Analysis of TCP Performance over Mobile Ad Hoc Networks*

Part II: Simulation Details and Results

Gavin Holland and Nitin Vaidya
{gholland,vaidya}@cs.tamu.edu

Dept. of Computer Science
Texas A&M University, College Station, TX 77843
Phone: (409)845-5534 Fax: (409)847-8758

February 15, 1999

Technical Report 99-005

Abstract

Mobile ad hoc networks have gained a lot of attention lately as a means of providing continuous network connectivity to mobile computing devices regardless of physical location. Recently, a large amount of research has focused on the routing protocols needed in such an environment. In this two-part report, we investigate the effects that link breakage due to mobility has on TCP performance. Through simulation, we show that TCP throughput drops significantly when nodes move because of TCP's inability to recognize the difference between link failure and congestion. We also analyze specific examples, such as a situation where throughput is zero for a particular connection. We introduce a new metric, *expected throughput*, for the comparison of throughput in multi-hop networks, and then use this metric to show how the use of explicit link failure notification (ELFN) techniques can significantly improve TCP performance. In Part I of this report, we presented the problem and an analysis of the simulation results. In this paper (Part II of the report), we present the simulation in more detail and provide additional results.

Keywords: Mobile Ad Hoc Networks, TCP/IP, Performance Analysis, Explicit Link Failure Notification (ELFN), Dynamic Source Routing (DSR)

*This work was supported in part by the Department of Education under Award No. P200A80305, the National Science Foundation under Grant No. 32525-46600, and the Texas Advanced Technology Program under Grant No. 010115-248.

1 Introduction

With the proliferation of mobile computing devices, the demand for continuous network connectivity regardless of physical location has spawned an interest in the use of mobile ad hoc networks. A mobile ad hoc network is a network in which a group of mobile computing devices communicate among themselves using wireless radios, without the aid of a fixed networking infrastructure. Their use is being proposed as an extension to the Internet, but their use can be extended to anywhere that a fixed infrastructure does not exist, or is not desirable. A lot of research of mobile ad hoc networks has focused on the development of routing protocols(e.g. [6, 7, 11, 13, 14, 15, 16, 18]. Our research is focused on the performance of TCP over mobile ad hoc networks.

Since TCP/IP is the standard network protocol stack for communication on the Internet, its use over mobile ad hoc networks is a certainty because of the number of applications that it leverages, and because it allows seamless integration with the fixed infrastructure, where available.

However, earlier research on TCP over cellular wireless systems has shown that TCP suffers poor performance because of packet losses and corruption caused by wireless induced errors. Thus, a lot of research has focused on mechanisms to improve TCP performance in cellular wireless systems (e.g. [2, 3]). Other studies have looked at the problem of bandwidth asymmetry and large round-trip times, prevalent in satellite networks(e.g. [8, 4]).

In this report we address another characteristic of mobile ad hoc networks that impacts TCP performance: link failures due to mobility. In Part I of the report [12], we presented a performance analysis of standard TCP over mobile ad hoc networks and an analysis of the use of explicit notification techniques to counter the affects of link failures. In this paper, Part II of the report, we present more details of the simulation environment and the scenarios that were used, as well as the complete TCP-Reno results for each scenario.

2 Paper Organization

In Section 3 we give an overview of the *ns* network simulator, followed in Section 4 by a description of the methodology that we used to generate the simulation results in the Appendix and in Part I [12] of this report. Section 5 concludes this report.

3 Overview of the *ns* Simulator

The *ns* simulator [10] is an object-oriented event-driven network simulator that is being developed by the VINT project (a DARPA funded collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC). It is a widely respected simulator that has been used extensively in the literature (e.g. [9], [3], [1]). It is designed for network protocol research, providing an architecture that allows the composition of network topologies and workloads at run-time. This is made possible through the integration of compiled C++ objects with interpreted OTcl [1] objects, forming a composite *mirrored object*. Each class that represents a composable resource, such as a transport protocol or data agent, in the C++ domain has a mirror class in the OTcl domain. This integration

[1] OTcl is an object-oriented version of TCL that was written by David Wetherall at MIT [19].

```
set tcp_(0) [$ns_ create-connection  TCP/Reno $node_(1) TCPSink $node_(2) 0]
$tcp_(0) set window_ 32
set ftp_(0) [$tcp_(0) attach-source FTP]
$ns_ at 0 "$ftp_(0) start"
```

