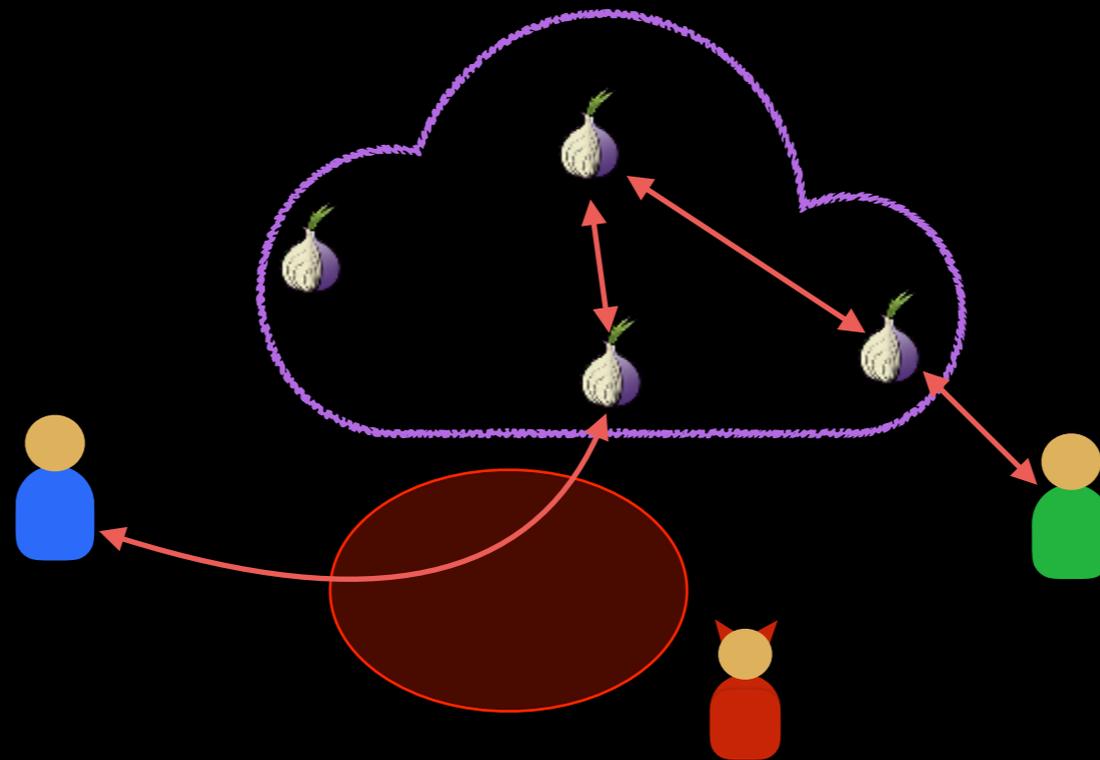


# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

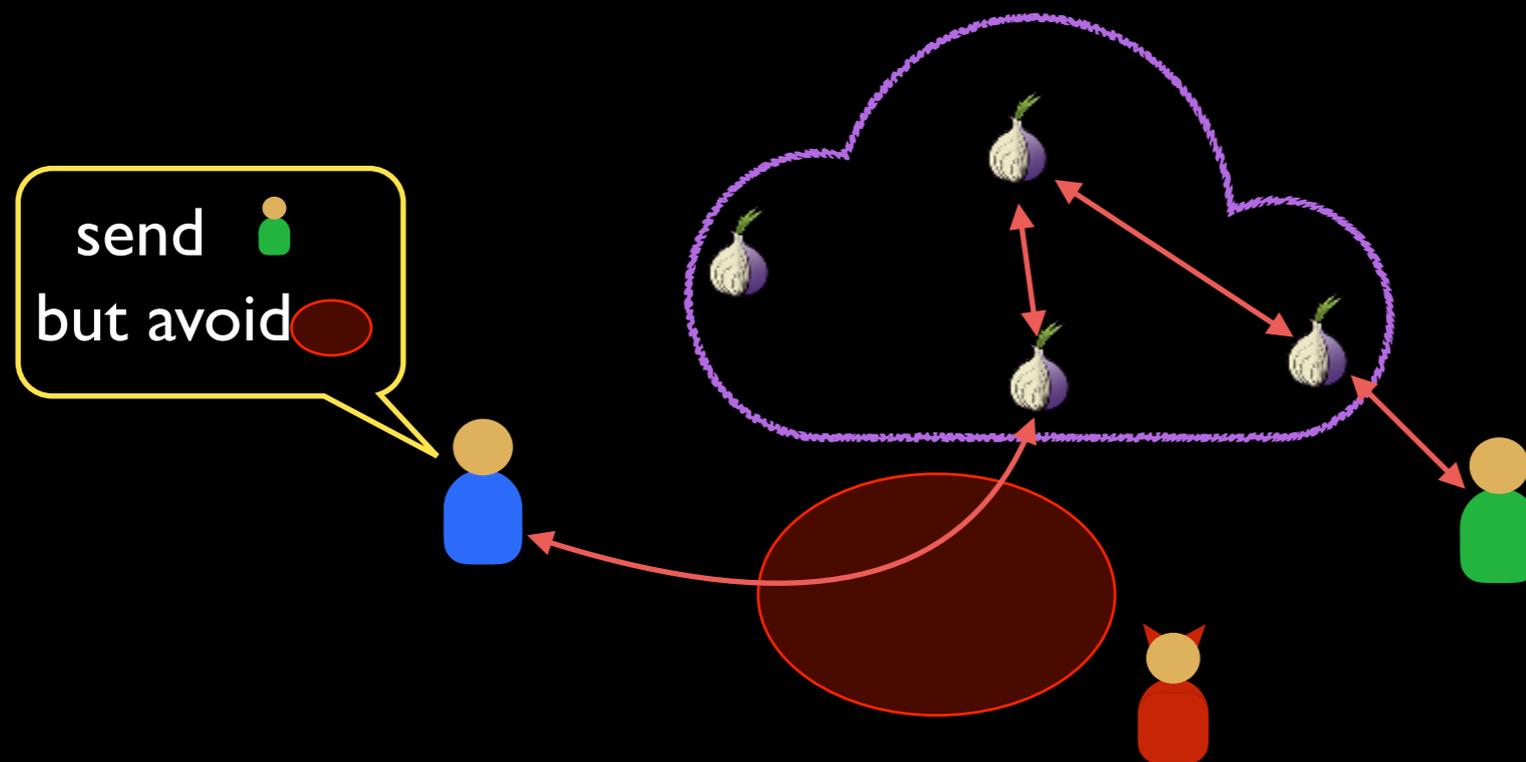


# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

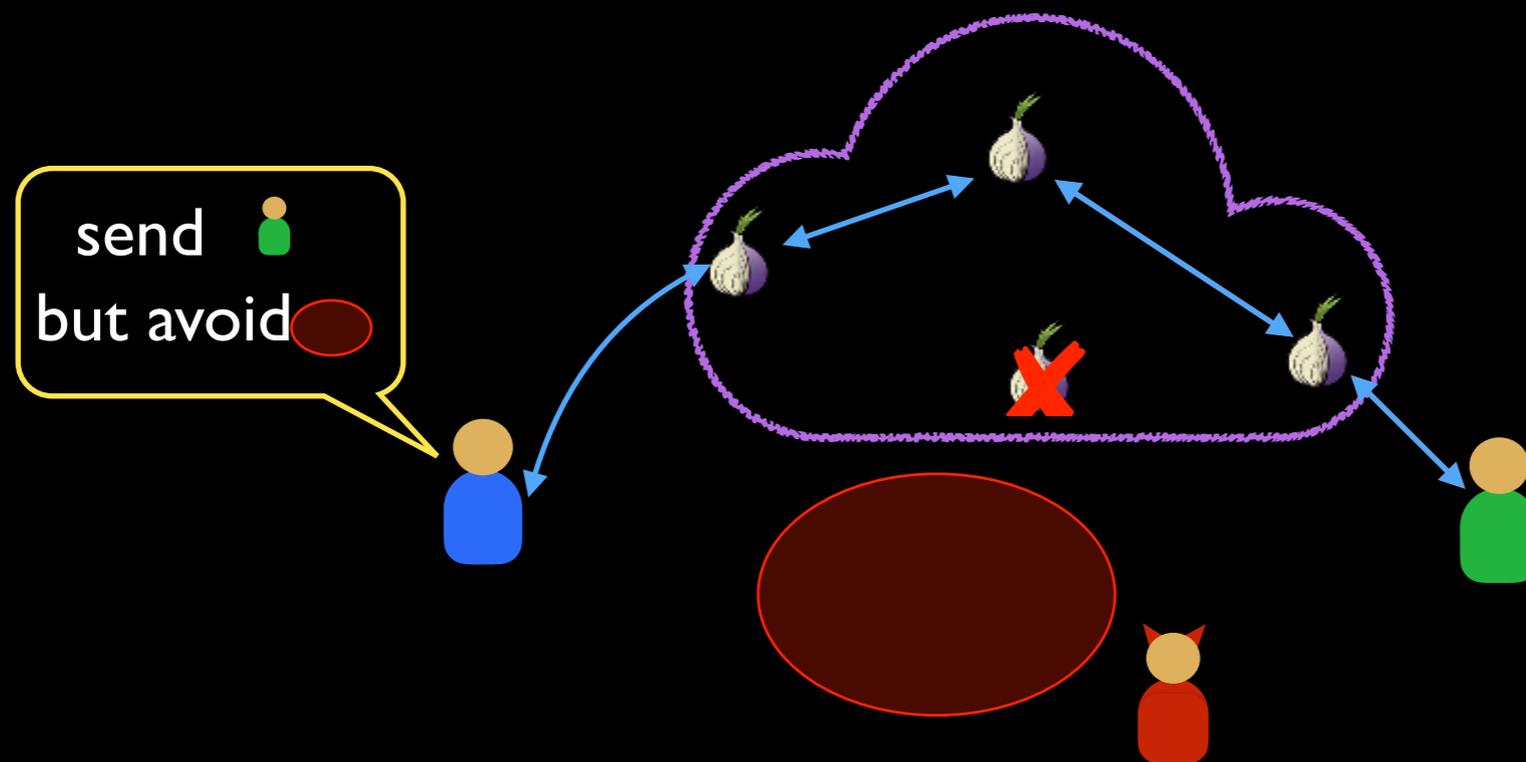


# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice



# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

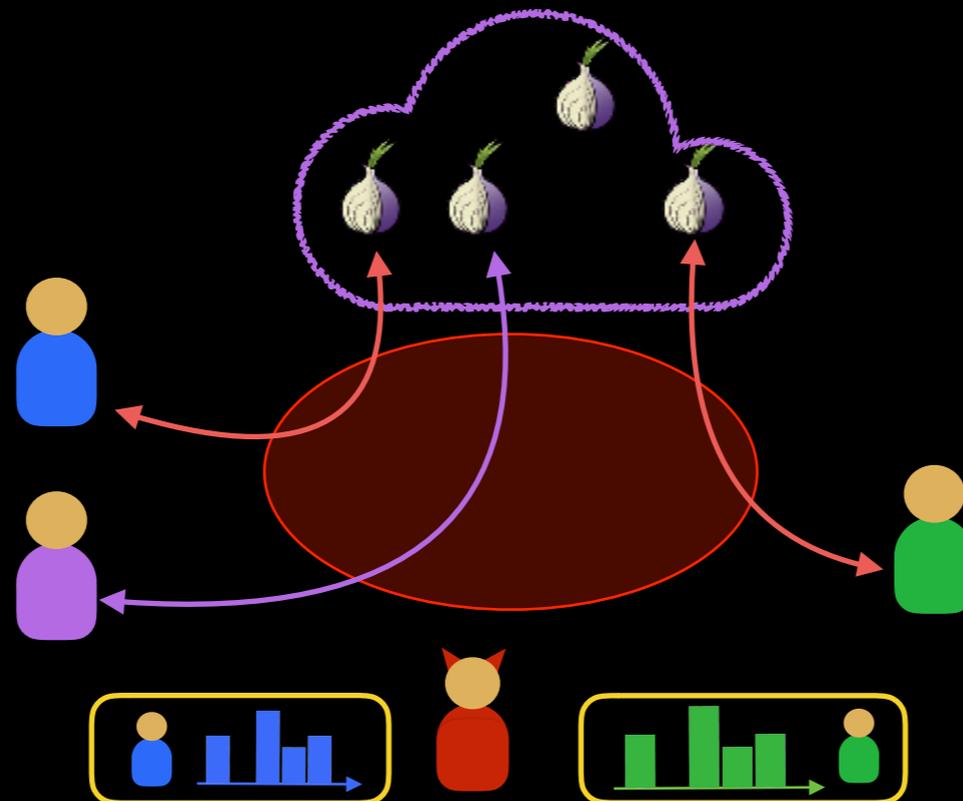
Never-twice

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

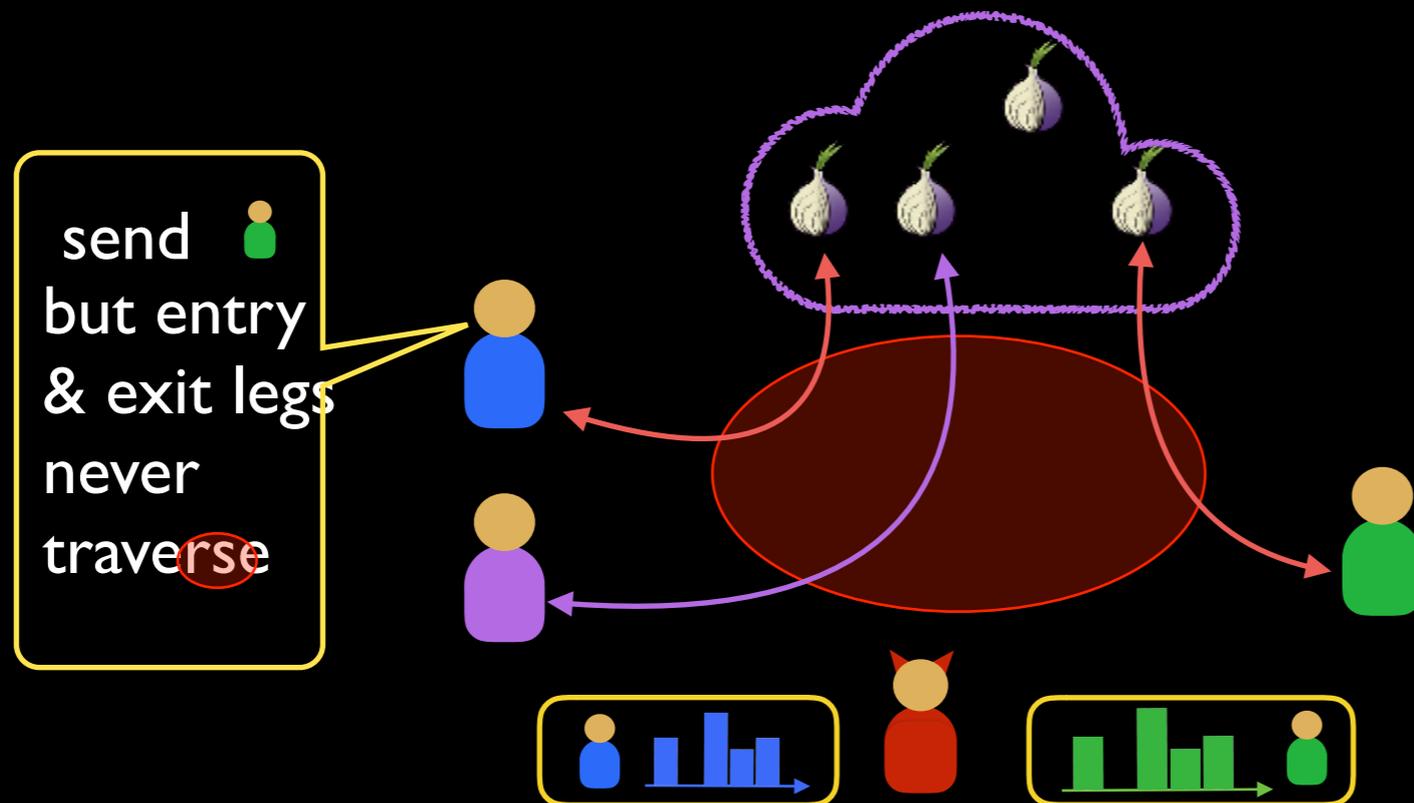


# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice

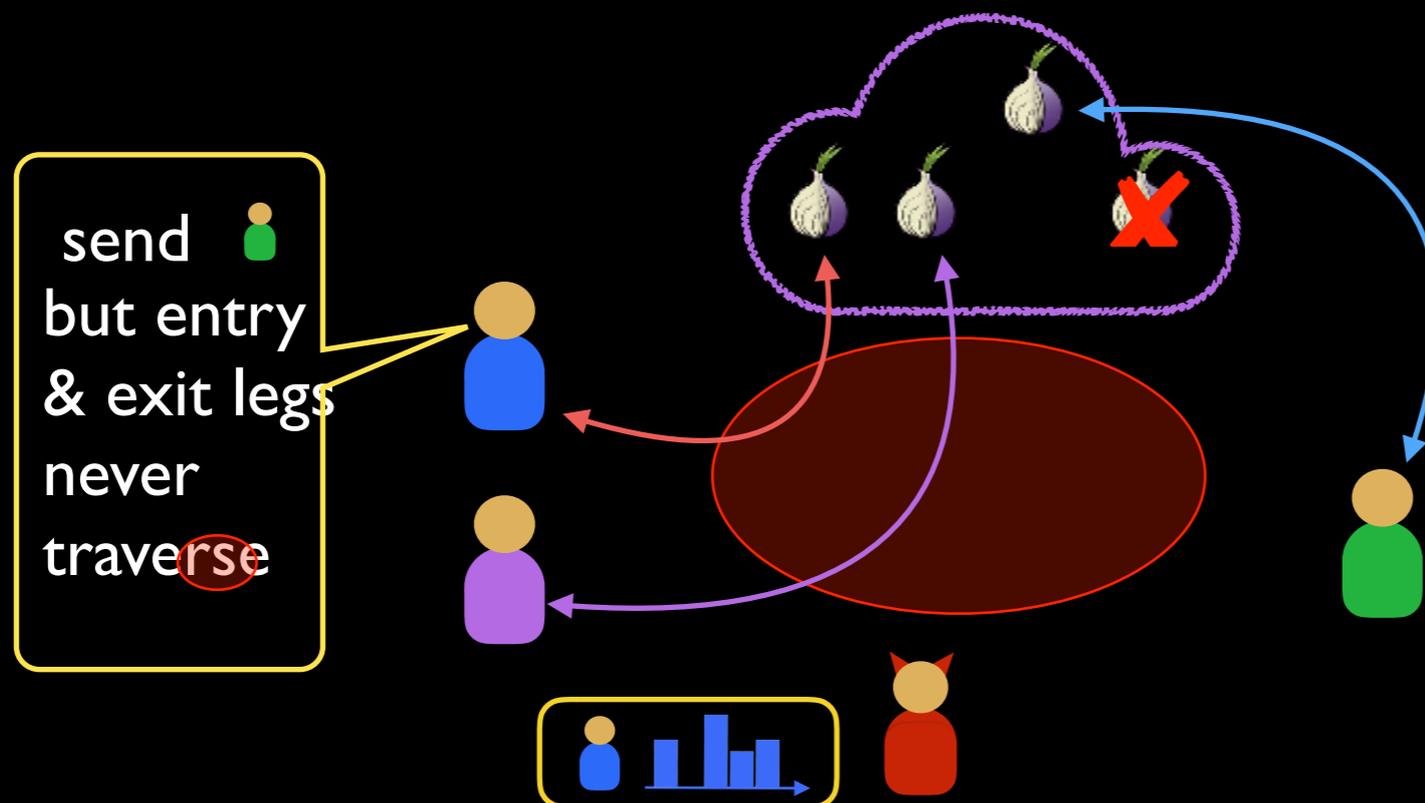


# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

Never-twice



# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry & exit legs  
never traverse

Provide per-packet  
proof of avoidance

# DeTor goals

Deployable

Allow users to avoid adversaries with smart circuits selection

Proof

Provide **proofs** of avoidance

# DeTor goals

Deployable

Allow users to avoid adversaries with smart circuits selection

# DeTor goals

Deployable

Allow users to avoid  
adversaries with smart circuits  
selection

Without having to  
know  
underlying routes

# DeTor goals

Deployable

Allow users to avoid adversaries with smart circuits selection

Without having to know underlying routes or modifications to Internet routers

# DeTor goals

Deployable

Allow users to avoid adversaries with smart circuits selection

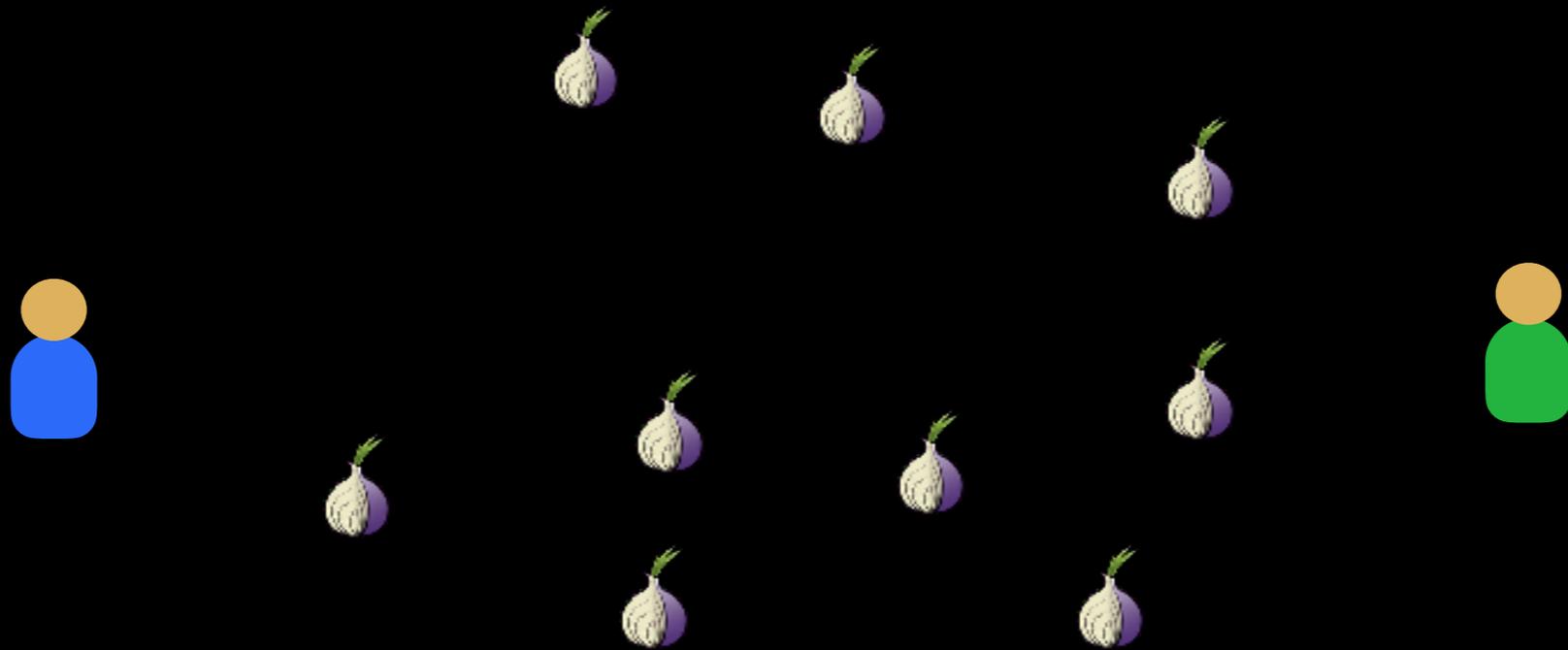
Without having to know underlying routes  
Without modifications to Internet routers  
Without changes to Tor's protocol

