

CORE

P2P-Based Connectionless Onion Router

Olaf Landsiedel

Protocol-Engineering & Distributed Systems Group

University of Tübingen

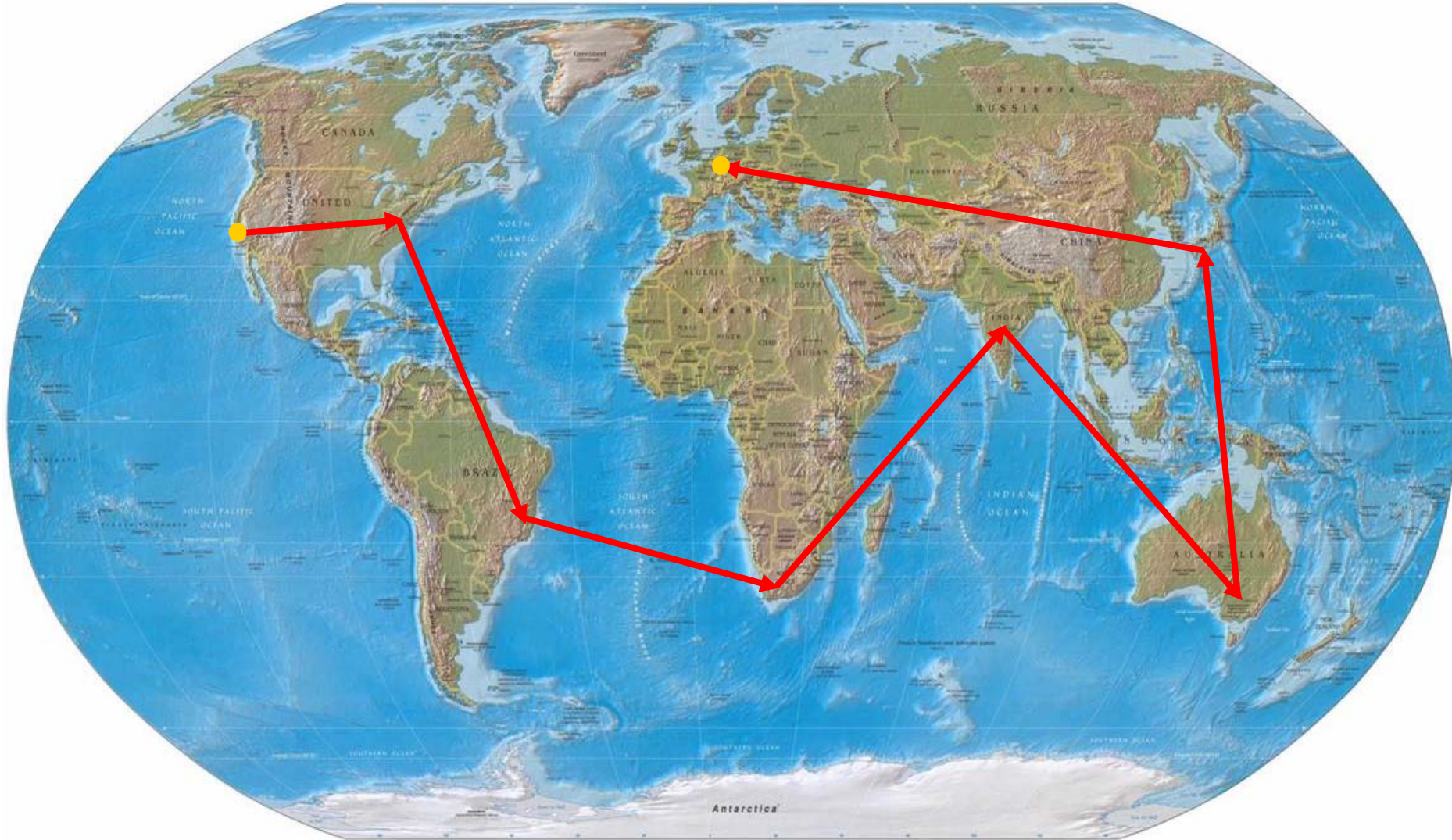
<http://ps.ri.uni-tuebingen.de>

- **Anonymous communication is crucial these days**
 - ▶ Reduced privacy
- **Onion Routing**
 - ▶ Near real-time communication
- **Today's Onion Routing**
 - ▶ Forward packets along one static path
 - ▶ Susceptible to attacks
 - Pattern matching