Figure 1: An example connection pattern script that creates a single TCP-Reno connection driven by an FTP data source.

allows C++ objects to be combined at run-time into the desired network simulation configuration through the manipulation of their OTcl object counterparts. The OTcl class is used at run-time to instantiate and configure its C++ counterpart, and it is used as an interface for data gathering. Data is shared between the two domains via *bound* variables. A variable in the C++ domain is *bound* to a variable in the OTcl domain through the `bind()` function call, which commutes changes to a variable in one domain to its counterpart in the other domain. This allows the user to trace or modify variables in the C++ domain through the OTcl domain without requiring that the user recompile the simulator. The simulator is freely available, and can be obtained in binary and source form from `http://www-mash.cs.berkeley.edu/ns/`. Further details on *ns* can be found in [10].

The simulations in this paper were generated using an extended version of *ns* that was developed by the Monarch Project at Carnegie Mellon University (`http://www.monarch.cs.cmu.edu/`). The extensions added to *ns* by the Monarch group provide the mechanisms necessary for simulating a mobile ad hoc network, including implementations of the 802.11 wireless MAC protocol, the BSD ARP protocol, several mobile ad hoc routing protocols, and a radio propagation model. It also provides mechanisms to simulate node mobility using precomputed files that specify node motion and peer to peer reachability for the duration of the simulation. Following is a brief overview of the input parameters and files that are required by the CMU extensions. The reader is referred to [17] for a more detailed description.

The CMU extensions to the *ns* simulator include a group of OTcl scripts that are processed as inputs to the simulator as well as a collection of programs that generate new input scripts. The main input file is an OTcl script that processes command-line arguments and initializes and configures the simulation environment. This is the `cmu/scripts/run.tcl`[2] script. In the process of configuring the simulation, this script calls several other OTcl scripts, some of which are named in command-line arguments. One of these is the routing protocol configuration script, which configures the simulated mobile nodes as routers that implement the named protocol (e.g., the DSR configuration script is `cmu/dsr/dsr.tcl`). Two other OTcl scripts, the connection pattern script and the mobility pattern script, designate the workload and network topology and dynamics for a single run of the simulation.

The *connection pattern* script designates the configuration and behavior of data connections in the network scenario to be simulated. For instance, it specifies when a data connection should be created and destroyed, the endpoints of the data connection (e.g. Node 1 to Node 2), when data flow across the connection should start and stop, the type of application data source that will send the data (e.g. FTP), and the transport protocol that the data source will use (e.g. TCP Reno). It may also specify other characteristics of the connection, such as the amount of data to send, the maximum TCP window size, and whether tracing should be enabled on variables related to the connection. An example script is shown in Figure 1. This script sets up a single TCP-Reno connection from Node 1

[2]All paths in this report are relative to the `ns-src/` directory in the 2.1b3 distribution of the CMU extensions.

```
$node_(1) set X_ 93.909
$node_(1) set Y_ 57.308
$node_(1) set Z_ 0.000
$node_(2) set X_ 325.905
$node_(2) set Y_ 122.463
$node_(2) set Z_ 0.000
$ns_ at 0.000 "$node_(1) setdest 80.985 310.405 1.962"
$ns_ at 0.000 "$node_(2) setdest 174.895 367.713 2.101"
$god_ set-dist 1 2 1
$ns_ at 129.151 "$node_(1) setdest 297.359 448.699 1.853"
$ns_ at 137.049 "$node_(2) setdest 94.547 123.594 1.959"
$ns_ at 230.673 "$god_ set-dist 1 2 16777215"
$ns_ at 267.718 "$node_(1) setdest 217.640 420.953 2.000"
$ns_ at 268.237 "$node_(2) setdest 151.138 15.501 1.901"
```

Figure 2: An example mobility pattern script for two nodes moving in a 500m x 500m area at a mean speed of 2 m/s for 300 seconds.

to Node 2 and then sets the maximum TCP window for that connection to 32 packets. (Note that `tcp_(0)` is an example of a *mirrored object* and `window_` is an example of a *bound* variable, both of which were described earlier.) The script then attaches an FTP data source to Node 1 and schedules the data source to start sending at time 0. A program to generate connection pattern scripts is provided in the CMU distribution (`cmu/scripts/cbrgen.tcl`).

