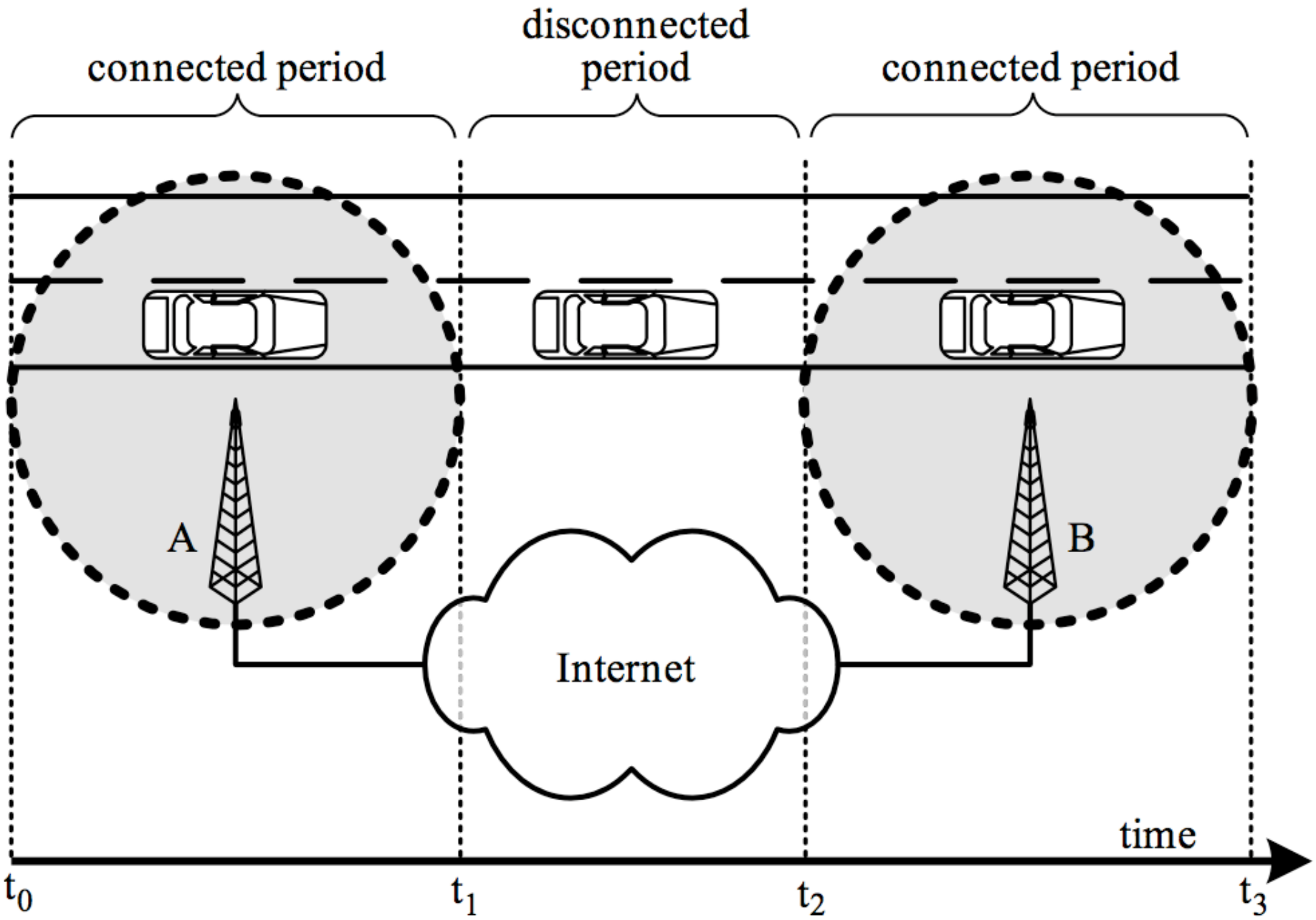
TCP Extensions for Intermittently Connected Hosts