# DeTor goals

Proof

Provide **proofs** of avoidance

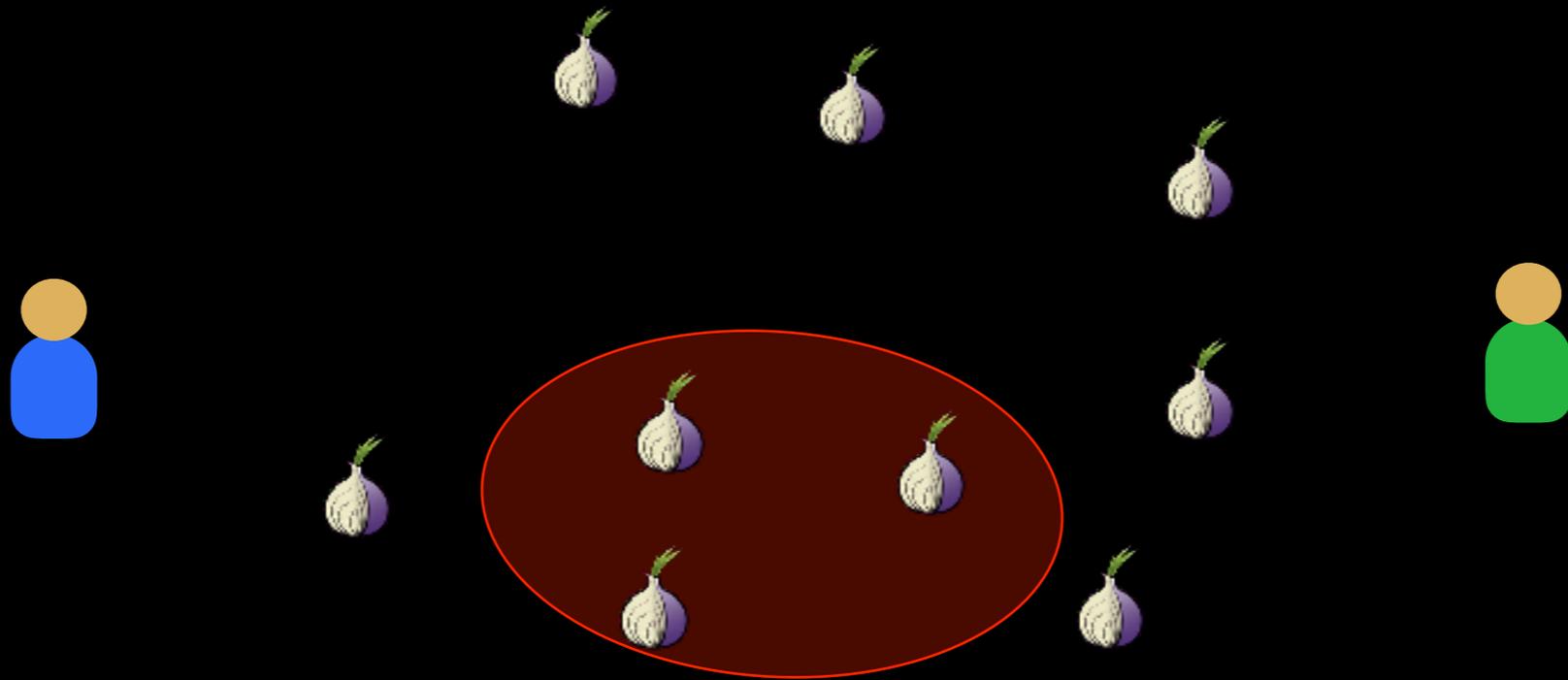
# DeTor goals



Proof

Provide **proofs** of avoidance

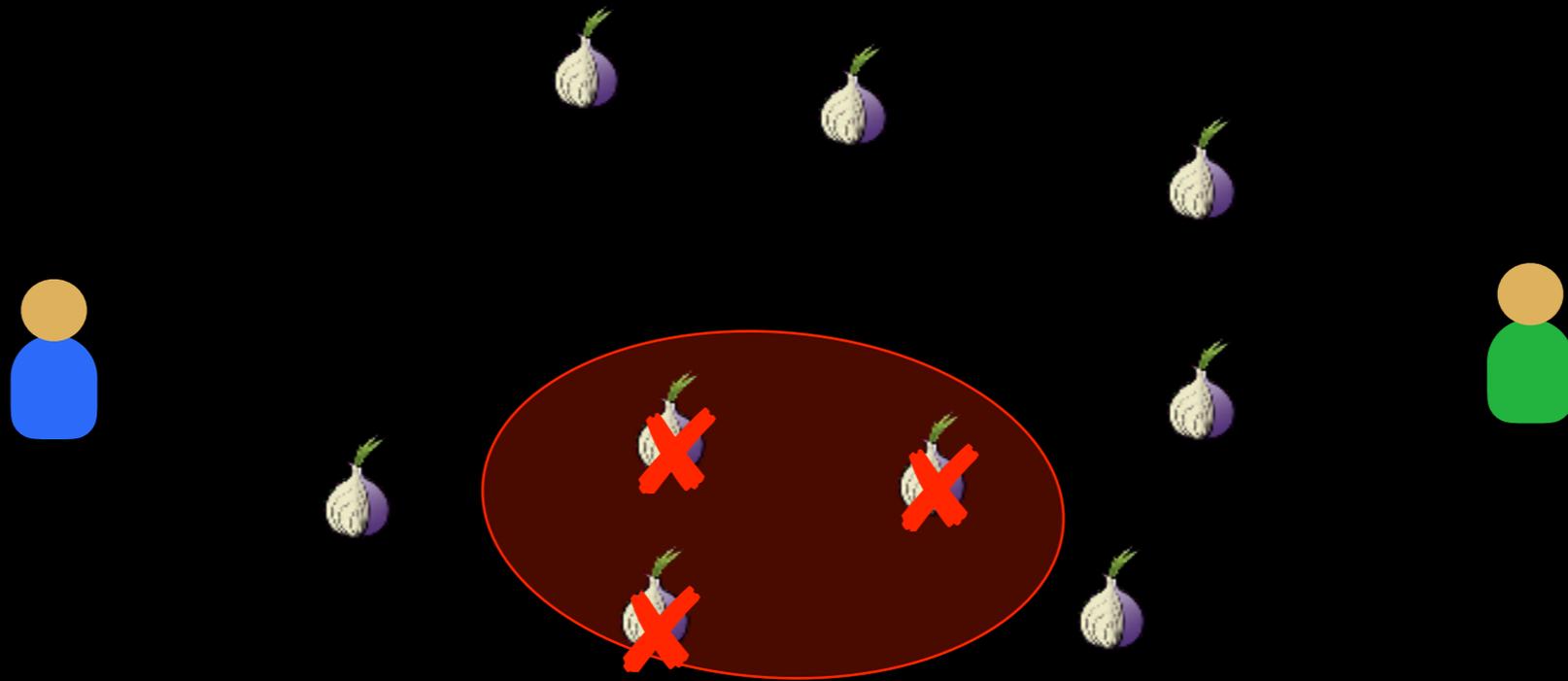
# DeTor goals



Proof

Provide **proofs** of avoidance

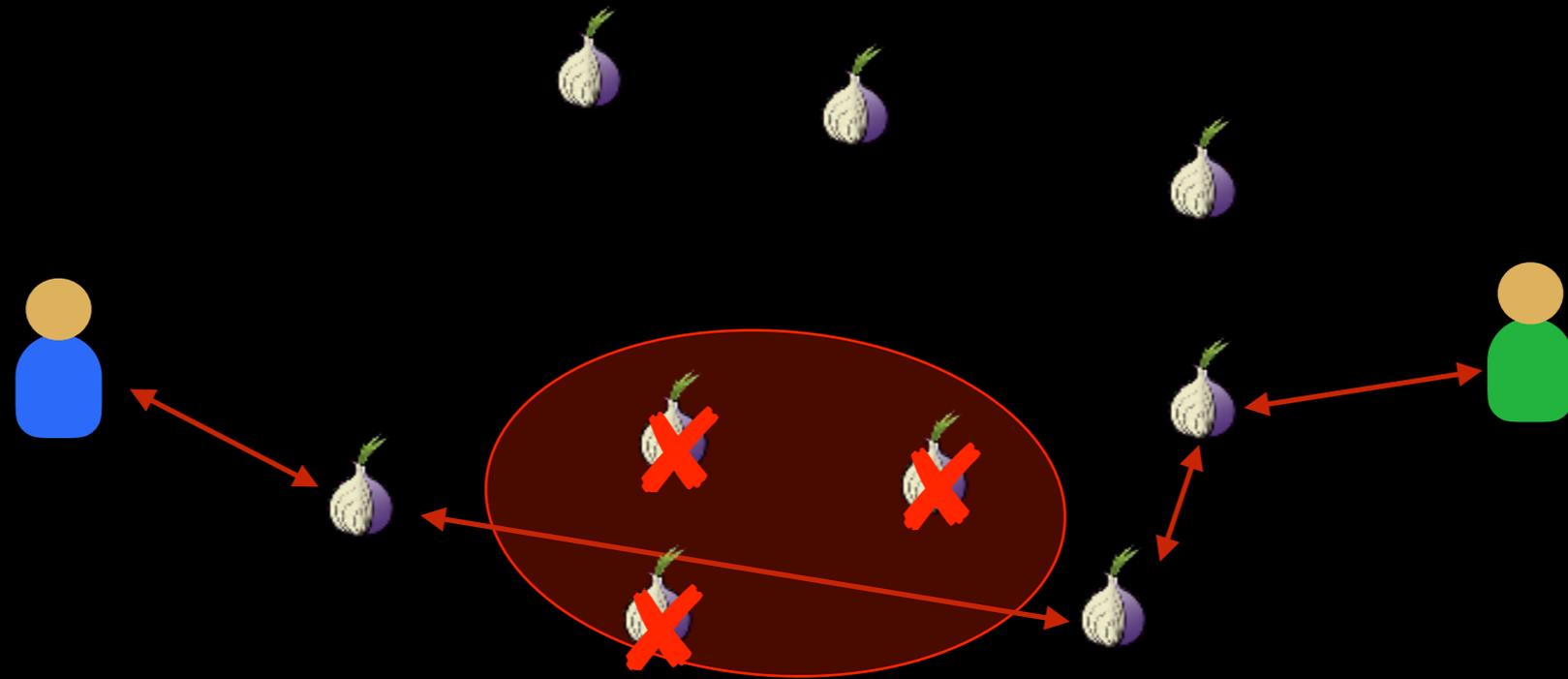
# DeTor goals



Proof

Provide **proofs** of avoidance

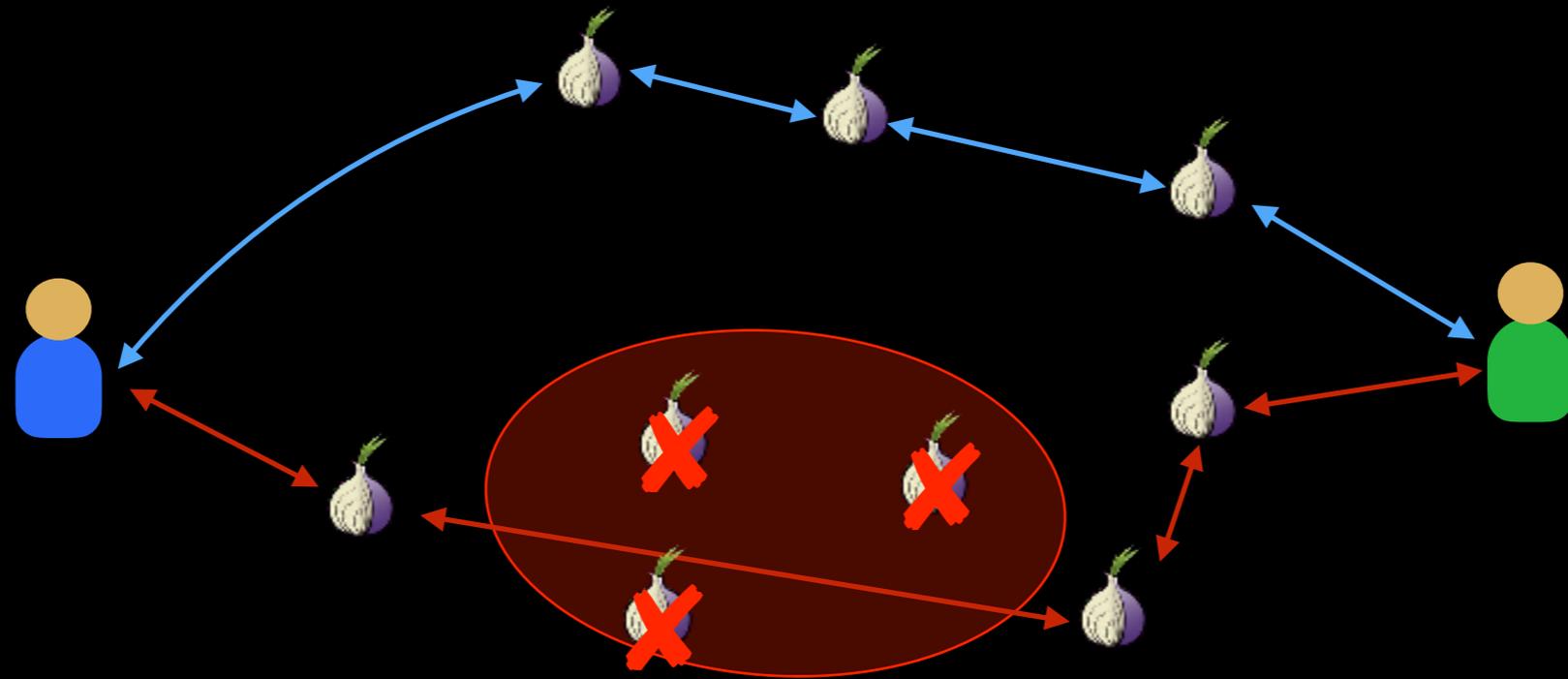
# DeTor goals



Proof

Provide **proofs** of avoidance

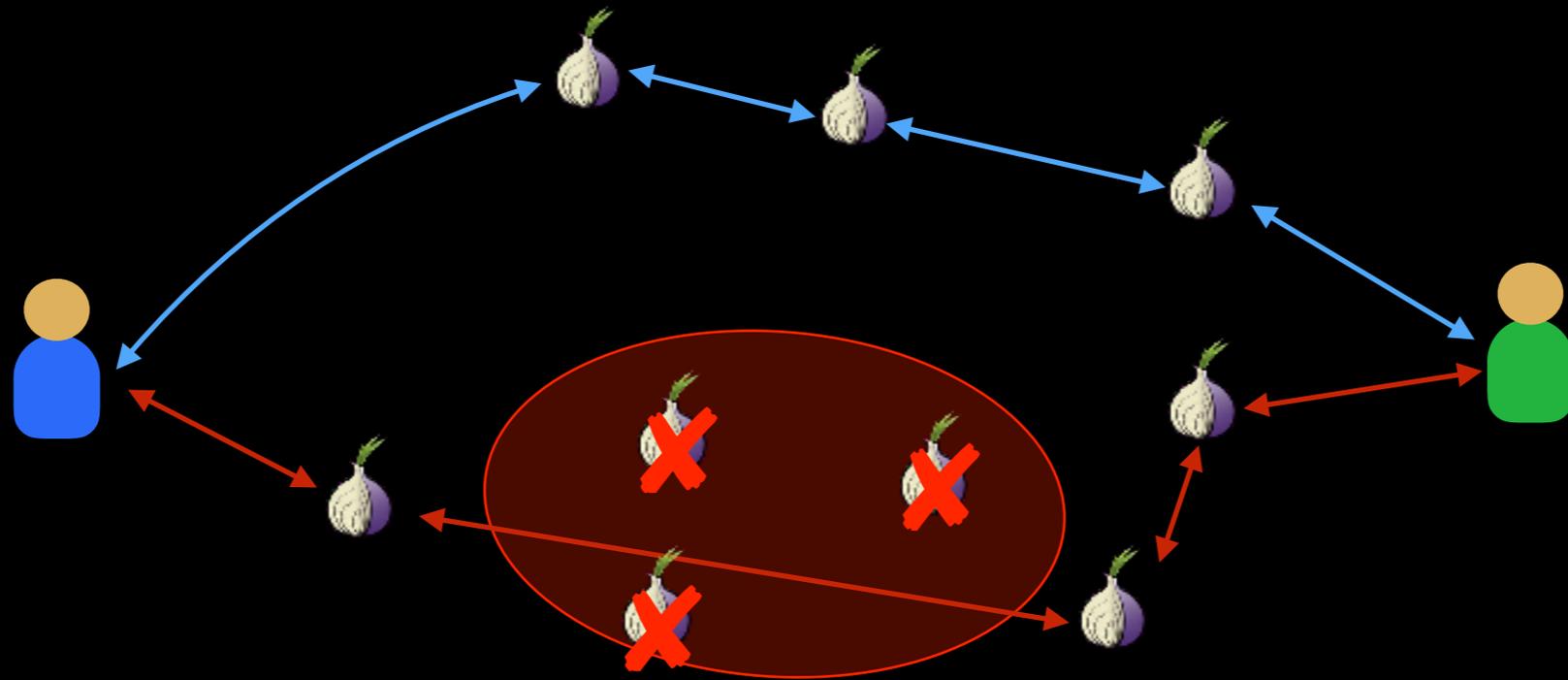
# DeTor goals



Proof

Provide **proofs** of avoidance

# DeTor goals



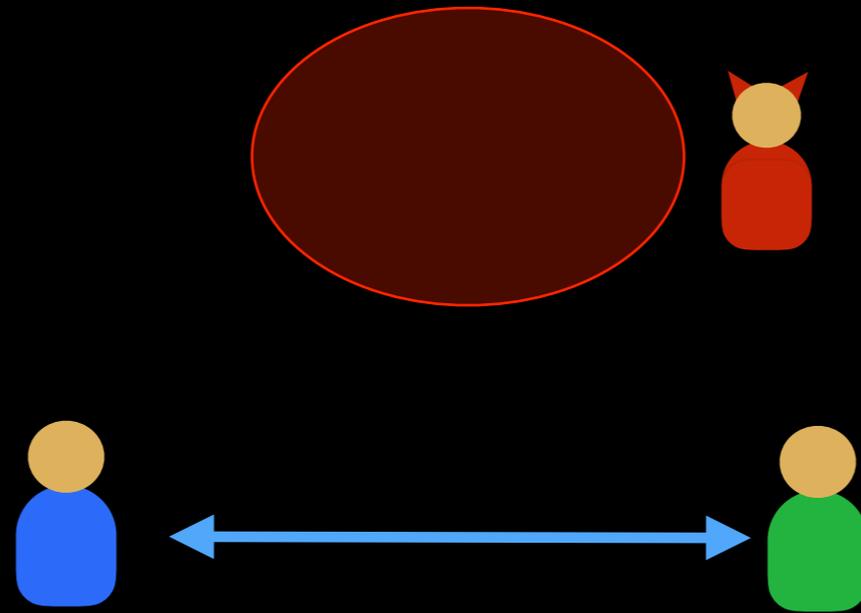
Proof

Provide proofs of avoidance

Measurement of roundtrip time

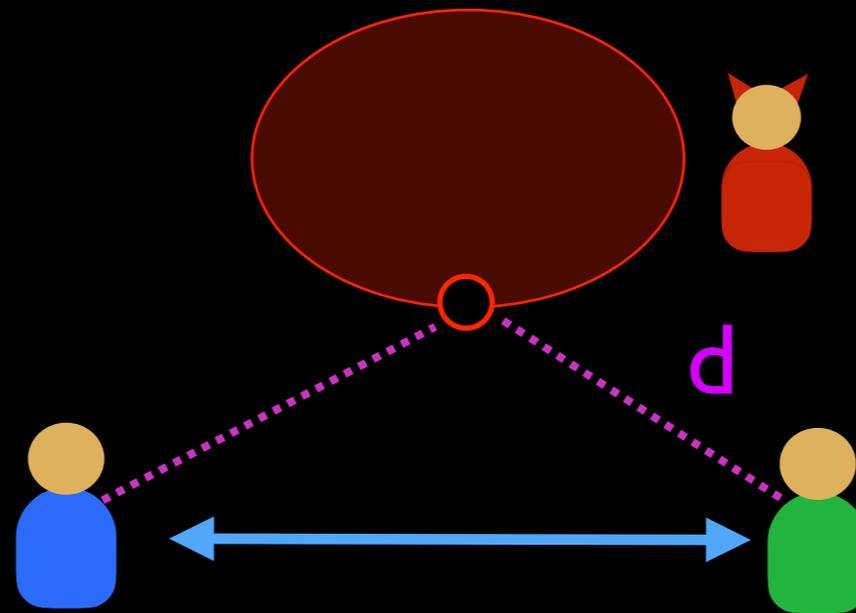
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



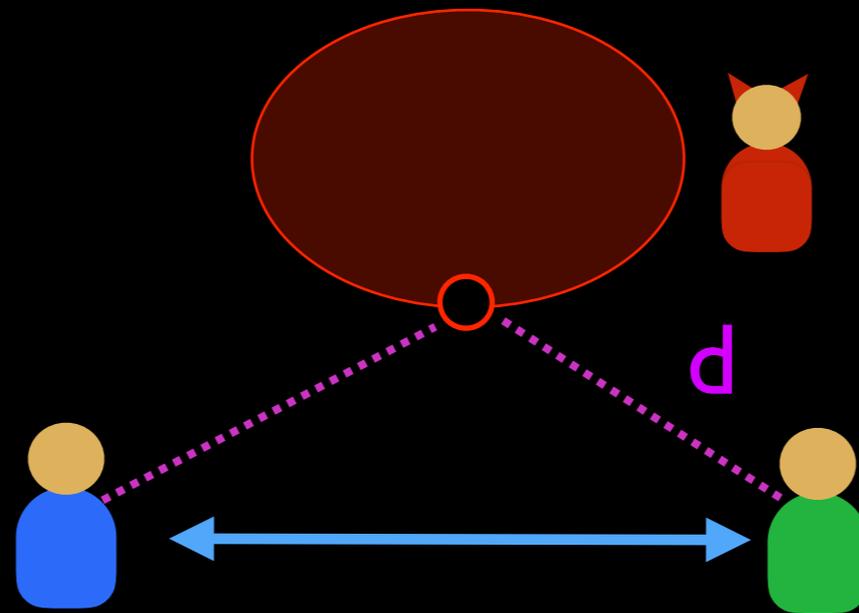
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



# Provable avoidance

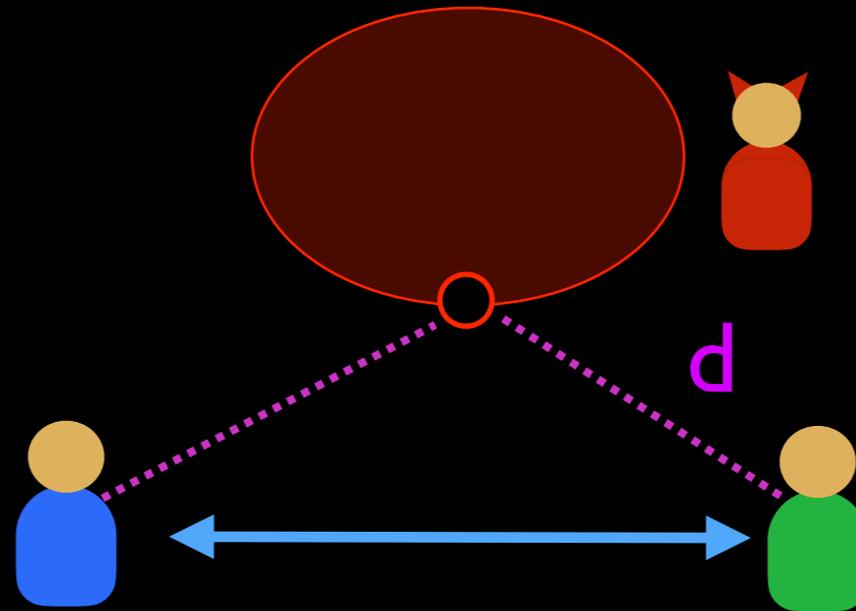
Alibi Routing by Levin et al. in SIGCOMM 2015



The shortest possible RTT  
thru  to  =  $2d/c$

# Provable avoidance

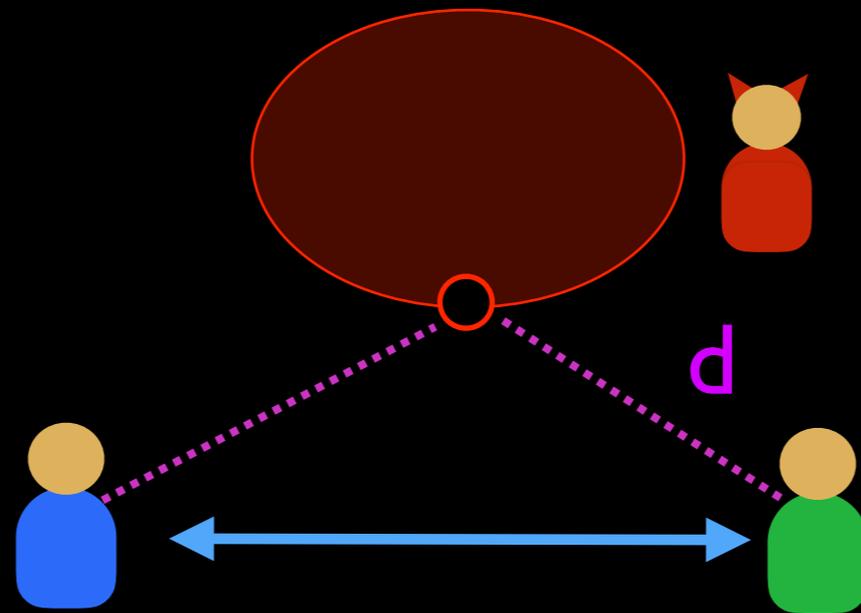
Alibi Routing by Levin et al. in SIGCOMM 2015



Measured RTT  $\ll$  The shortest possible RTT =  $2d/c$   
thru  to 

# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015

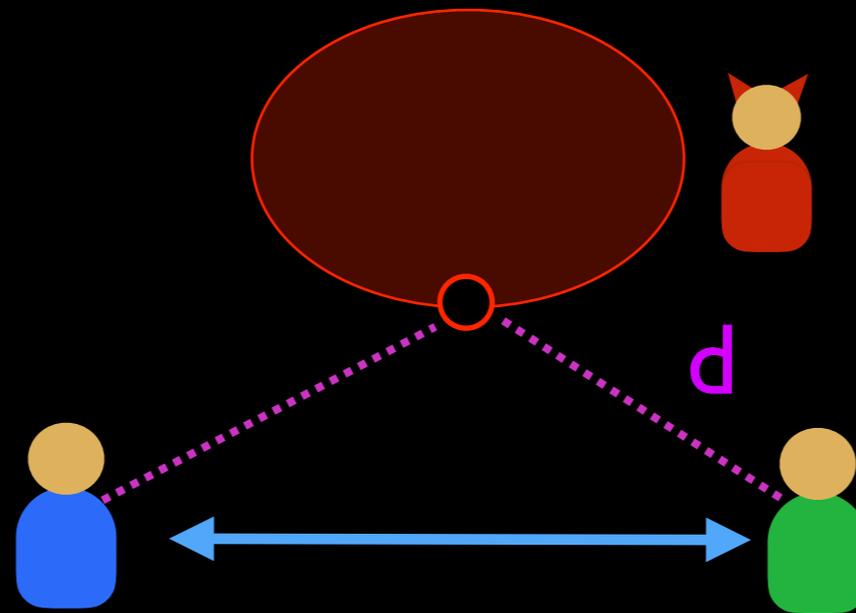


Measured RTT  $\ll$  The shortest possible RTT thru  to  =  $2d/c$

$\Rightarrow$  The packet could not have traversed   to

# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



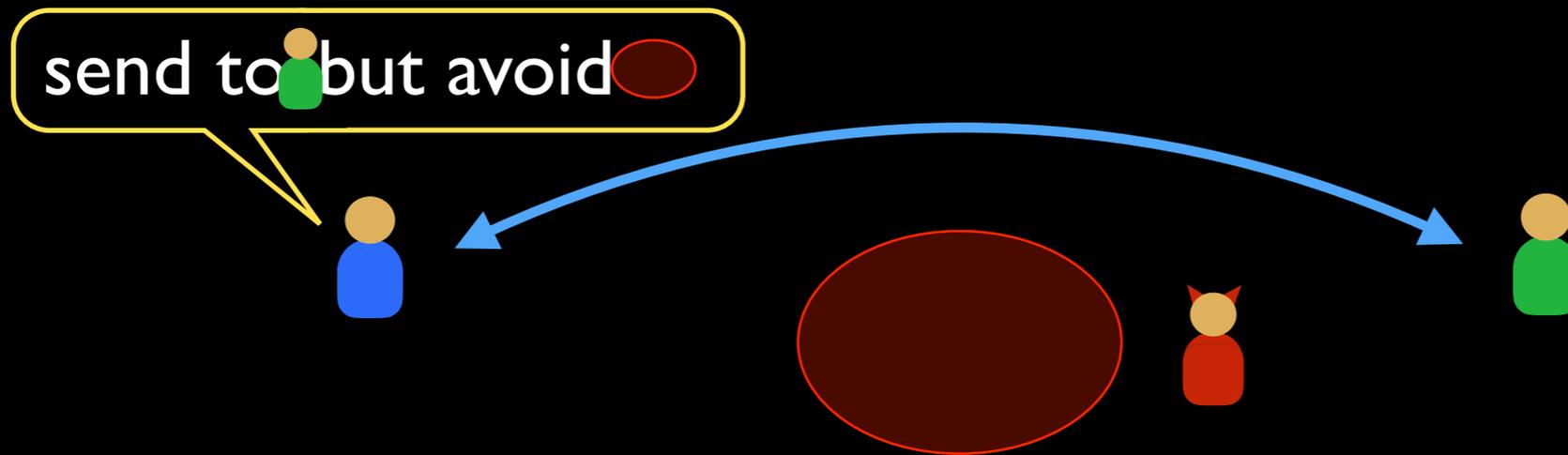
Measured RTT  $\ll$  The shortest possible RTT thru  to  =  $2d/c$

$\Rightarrow$  The packet could not have traversed   to

**Alibi**

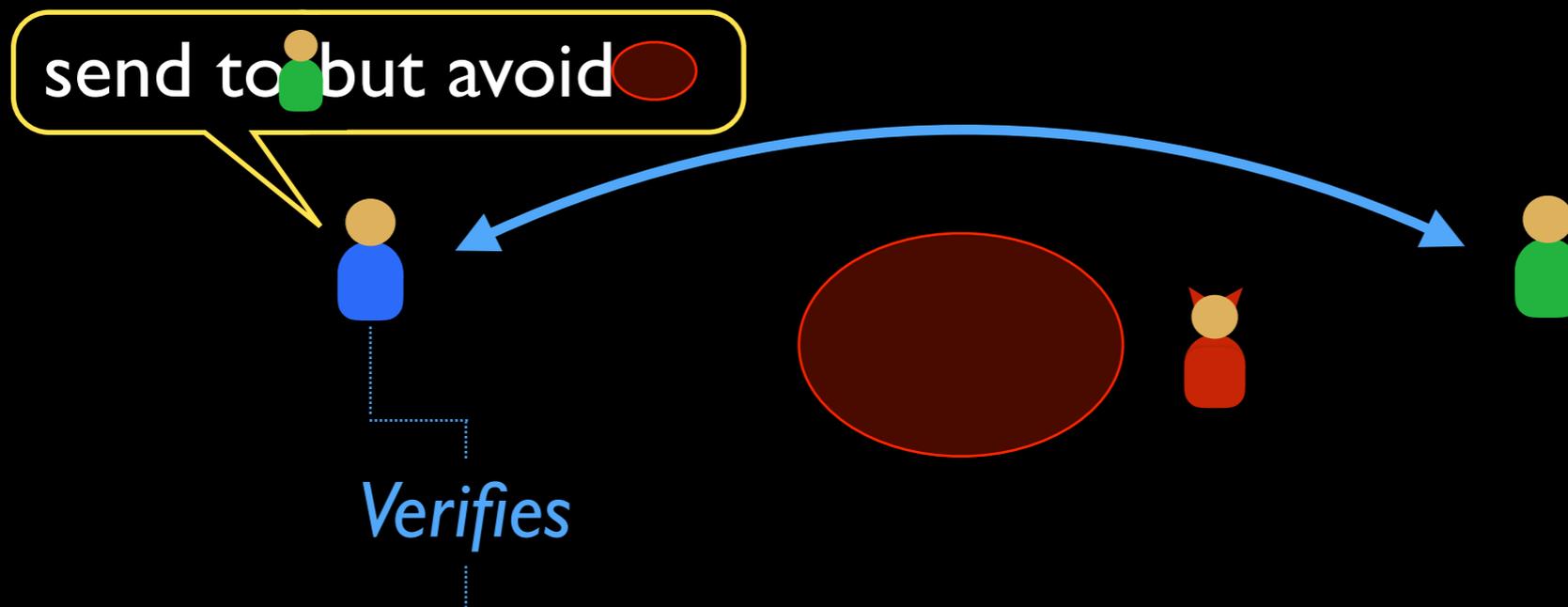
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



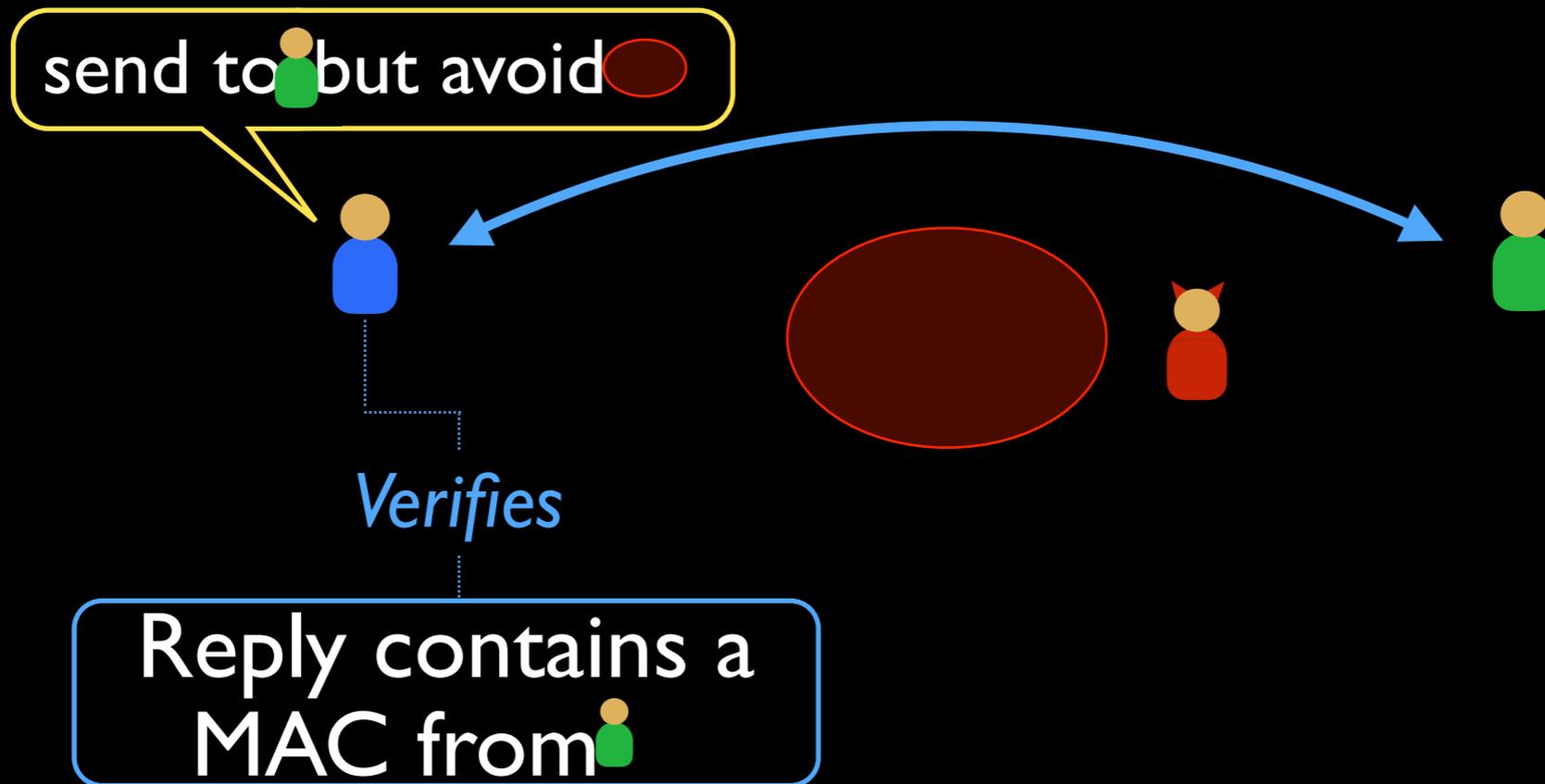
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



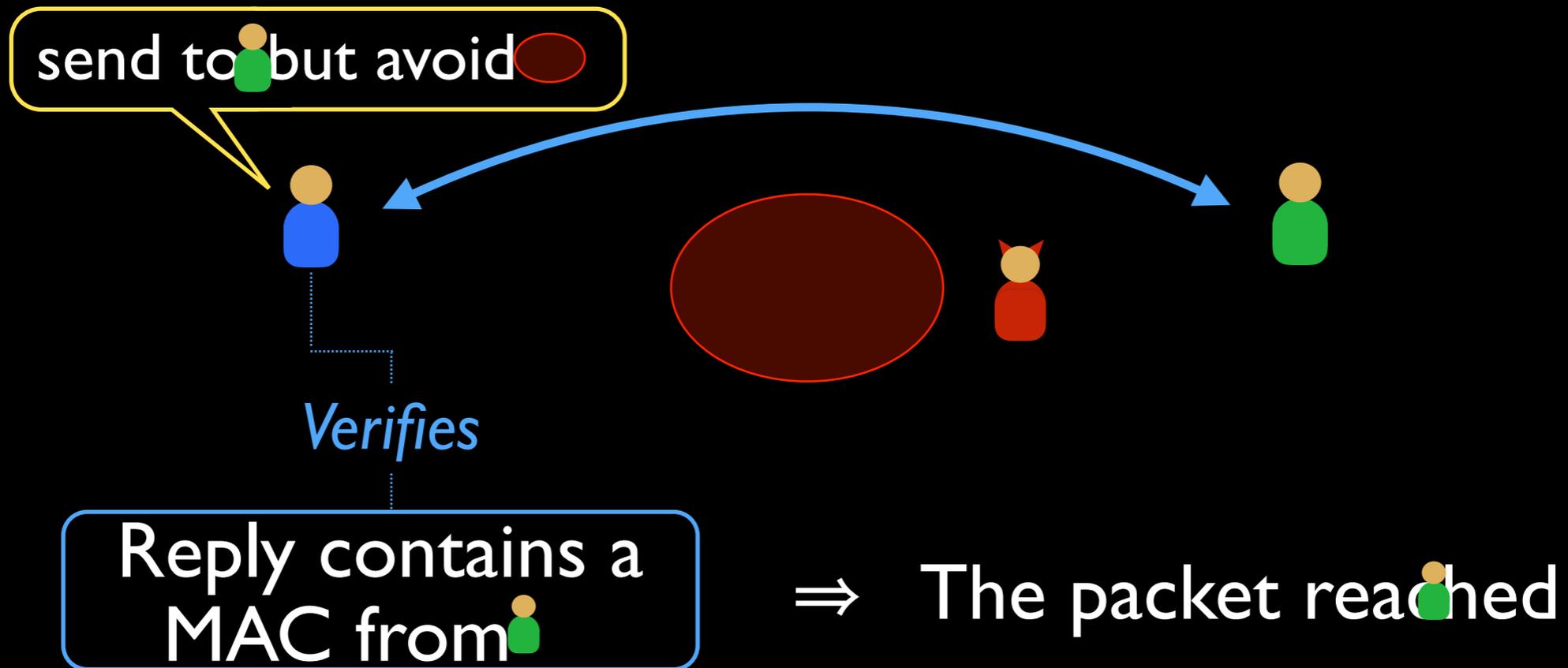
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