The *mobility pattern* script (also called a *scenario file* in [17]) designates the motion of the nodes in the network and the changes in paths between the nodes over time. Associated with each node x are a set of coordinates (x_l, y_l, z_l) that designate x's location in the area of motion, a set of coordinates (x_d, y_d, z_d) that designate x's next destination, and x's current speed. The initial coordinates of the nodes and their patterns of destinations and speeds for the duration of the simulation are all designated in the mobility pattern script. An example script is shown in Figure 2, which contains commands designating the motion of two nodes moving at a mean speed of 2 m/s for 300 seconds in a square 500m x 500m area. The first sequence of commands (i.e. `$node_(1) set ...`) designate the initial placement of the two nodes at the start of the simulation. The next two commands (i.e. `$ns_ at 0.000 "$node_(1) setdest ...`) designate the (x_d, y_d) coordinates ($z_d = 0$ in this version) for each node's next destination, as well as the speed at which they will travel. (The "`at 0.000`" portion of the entry designates when (in simulated time) the command will be executed.) The `setdest` commands that appear later in the script designate new destinations and speeds as dictated by the mobility model that was used to create the script. In this instance, the nodes are moving in a random-walk. Also included in the script are commands that indicate changes in the optimal path lengths (in hops) between nodes in the network (i.e. `...$god_ set-dist...`). These commands are used to track the efficiency of the routing protocol. Each "`set-dist <i> <j> <d>`" command designates the time at which the optimal (shortest) path length between Node i and Node j becomes d hops. A value of $d = 16777215$ indicates an infinite path length, meaning no path exists between the nodes (the network is *partitioned*). A program to generate mobility pattern scripts is also provided in the CMU distribution (`cmu/setdest/setdest`). By default, this program assumes a fixed maximum transmission of 250m (the length of one hop), which is consistent with the radio that was simulated (914Mhz WaveLan).

4 Simulation Methodology

In this section we present the simulation parameters and procedures that were used for this report. Those parameters that are not mentioned here were chosen to be identical to those used in [5]. Unless otherwise stated, all procedures presented in this section apply to all of the simulations in this report.

4.1 Network and Node Models

Number of nodes	30
Mobility model	random-walk
Area of mobility	1500m x 300m
Topography	flat
Mean node speeds (μ)	2,10,20,30 m/s
Variation in node speeds	$0.9\mu - 1.1\mu$

Table 1: The network model.

The characteristics of our network model are shown in Table 1. The network consists of 30 nodes in a 1500m x 300m flat rectangular area. At the start of each simulation, the nodes were placed in random starting positions within the rectangular area. From there, they moved around in a continuous random-walk pattern for the duration of the simulation. This is equivalent to the *random waypoint* mobility model used in [5] with a pause time of zero. In the random waypoint model, each node x picks a random set of coordinates as its next destination and then travels in a straight line to those coordinates at some randomly chosen speed. Once x arrives at this destination, it pauses before picking another set of coordinates and continuing onward. Obviously, a pause time of zero results in constant motion. The randomly chosen speeds were uniformly distributed in an interval of $0.9\mu - 1.1\mu$ for some mean speed μ. We generated results for four different mean speeds: 2, 10, 20, and 30 m/s.

Application protocol	FTP (w/ unlimited data)
Transport protocol	TCP-Reno (w/o delayed ACKs)
Data packet size	1460 bytes
Maximum TCP window	32 packets
Routing protocol	Dynamic Source Routing (DSR)
Link-layer protocols	FIFO queues, BSD ARP, 802.11 MAC
Wireless interface	914MHz WaveLan, 2Mbps, 250m
Antenna	Omnidirectional, 1.5m above ground
Propagation model	Friis free-space, Two Ray Ground

Table 2: The node model.