Simon Schütz, Lars Eggert, Stefan Schmid and Marcus Brunner

NEC Network Laboratories
Heidelberg, Germany

Scenario

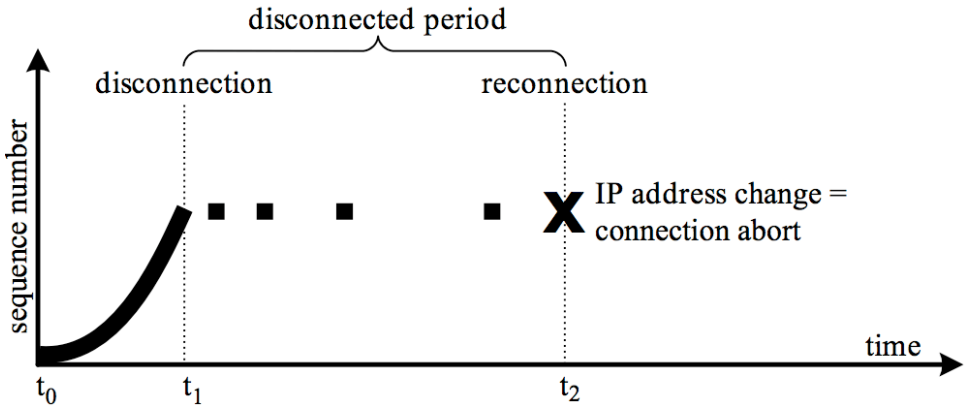

connected period
disconnected period
connected period
A
B
Internet
time
t_0
t_1
t_2
t_3

TCP Problems with Intermittent Connectivity

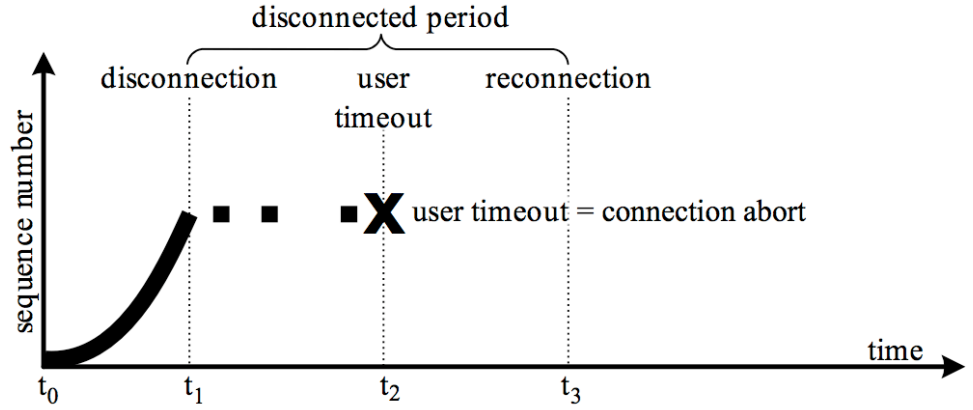
- **connections abort**
  - due to IP address changes
  - during long periods of disconnection
- **poor performance**
  - due to “exponential backoff” retransmission behavior
  - across wireless links: not looked at here