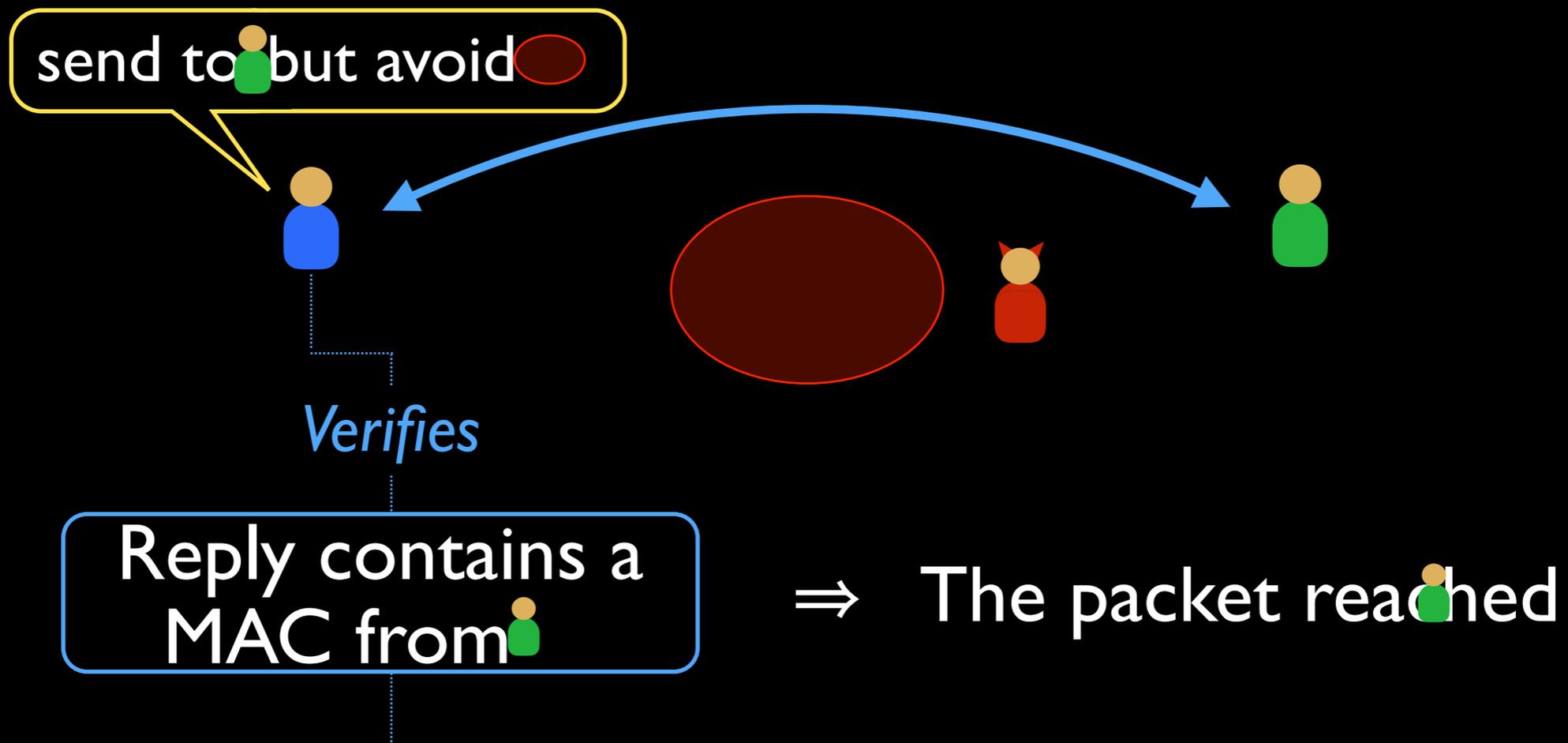
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



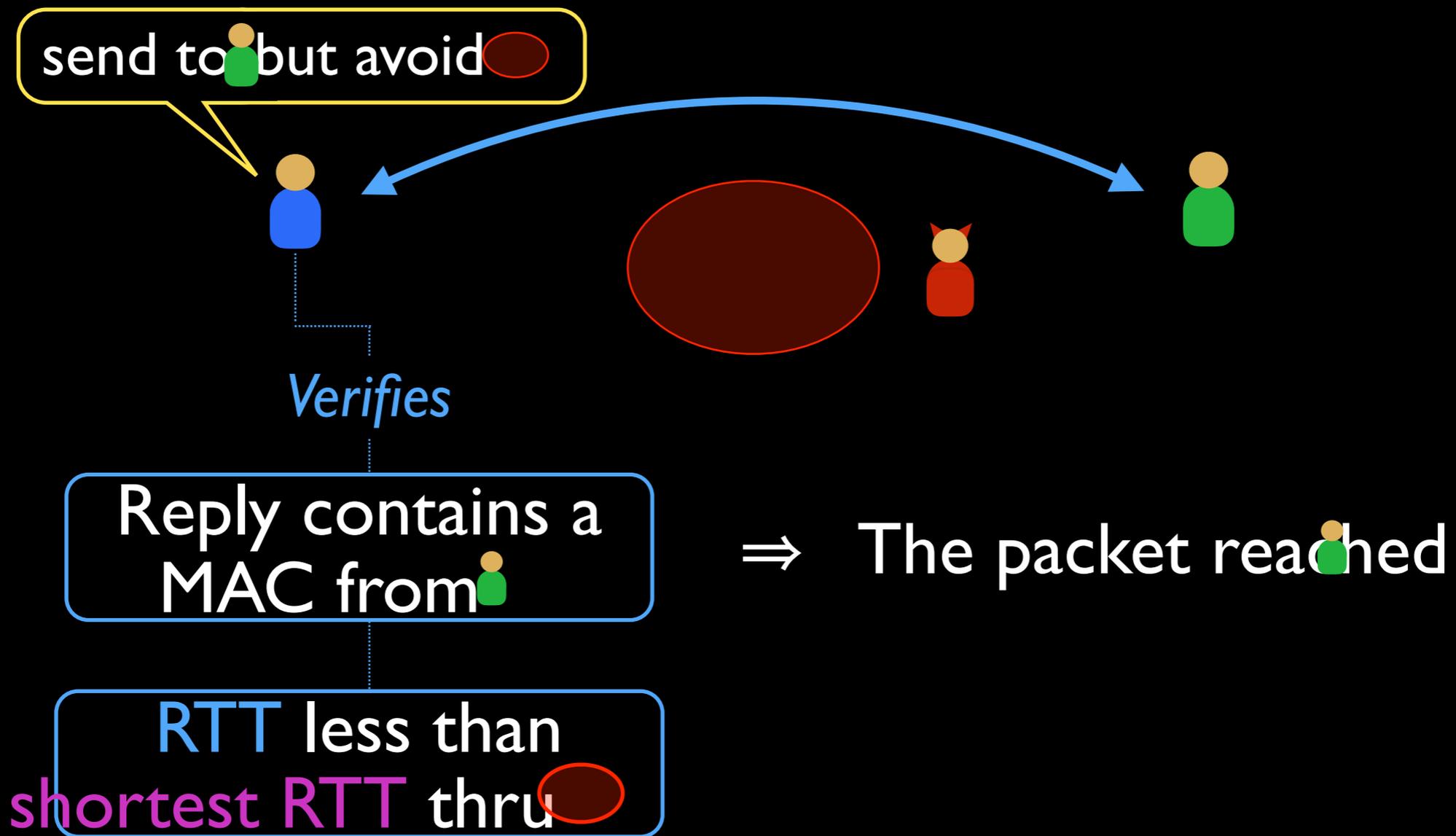
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



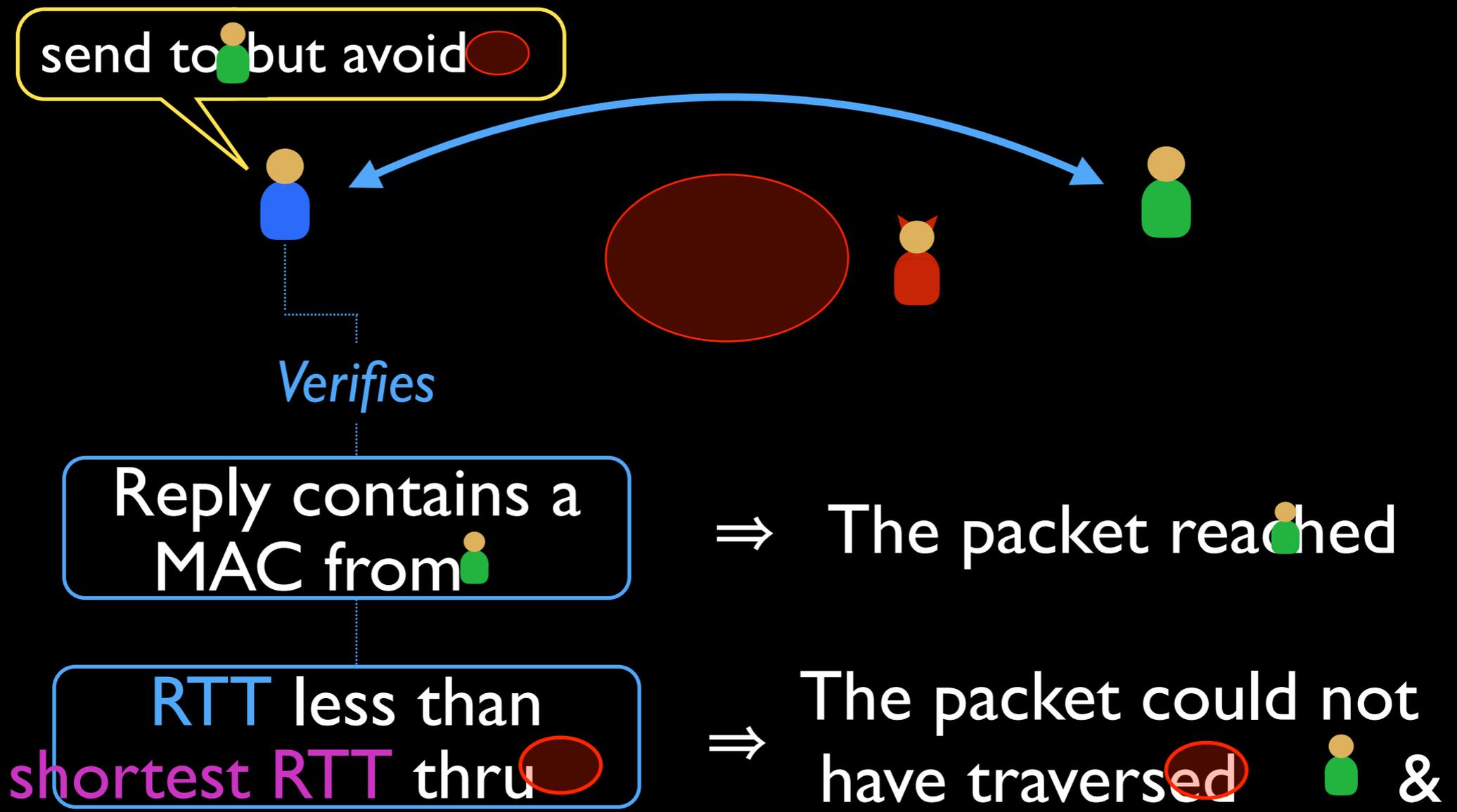
# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



# Provable avoidance

Alibi Routing by Levin et al. in SIGCOMM 2015



# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry&exit legs  
never traverse

Provide per-packet  
proof of avoidance

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

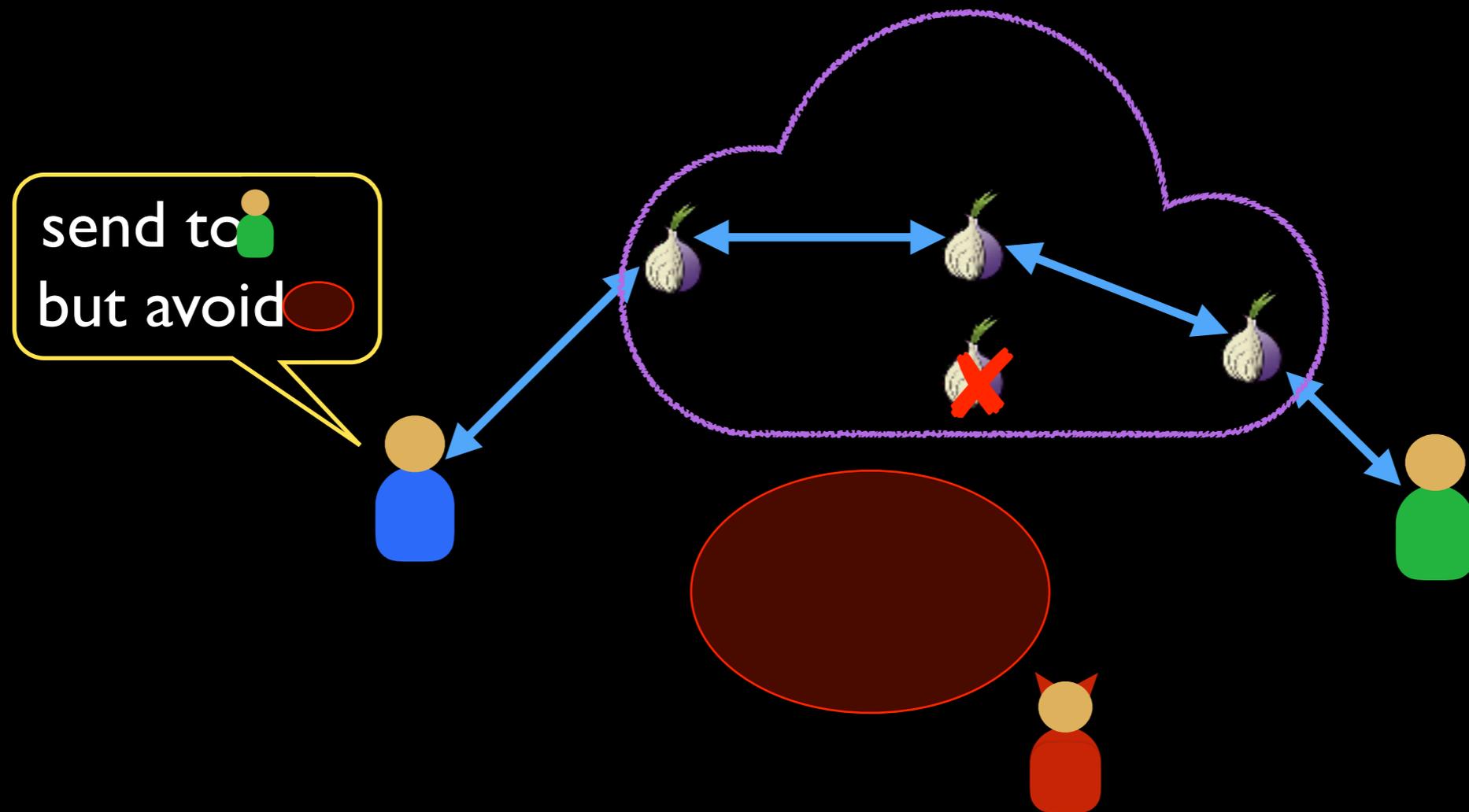
Never-twice

entry&exit legs  
never traverse

Provide per-packet  
proof of avoidance

# DeTor: never-once avoidance

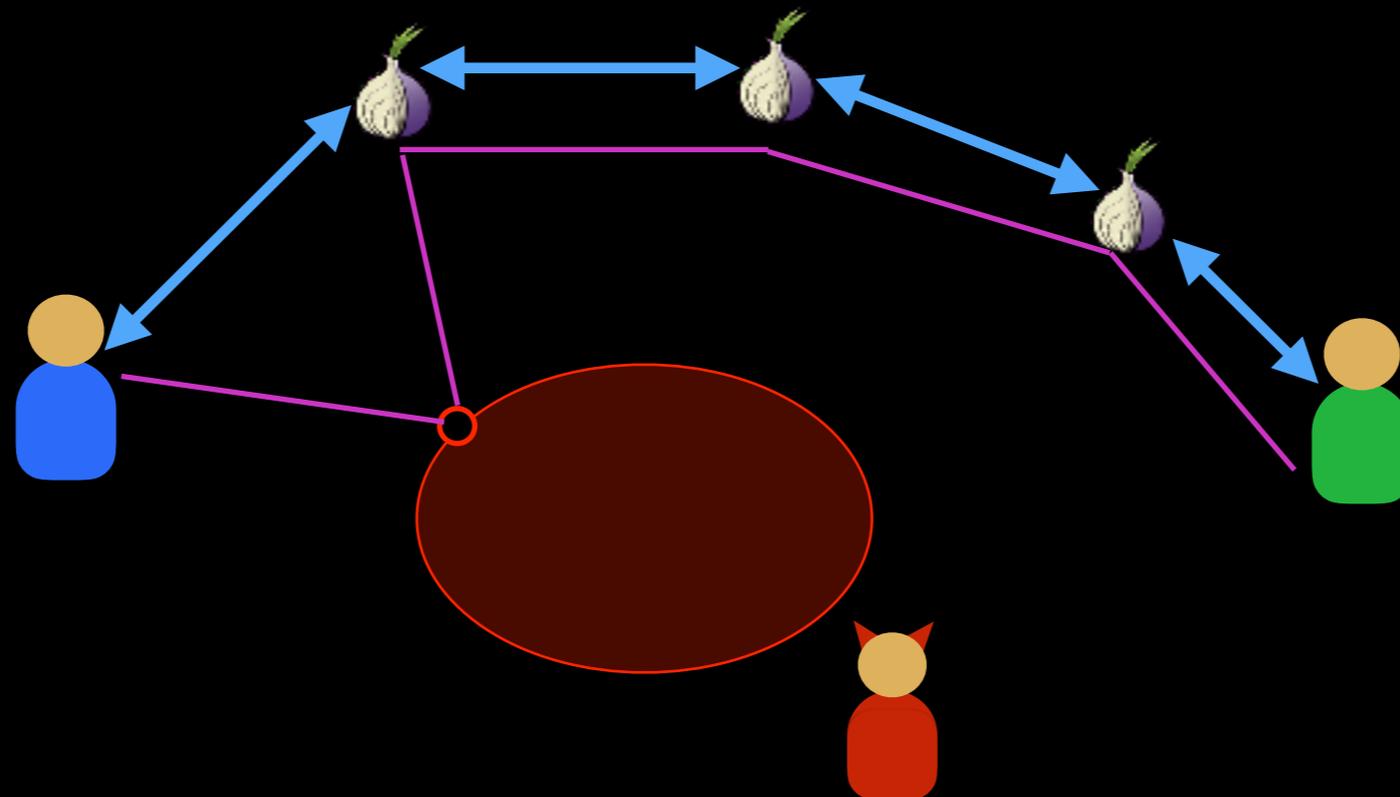
Avoid user specified geographic regions



# DeTor: never-once avoidance

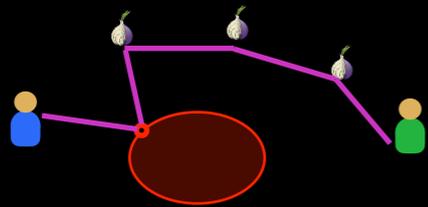
The shortest possible RTT through  and  to 

shortest distance =  $d_1$

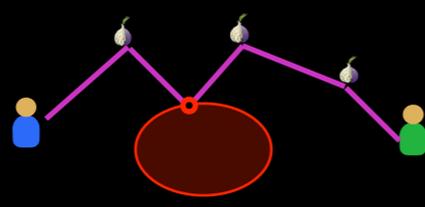


# DeTor: never-once avoidance

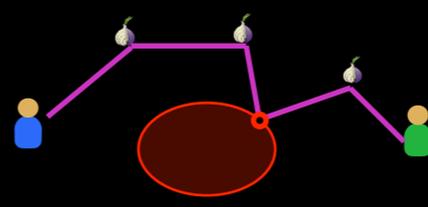
The shortest possible RTT through  and  to



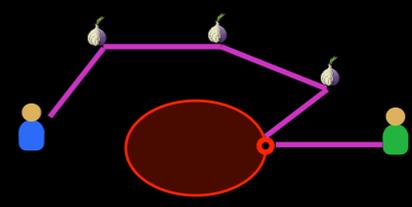
$d_1$



$d_2$



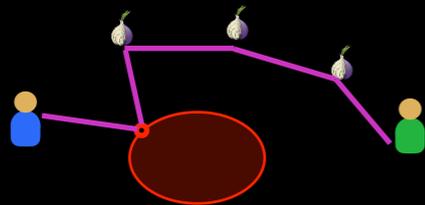
$d_3$



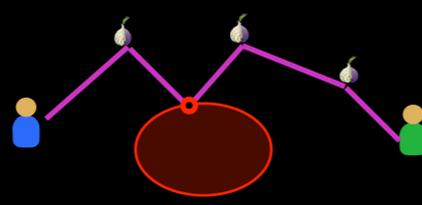
$d_4$

# DeTor: never-once avoidance

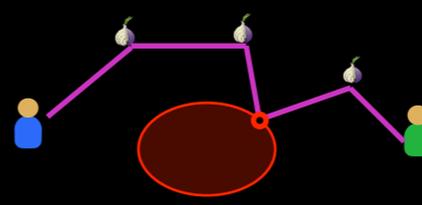
The shortest possible RTT thru  and  to 



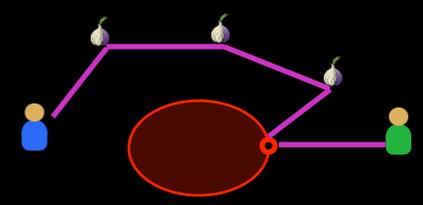
$d_1$



$d_2$



$d_3$

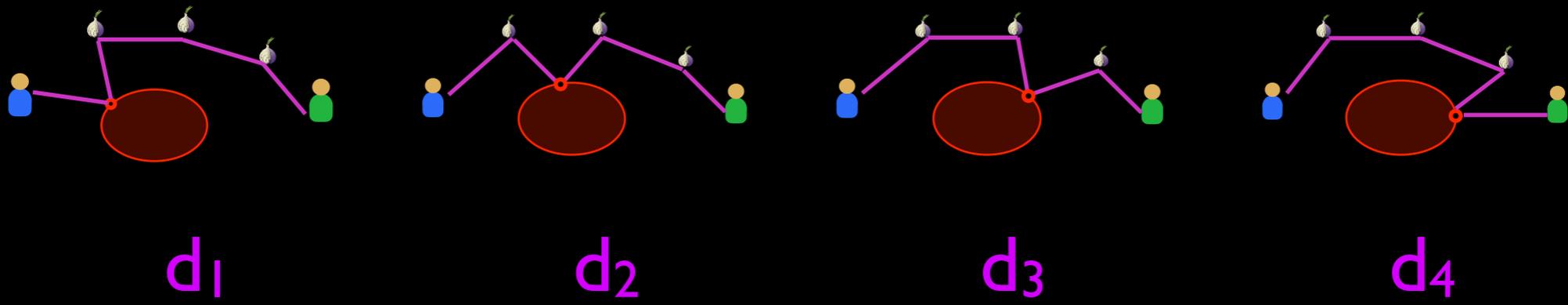


$d_4$

The shortest possible RTT thru  and  to  =  $2 \min\{d_i\} / c$

# DeTor: never-once avoidance

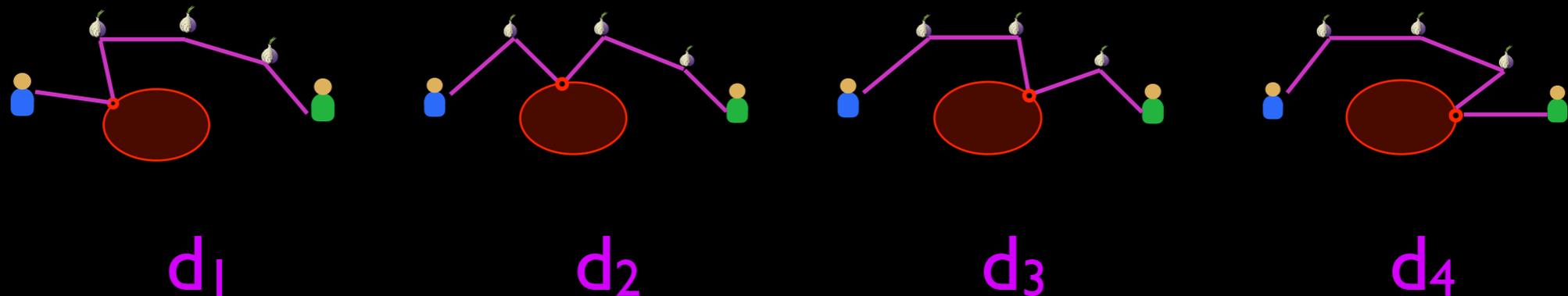
The shortest possible RTT thru  and  to 



Measured RTT  $\ll$  The shortest possible RTT thru  and  to  =  $2 \min\{d_i\} / c$

# DeTor: never-once avoidance

The shortest possible RTT thru  and  to 

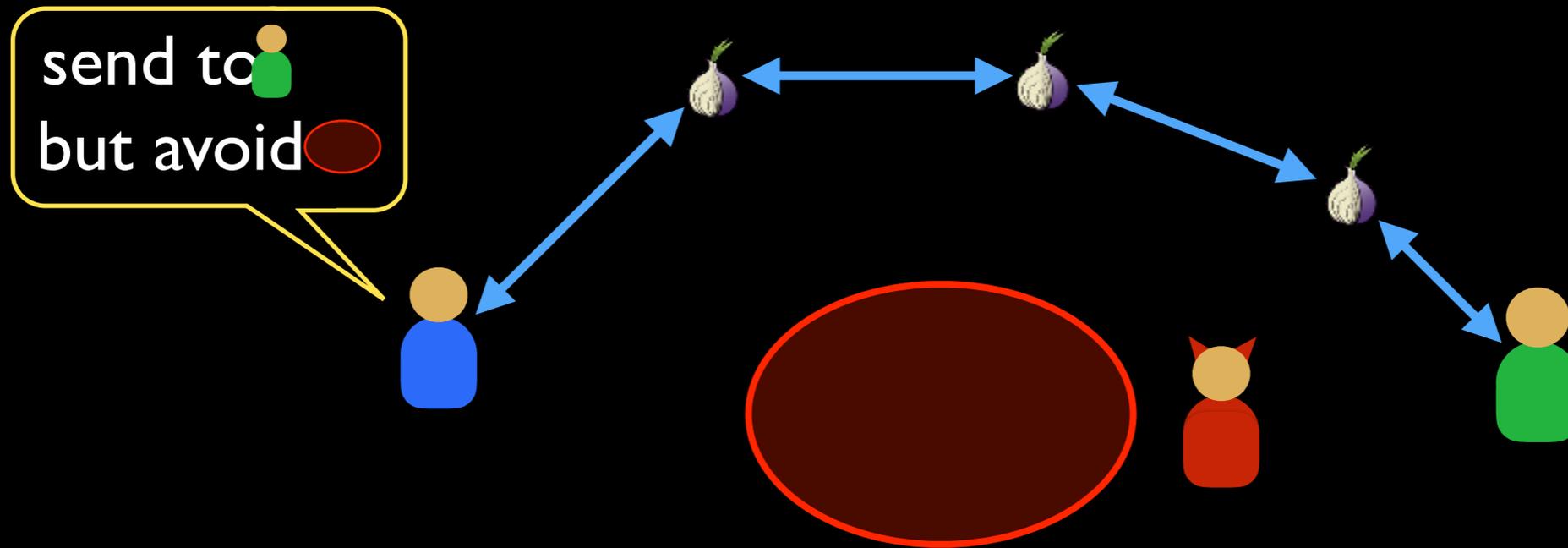


Measured RTT  $\ll$  The shortest possible RTT thru  and  to  =  $2 \min\{d_i\} / c$