Our node model is shown in Table 2, which is similar to that used in [5]. Each node in the system was identically configured as a mobile wireless terminal with a single wireless interface and omnidirectional antenna, communicating on a single frequency. All nodes acted as IP routers, using the Dynamic Source Routing (DSR) routing protocol. The choice of DSR as the routing protocol was primarily based on the availability of the *ns* extensions at the time when we started this study. Our goal was only to observe TCP's performance in the presence of mobility induced failures in a plausible network environment for which any of the proposed mobile wireless ad-hoc routing protocols would have sufficed. We used the protocol as implemented in the CMU extensions for the base TCP measurements presented in this report. The underlying link-layer used a typical FIFO queuing discipline and the BSD Address Resolution Protocol (ARP) was used for resolving IP addresses to MAC addresses. Each node's interface was configured to emulate an 802.11-based 914MHz WaveLan half-duplex wireless radio which has a bandwidth of 2Mbps and a transmission range of 250m. Medium access was controlled using both virtual and real carrier sensing, as specified by the 802.11 standard. Virtual carrier sensing was executed according to the Distributed Coordination Function (DCF) of the 802.11 standard, which uses a sequence of messages (RTS,CTS,ACK) to reserve the wireless medium for contention free transmission and for error detection. Signal attenuation was modeled using a combination of the Friis free-space model at close range and the Two-Ray Ground model at long range. The models assumed that the antenna of the mobile terminal traveled at a fixed distance of 1.5m above the ground.

The two nodes that were the endpoints of the TCP connection was simulated were additionally configured as a TCP source and sink. The TCP protocol that we simulated was TCP Reno without delayed acknowledgments. The data packet size (TCP payload) was a fixed 1460 bytes and the receiver-advertised window for the connection was set to 32 packets. The application that provided the data stream for the connection was an FTP agent with an unlimited amount of data to send.

4.2 Mobility Patterns

Our simulation results are based on 50 randomly generated mobility patterns. Each mobility pattern designates an initial placement and sequence of moves for the nodes in the simulated network. The initial placement was random, and the move sequences were generated according to the random-walk mobility model described in the previous section. The base patterns were generated for a network of 30 nodes moving at a mean speed of 2 m/s for 1800 seconds. Individual node speeds varied uniformly in an interval of $0.9\mu - 1.1\mu$ around the mean speed $\mu = 2m/s$. The node placement and move sequences of these base patterns were then used to generate the mobility pattern scripts, described in Section 3, for different values of μ. Thus, for two pattern scripts I and J, where nodes in script I have a mean speed of $\mu = v$ and nodes in script J have a mean speed of $\mu = 2v$, a node x will execute the exact same sequence of moves in J as it does in I, just at twice the rate. Thus, if node x takes time t to move from point A to point B in I, then it will take $t/2$ to travel from $A - B$ in J.

The network conditions experienced by the simulated TCP connection in each of the 50 mobility patterns is presented in Table 3. Shown in Table 3 are the expected throughput (*Exp Tput*), the number of path changes (*Path Chgs*) and hop changes (*Hop Chgs*), the number of partitions (*Path N/A*), and the percentage of time the endpoints spent at various distances (in hops).

The expected throughput (*Exp Tput*) is an estimate of the upper bound of the throughput achievable across the TCP connection within the mobility pattern, and is calculated as

$$\text{expected throughput} = \frac{\sum_{i=1}^{\infty} t_i * T_i}{\sum_{i=1}^{\infty} t_i} \tag{1}$$

where t_i is the time that the minimum distance between the endpoints of the TCP connection was i hops ($1 \leq i \leq \infty$), and T_i is the TCP throughput obtained over a fixed linear i hop network (note that $T_i = 0$ for $i = \infty$). See Part I of this report for a more complete description of the expected throughput [12].