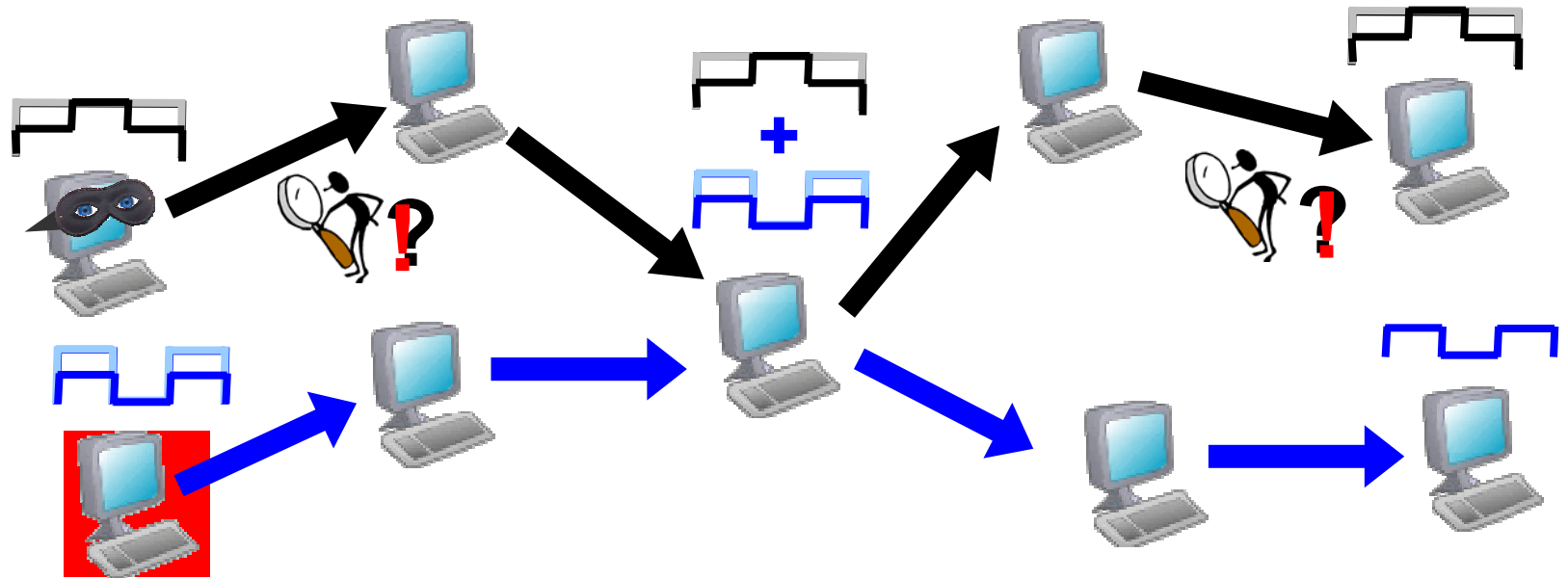
Outline

- **Background**
 - ▶ Onion Routing
 - ▶ Pattern matching attack
- **Core architecture**
 - ▶ Path concatenation scheme
 - ▶ Key cache
 - ▶ Transparent application support
- **Example: anonymous web-browsing**
- **Security analysis**
- **Implementation**
 - ▶ Performance
 - ▶ Challenge: make TCP and CORE friends
- **Conclusion**

Onion Routing: Idea



Pattern Attack



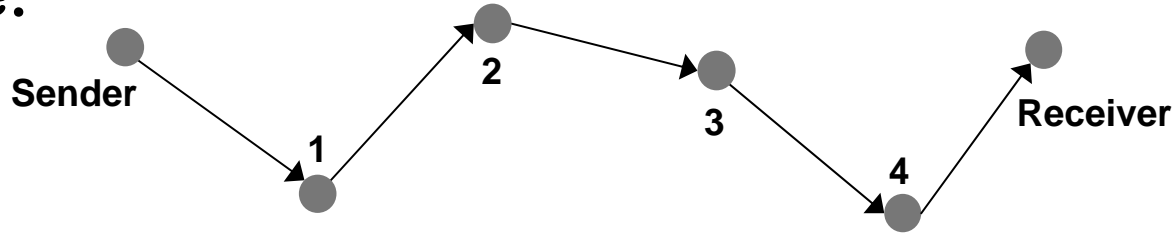
- “Low-Cost Traffic Analysis of Tor”
 - ▶ (S. J. Murdoch and G. Danezis)
- Problem: static route
- Our solution: one route per packet

Observations

- Immune to these pattern matching attacks
- **This cannot work...**
 - ▶ Asymmetric cryptography
 - Performance bottleneck
 - Bandwidth bottleneck
 - ▶ Highly dynamic round-trip time
 - TCP over dynamic overlay
- **...but it does!**