$\Rightarrow$  The packet could not have traversed  to 

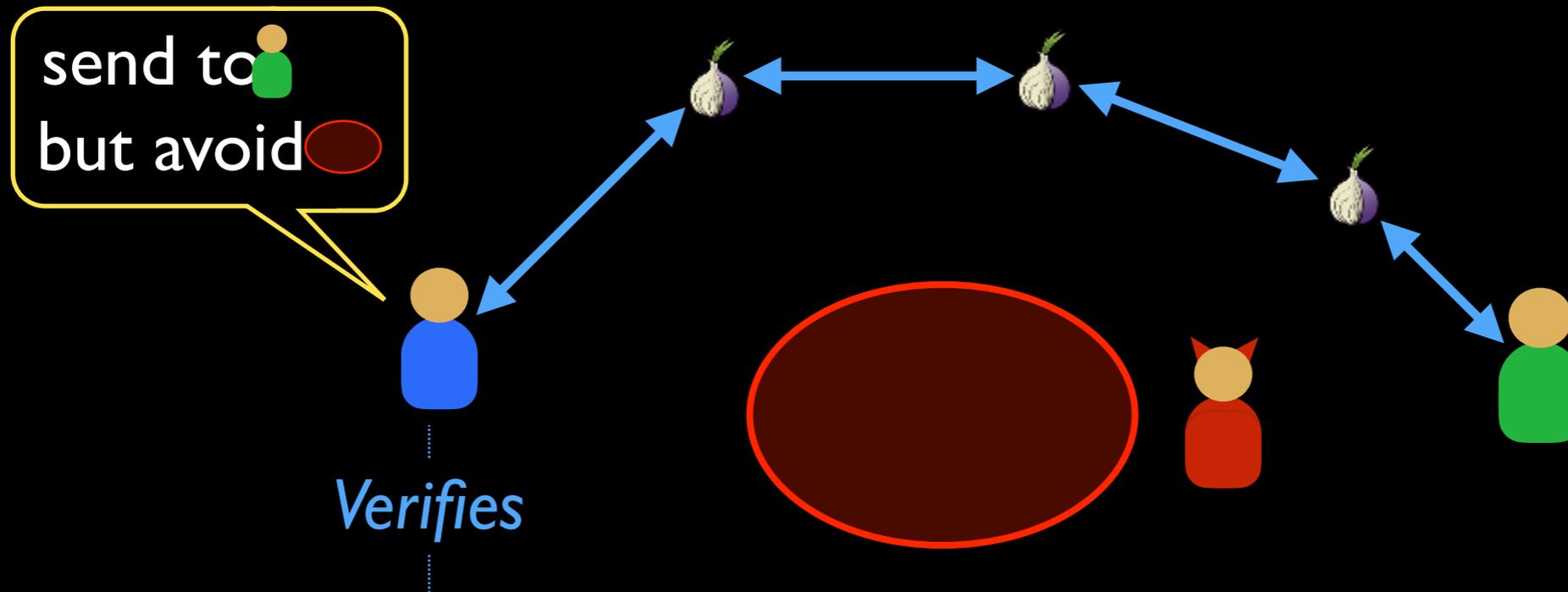
# DeTor: never-once avoidance

Achieving provable avoidance



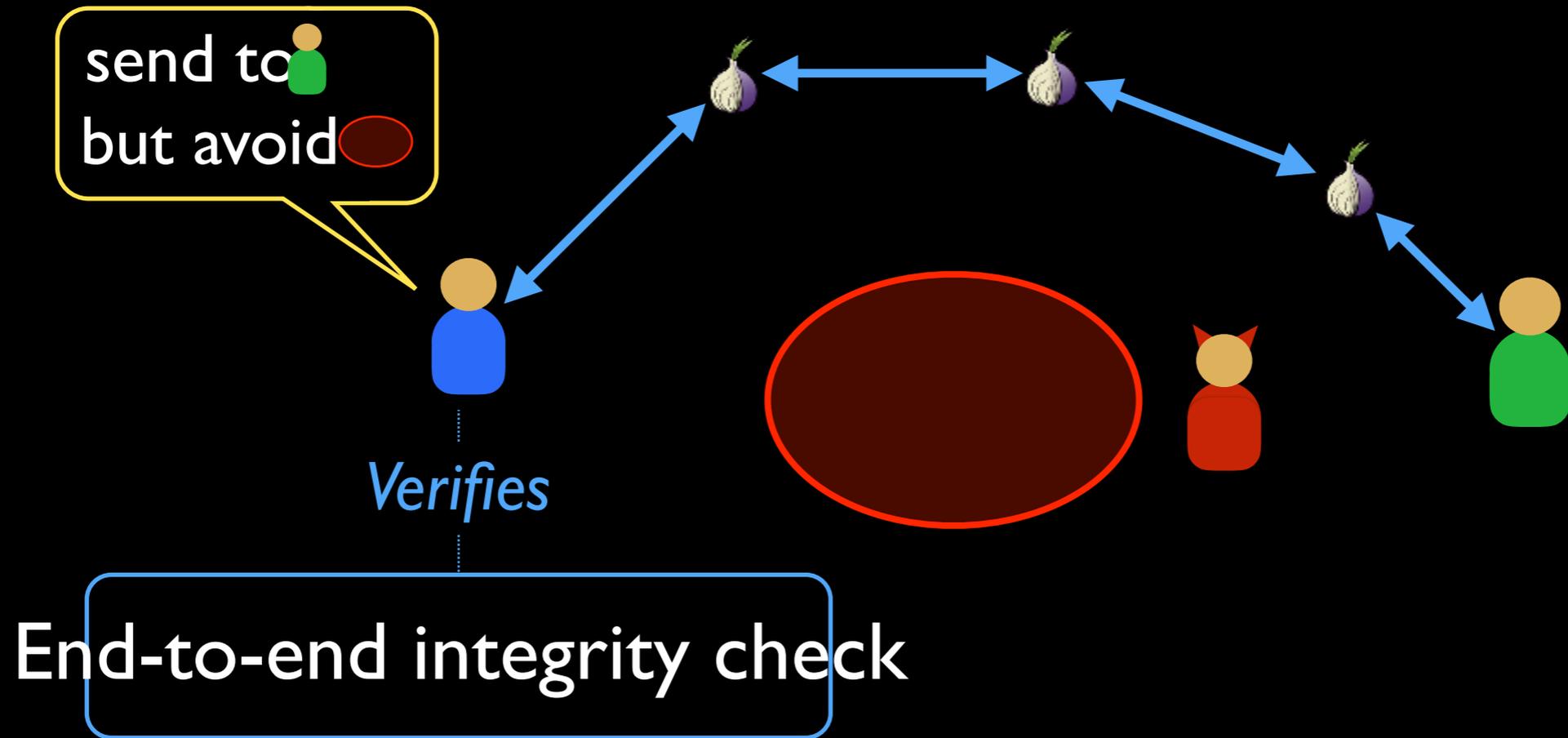
# DeTor: never-once avoidance

Achieving provable avoidance



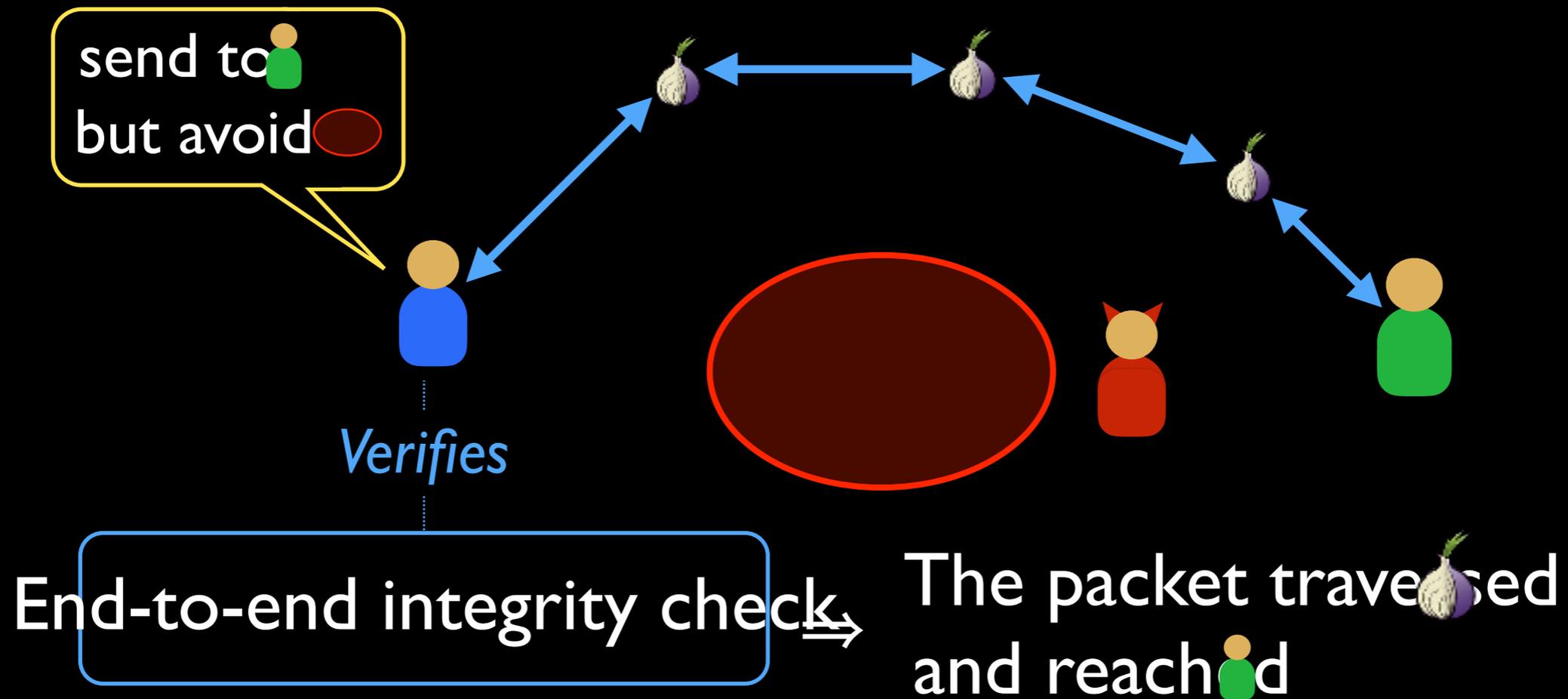
# DeTor: never-once avoidance

Achieving provable avoidance



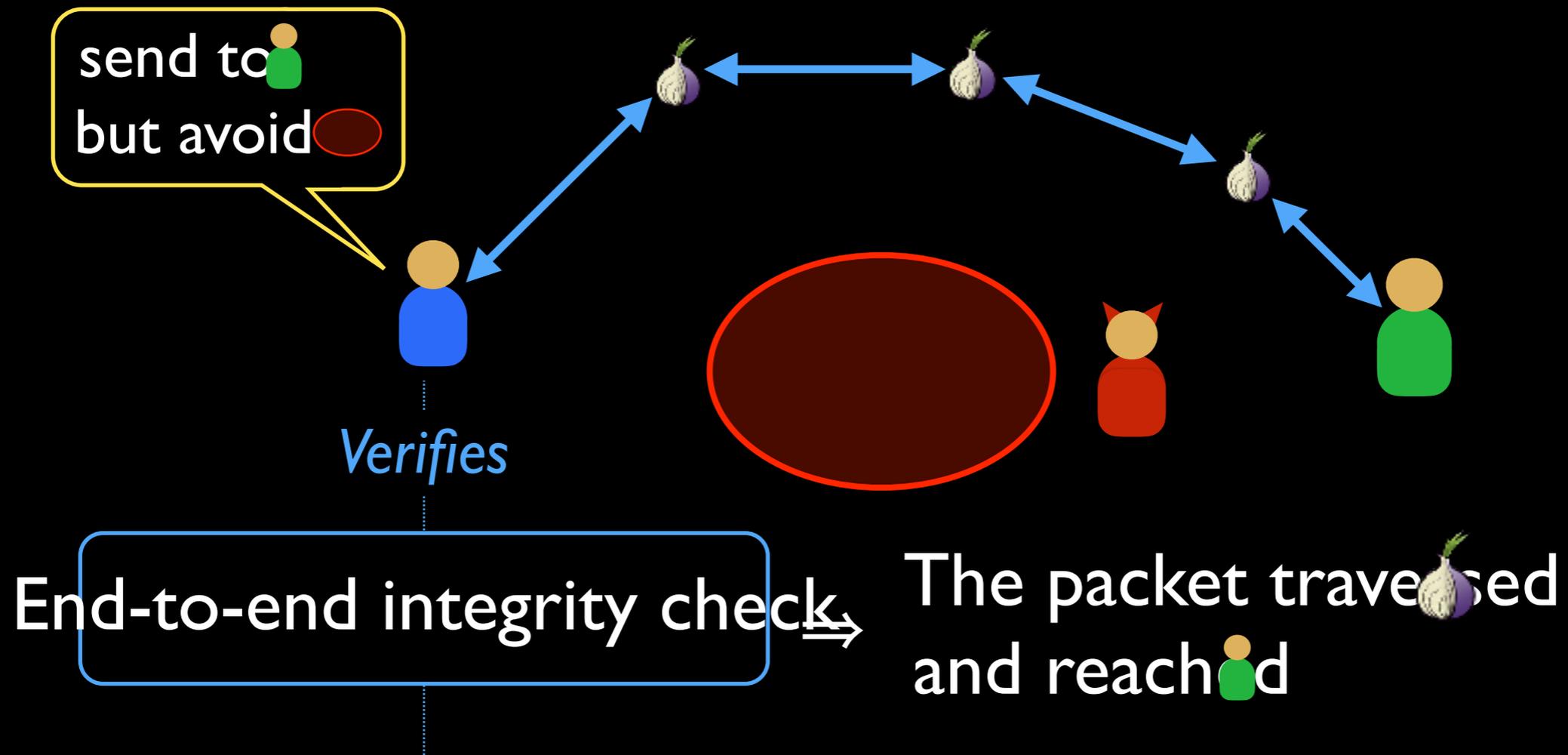
# DeTor: never-once avoidance

Achieving provable avoidance



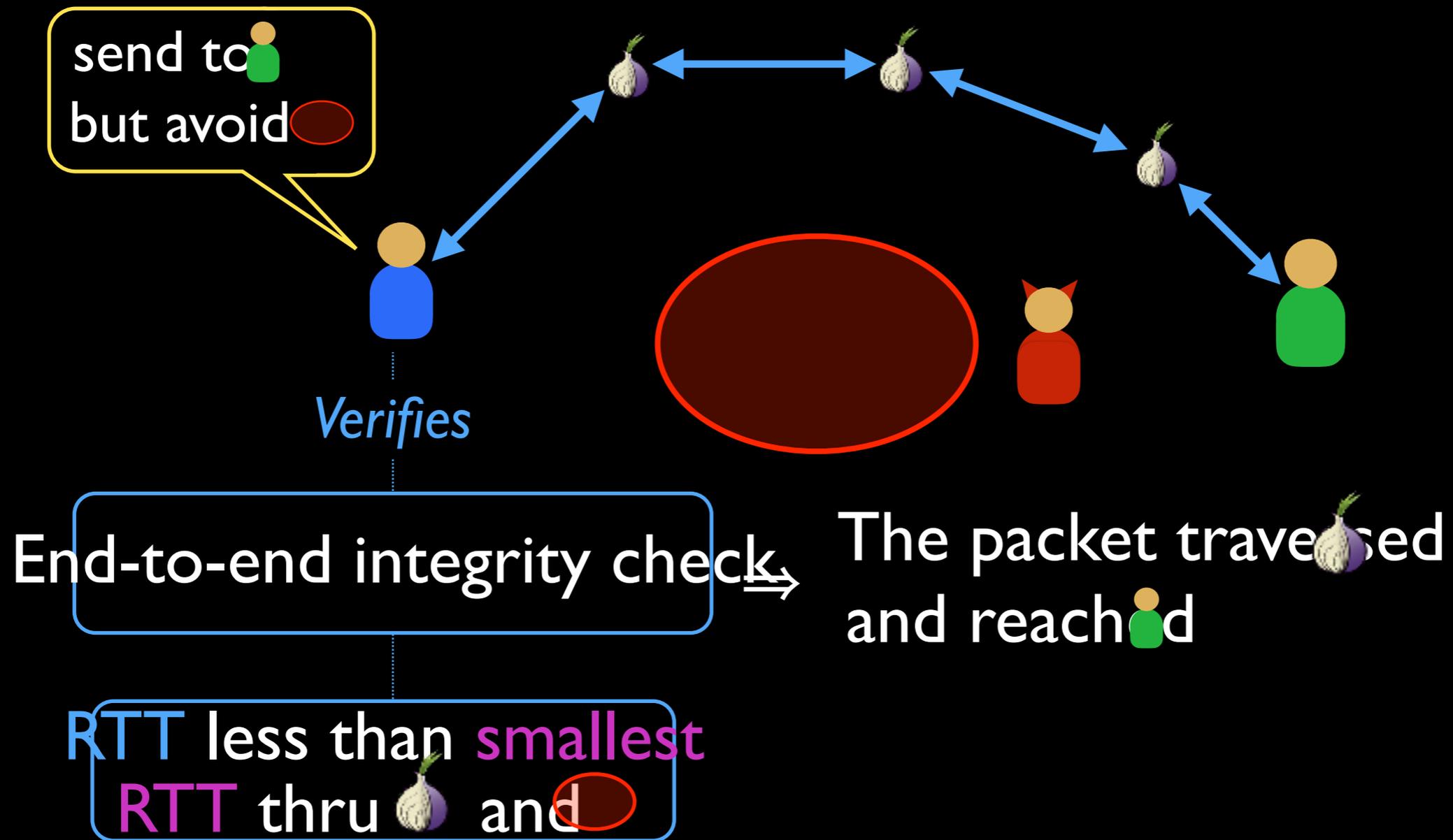
# DeTor: never-once avoidance

Achieving provable avoidance



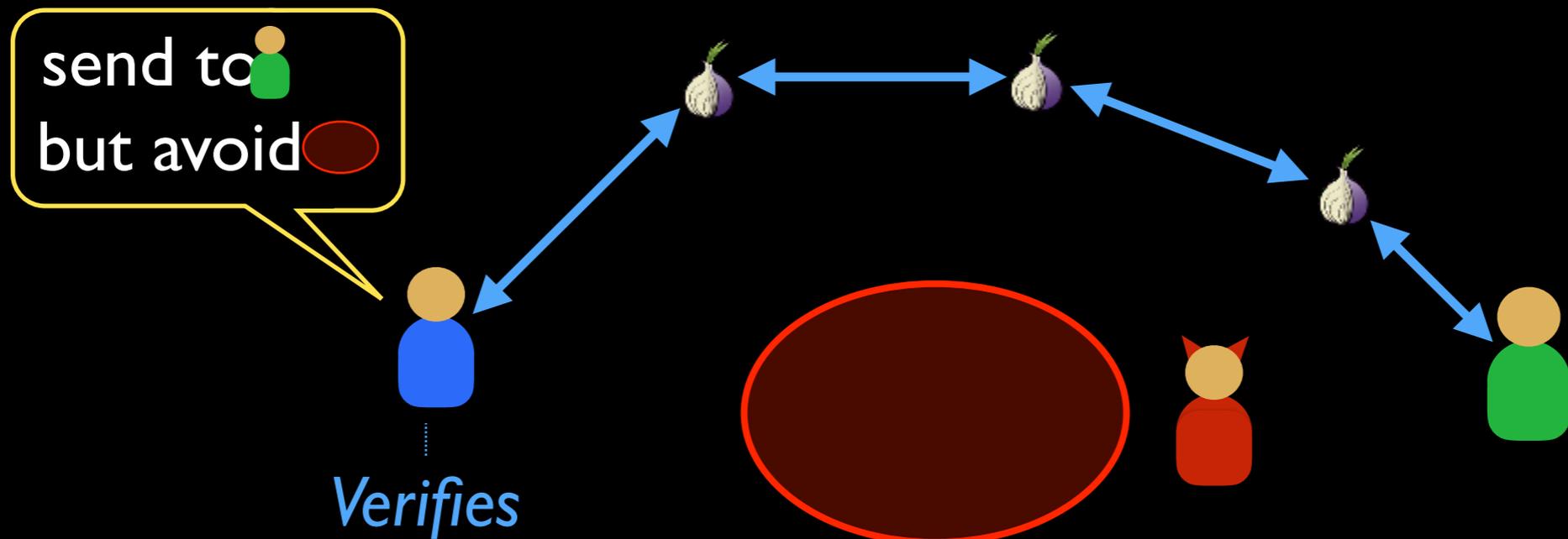
# DeTor: never-once avoidance

Achieving provable avoidance



# DeTor: never-once avoidance

Achieving provable avoidance



End-to-end integrity check  $\Rightarrow$

The packet traversed  and reached 

RTT less than **smallest**  
RTT thru  and 

$\Rightarrow$  The packet could not have traversed  and 

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry&exit legs  
never traverse

Provide per-packet  
proof of avoidance

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry&exit legs  
never traverse