No.	Exp Tput (Kbps)	Path Chgs	Hop Chgs	Path N/A	Pctg of Time TCP Endpoints Were x Hops Apart							
					1	2	3	4	5	6	> 6	N/A
1	712.4	45	37	1	27.4	32.0	18.5	7.9	11.7	1.0	0.0	1.5
2	853.9	23	14	1	36.8	45.5	8.8	4.4	0.0	0.0	0.0	4.6
3	823.8	52	40	2	49.8	8.7	6.2	11.3	10.2	4.3	5.9	3.7
4	632.7	34	30	2	22.1	20.6	34.5	9.4	8.1	0.6	0.0	4.7
5	494.1	85	65	1	11.7	11.5	34.8	32.0	6.8	1.9	0.9	0.3
6	972.2	39	35	0	52.2	23.3	24.5	0.0	0.0	0.0	0.0	0.0
7	676.6	54	41	2	19.5	33.2	40.9	2.9	0.0	0.0	0.0	3.5
8	851.9	33	26	0	37.3	37.0	16.7	9.0	0.0	0.0	0.0	0.0
9	600.7	51	43	0	21.7	21.1	20.3	12.7	12.5	7.1	4.7	0.0
10	602.6	45	33	2	18.4	30.6	14.8	27.1	4.8	0.4	0.0	3.8
11	933.8	19	9	0	42.5	45.6	11.9	0.0	0.0	0.0	0.0	0.0
12	611.5	54	48	0	21.1	20.2	25.8	18.2	8.0	6.6	0.0	0.0
13	835.6	34	29	0	40.8	32.6	5.3	4.5	10.7	6.2	0.0	0.0
14	721.5	51	42	2	36.7	15.0	13.2	14.4	5.3	5.5	5.5	4.5
15	634.6	47	38	1	16.4	33.9	30.8	16.9	1.5	0.0	0.0	0.5
16	550.6	64	55	2	20.9	12.6	16.9	29.8	12.2	3.0	0.0	4.5
17	921.8	19	18	2	48.4	29.3	8.8	10.0	1.9	0.0	0.0	1.5
18	686.8	65	47	2	26.1	30.8	18.8	9.5	9.2	0.4	0.0	5.1
19	691.5	47	40	2	20.4	43.9	17.6	12.2	5.4	0.0	0.0	0.5
20	784.9	39	32	0	25.3	48.6	26.1	0.0	0.0	0.0	0.0	0.0
21	791.1	45	34	1	40.2	21.8	8.2	16.7	4.8	2.1	4.9	1.4
22	541.3	54	43	4	13.4	24.8	35.7	5.7	5.7	3.9	1.7	9.1
23	1029.1	15	11	0	53.7	42.6	3.7	0.0	0.0	0.0	0.0	0.0
24	552.0	56	44	1	15.0	28.6	15.8	15.9	18.3	6.2	0.0	0.2
25	565.3	74	61	0	24.4	11.8	12.2	13.1	22.2	15.3	1.1	0.0
26	488.0	51	36	1	12.6	20.7	15.5	24.4	20.1	4.5	0.0	2.1
27	729.6	48	37	2	34.1	18.5	21.2	10.2	5.5	4.2	3.1	3.1
28	970.9	42	36	1	60.5	6.8	11.5	19.9	1.0	0.0	0.0	0.3
29	1058.1	26	21	0	65.8	10.9	18.1	5.2	0.0	0.0	0.0	0.0
30	991.2	24	21	1	51.2	37.6	9.9	0.4	0.0	0.0	0.0	0.8
31	749.6	32	25	0	29.5	42.9	2.7	7.4	13.8	3.8	0.0	0.0
32	610.9	49	41	1	18.7	18.4	41.6	18.1	0.0	0.0	0.0	3.3
33	786.4	48	33	0	34.7	35.9	8.1	4.7	3.9	7.7	5.1	0.0
34	752.2	52	44	1	37.3	15.6	14.9	22.0	9.5	0.0	0.0	0.7
35	785.4	51	38	1	29.1	38.2	30.2	0.3	0.0	0.0	0.0	2.2
36	614.7	52	40	2	20.2	24.7	22.6	20.9	8.7	1.5	0.0	1.3
37	839.7	37	31	1	33.2	46.1	16.0	3.0	1.2	0.0	0.0	0.5
38	1022.0	22	20	0	62.3	10.4	20.1	7.2	0.0	0.0	0.0	0.0
39	831.1	45	37	0	46.5	10.5	12.7	22.6	7.6	0.1	0.0	0.0
40	616.9	42	36	0	15.2	32.9	32.6	11.8	7.4	0.1	0.0	0.0
41	608.6	57	44	1	13.3	44.0	14.7	20.5	5.1	2.4	0.0	0.1
42	944.3	7	5	0	41.3	54.9	3.8	0.0	0.0	0.0	0.0	0.0
43	863.2	24	20	0	39.2	33.9	18.5	8.4	0.0	0.0	0.0	0.0
44	585.8	55	44	3	20.9	23.7	16.2	14.9	10.9	5.2	2.0	6.1
45	659.4	50	37	0	20.6	40.9	9.0	14.4	12.4	2.6	0.0	0.0
46	918.1	26	22	0	47.6	28.0	13.7	6.0	4.7	0.0	0.0	0.0
47	681.1	63	52	0	27.6	18.1	25.7	23.1	4.5	1.1	0.0	0.0
48	821.4	52	45	0	44.2	9.7	23.4	15.0	6.3	1.4	0.0	0.0
49	888.1	26	20	0	44.8	21.1	27.3	6.8	0.0	0.0	0.0	0.0
50	869.2	29	24	0	40.1	35.9	14.6	4.0	5.4	0.0	0.0	0.0
Avg	755.2	43.1	34.5	0.9	32.7	27.7	18.3	11.5	5.7	2.0	0.7	1.4

Figure 3: Network conditions experienced by the measured TCP connection for the 50 mobility patterns.