Core Routing

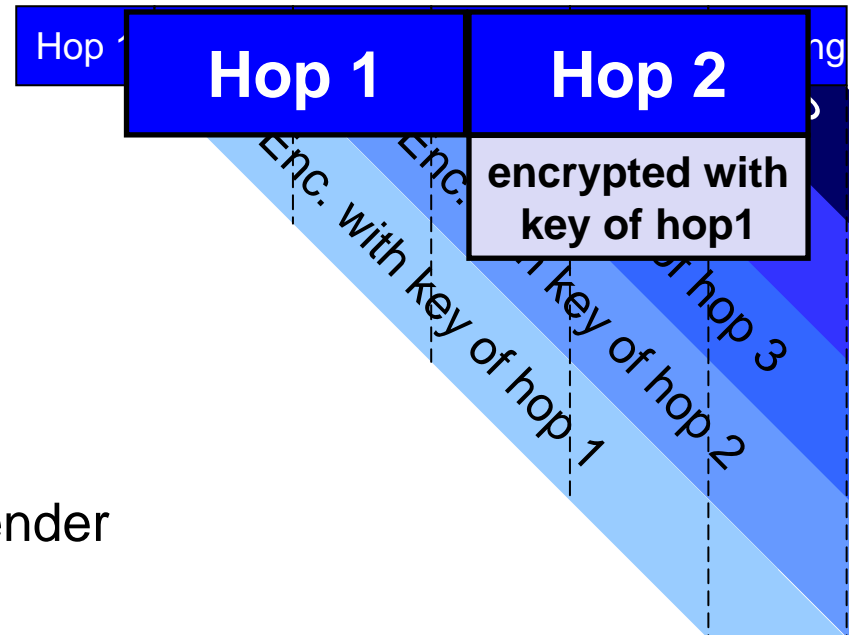
- **Example:**



- **Sender selects an anonymous path**

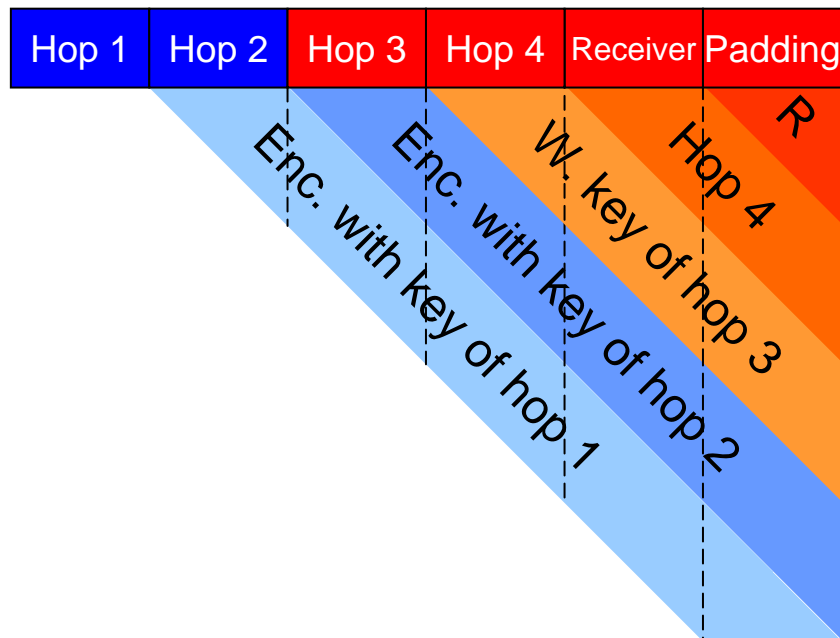
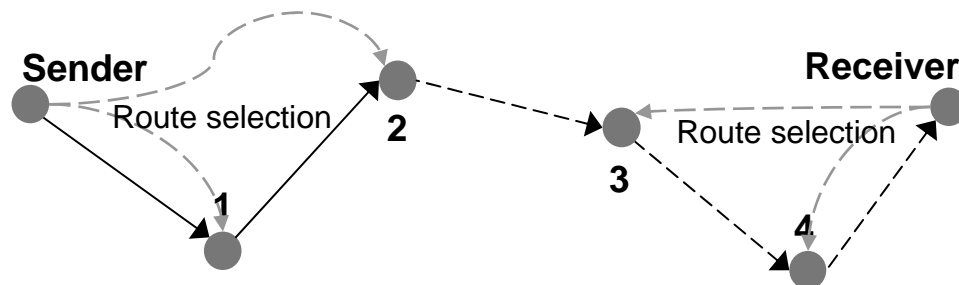
- **Layered encryption**

- ▶ One hop can only decrypt its successor
- ▶ Each hop removes a layer of encryption
- ▶ Intermediate nodes and receiver have no information about the sender
- ▶ Padding



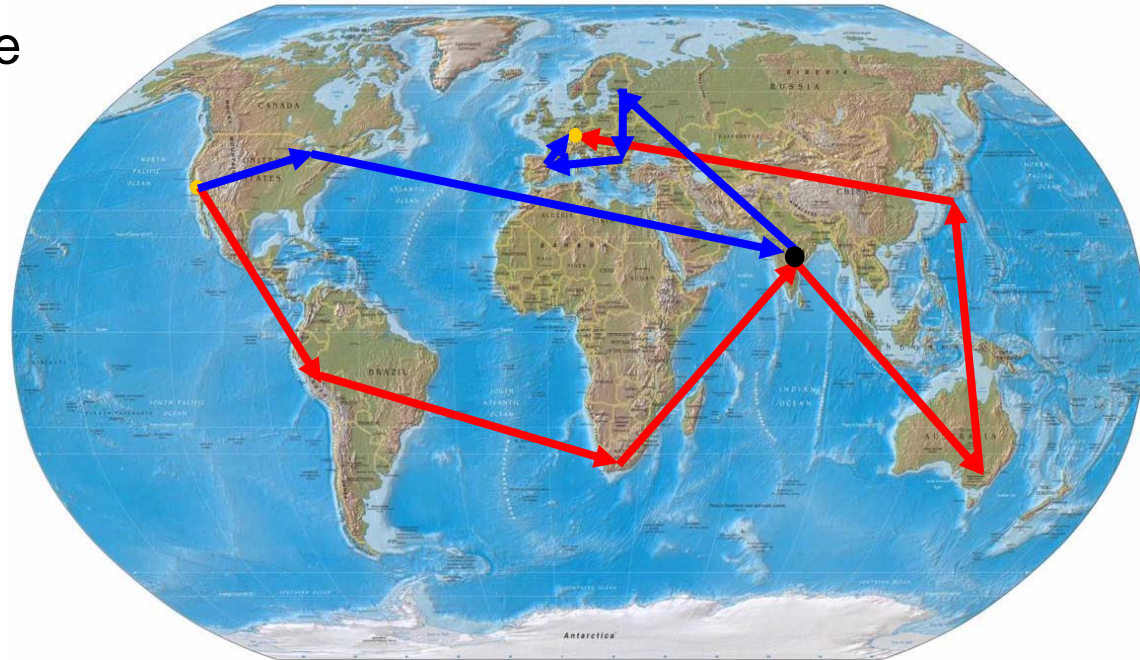
Receiver Anonymity

- **Sender and relationship anonymity**
 - ▶ No receiver anonymity, yet
- **Path selection**
 - ▶ Head by sender
 - ▶ Tail by receiver
- **Receiver publishes**
 - ▶ Path entry point
 - ▶ Path as layered encryption
- **Sender concatenates to anonymous path**
 - ▶ Piggyback return path



Overhead

- **Bandwidth overhead**
 - ▶ 2048 bit RSA keys: 256 byte per hop
 - ▶ 192 bit ECC keys: 28 byte per hop
- **Computational overhead**
 - ▶ Asymmetric cryptography is computationally expensive
 - To create symmetric keys
 - ▶ Key caching
 - Reuse symmetric key