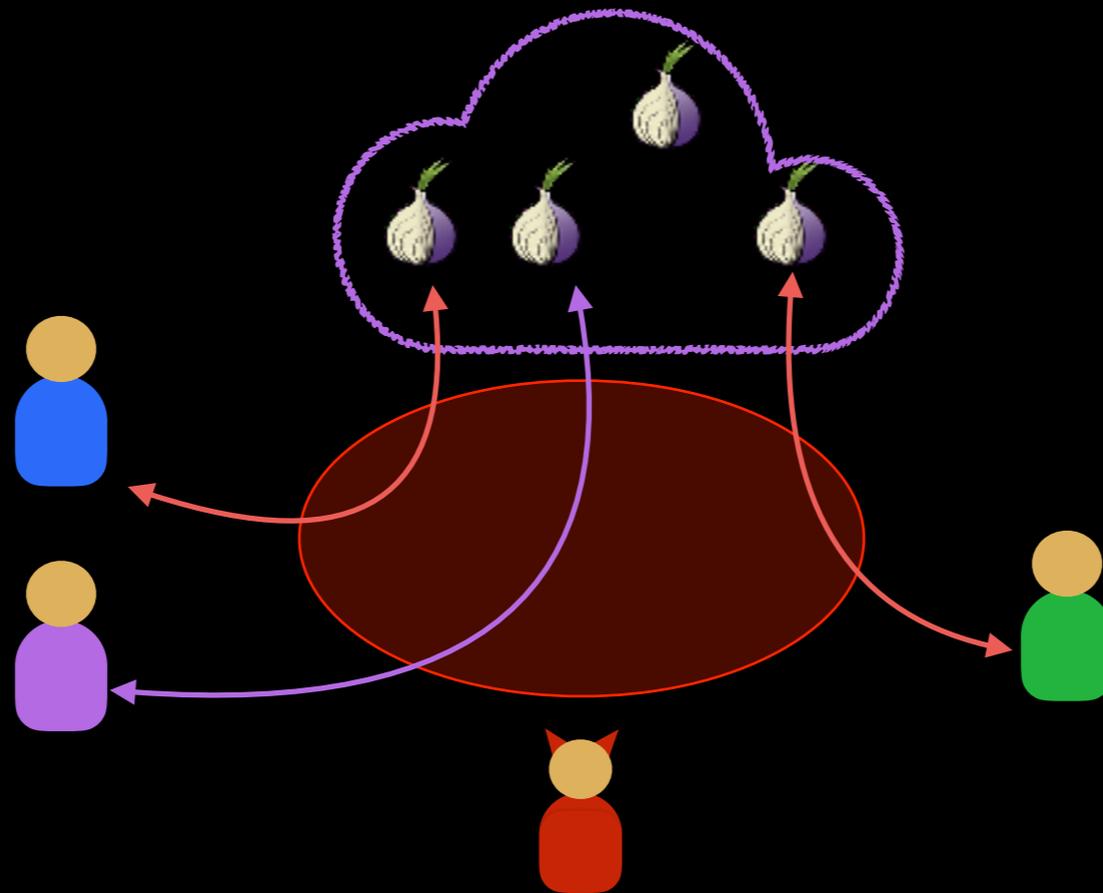
Provide per-packet  
proof of avoidance

# DeTor: never-twice avoidance

Entry and exit legs never traverse the same region

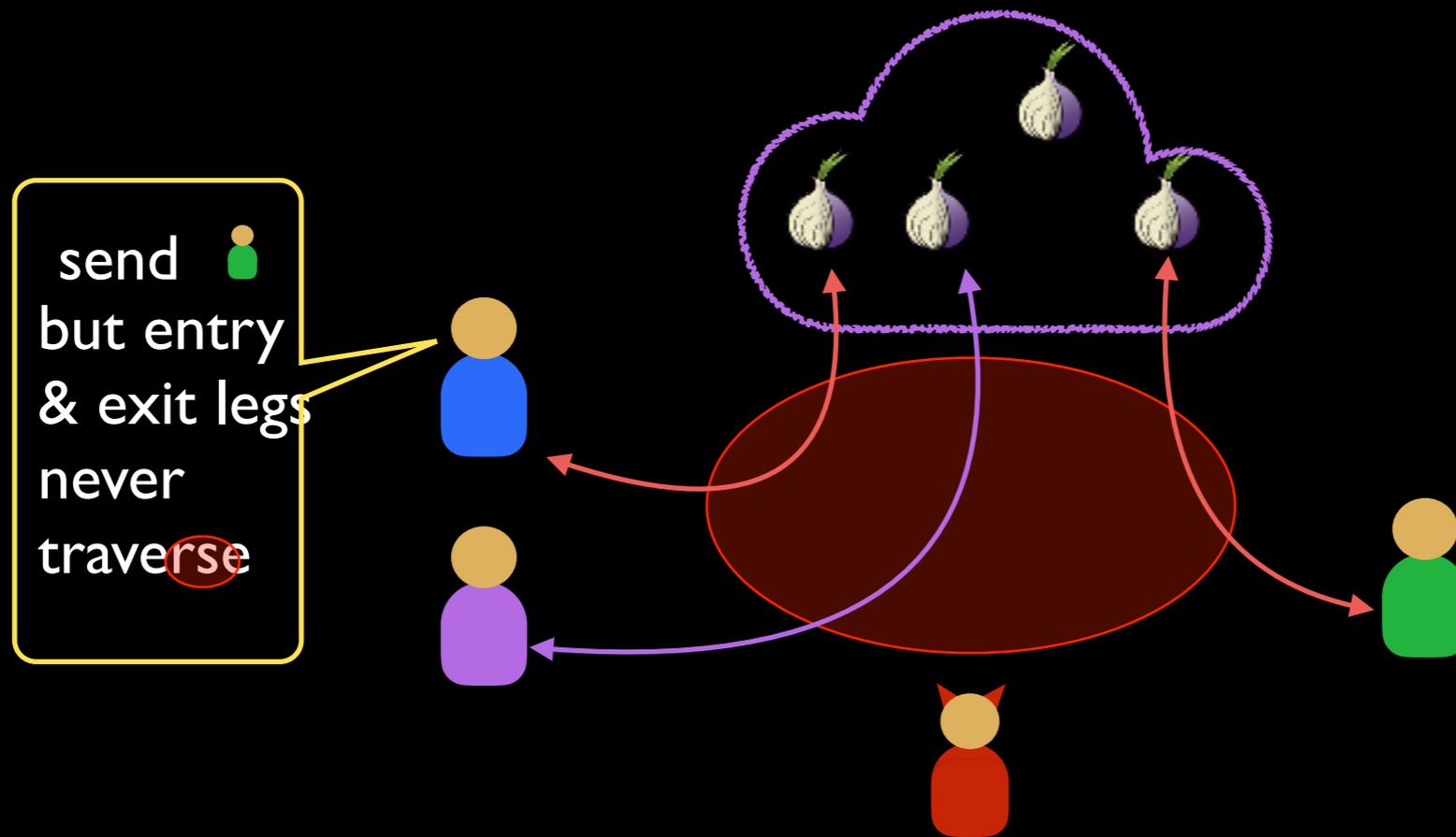
# DeTor: never-twice avoidance

Entry and exit legs never traverse the same region



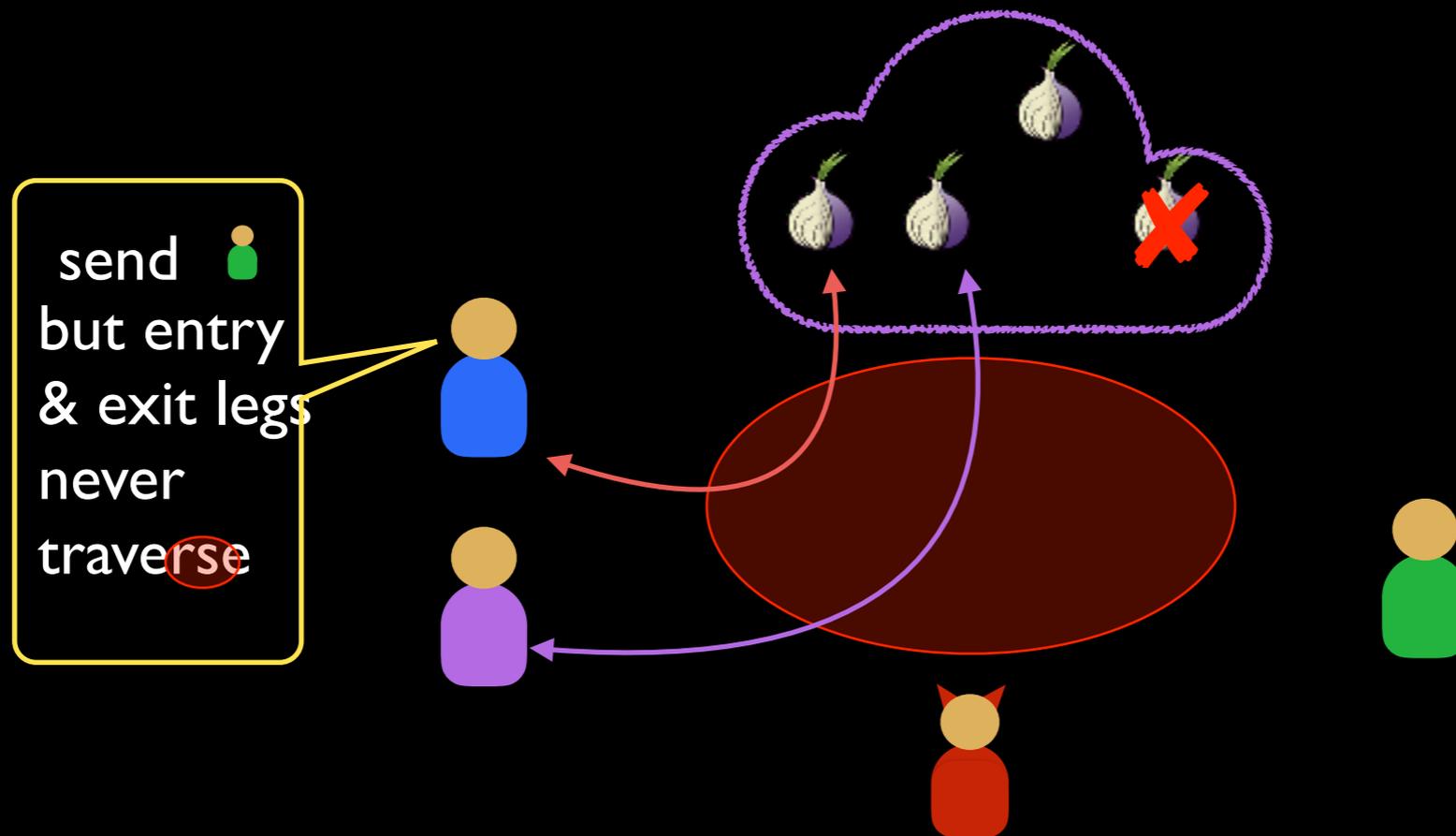
# DeTor: never-twice avoidance

Entry and exit legs never traverse the same region



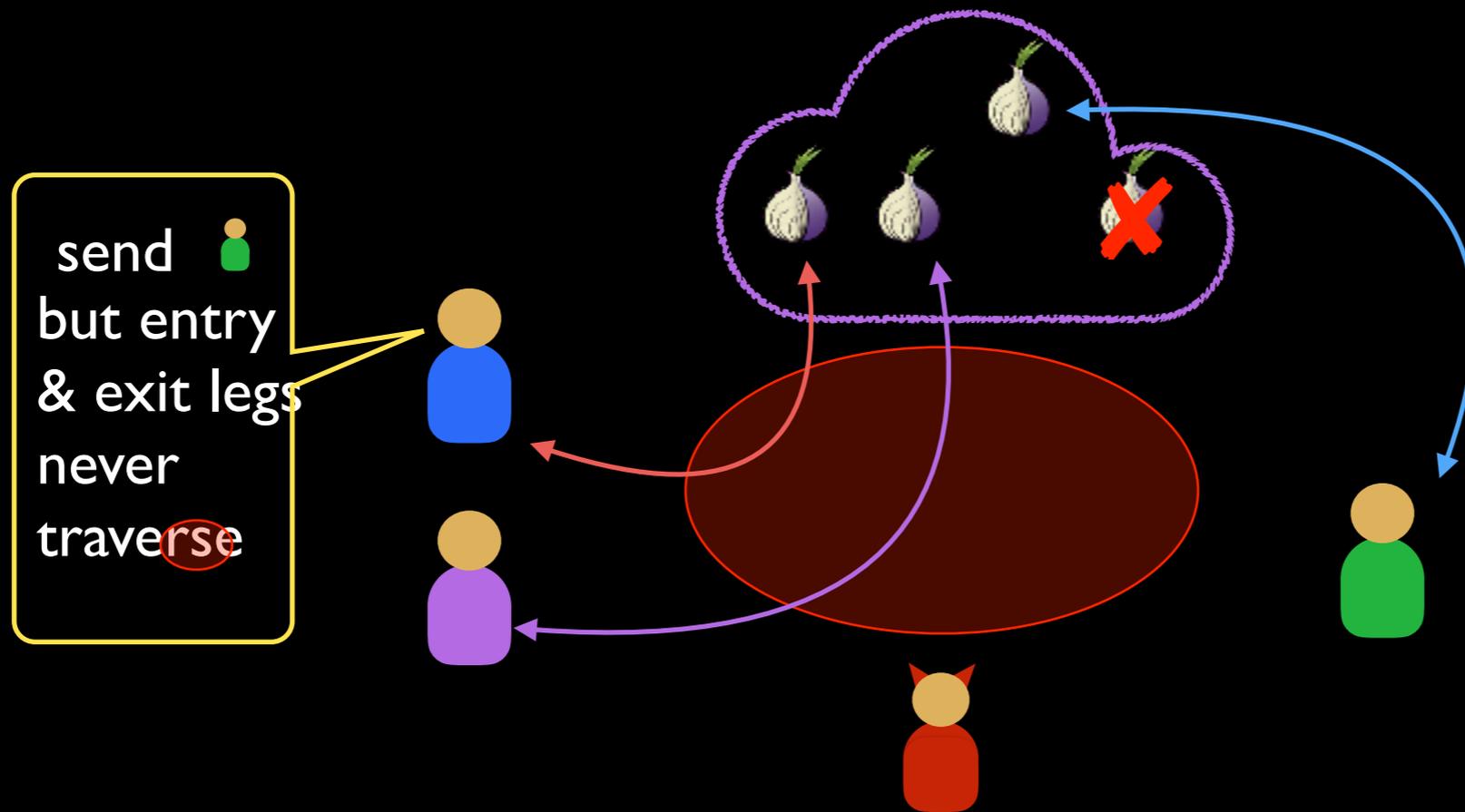
# DeTor: never-twice avoidance

Entry and exit legs never traverse the same region



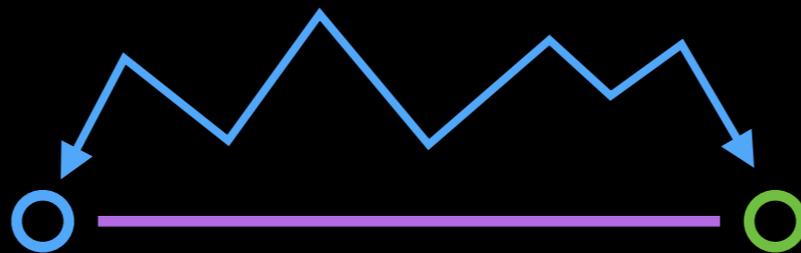
# DeTor: never-twice avoidance

Entry and exit legs never traverse the same region



# DeTor: never-twice avoidance

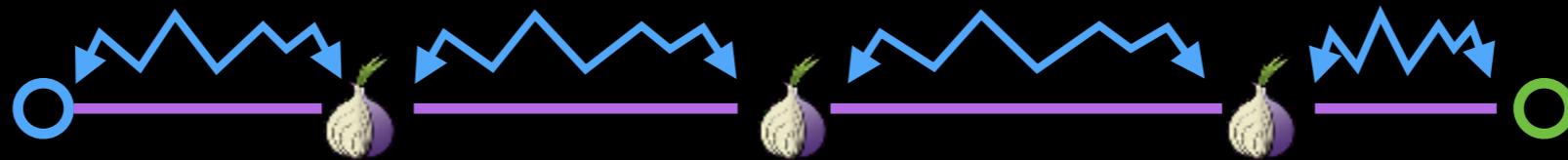
Where could packets possibly reach



Measured RTT = shortest possible RTT + extra

# DeTor: never-twice avoidance

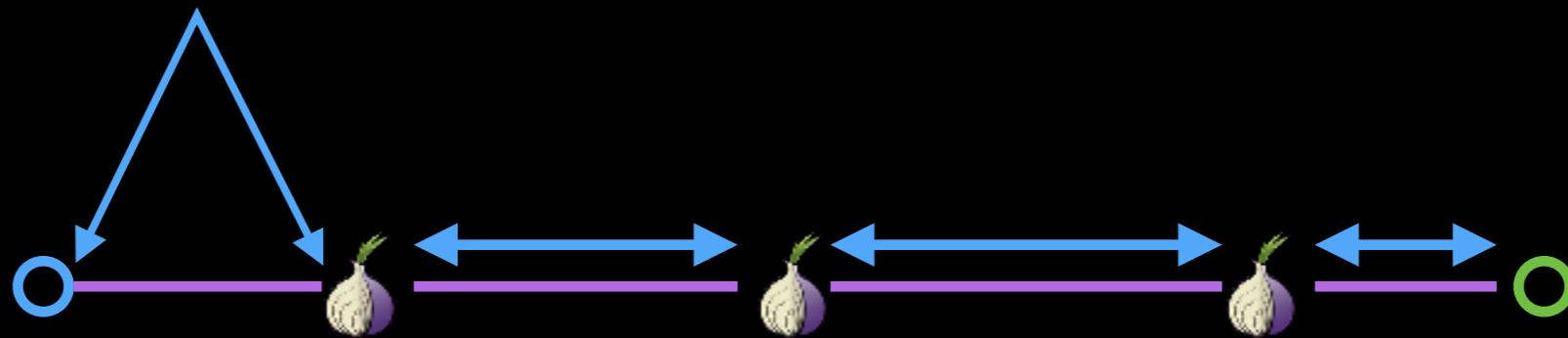
Where could packets possibly reach



Measured RTT  $\geq$  shortest possible RTT  $\geq$  extra

# DeTor: never-twice avoidance

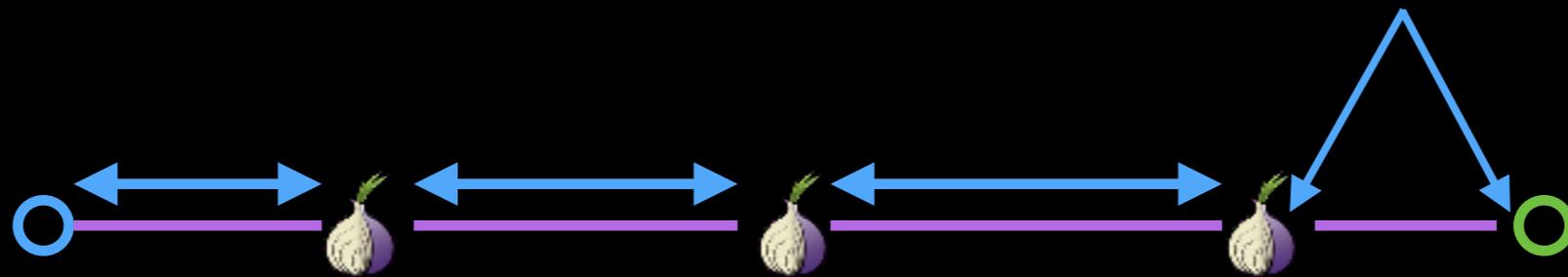
Where could packets possibly reach



Measured RTT  $\geq$  shortest possible RTT  $\geq$  extra

# DeTor: never-twice avoidance

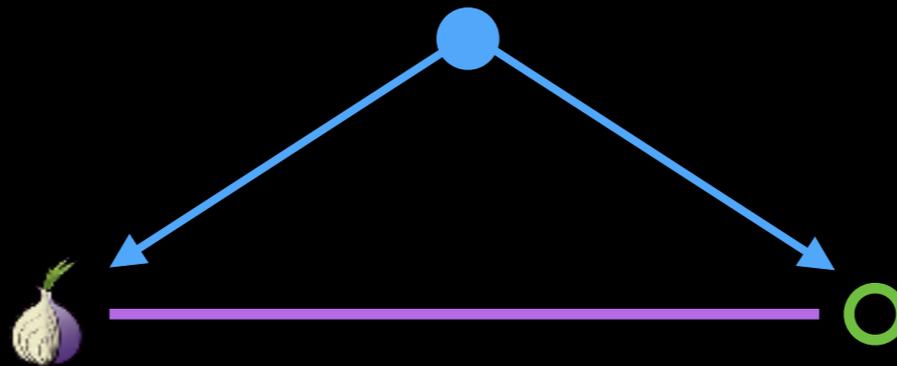
Where could packets possibly reach



Measured RTT  $\sum$  shortest possible RTT  $\sum$  extra

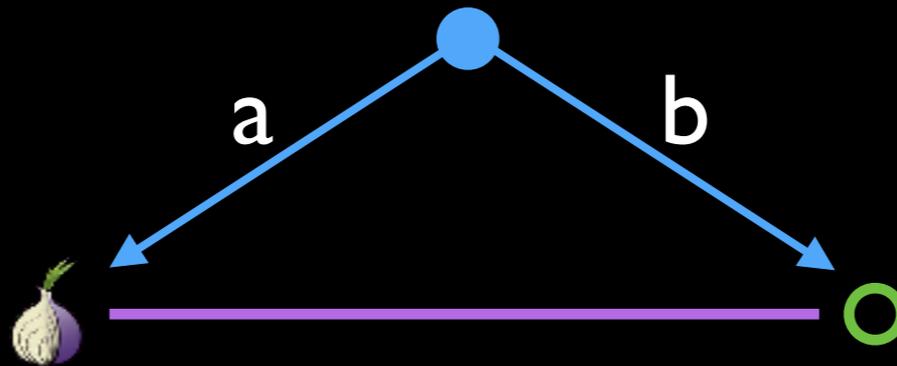
# DeTor: never-twice avoidance

Where could packets possibly reach



# DeTor: never-twice avoidance

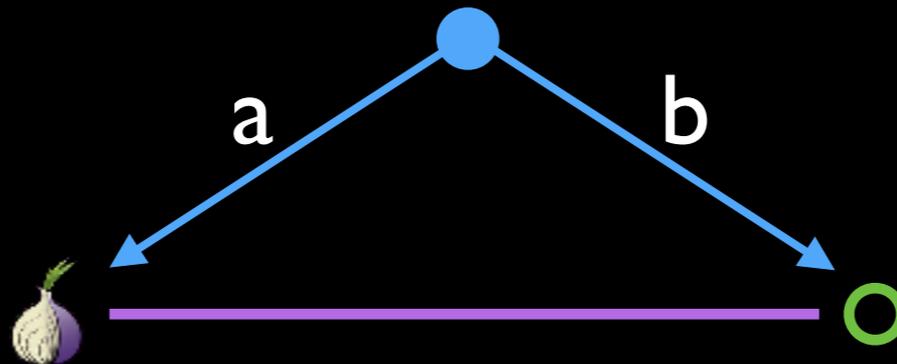
Where could packets possibly reach



# DeTor: never-twice avoidance

Where could packets possibly reach

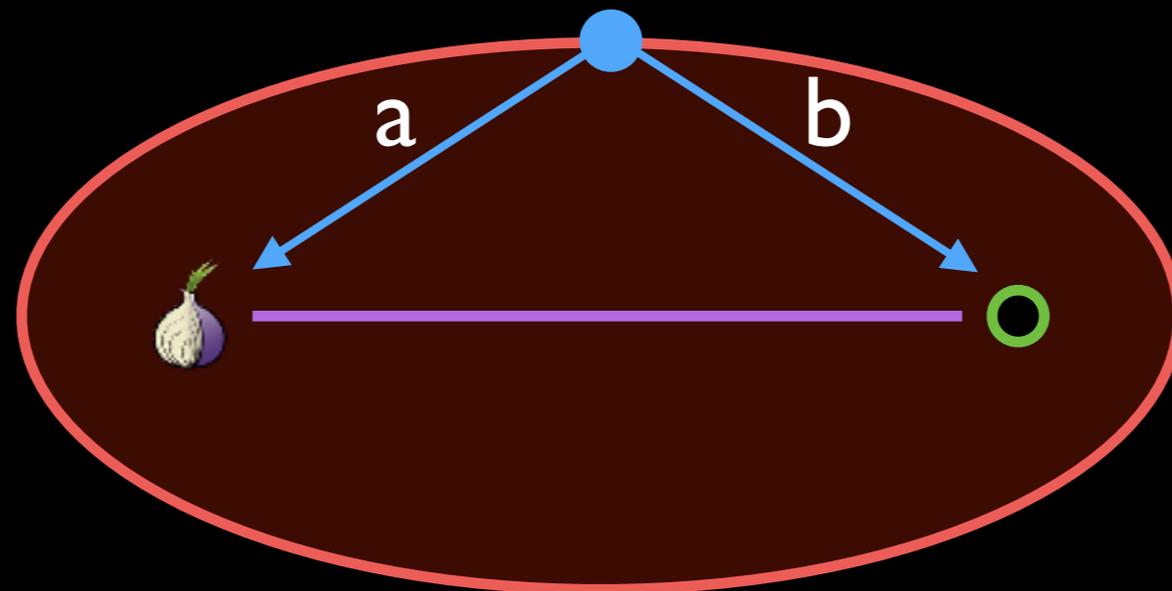
Upper bound  $RTT \geq 2(a+b) / c$



# DeTor: never-twice avoidance

Where could packets possibly reach

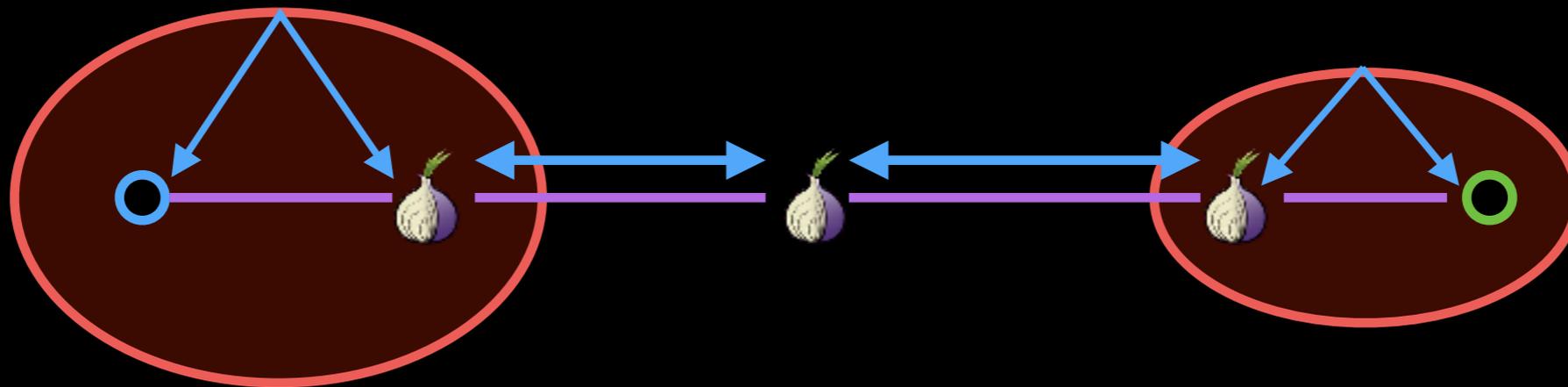
$$\text{Upper bound RTT} \geq 2(a+b) / c$$



The packet could possibly reach any point  
in the ellipse

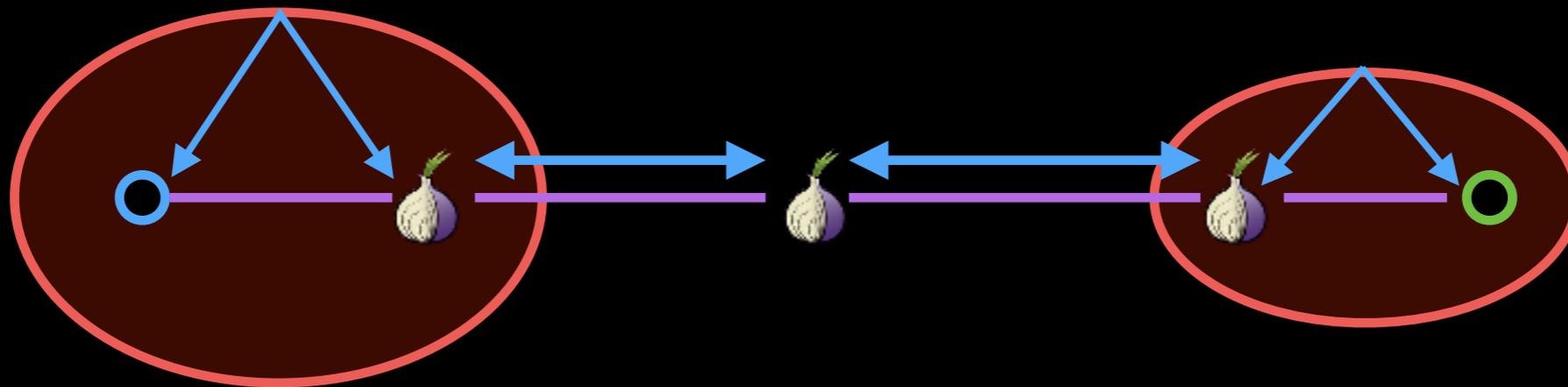
# DeTor: never-twice avoidance

Where could packets possibly reach



# DeTor: never-twice avoidance

Where could packets possibly reach



Compute the **worst-case scenarios**  
for *both* entry and exit legs, separately

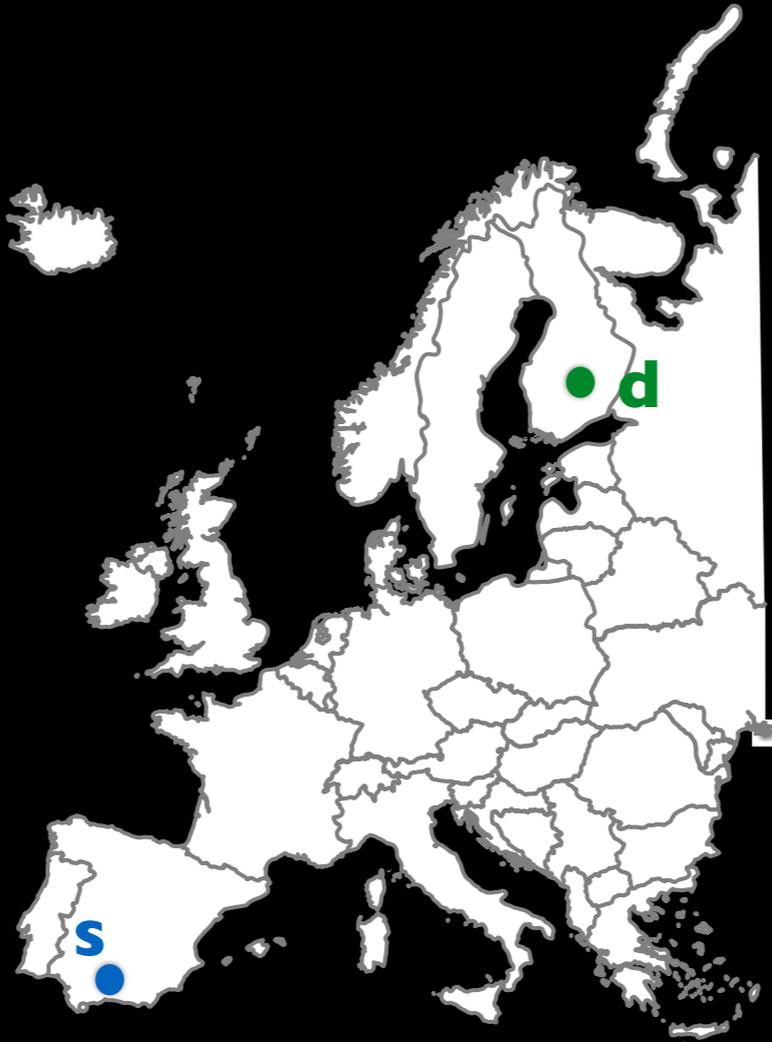
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



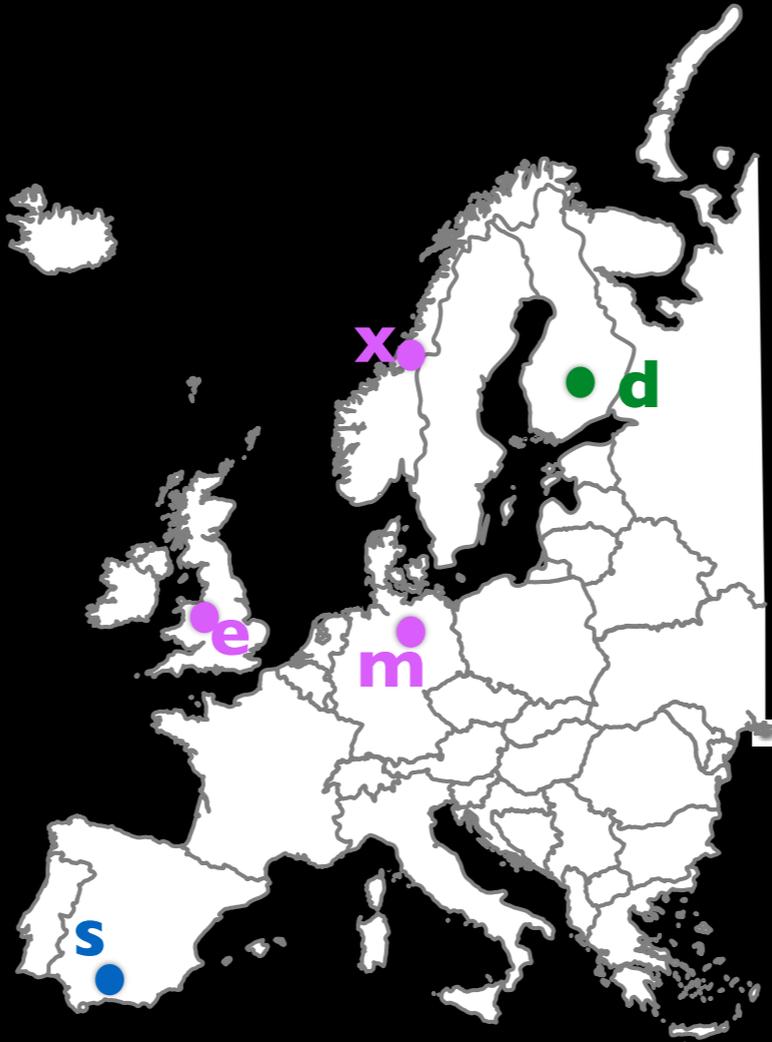
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



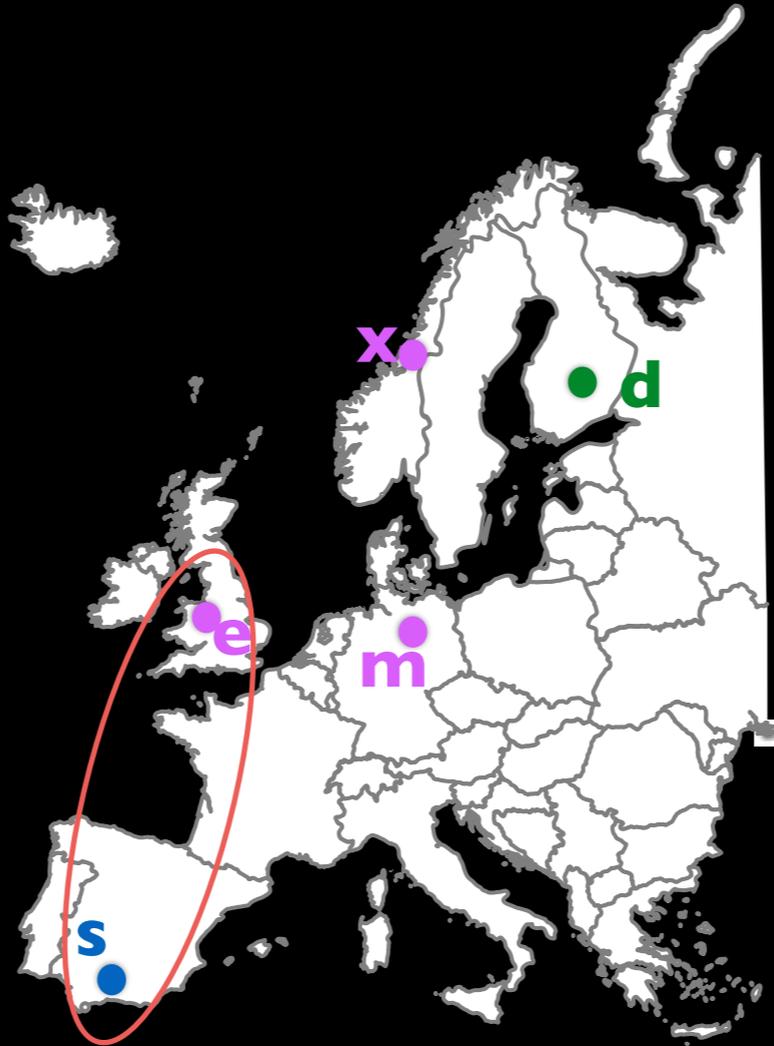
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



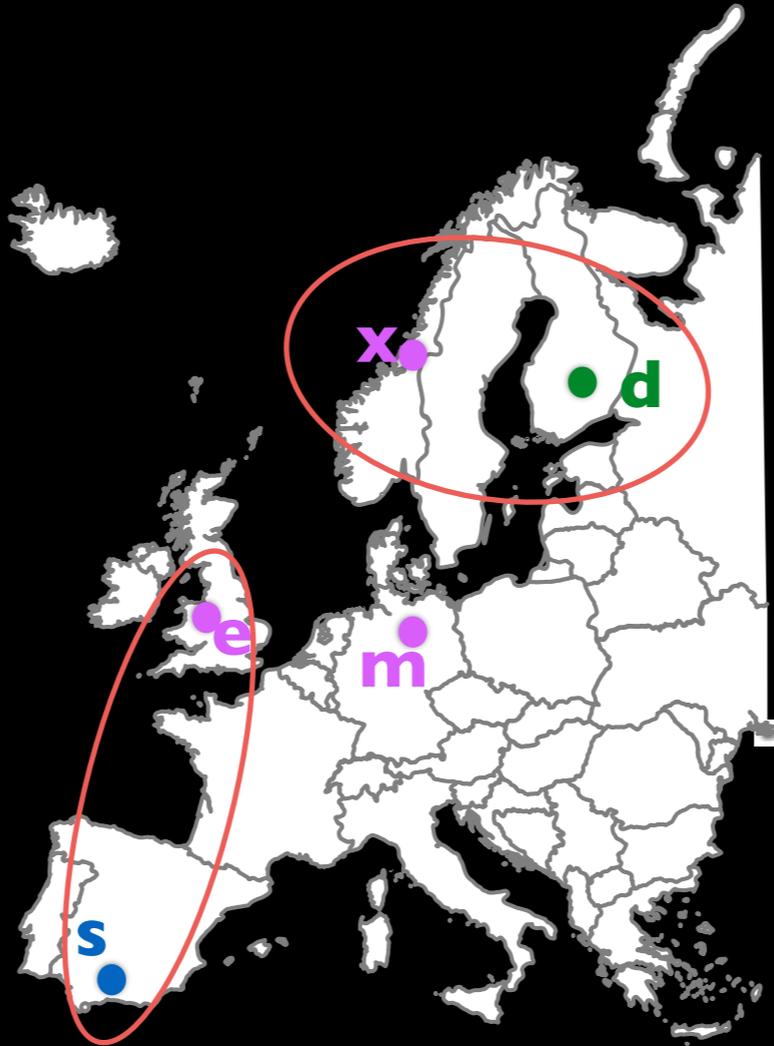
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



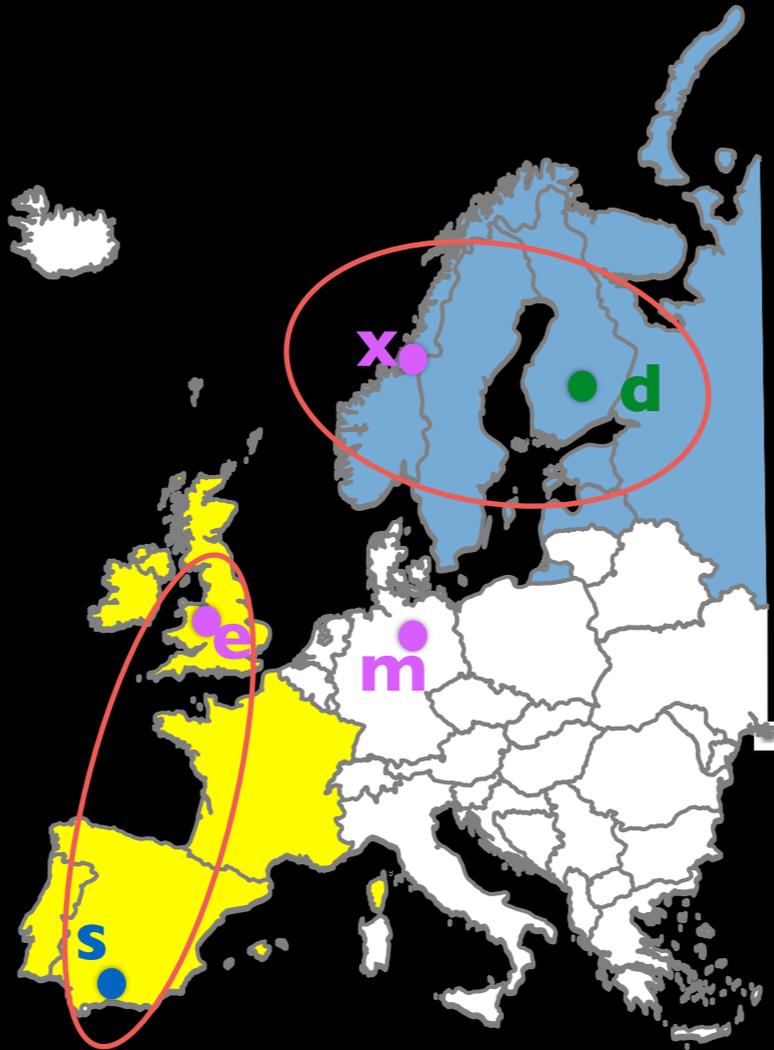
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



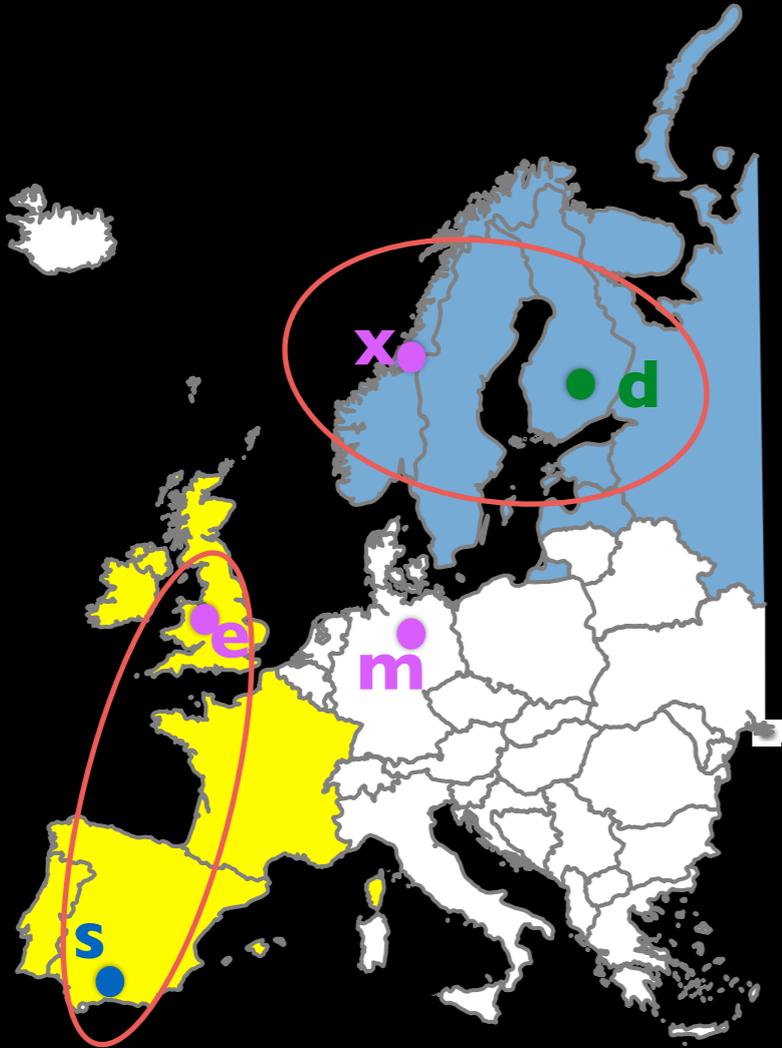
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