The *Path Chgs* column in Table 3 shows the number of times the shortest path (in hops) between the TCP endpoints changed over the length of the pattern. This differs from changes in the number of hops between the endpoints (shown in the *Hop Chgs* column) in that a change in the shortest path may occur even though the number of hops stays the same. For example, consider a simple two-hop shortest path ($a \rightarrow b \rightarrow c$) where nodes a and c are the TCP endpoints. If node b is replaced by another node d then the path will change ($a \rightarrow d \rightarrow c$) but the number of hops will stay the same. This is an important distinction, since any change in the path will cause a change in route (assuming the routing protocol prefers routes along shortest paths) but not all path changes are caused by a change in the number of hops. Thus, path changes are a more accurate indicator of the variability of the network topology than are hop changes (which was used in an earlier study [5]). However, tracking path changes is more complicated than tracking hop changes because there may be a set of shortest paths between the two TCP endpoints at any point in time. Thus, we define the occurrence of a path change to mean the point in time at which a change in the set of shortest paths between two nodes occurs such that no path exists in both the old and the new sets. More precisely, let $S^i_{(a,b)}$ be the set of shortest paths at time t_i between nodes a and b. A *path change* is said to occur at time t_i if $S^{i-1}_{(a,b)} \cap S^i_{(a,b)} = \emptyset$ and $S^{i-1}_{(a,b)} \cup S^i_{(a,b)} \neq \emptyset$. The *Path N/A* column shows the number of times that this occurs for the two TCP endpoints.

The last set of columns in Table 3 gives the percentage of time that the endpoints spent separated by selected distances (in hops). These times correspond to the t_i values in Equation 1. The last column *N/A* gives the duration of time the endpoints were in separate partitions (which is equivalent to saying that the distance in hops is ∞ for the purposes of Equation 1).

4.3 Simulation Platform

The results presented in this report are based on the 2.1b3 release of the CMU extensions with some minor bug fixes that were necessary to compile and run the simulator on our computing platform. All of the results were generated on a PC running the Sun Solaris 2.6 operating system. The code was compiled using the Sun Sunpro C/C++ suite of compilers.

5 Concluding Remarks

This paper concludes our report on TCP performance in ad hoc networks. It is hoped that the simulation details provided herein provide a more complete picture of the mechanisms and methodology that was used for this study, and that the additional results will spur additional research into optimizations for TCP in ad hoc networks. To assist those that wish to pursue such research, the ELFN code is available from the primary author.

References

[1] B. Bakshi, P. Krishna, N. Vaidya, and D. Pradhan, "Improving performance of TCP over wireless networks," in *Proceedings of the International Conference on Distributed Computing Systems*, may 1997.

[2] H. Balakrishnan and R. Katz, "Explicit loss notification and wireless web performance," in *IEEE Globecom Internet Mini-Conference, Sydney*, Oct. 1998.

[3] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," in *ACM SIGCOMM, Stanford, CA*, Aug. 1996.

[4] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on TCP performance," in *Proceedings of the IEEE Mobicom'97*, (Budapest, Hungary), pp. 77–89, sep 1997.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pp. 85–97, Oct. 1998.

[6] M. S. Corson and A. Ephremides, "A distributed routing algorithm for mobile wireless networks," *ACM J. Wireless Networks*, vol. 1, no. 1, pp. 61–81, 1995.

[7] B. Das, E. Sivakumar, and V. Bhargavan, "Routing in ad-hoc networks using a virtual backbone." manuscript, 1997.

[8] R. C. Durst, G. J. Miller, and E. J. Travis, "TCP extensions for space communications," in *Proceedings of MOBICOM '96*, 1996.

[9] K. Fall and S. Floyd, "Simulation-based comparisons of tahoe, reno and tcp sack," pp. 5–21, July 1996.

[10] K. Fall and K. Varadhan, ns *Notes and Documentation.* LBNL, August 1998. http://www-mash.cs.berkeley.edu/ns/.

[11] Z. J. Haas and M. R. Pearlman, "The zone routing protocol (ZRP) for ad hoc networks (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Aug. 1998.