Problem Illustration

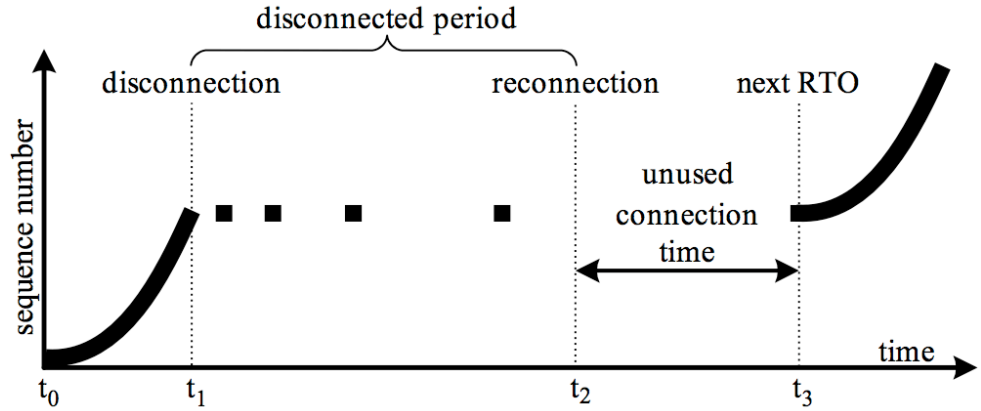
abort due to
IP address change



abort due to
user timeout



poor performance



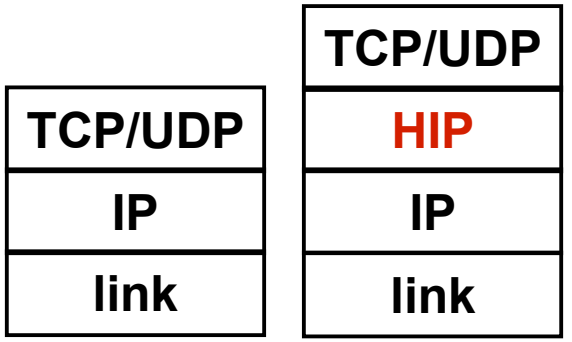
Solution

- three complementary components:
  1. **Host Identity Protocol (HIP)**
     decouple transport layer from IP addresses
  2. **new TCP User Timeout Option**
     tolerate disconnected periods up to O(days)
  3. **new TCP Retransmission Trigger**
     improve retransmission behavior = performance

- usual benefits/drawbacks of TCP modifications
  - \+ end-to-end ➔ no changes to infrastructure
  - \+ existing apps benefit without modification
  - \- danger of optimizing TCP for specific use
    (but we'd argue that these extensions are generally useful)

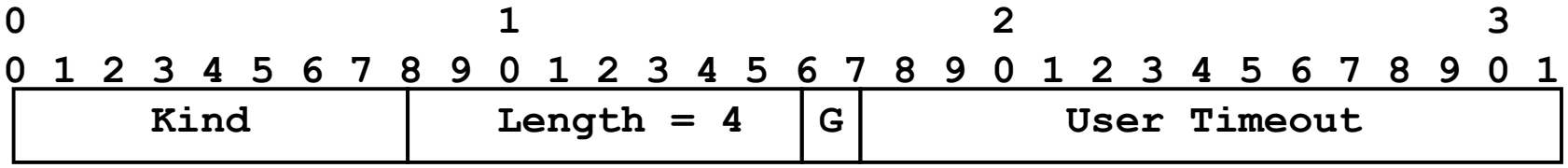
Component 1: Host Identity Protocol

- new layer between network and transport layers
  - connections bind to host identifiers instead of IP addresses
- HIP layer maps host identifiers into IP addresses
- mobility mechanism
  - dynamically update HIP→IP mapping
- intrinsic security
  - host identifiers are cryptographic keys
  - use for authentication
  - use with IPsec for encryption



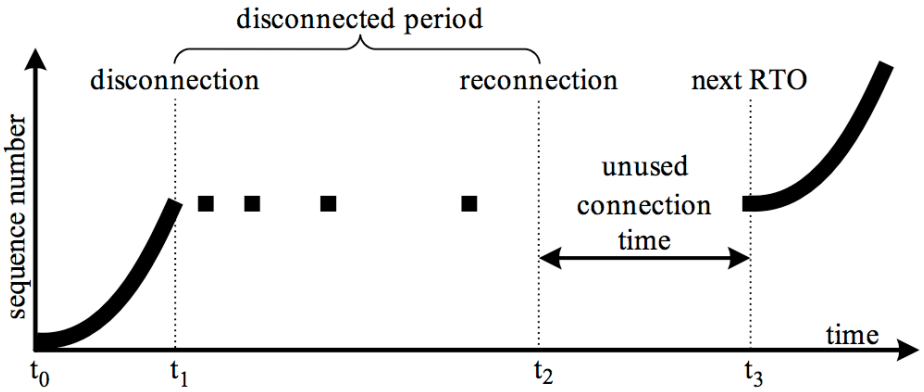
Component 2: TCP User Timeout Option

- enable **per-connection user timeouts**
  - tolerate longer disconnected periods
  - instead of system-wide default of O(minutes)
- exchange user timeout values between peers
  - don't need to establish full agreement
- shorter- and longer-than-default timeouts
  - maximum is 2^{15} minutes > 22 days
  - minimum can be O(seconds)