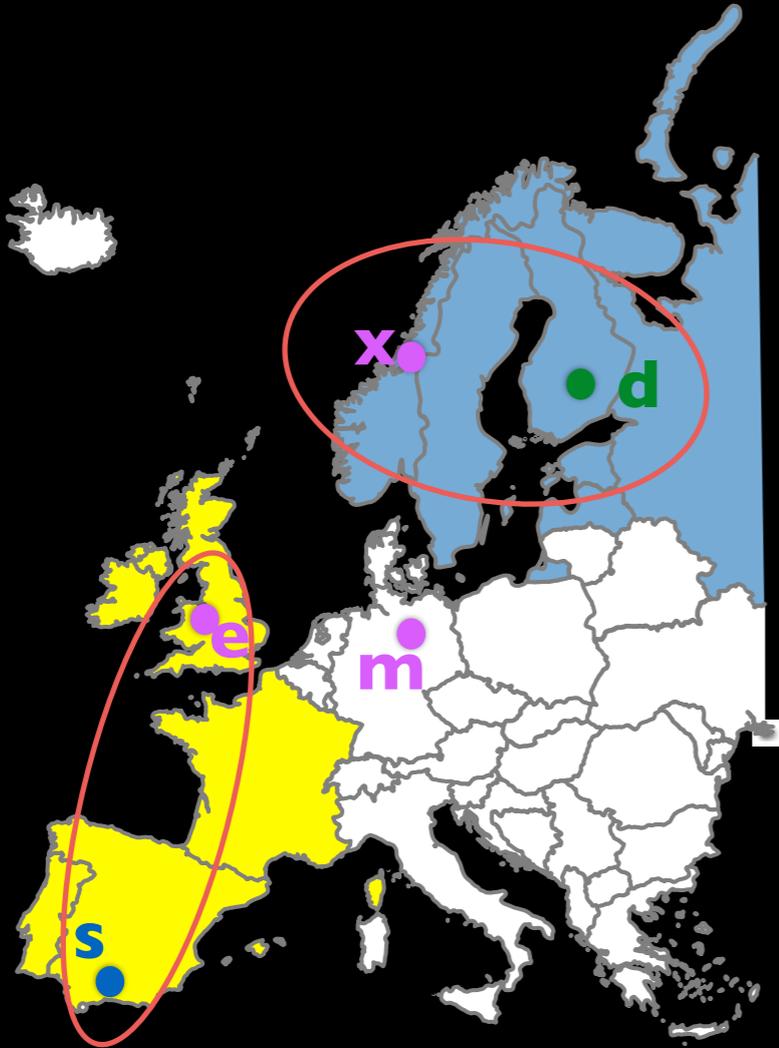
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



# DeTor: never-twice avoidance

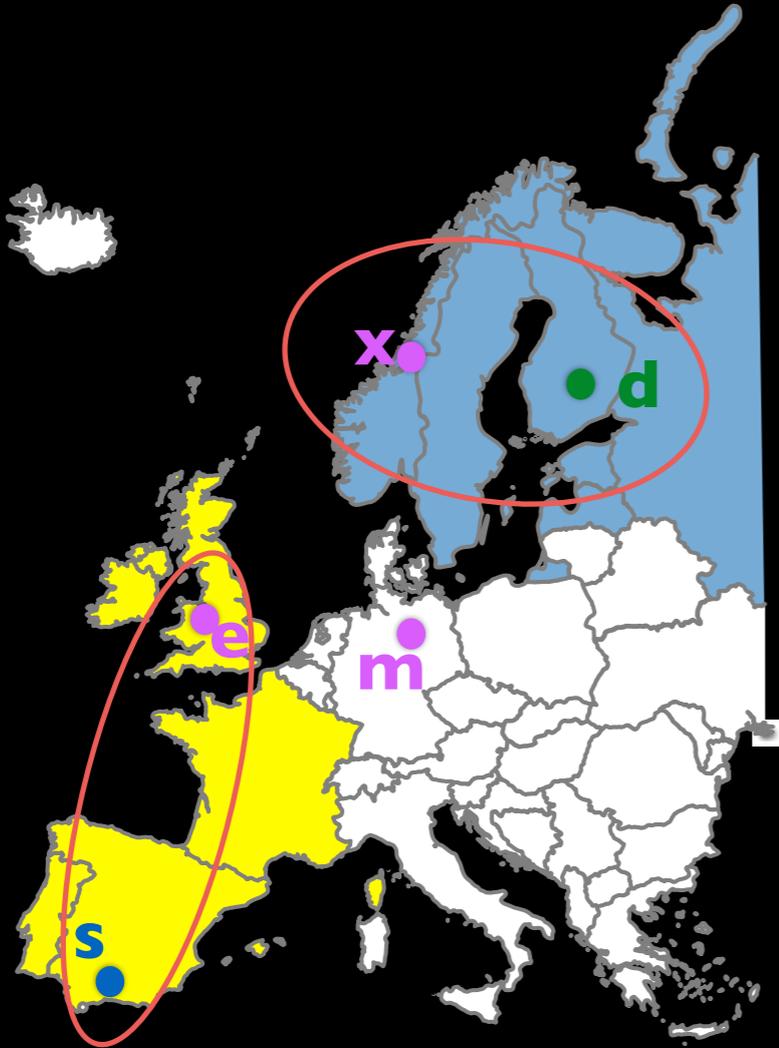
Which countries can entry & exit legs reach



no country  
intersects  
with both ellipses

# DeTor: never-twice avoidance

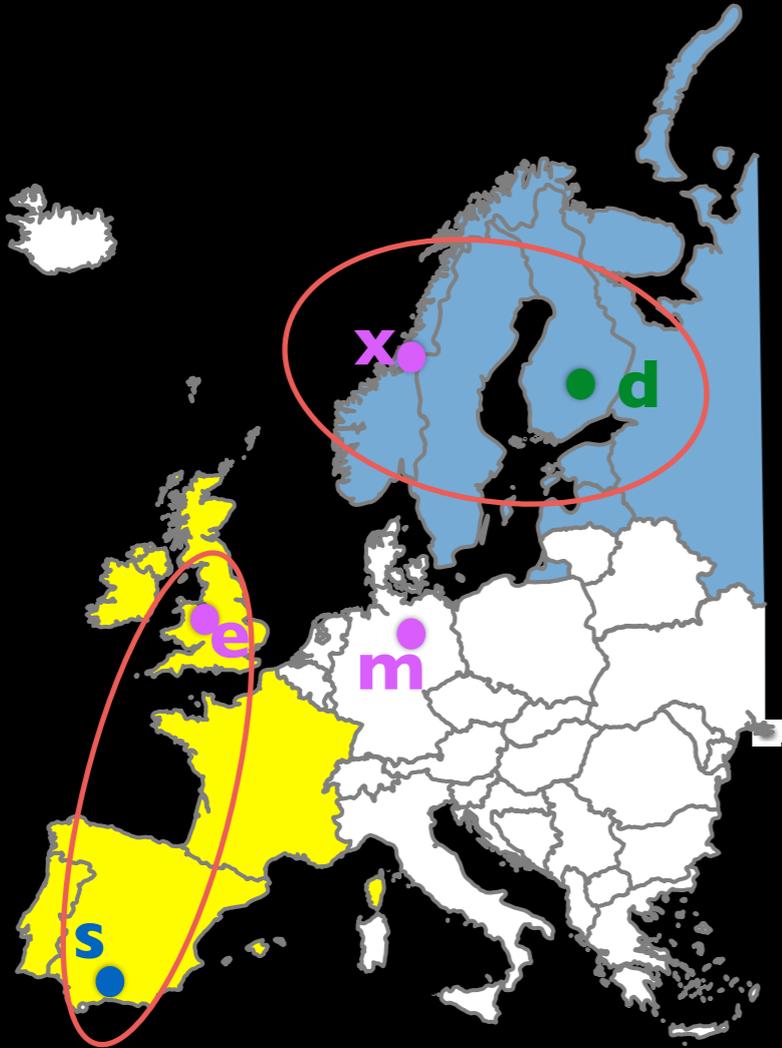
Which countries can entry & exit legs reach



no country  
intersects  
with both ellipses  
↓

# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



no country  
intersects  
with both **ellipses**  
packet over entry/exit legs  
**could not**  
have traversed the same  
country

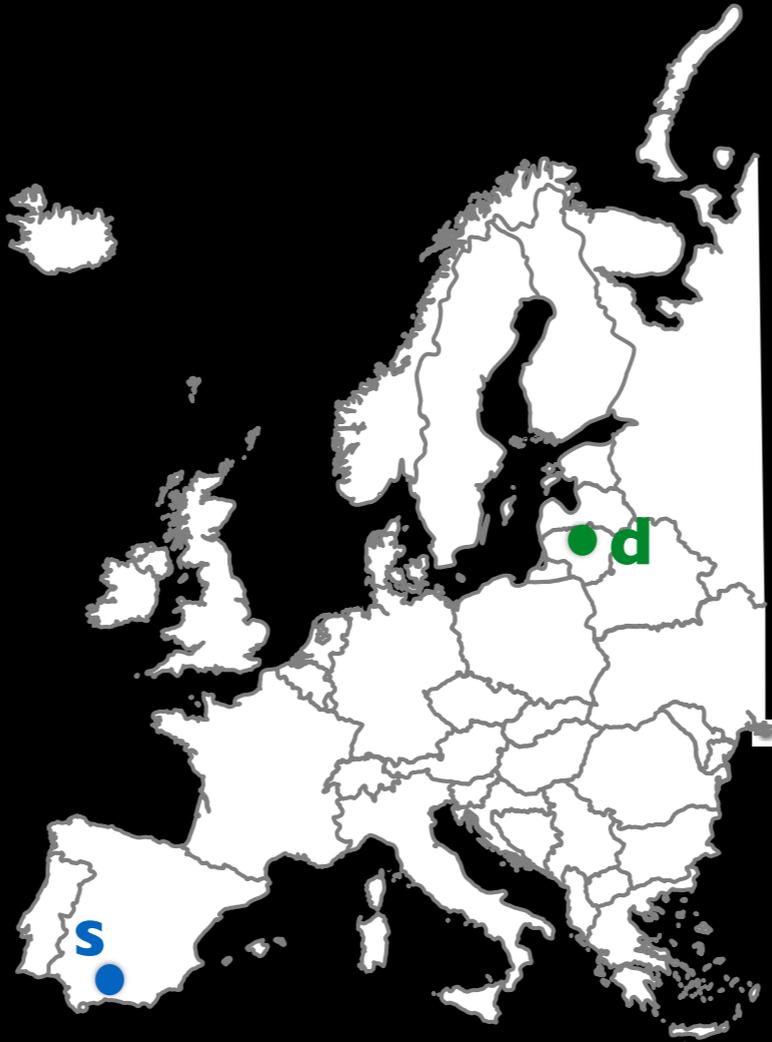
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



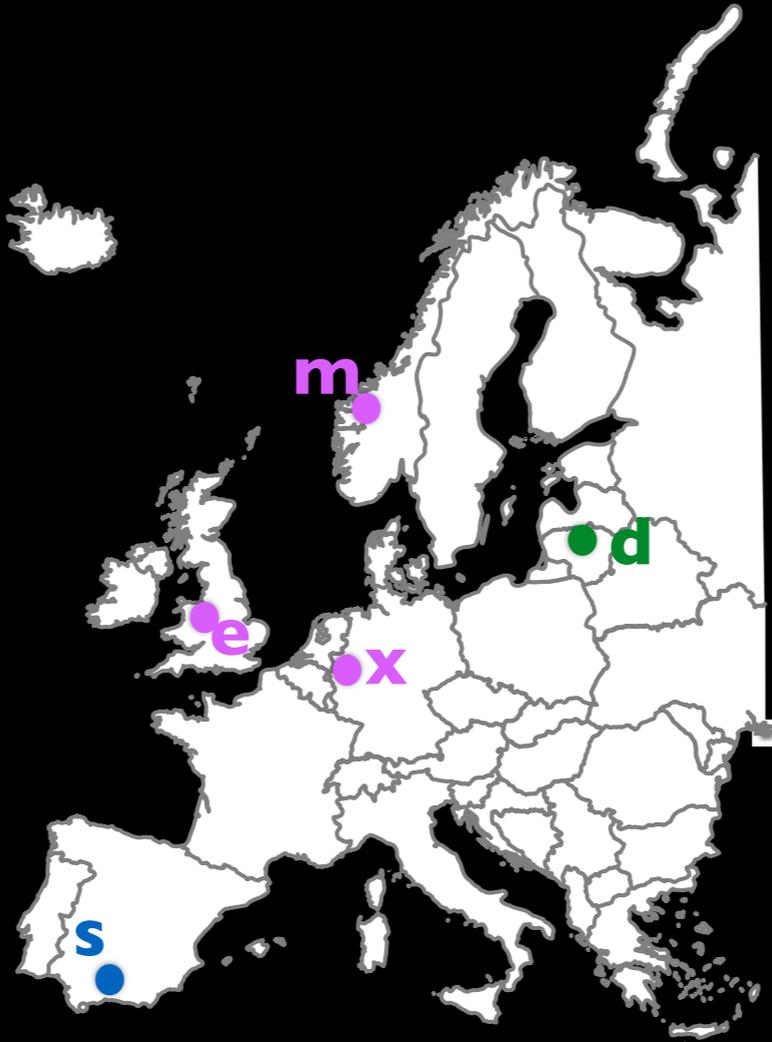
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



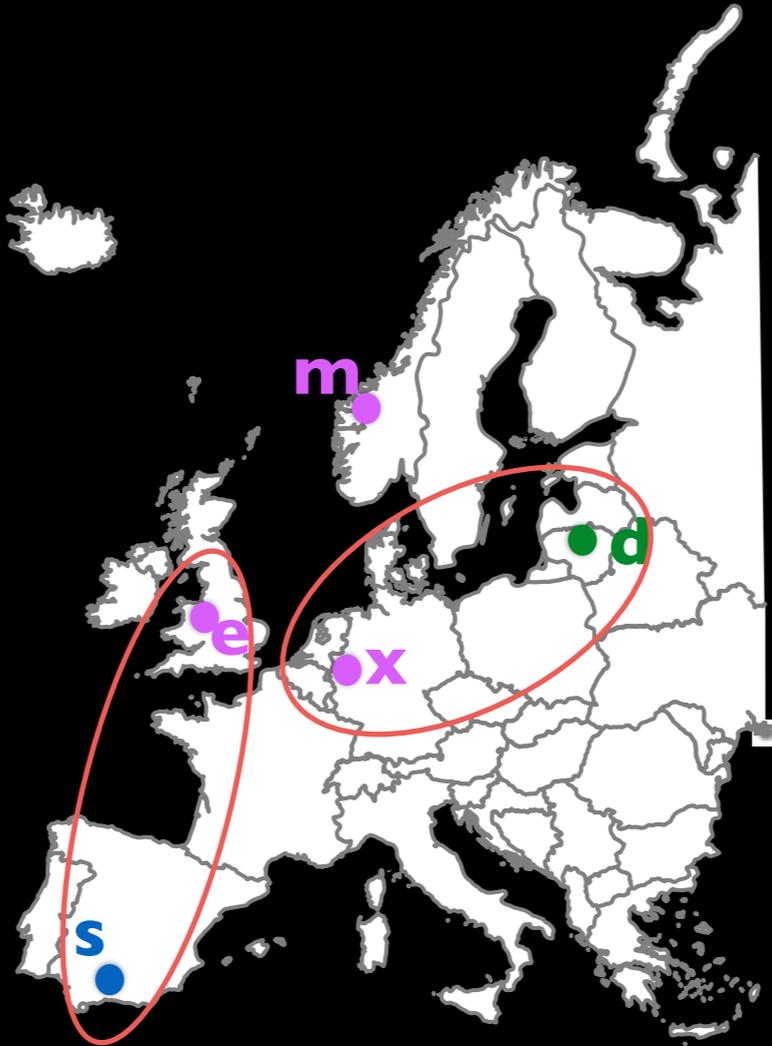
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



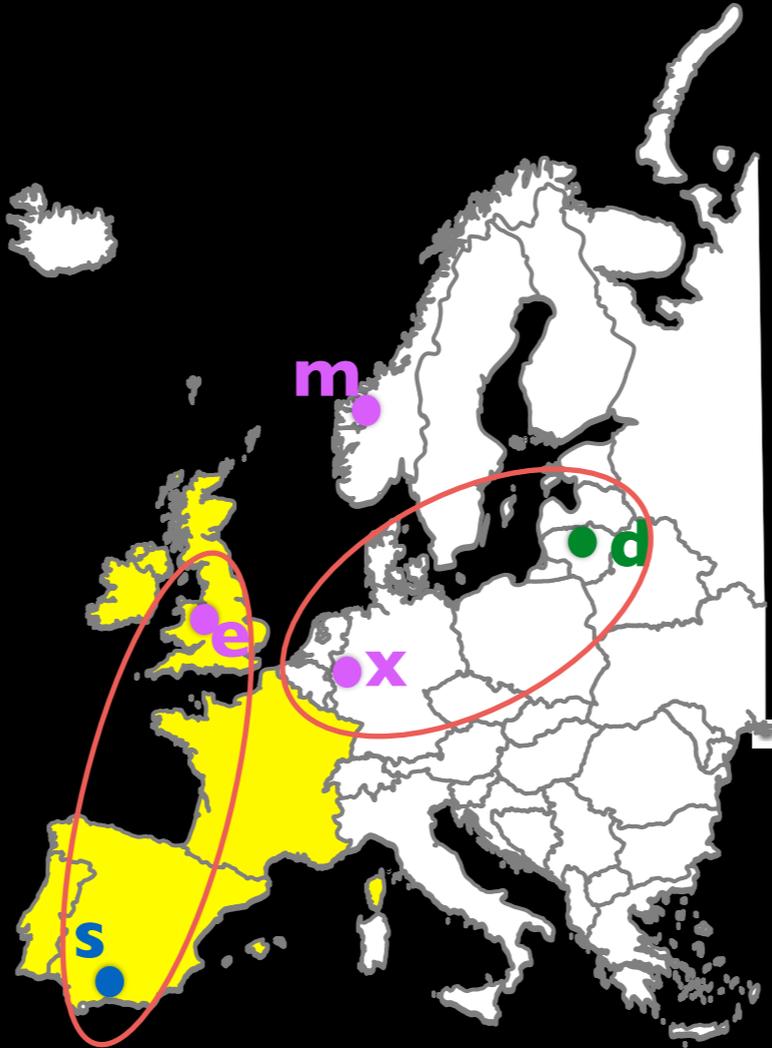
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



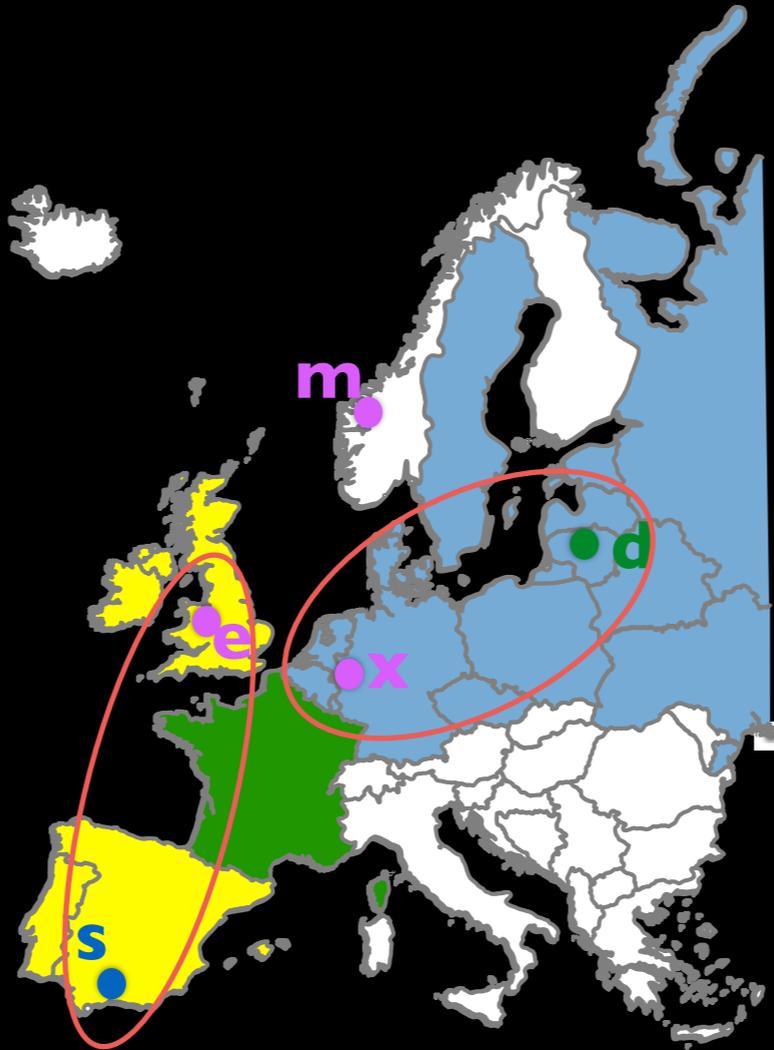
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



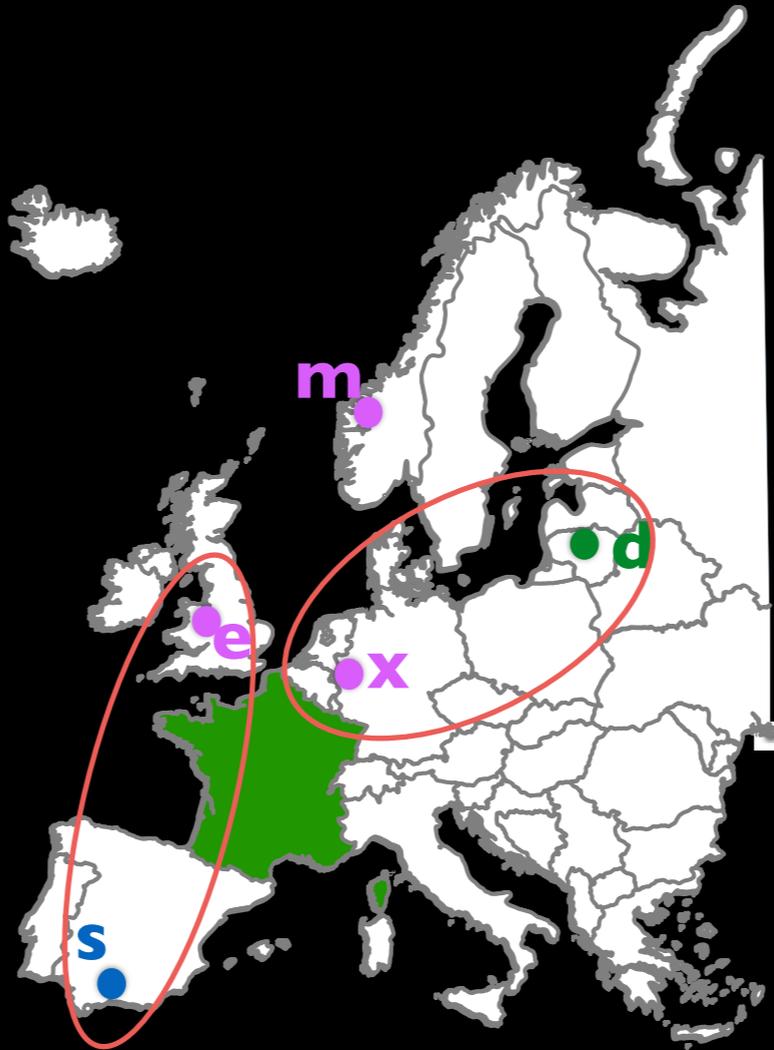
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



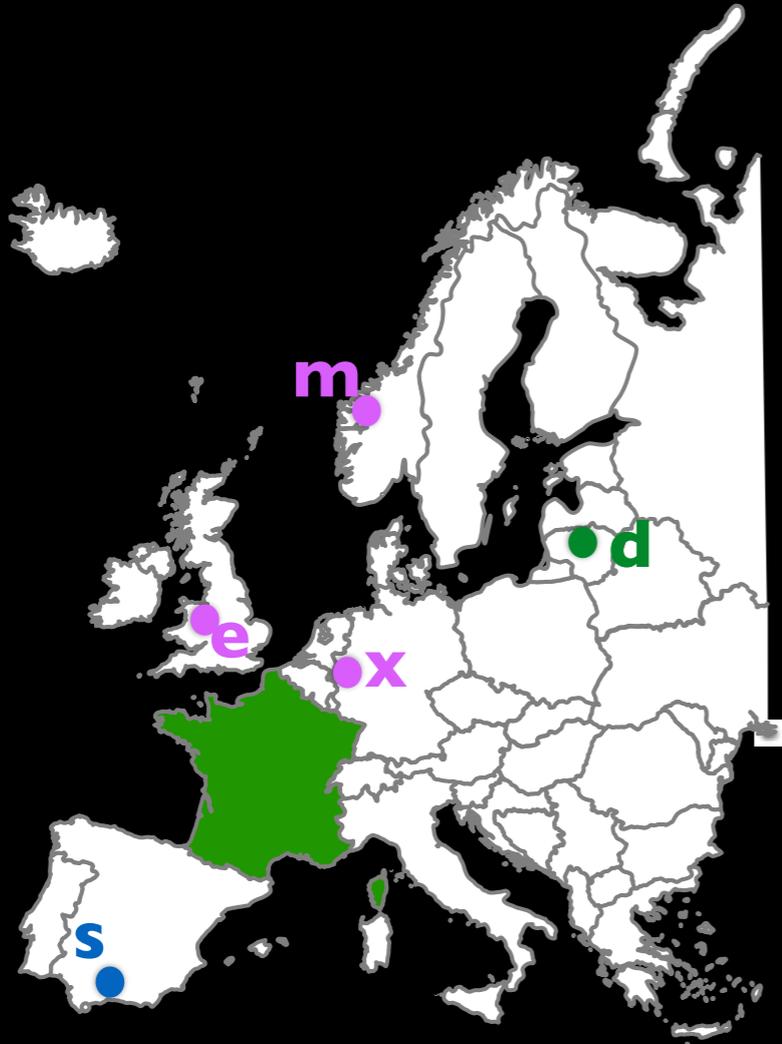
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



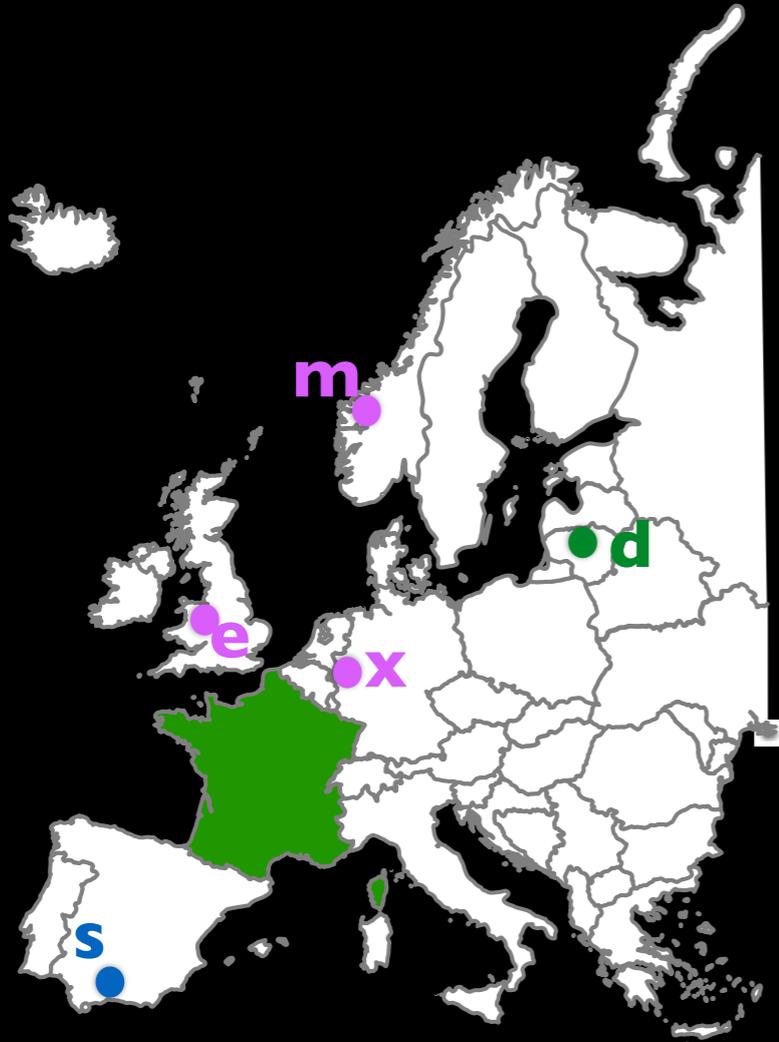
# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



# DeTor: never-twice avoidance

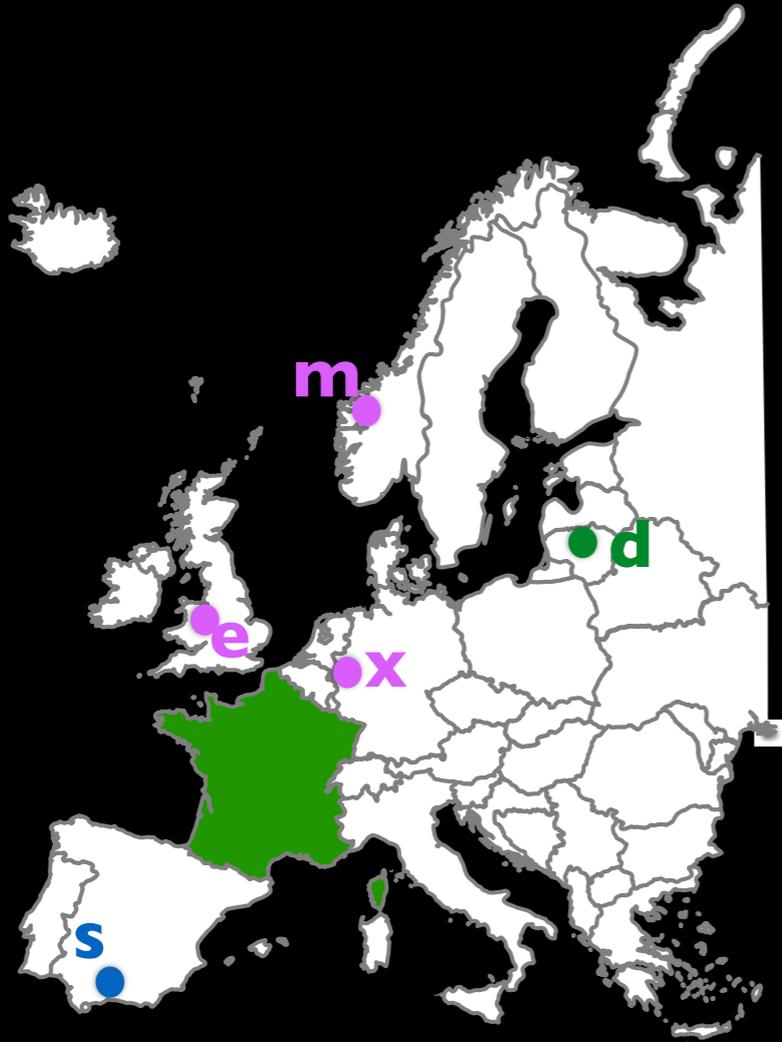
Which countries can entry & exit legs reach