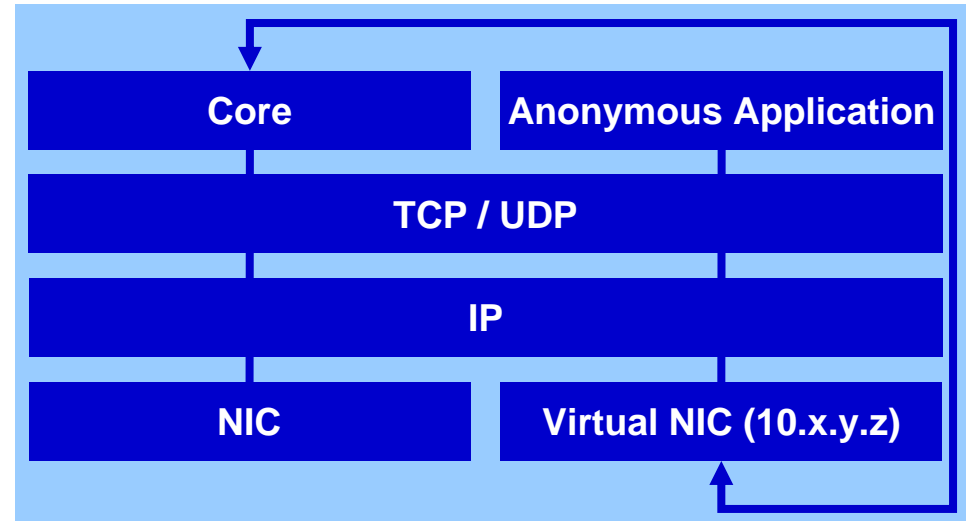


- **Transparent Application Support**

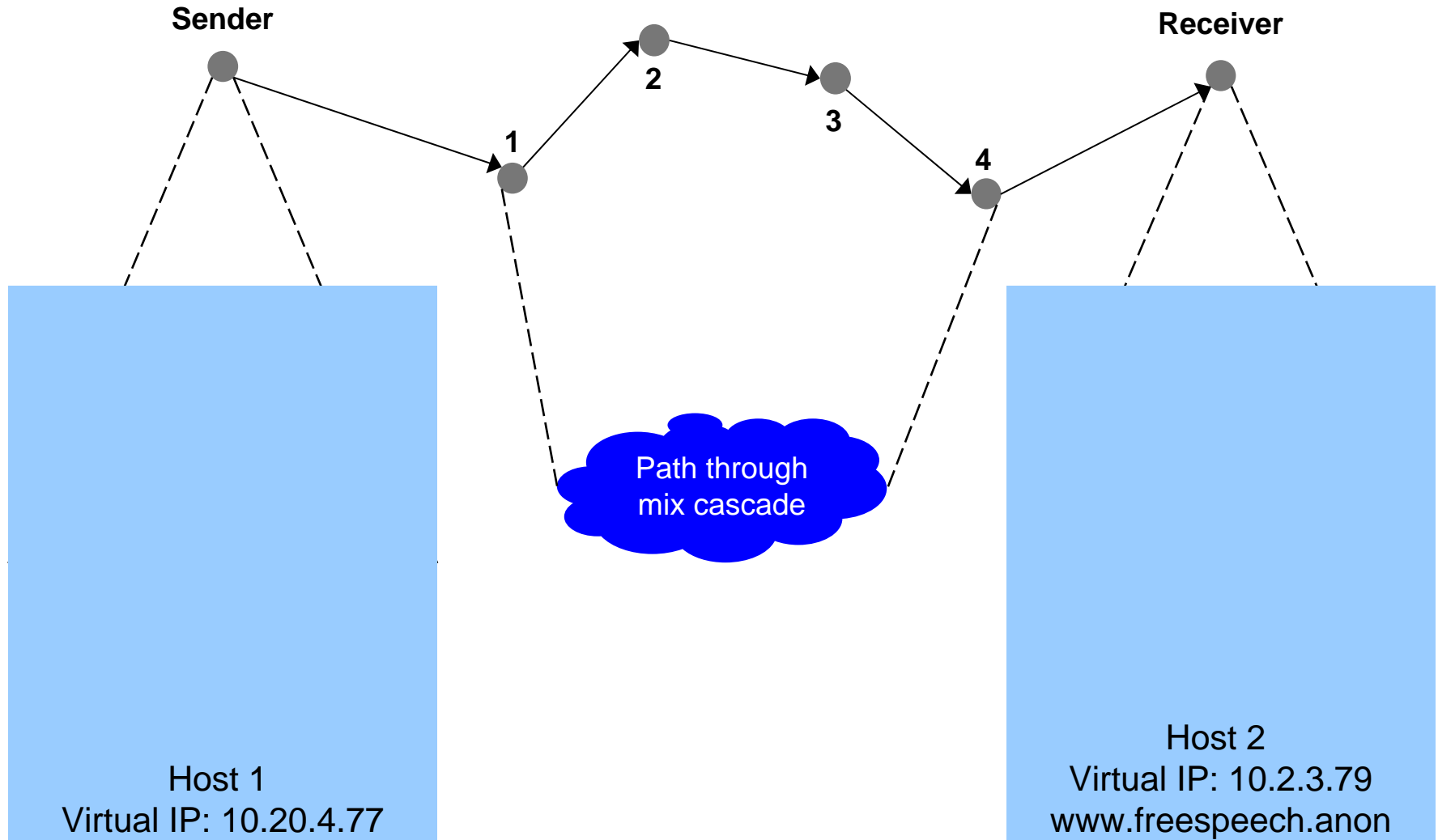
- ▶ Virtual Network Interface
 - IP level tunneling
- ▶ Legacy support
 - No changes to applications