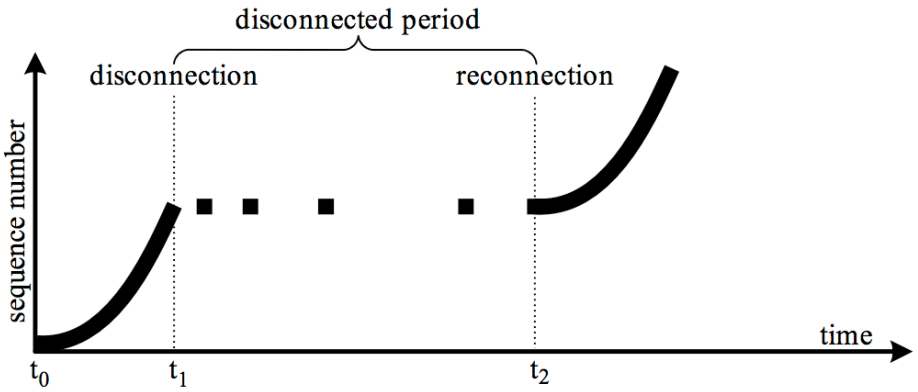


Component 3: TCP Retransmission Trigger

- **reduce unused connection time after reconnect**

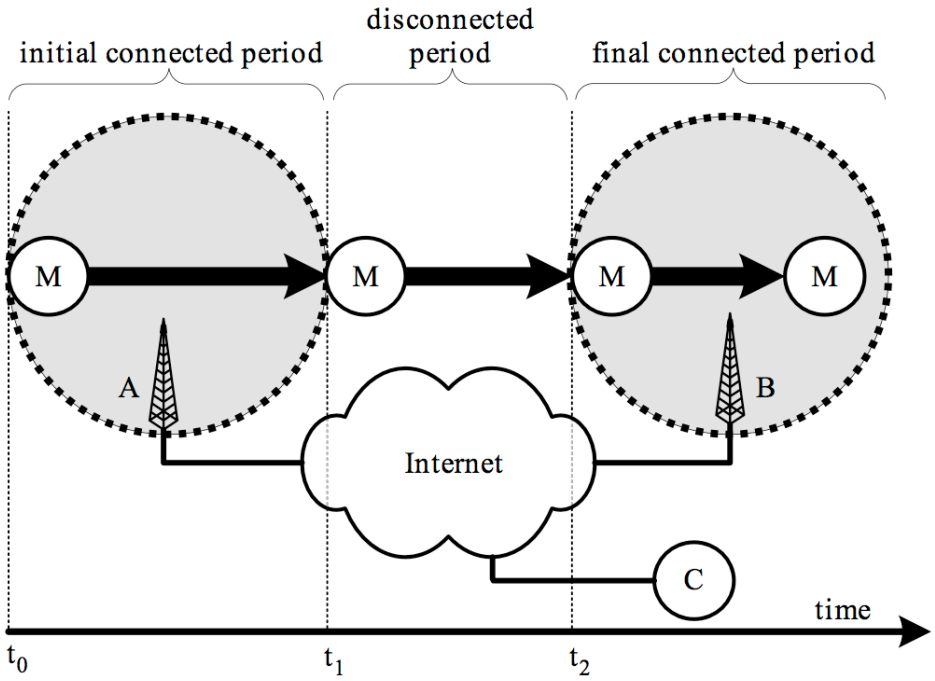


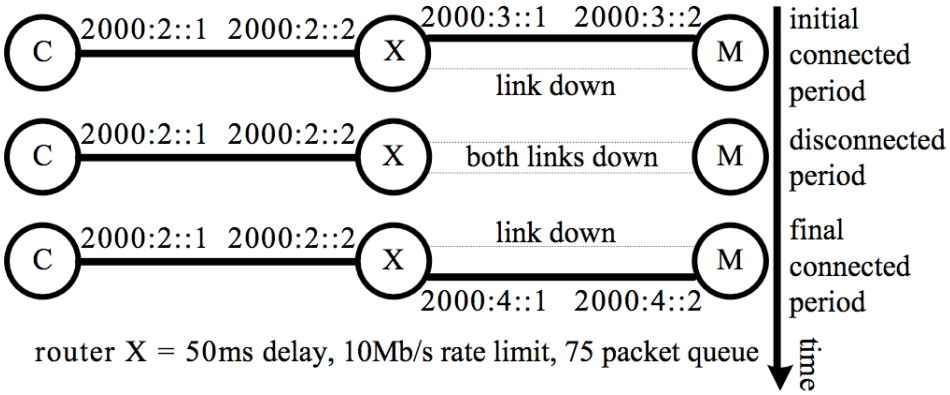
- trigger additional, speculative TCP retransmission upon “connectivity indicator”
  - purposefully vague
  - here: based on HIP mobility mechanism
  - other possibilities



Experimental Evaluation

- single bulk data transfer between M and C
  - 25 MB in ~22 sec
- **M "moves" from access point A to B, then stops**
  - intermediate disconnection
- emulate mobility
  - through dynamic reconfiguration of Ethernet interfaces
  - not interested in the wireless effects here