For each **country** intersects with both **ellipses**

# DeTor: never-twice avoidance

Which countries can entry & exit legs reach

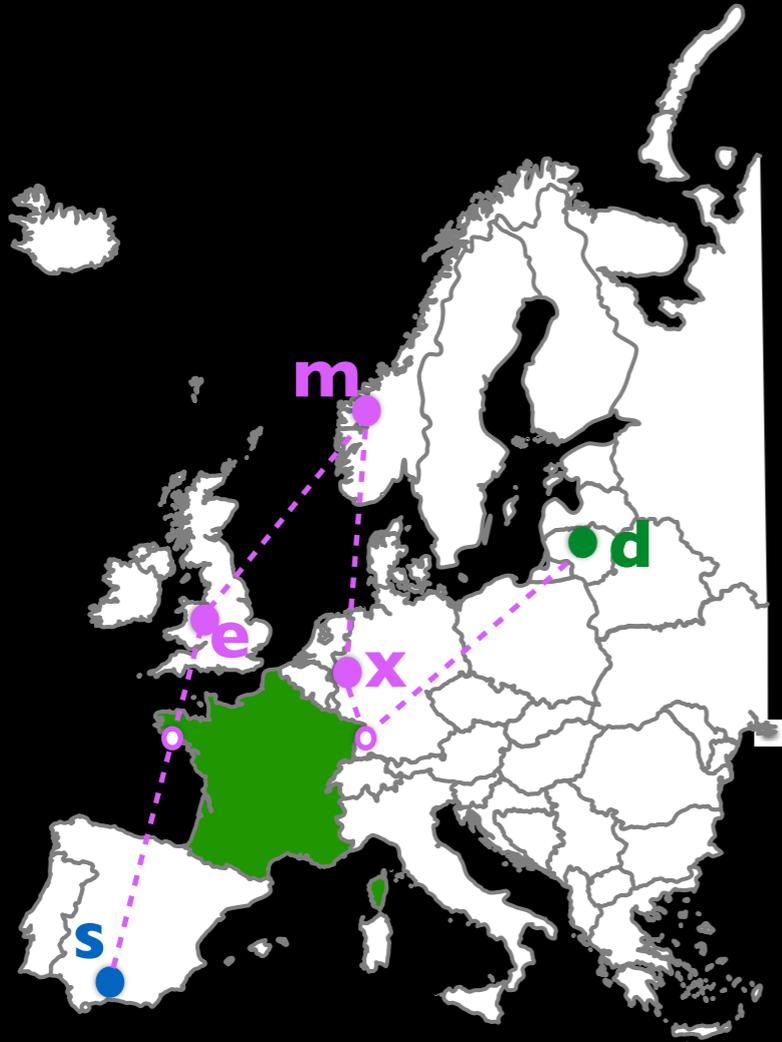


For each **country** intersects with both **ellipses**

The **shortest possible RTT** thru Tor and entry & exit legs **traverse**

# DeTor: never-twice avoidance

Which countries can entry & exit legs reach

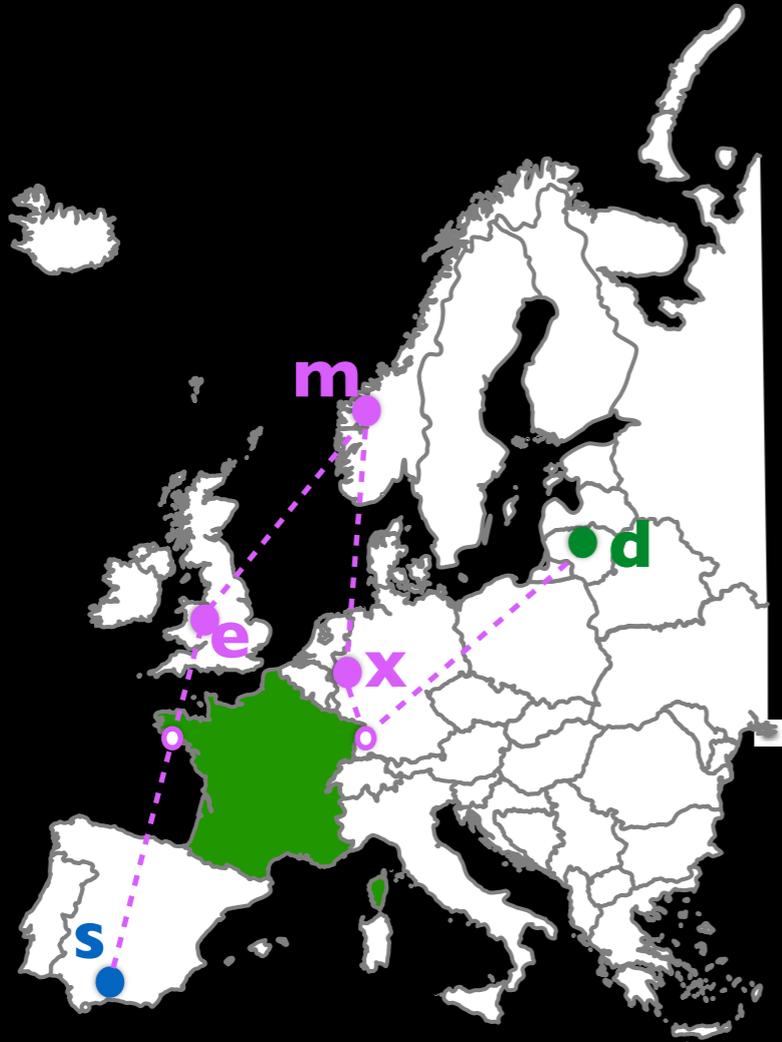


For each **country** intersects with both **ellipses**

The **shortest possible RTT** thru Tor and entry & exit legs **traverse**

# DeTor: never-twice avoidance

Which countries can entry & exit legs reach

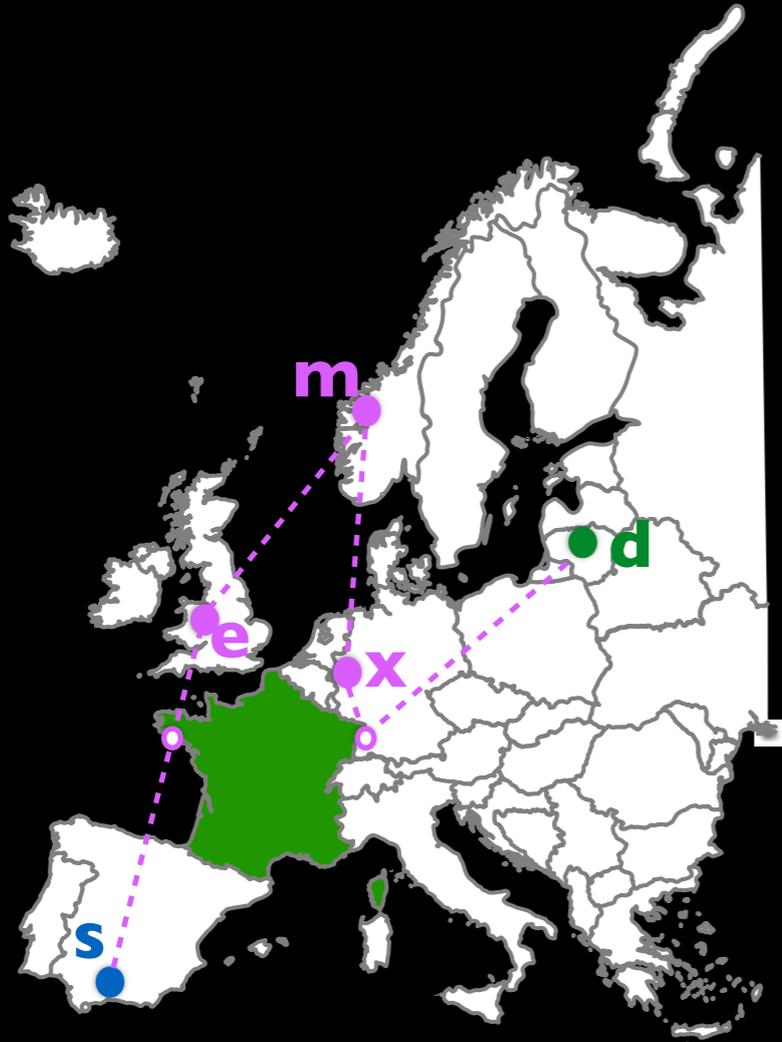


For each **country** intersects with both **ellipses**

Measured RTT  $\ll$  The shortest possible RTT thru Tor and entry & exit legs traverse

# DeTor: never-twice avoidance

Which countries can entry & exit legs reach



For each country intersects with both ellipses

Measured RTT  $\ll$  The shortest possible RTT thru Tor and entry & exit legs

The packet could not have traversed over entry & exit legs

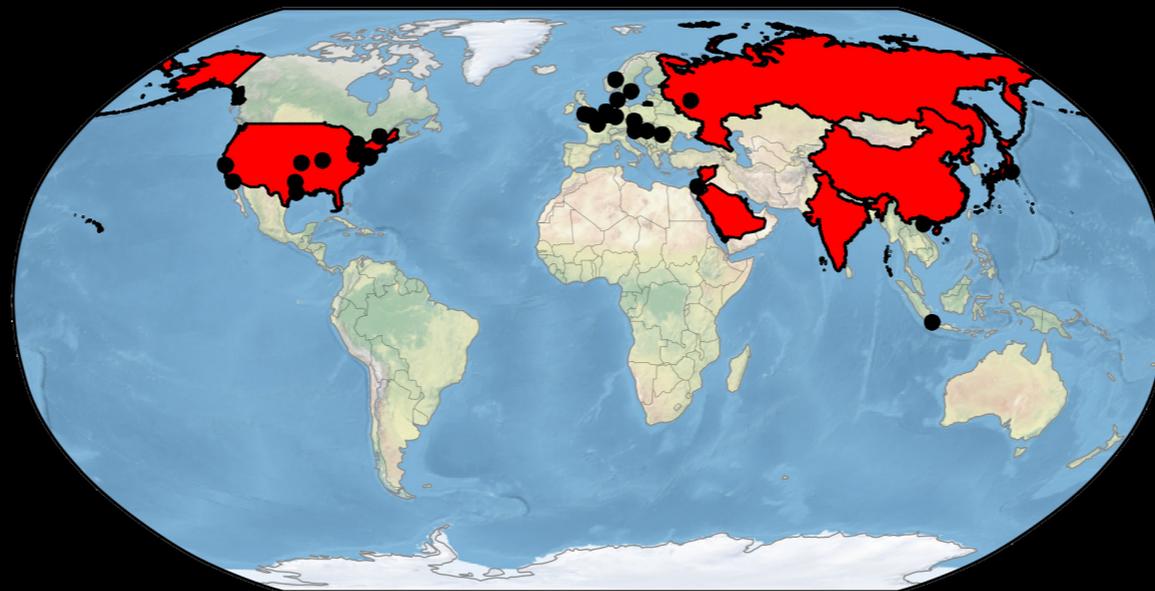
# Evaluation

Through simulation

# Evaluation

## Through simulation

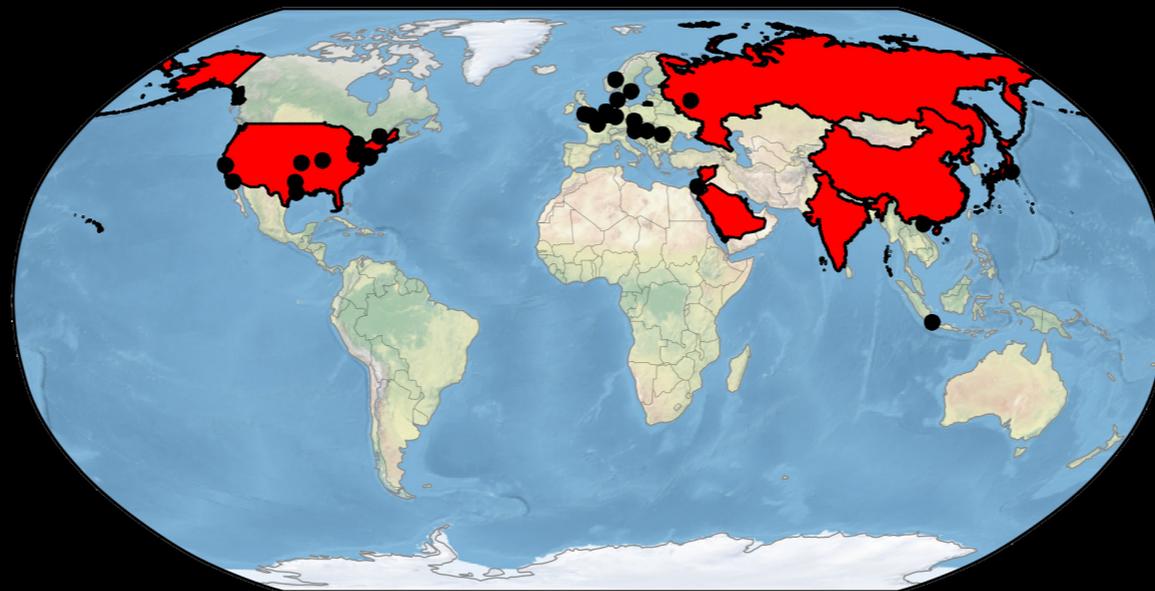
- 50 random real Tor nodes
  - with GPS locations and pair-wise RTTs using Ting
  - choose sources and destinations among these nodes



# Evaluation

## Through simulation

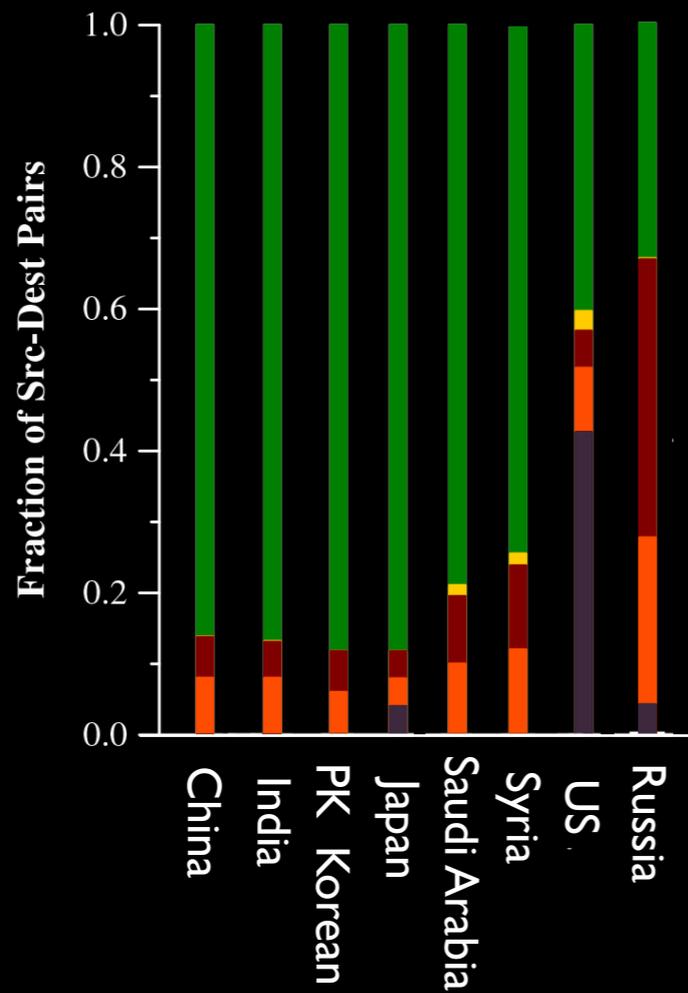
- 50 random real Tor nodes
  - with GPS locations and pair-wise RTTs using Ting
  - choose sources and destinations among these nodes



# Evaluation

- How successful is DeTor?
- How well do DeTor circuits perform?

# Never-once success rate



Successful with DeTor

Theoretically avoid, but failed with real RTTs

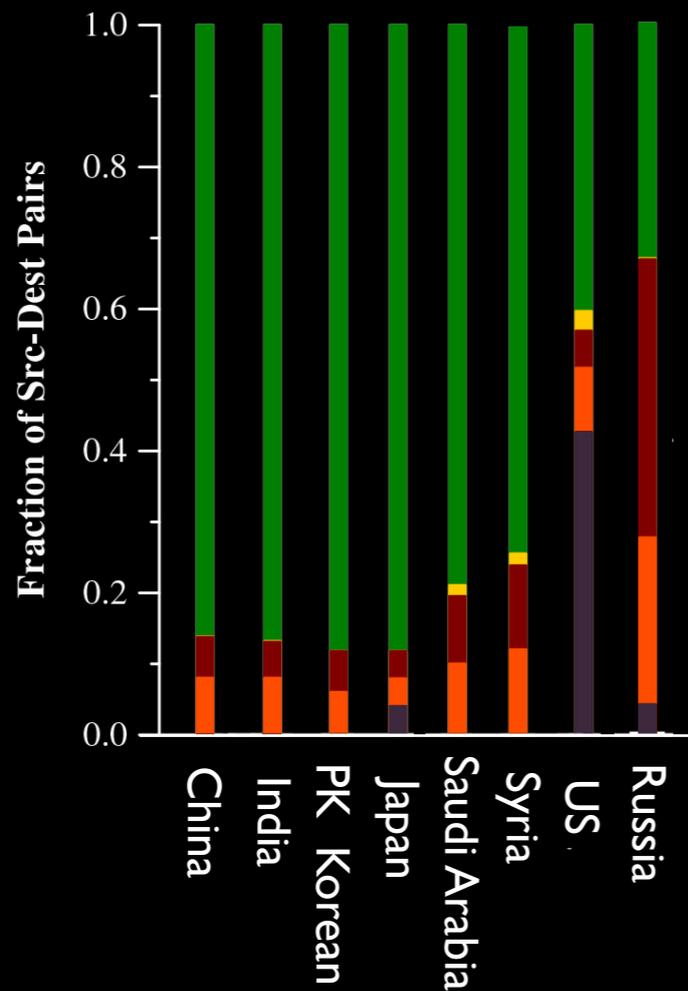
No circuits could provably avoid

No trusted Tor nodes

Source/Destination in Forbidden region

# Never-once success rate

Most src-dst pairs can successfully find never-once circuits



Successful with DeTor

Theoretically avoid, but failed with real RTTs

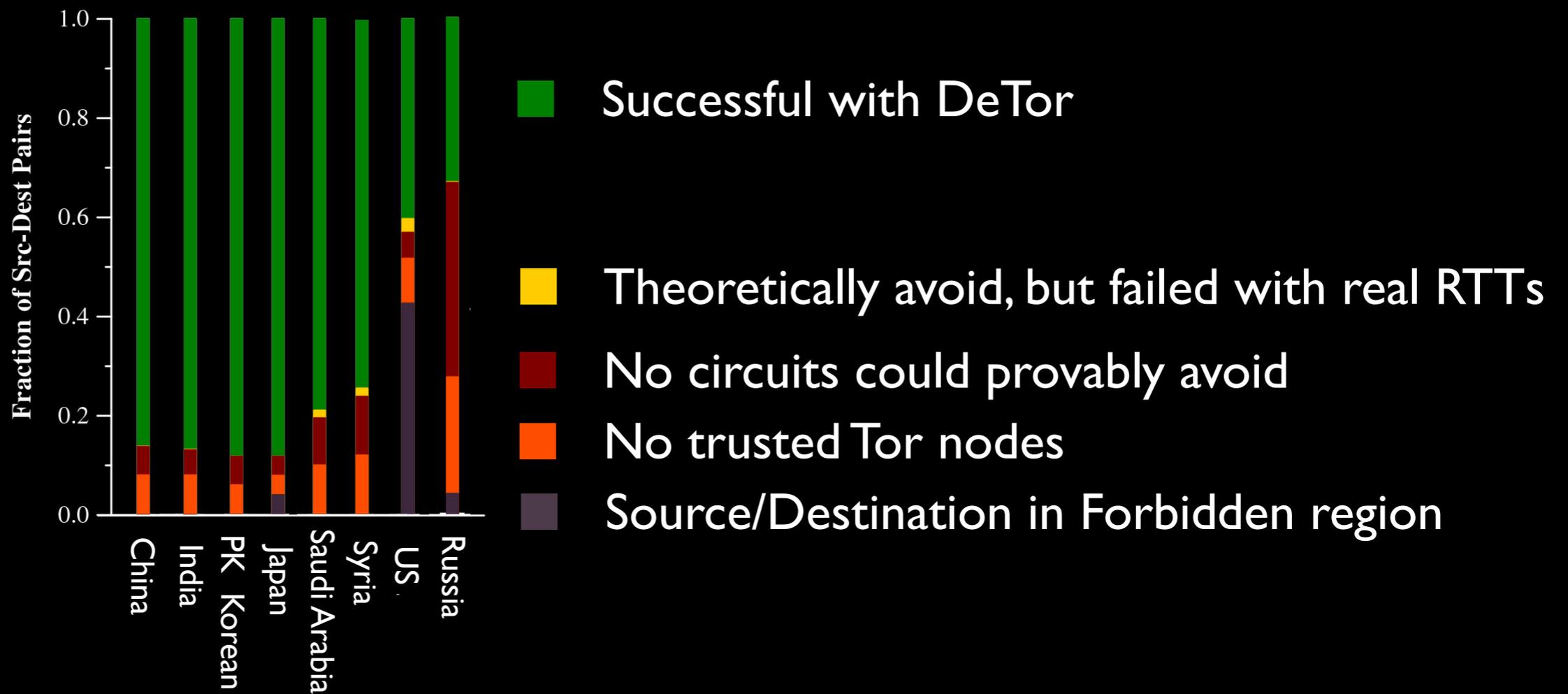
No circuits could provably avoid

No trusted Tor nodes

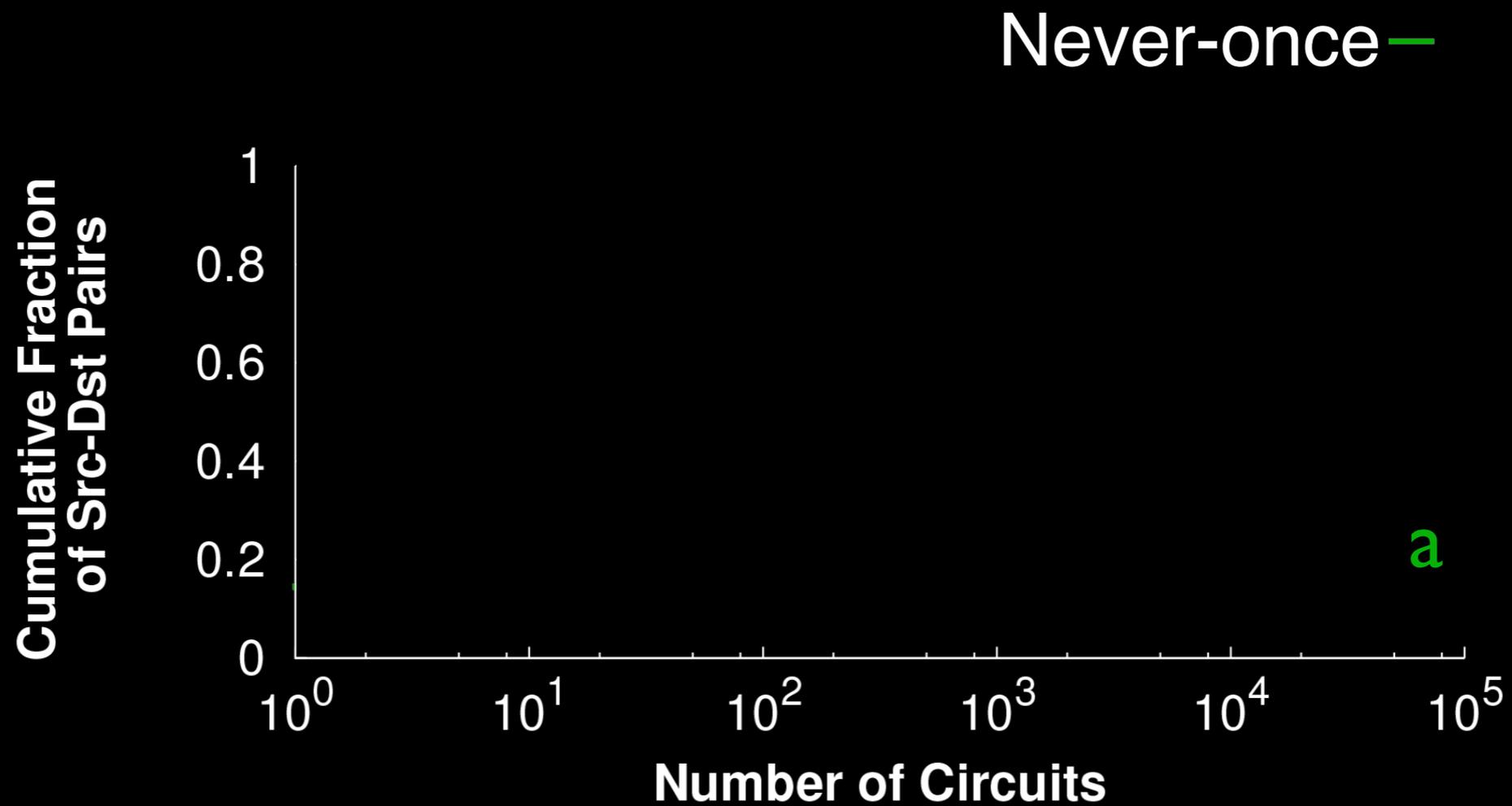
Source/Destination in Forbidden region

# Never-once success rate

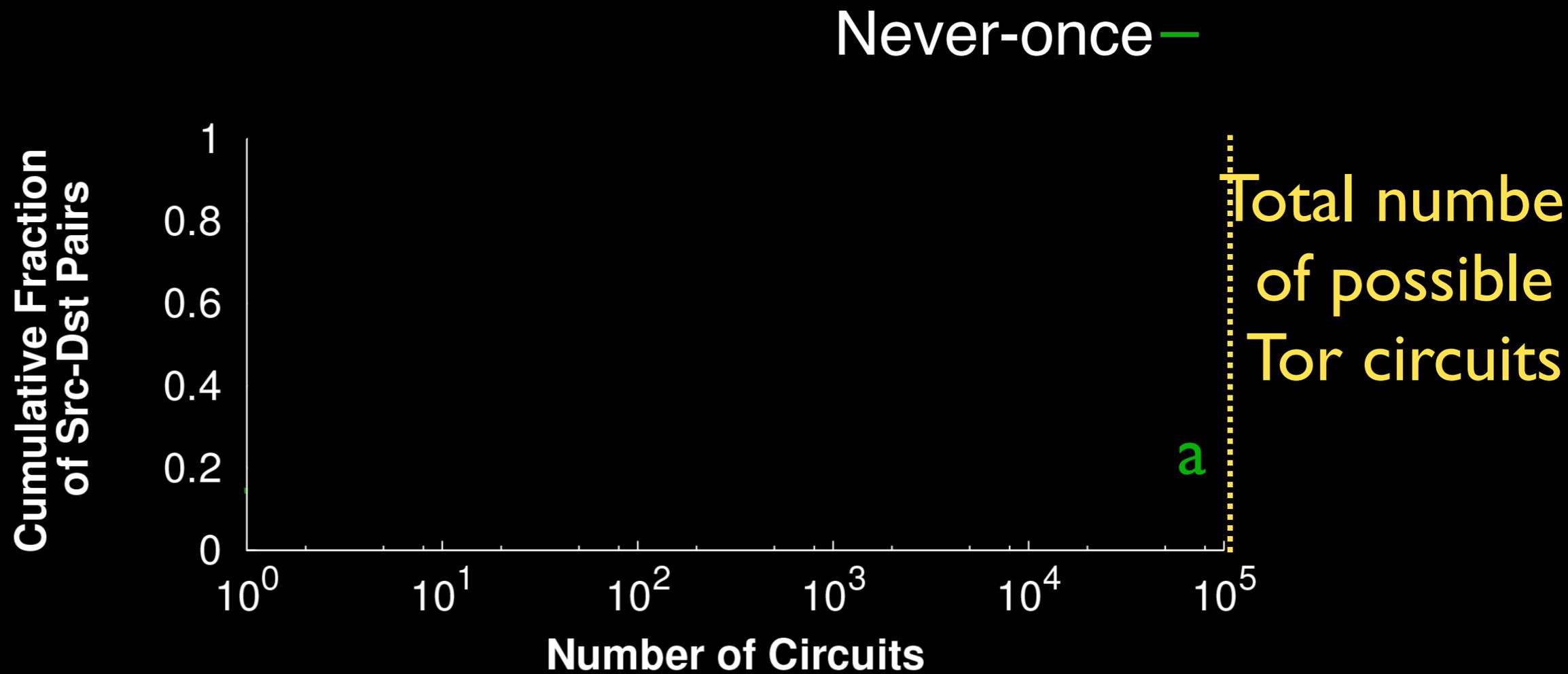
Failure typically arises when users are in or close to the regions to avoid