- **Service Directory**

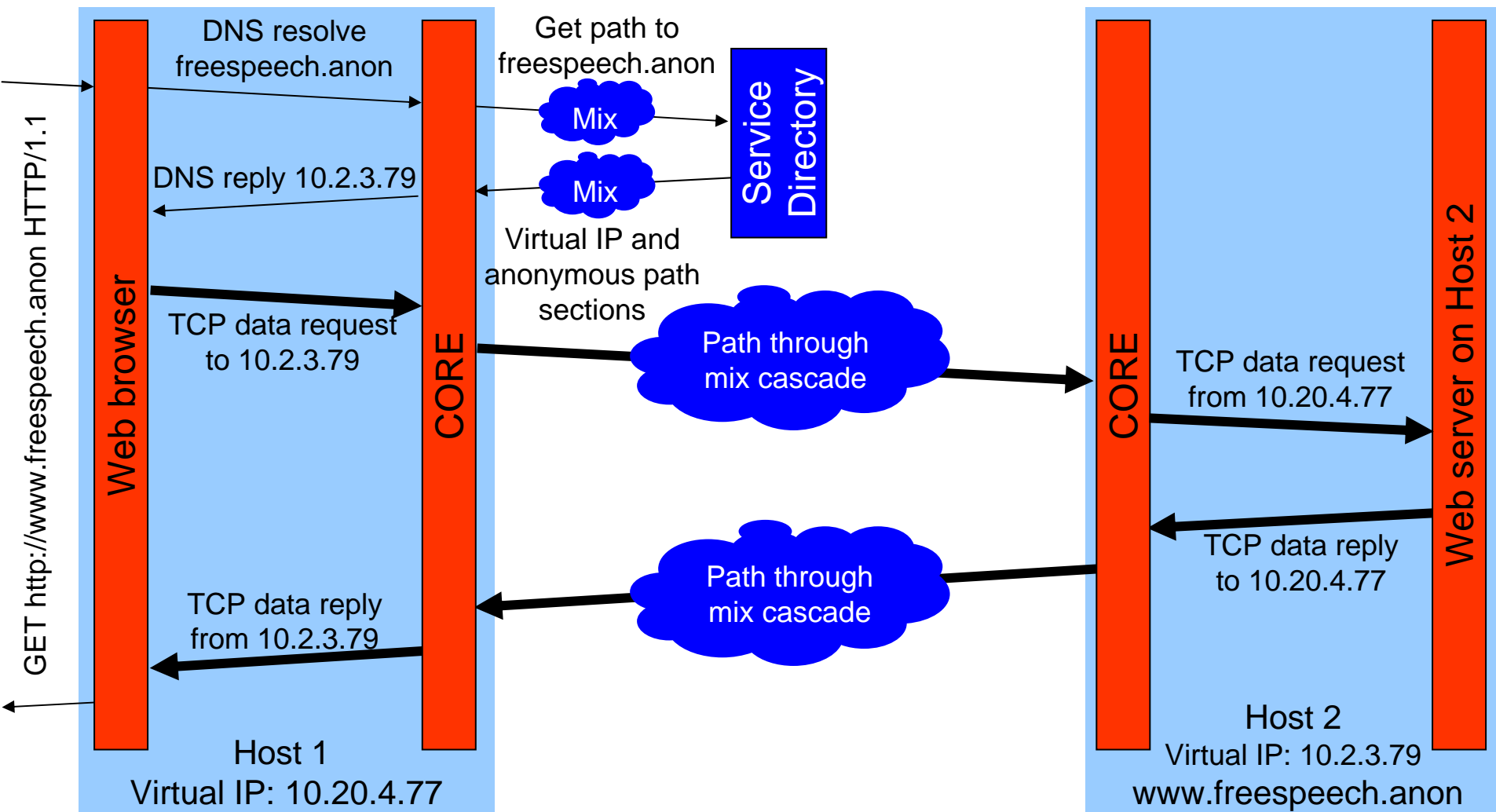
- ▶ Provides participating hosts
- ▶ Assigns IP addresses
- ▶ Provides anonymous path sections
- ▶ Move to DHT...