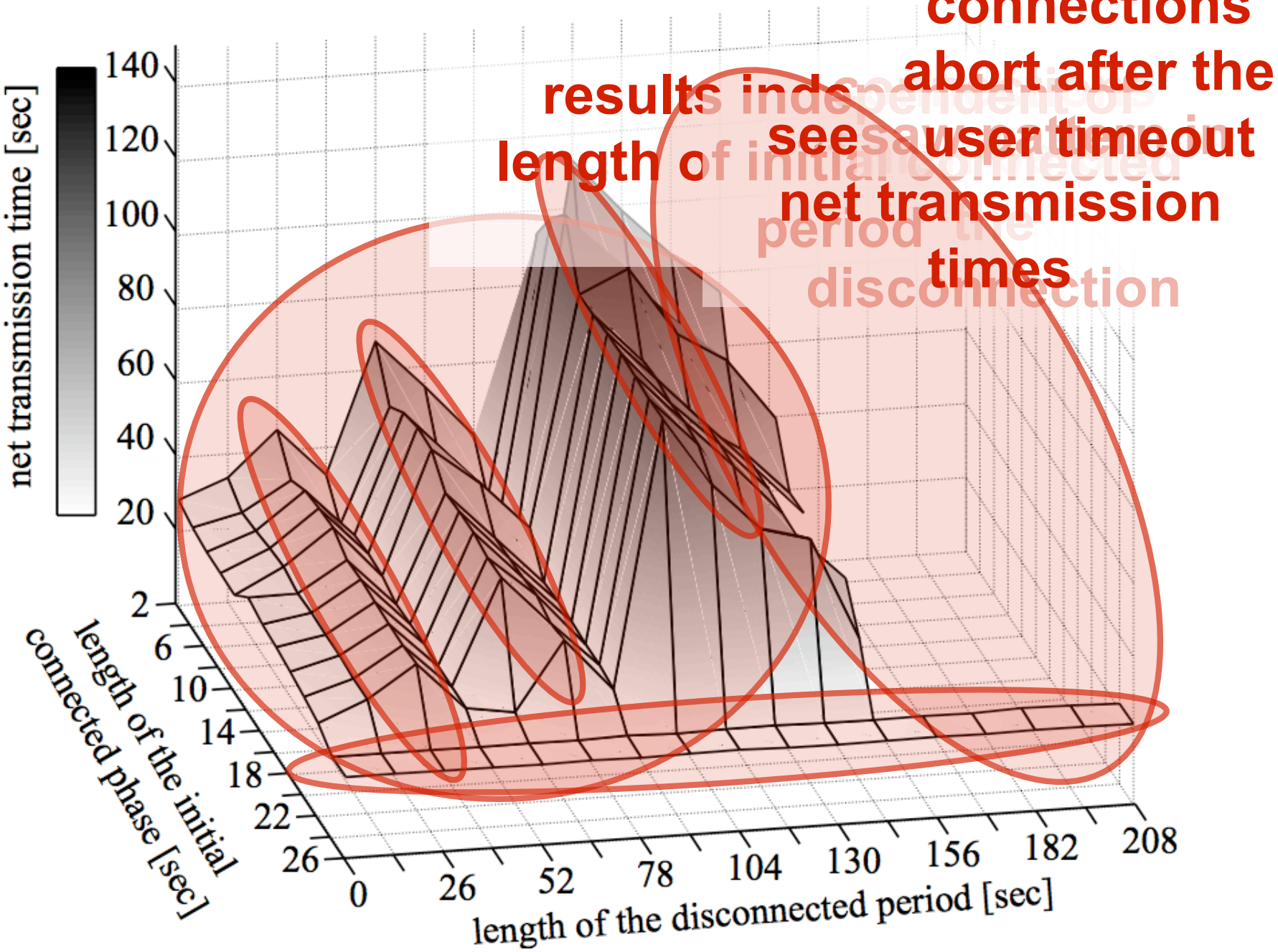




Parameters and Metric

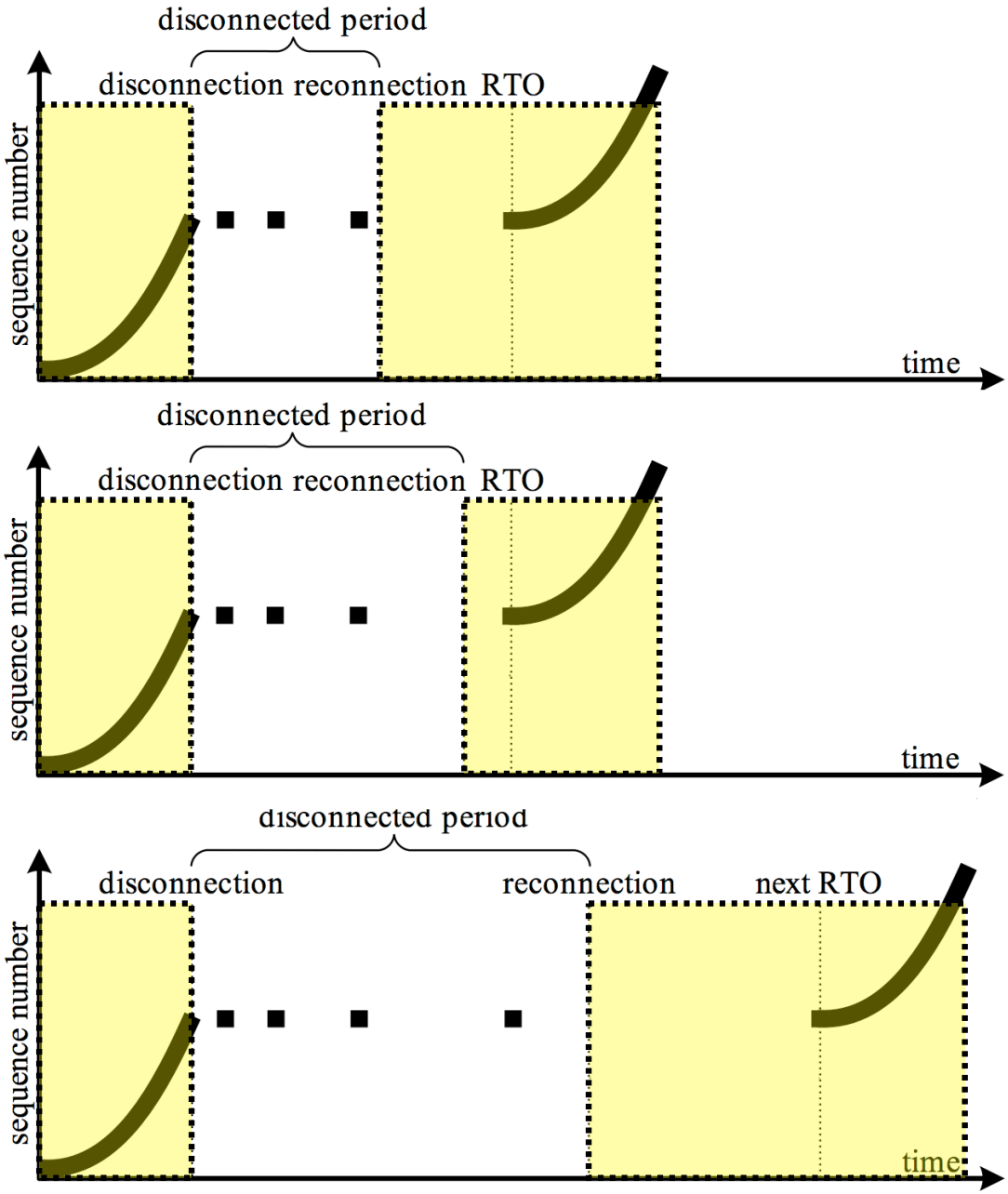
- **parameters**
  - length of initial connected period: 2-26 sec
  - length of disconnected period: 0-208 sec
- **performance metric**
  - **net transmission time = total connection time – length of disconnected period**
  - factor out disconnected periods
  - compare efficiency during connected periods
- **experimental details**
  - median and inter-quartile ranges
  - 10 runs per data point
  - results shown as 3D surface plots