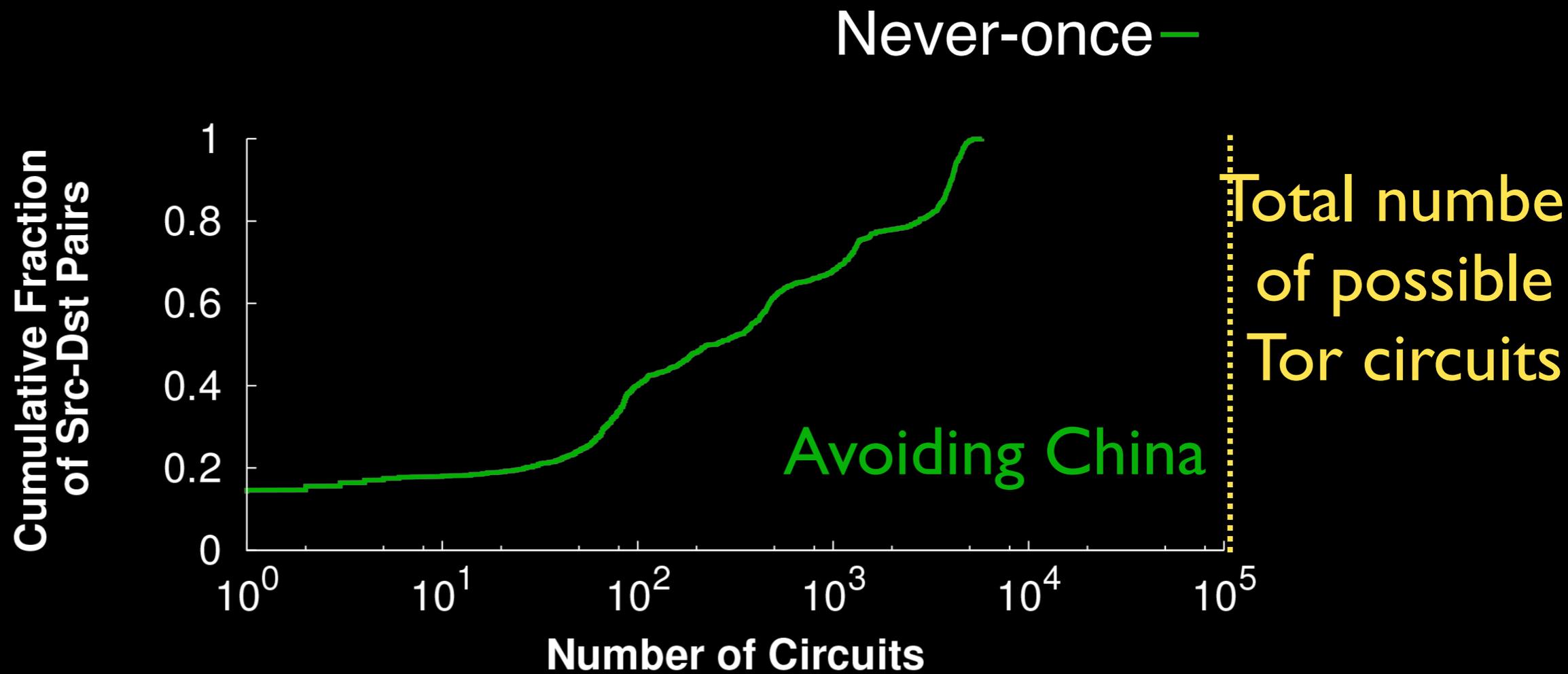
# Number of never-once circuits



# Number of never-once circuits

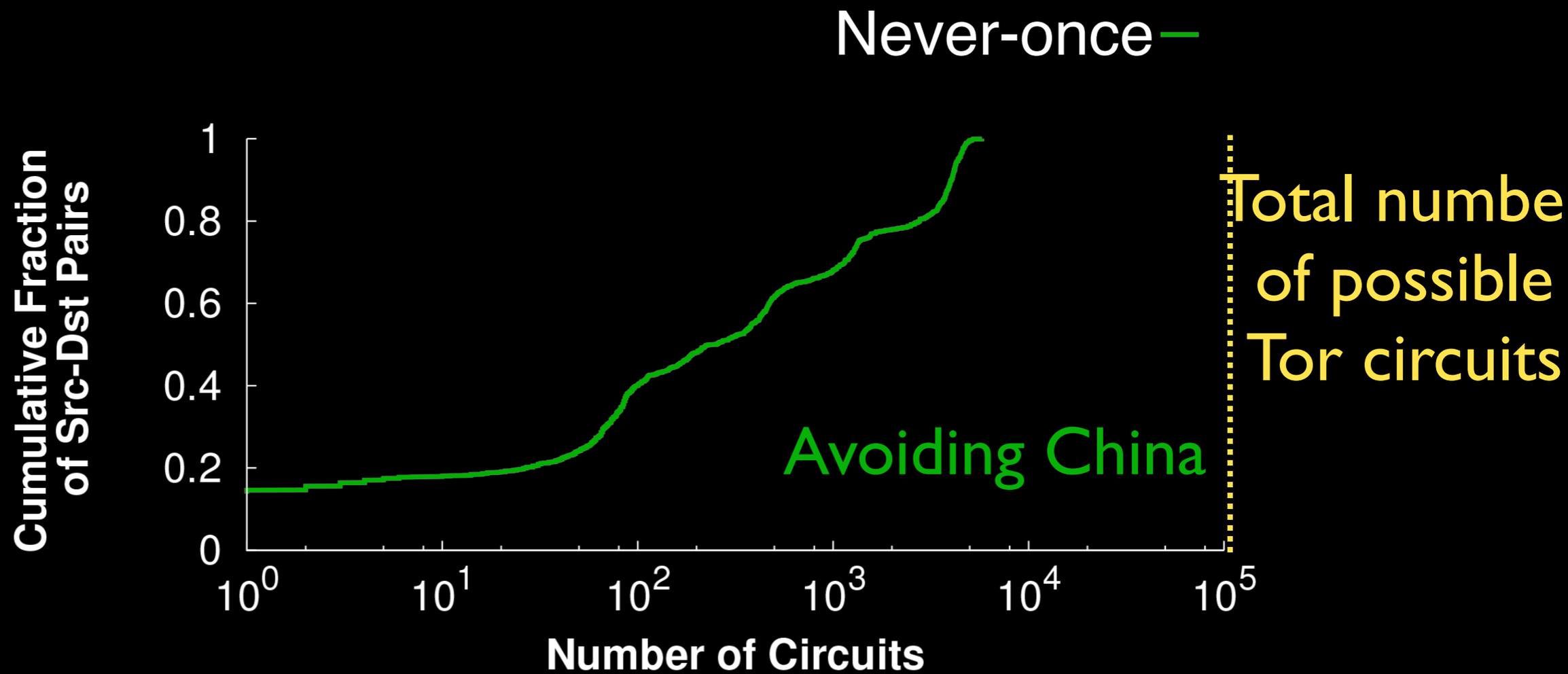


# Number of never-once circuits



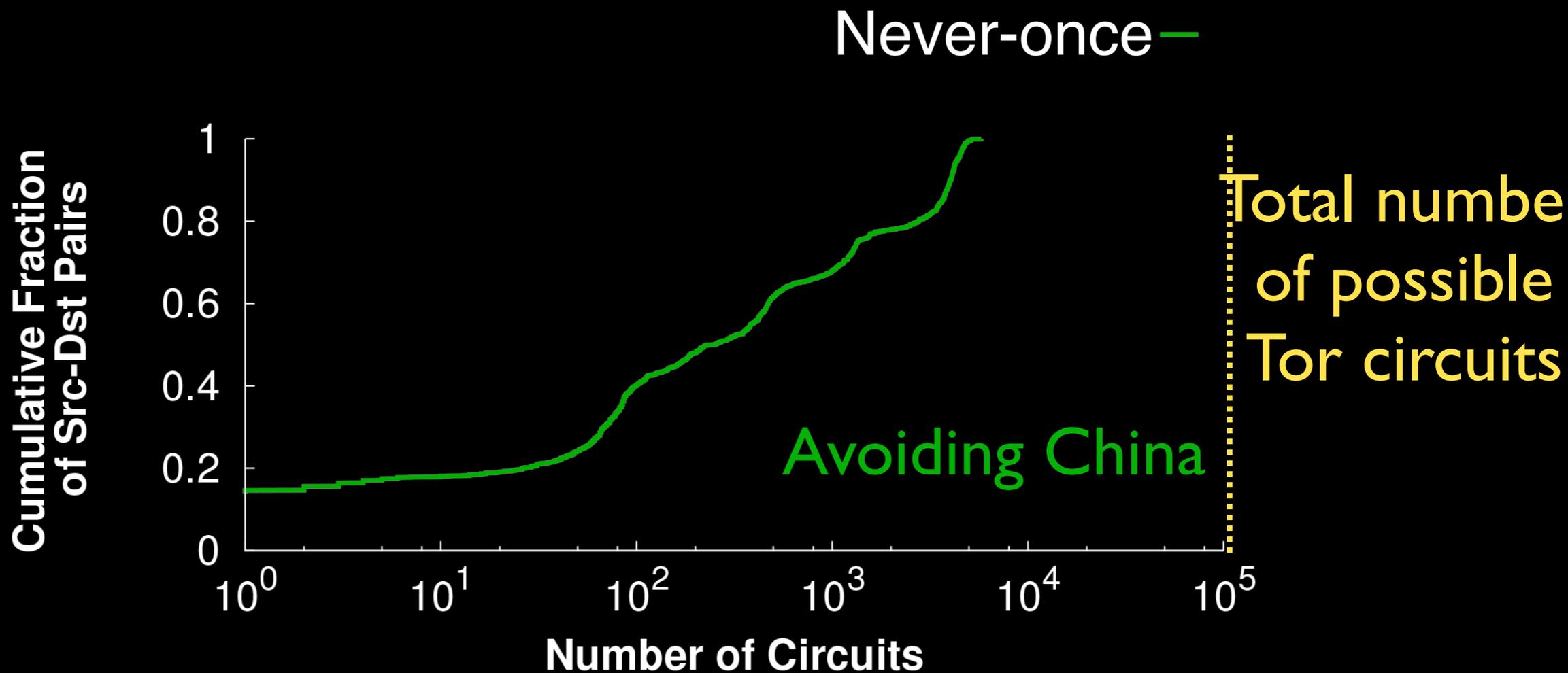
# Number of never-once circuits

Half of src-dst pairs have over 500 never-once circuits

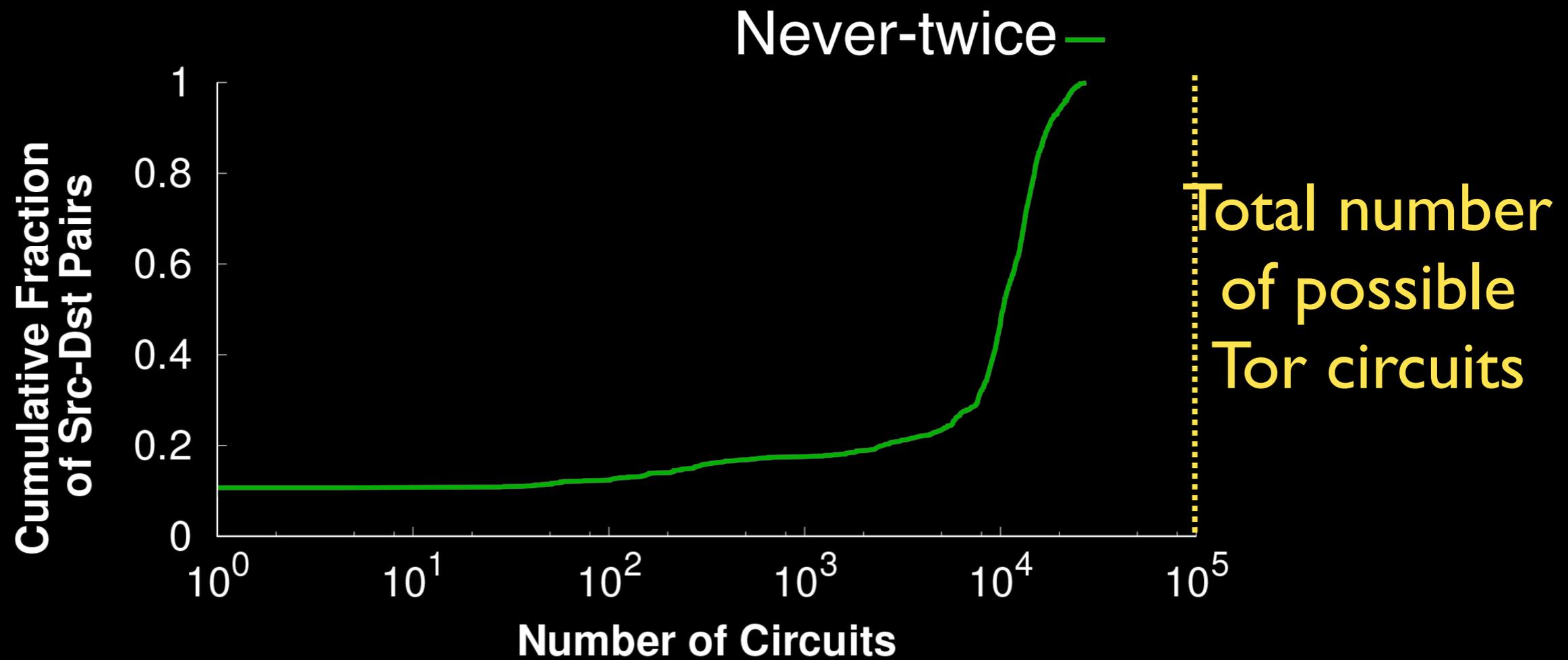


# Number of never-once circuits

Tor with no Chinese relays provably avoids China  
less than 10% of the time

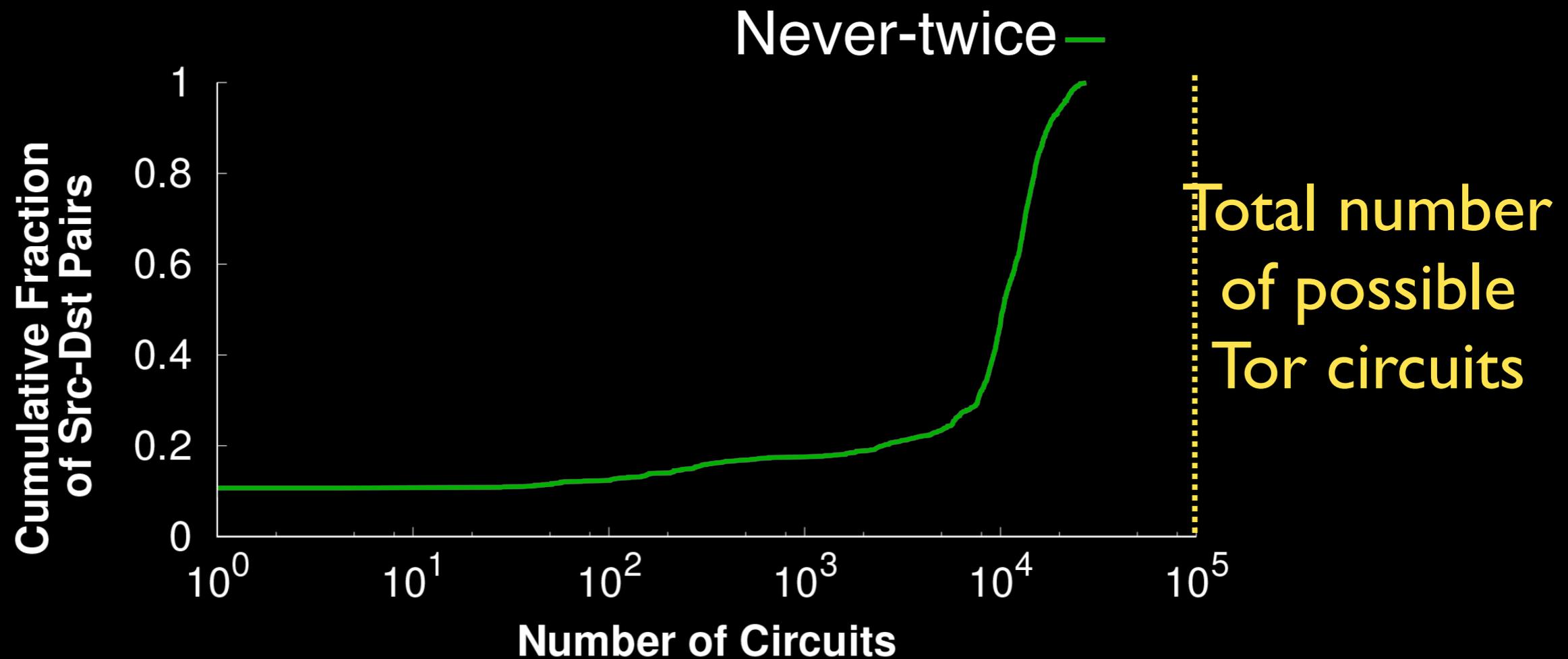


# Number of never-twice circuits

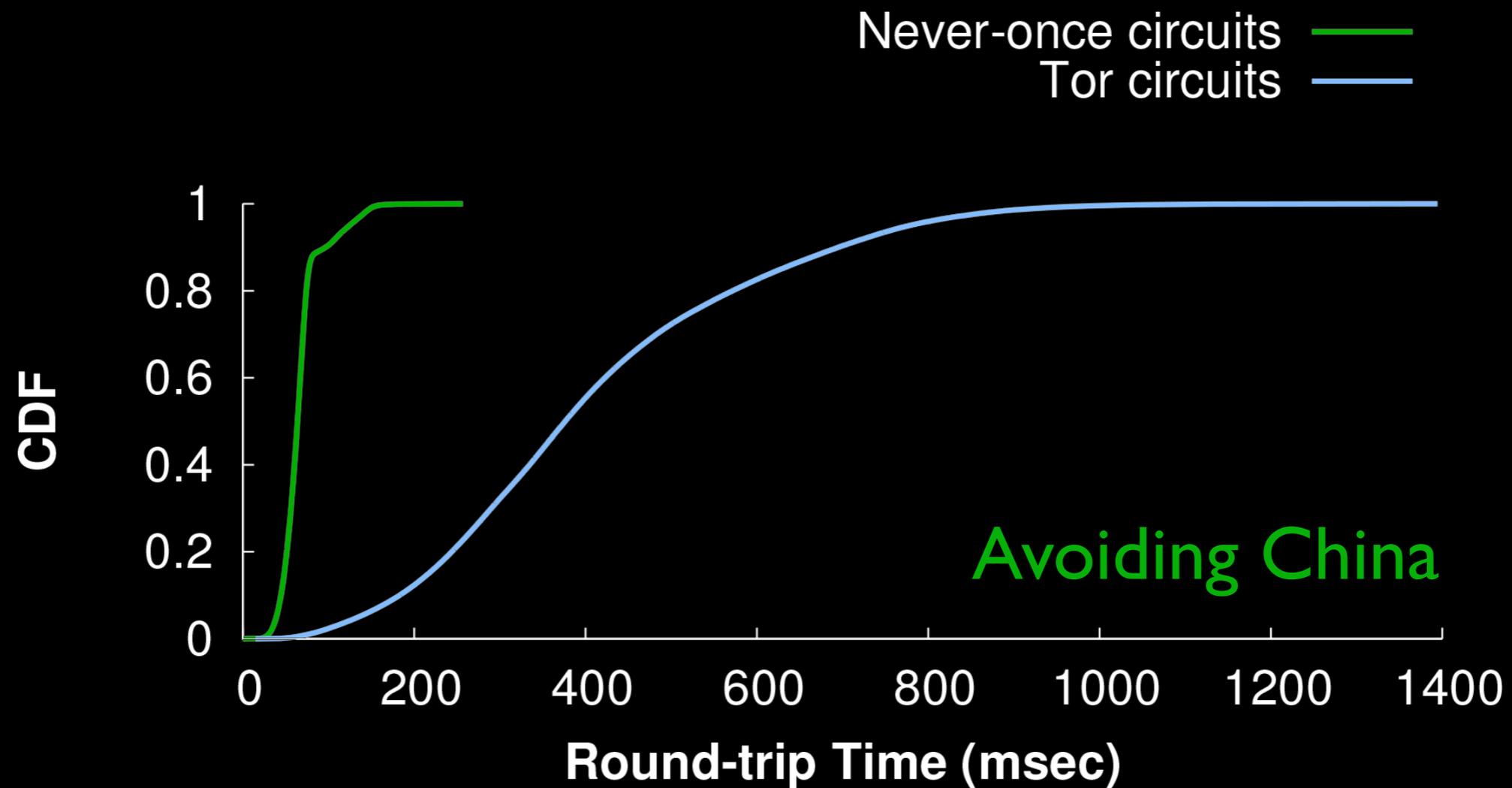


# Number of never-twice circuits

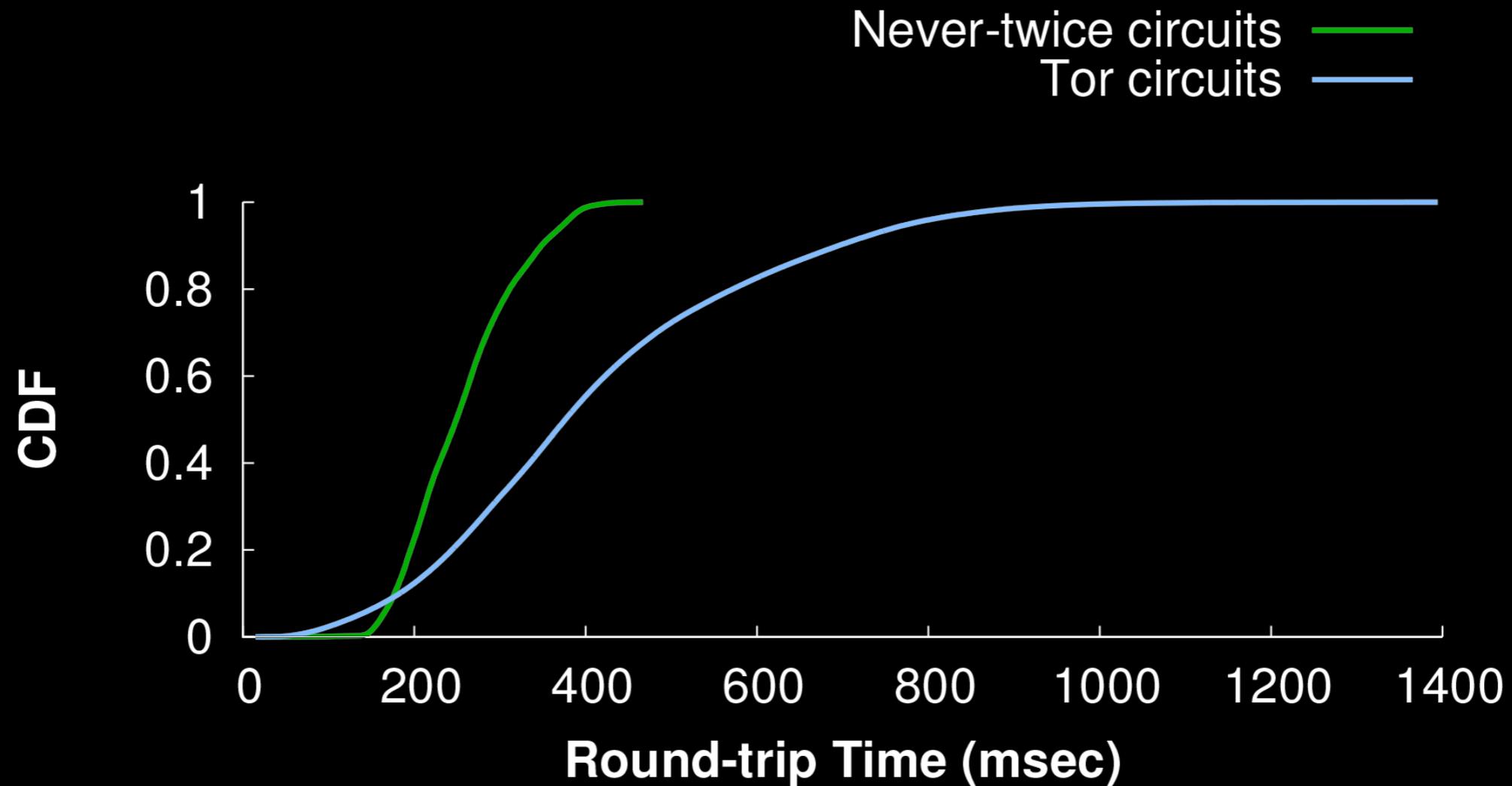
Client-side RTTs might be enough to address many attacks



# DeTor circuits tends to have lower RTTs



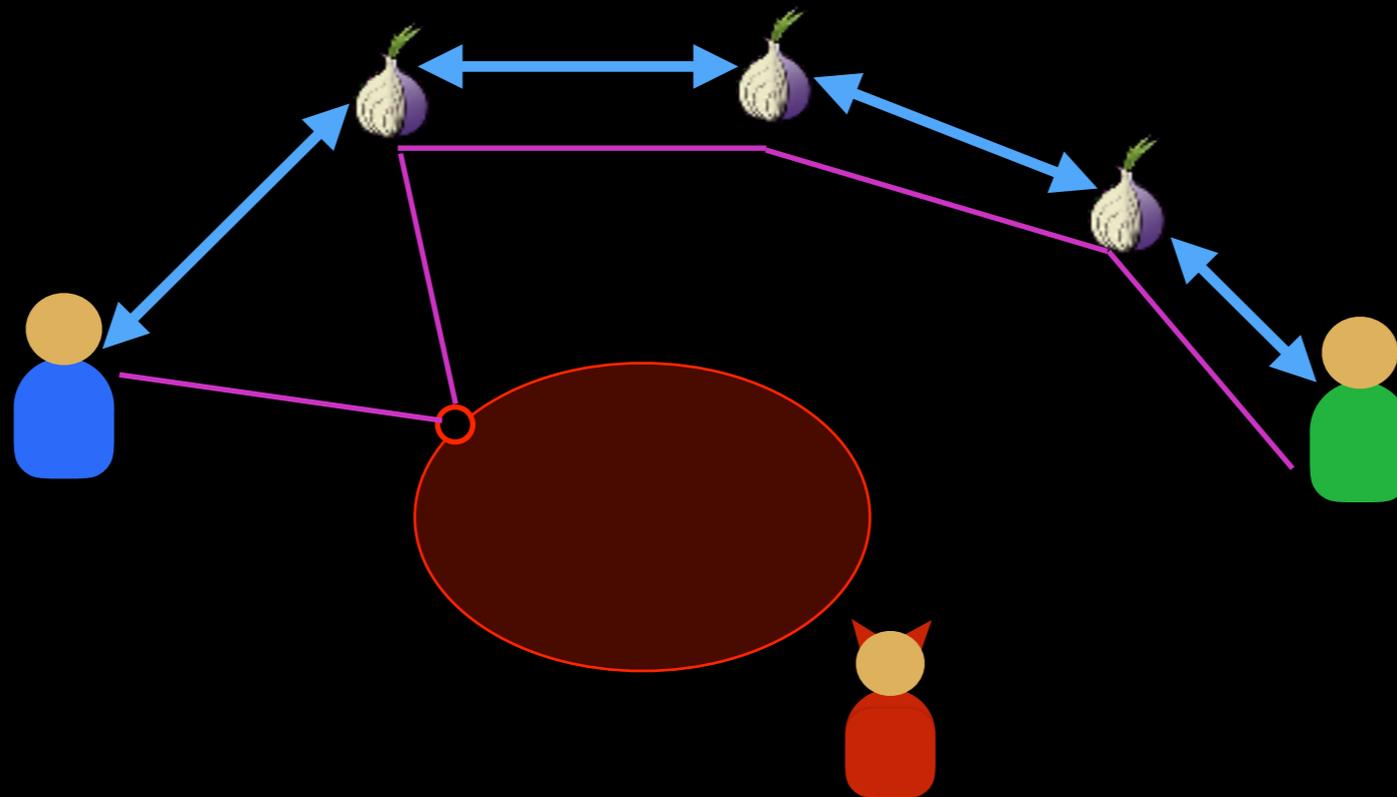
# DeTor circuits tends to have lower RTTs



# DeTor: never-once avoidance

Achieving provable avoidance

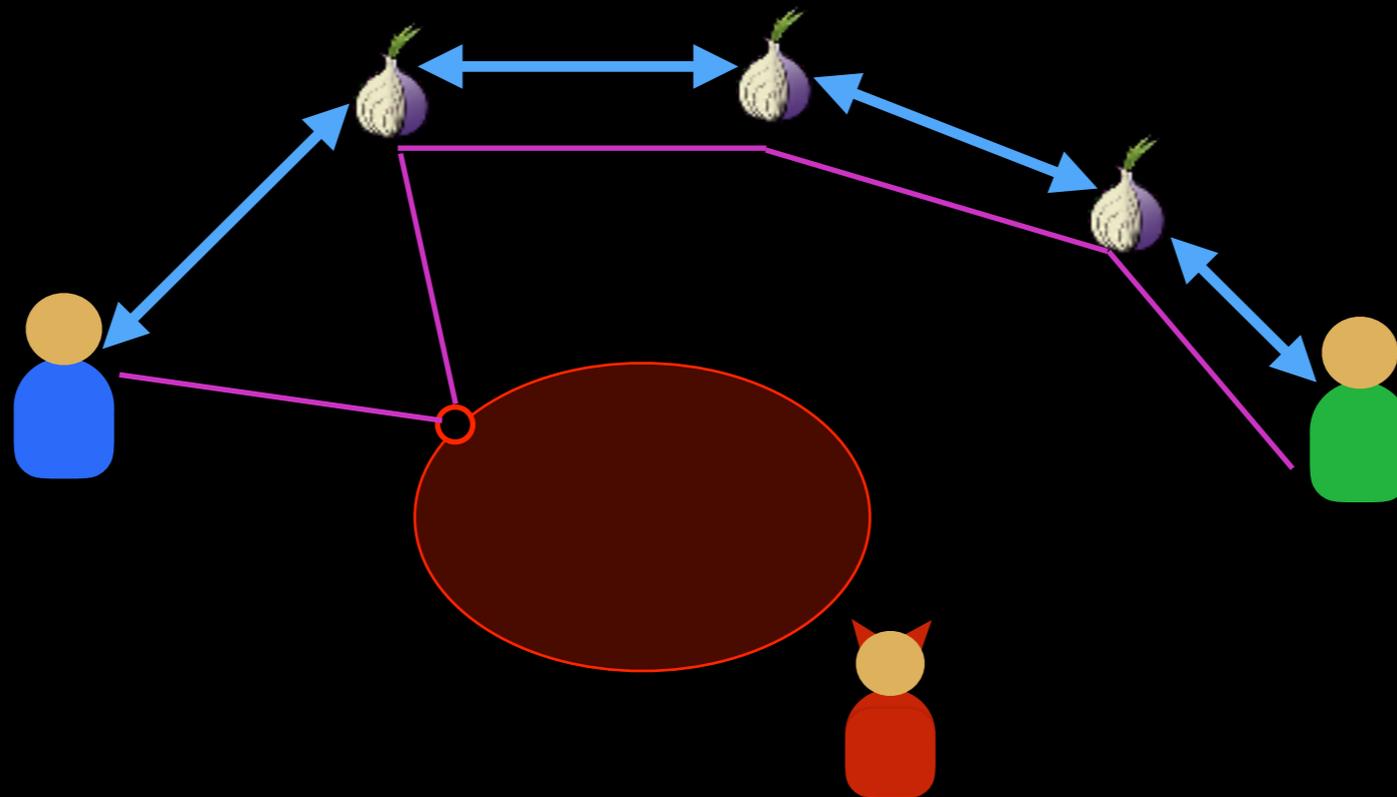
Measured RTT  $\ll$  The shortest possible RTT thru onion and  =  $2 \min\{d_i\} / c$



# DeTor: never-once avoidance

Achieving provable avoidance

Measured RTT  $\ll$  The shortest possible RTT  $= 2 \min\{d_i\} / c$   
thru onion and 



# Other results

- DeTor circuits usually have **higher bandwidth**
- DeTor introduces **slight node selection bias**
- Most nodes serve on few DeTor circuits
- Possible to **predict** whether a circuit will

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry & exit legs  
never traverse

- **Proofs of avoidance** verify that packets over DeTor circuits have avoided geographic regions

- DeTor circuits

- are **successful** for most src-dst pairs
- have **better performance**
- introduce small node selection bias

Code and data available at:  
[deter.cs.umd.edu](http://deter.cs.umd.edu)

# DeTor

With **smart circuit selection**, it is possible to *provably* avoid geographic regions with Tor

Never-once

never traverse  
specified regions

Never-twice

entry & exit legs  
never traverse

- **Proofs of avoidance** verify that packets over DeTor circuits have avoided geographic regions

- DeTor circuits

- are **successful** for most src-dst pairs
- have **better performance**
- introduce small node selection bias

Code and data available at:  
[detor.cs.umd.edu](http://detor.cs.umd.edu)