Example: Using a Web Browser



Example: Using a Web Browser



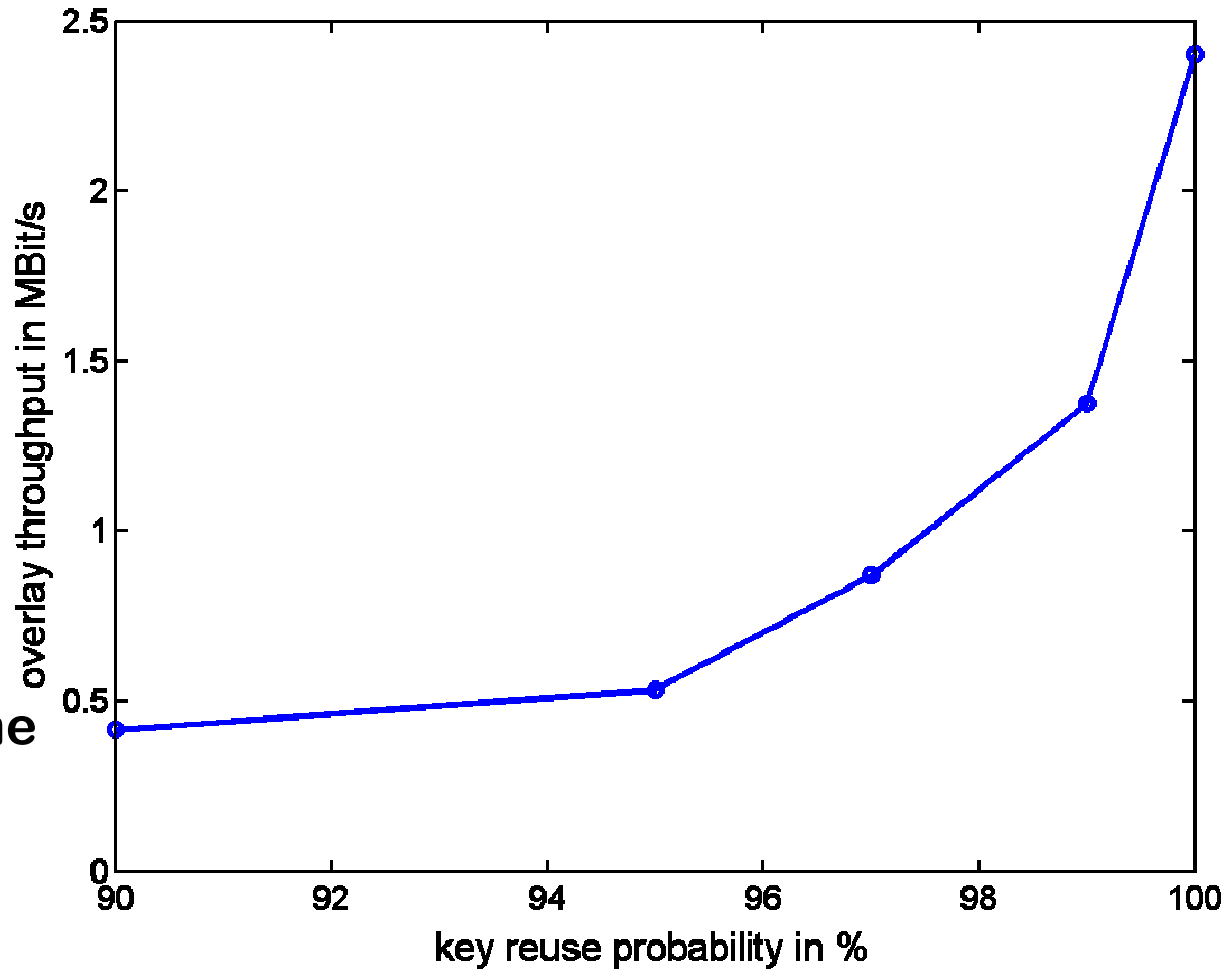
- **Practical adversary**

- ▶ No global observer
 - Observe only some part of the network
- ▶ Participate actively
 - Relaying traffic of other nodes
 - Offer service, e.g. web server
 - Access content
- ▶ Compromise a limited number of nodes
- ▶ Influence communications
 - Generating,
 - Delaying,
 - Modifying traffic content and patterns

- **Mitigating source / destination observation**
 - ▶ Traffic relaying
 - ▶ Message padding to constant length
 - ➡ It is not possible to determine via observation whether a node is sender, relay or receiver of a message
- **Mitigating pattern attacks**
 - ▶ Breaking pattern through multi-path routing
- ...

Evaluation

- **Testlab:**
 - ▶ 8 machines
 - ▶ Sender and receiver path length
 - Each 3 hops
 - ▶ Cell size: 700 bytes
- **Overlay throughput**
 - ▶ Up to 2.5 Mbit/s TCP
 - ▶ 5 Mbit/s UDP
- **Dynamic round trip time**
 - ▶ High CPU load on end-hosts
- **Out of order delivery**



Can CORE and TCP Become Good Friends?

- **PlanetLab!**

- ▶ Expect highly dynamic RTT
- ▶ Challenge to TCP
 - May need to smoothen TCP packet playout
 - Reduce dynamics of RTT
 - Reduce “out of order delivery”
 - Reduce fast retransmissions
 - Per hop acknowledgements
- ▶ Ongoing work

- **Lessons learned**

- ▶ Connectionless Onion Routing is possible
- ▶ TCP connections over such dynamic RTTs are possible
 - Improve throughput by tuning
- ▶ Asymmetric key overhead
 - Bandwidth
 - Low due to elliptic curve cryptography
 - Computational
 - Key caching

- **Future work**

- ▶ PlanetLab!
- ▶ 100 bit ECC keys instead of 192 bit
 - 16 byte per hop overhead

Conclusion

- **This cannot work...**
 - ▶ Asymmetric cryptography
 - Performance bottleneck ✓
 - Bandwidth bottleneck ✓
 - ▶ Highly dynamic round-trip time
 - TCP over highly dynamic overlay
 - Testlab ✓
 - Planetlab ?

Time for questions



<http://ps.ri.uni-tuebingen.de>

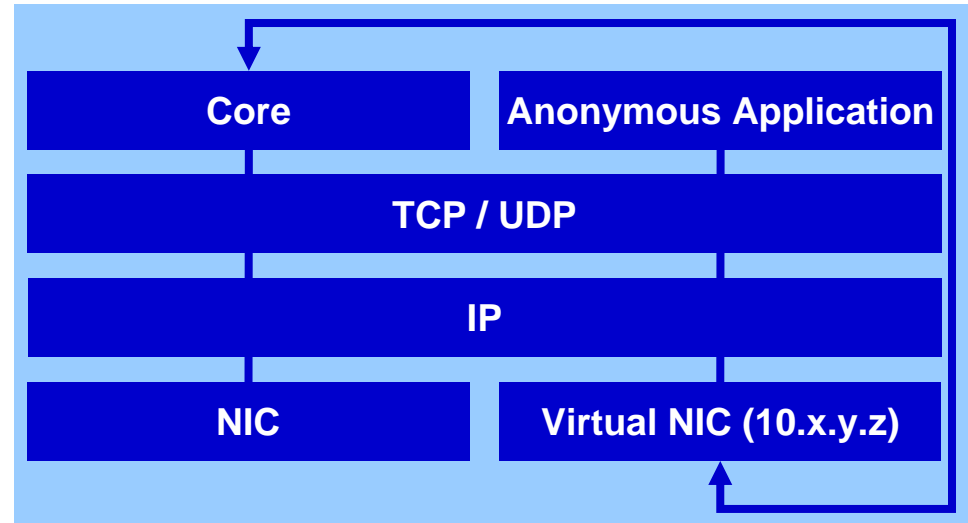
- **Treat the mix network as black box**
 - ▶ “Message Splitting Against the Partial Adversary”
(A. Serjantov and S. Murdoch)