Baseline Case: Only HIP

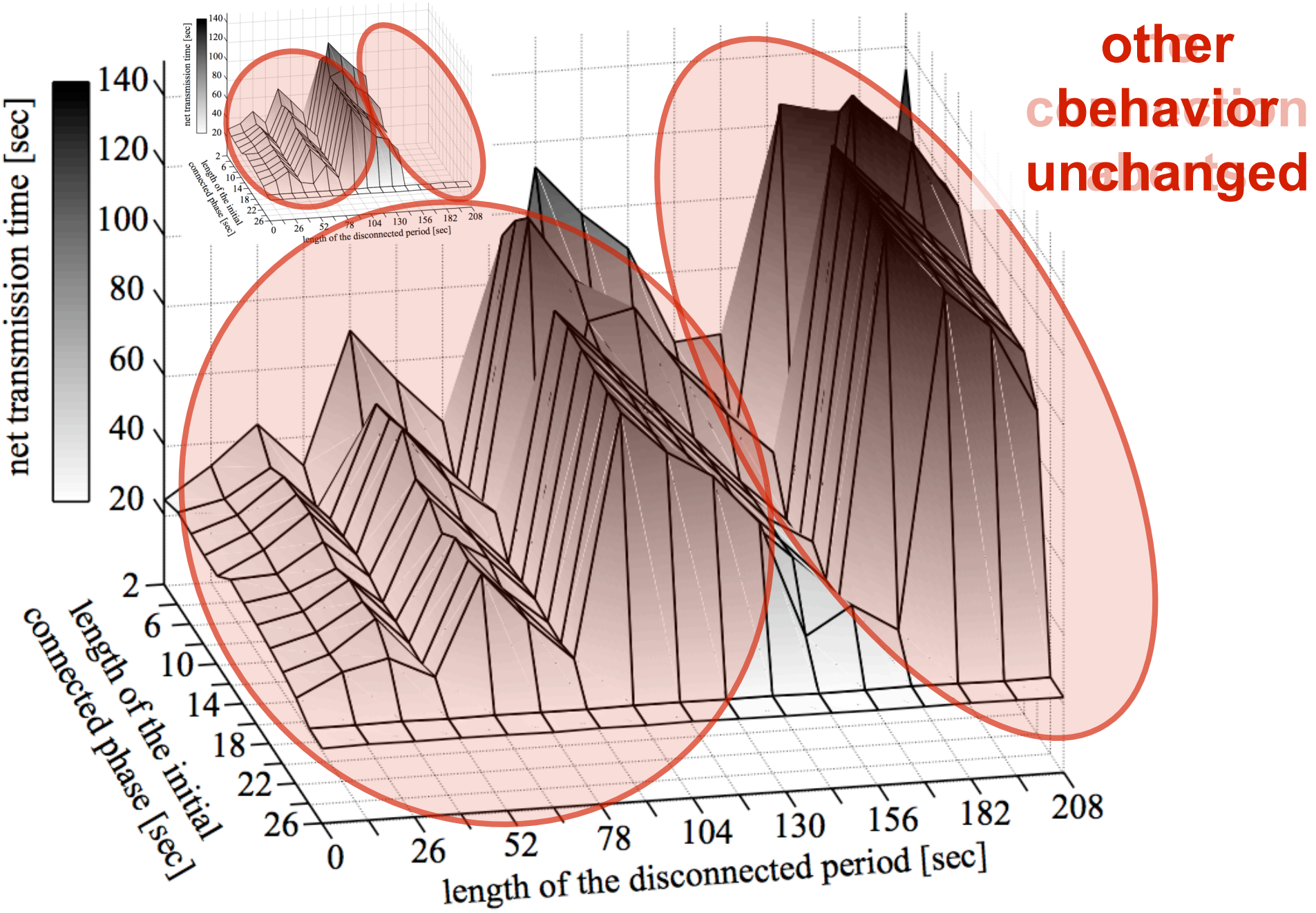


Seesaw Effect

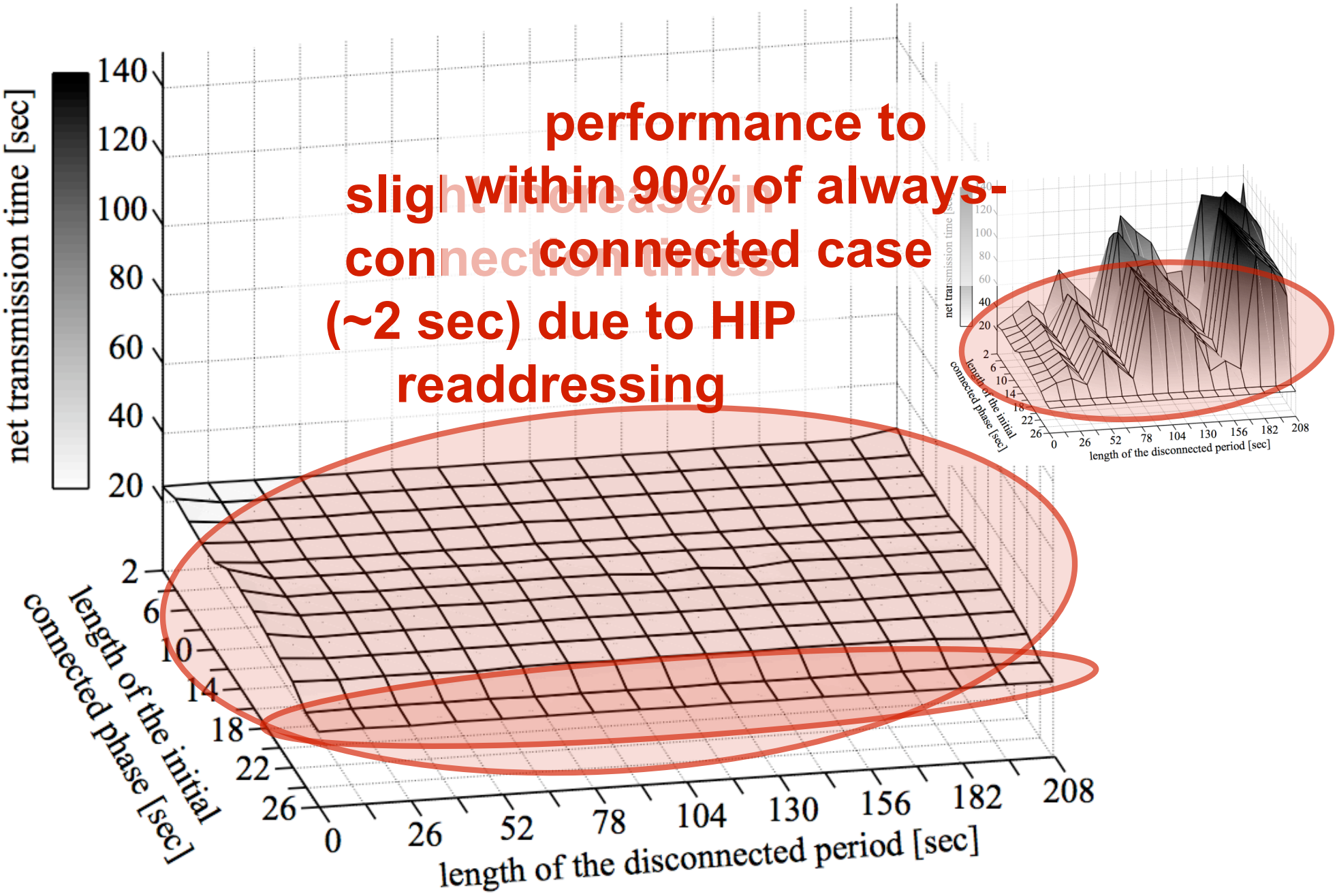
- net transmission times depend on the timing of reconnections and retransmission attempts
- longer disconnections can shorten net transmission



HIP + TCP User Timeout Option



HIP + TCP ATO + TCP Retransmission Trigger



performance to within 90% of always-connected case

slight increase in connection times (~2 sec) due to HIP readdressing

Related Work

- **delay-tolerant networking**
  - more complex: no end-to-end path even when connected
- **other mobility solutions**
  - Mobile IP, Virtual IP
- **disconnection tolerance**
  - Mobile TCP Socket
  - ROCKS/RACKS
  - Migrate
  - Drive-Thru Internet (proxy solution)
- **performance enhancements**
  - Implicit Link-Up Notification
  - Smart Link Layer
  - TCP-F, ELFN

Future Work

- improve efficiency during connected periods
  - improve slow-start after reconnect, *e.g.*, pacing
  - alternative transports, *e.g.*, XCP
  - combine with proxy approaches
- finish prototype implementation
- more experiments
  - different mobility scenarios, apps, traffic mixes
  - evaluate TCP enhancements with other mobility solutions, *e.g.*, MobileIP

Summary

- **analysis of TCP under intermittent connectivity**
  A. connections break due to IP address changes
  B. connections break due to connectivity disruptions
  C. poor performance

- **evaluated solution**
  A. solved by HIP
  B. solved by TCP User Timeout Option
  C. improved by TCP Retransmission Trigger
     to within 90% of constant connectivity

Thank You!