- **Service Directory**

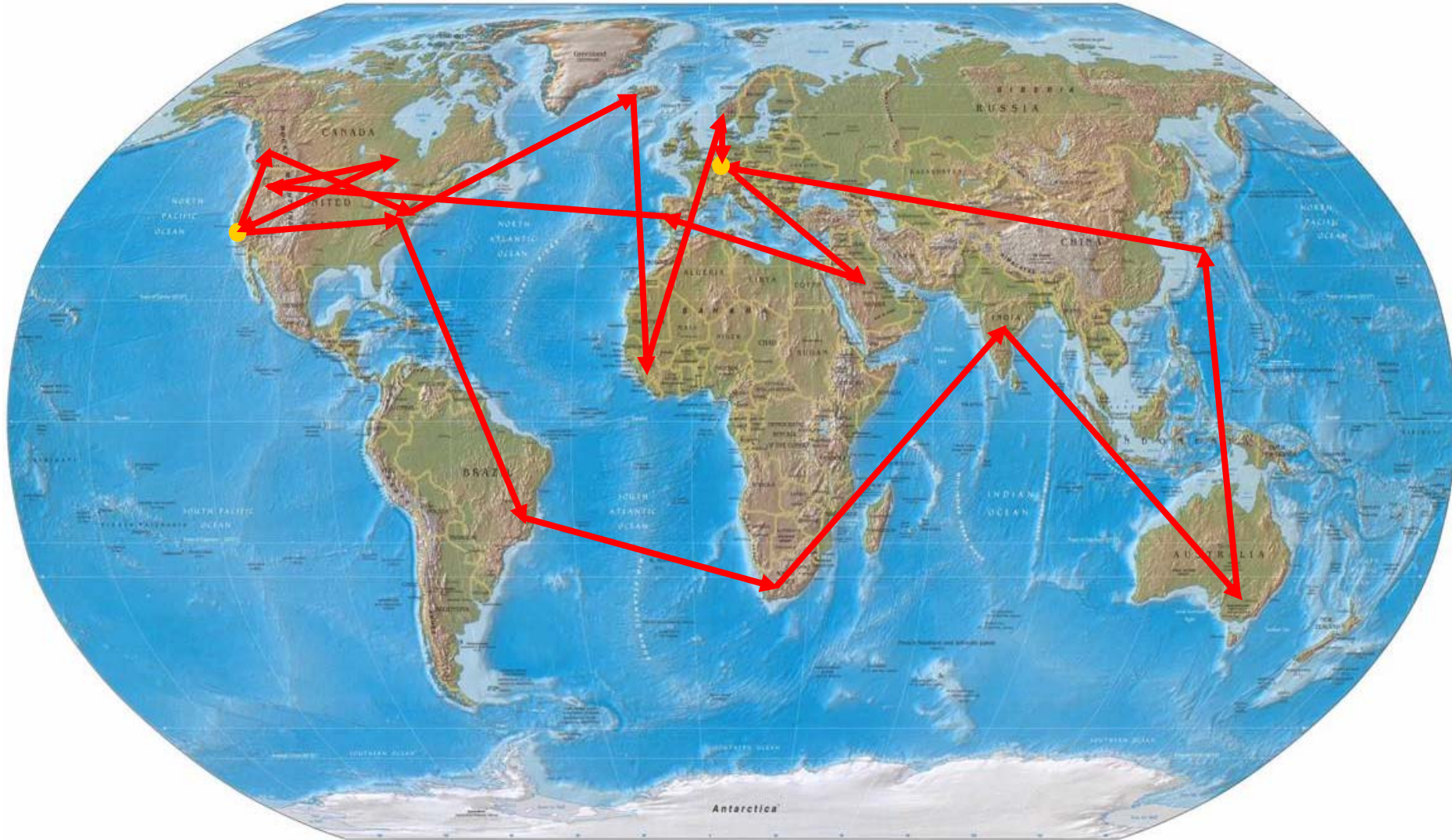
- ▶ Provides participating hosts
- ▶ Assigns IP addresses
- ▶ Provides anonymous path sections
- ▶ Move to DHT...

- **Transparent Application Support**

- ▶ Virtual Network Interface
 - IP level tunneling
- ▶ Legacy support
 - No changes to applications
- ▶ In-band signaling
- ▶ Enhancement via proxy possible



Solution: Connectionless Onion Routing



Session based!

- **Various Systems**

- ▶ Tarzan
- ▶ Tor
- ▶ MorphMix
- ▶ ...

- **(Most) enable sender and receiver anonymity**

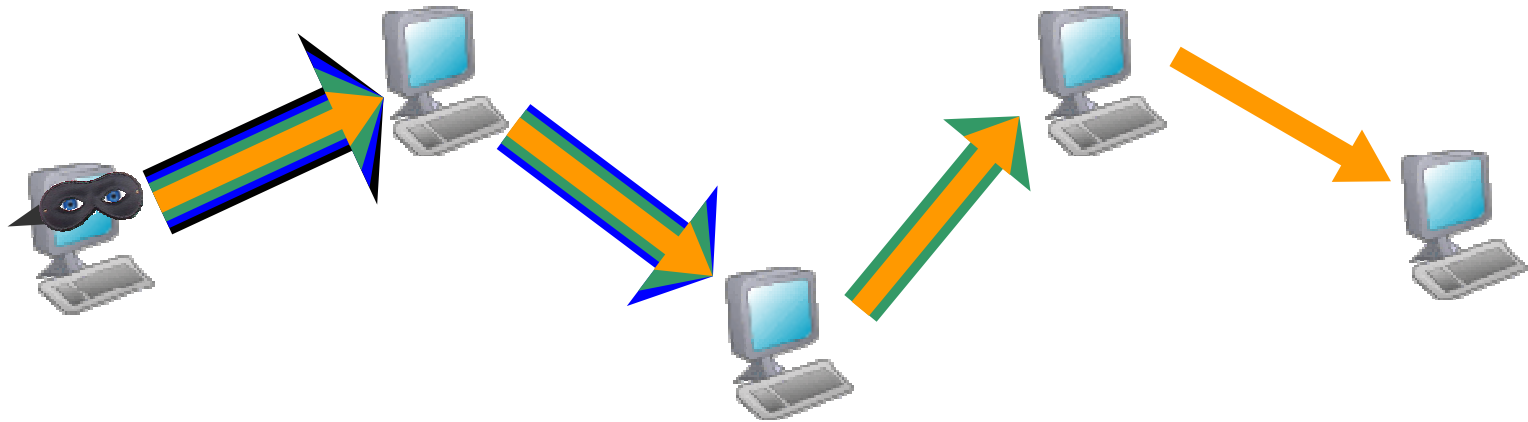
- ▶ By rendezvous point

- **All are session based**

- ▶ A channel / circuit is set up for each destination
- ▶ Allows pattern analysis

Classic Onion Routing

- For simplicity: no receiver anonymity



- All traffic travels along one path !!!
- Receiver anonymity
 - ▶ Rendezvous point

CORE: Connectionless Onion Router

- **Assumption**

- ▶ If each packet travels on a different way through the network this class of attacks are not possible
 - “Message Splitting Against the Partial Adversary” (A. Serjantov and S. Murdoch) -> black box

- **Requirements**

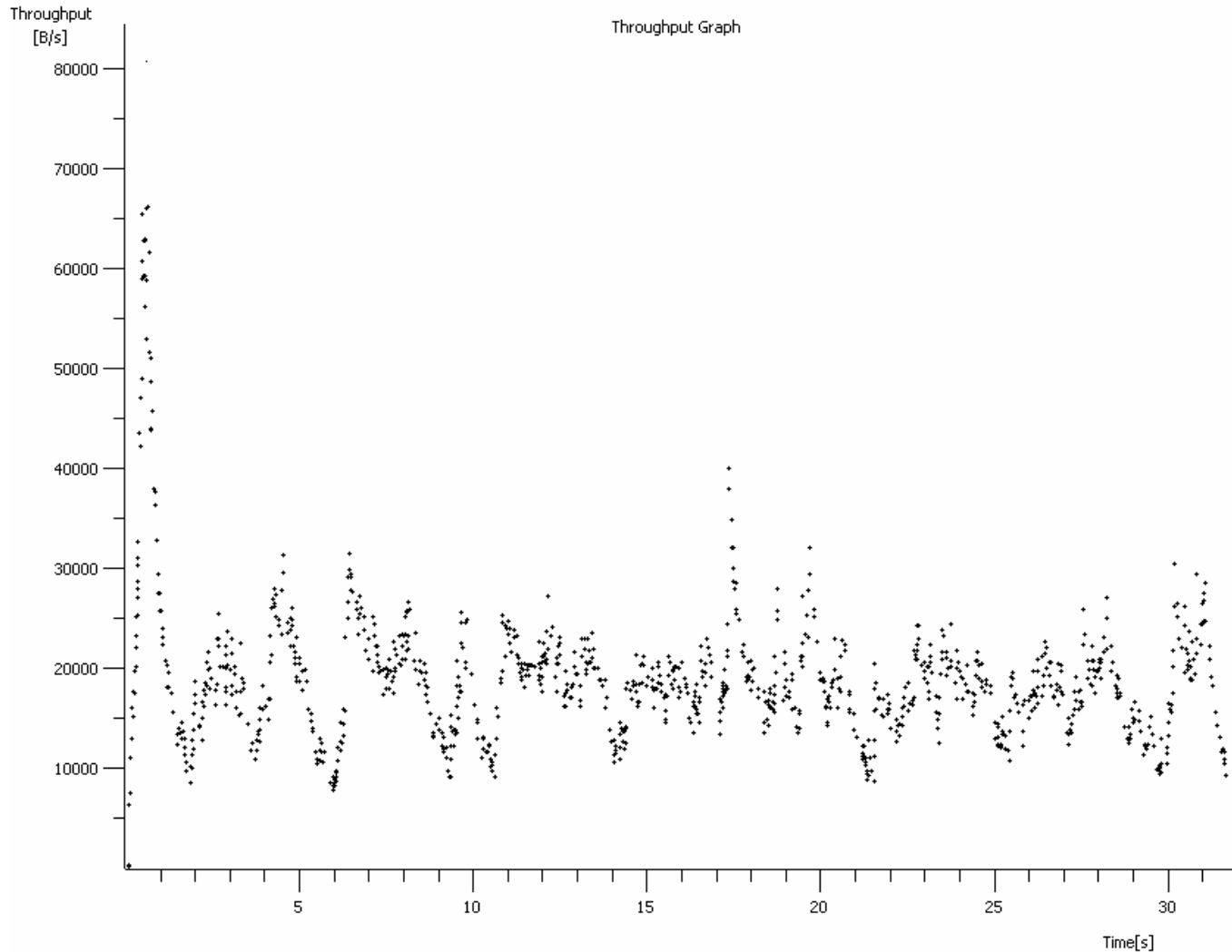
- **Consequence**

- ▶ Pic: UDP for rely, End-to-end TCP
- ▶ Each packet needs to contain a route description
 - Similar to ad-hoc network routing
- ▶ Needs asymmetric encryption
 - Throughput?
- ▶ Multi-path delay?
 - What will TCP do?

- **Connectionless onion routing will not be for free**

- ▶ Should be worth some overhead

Throughput



- Overlay throughput: $\sim 20\text{kbyte/s} \rightarrow 2.3\text{ Mbits/s}$

Implementation

- **Status**
- **ECC**
- **Header overhead**
- **Frame sizes**
 - ▶ Fully implemented on Linux
 - ▶ Cryptography
 - Elliptic-curve based DH
 - Short key length (192 bit <-> 2048 bit) RSA
 - Computing: EC is faster than RSA
 - ▶ 32 byte per relay host
 - ▶ Constant Frame/cell size 1000 bytes