[12] G. Holland and N. Vaidya, "Analysis of tcp performance over mobile ad hoc networks – part i: Problem discussion and analysis of results," Tech. Rep. 99-004, Texas A&M University, Dept. of Computer Science, MS/3112, Texas A&M University, College Station, TX 77843, February 1999.

[13] D. Johnson, D. A. Maltz, and J. Broch, "The dynamic source routing protocol for mobile ad hoc networks (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Mar. 1998.

[14] C. E. Perkins and E. M. Royer, "Ad hoc on demand distance vector (AODV) routing (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Aug. 1998.

[15] S. Ramanathan and M. Steenstrup, "A survey of routing techniques for mobile communication networks," *Mobile Networks and Applications*, pp. 89–104, 1996.

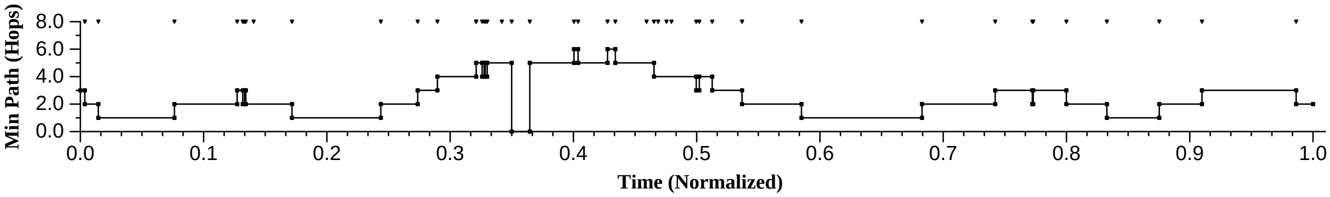
[16] R. Sivakumar, P. Sinha, and V. Bharghavan, "Core extraction distributed ad hoc routing (CEDAR) specification (Internet-Draft)," *Mobile Ad-hoc Network (MANET) Working Group, IETF*, Oct. 1998.

[17] T. C. M. Team, *The CMU Monarch Project's Wireless and Mobility Extensions to* ns. Computer Science Department, Carnegie Mellon University, August 1998. http://www.monarch.cs.cmu.edu/.

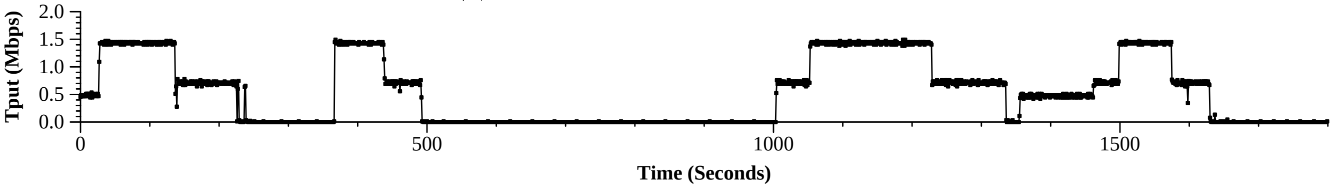
[18] C. Toh, "A novel distributed routing protocol to support ad-hoc mobile computing," *Wireless Personal Communication*, Jan. 1997.

[19] D. Wetherall, *OTcl – MIT Object Tcl: FAQ and Manual.* MIT Lab for Computer Science, September 1995. ftp://ftp.tns.lcs.mit.edu/pub/otcl/.

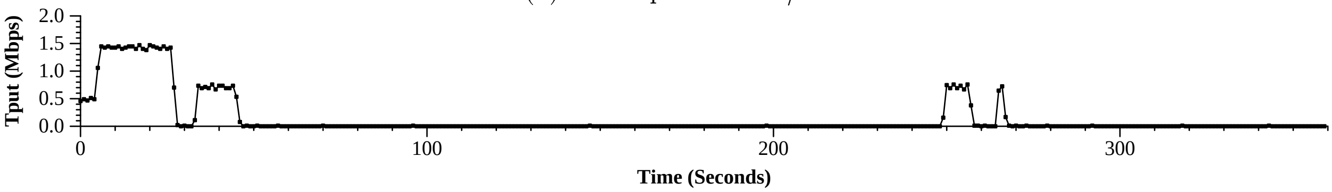
Appendix: TCP-Reno Results



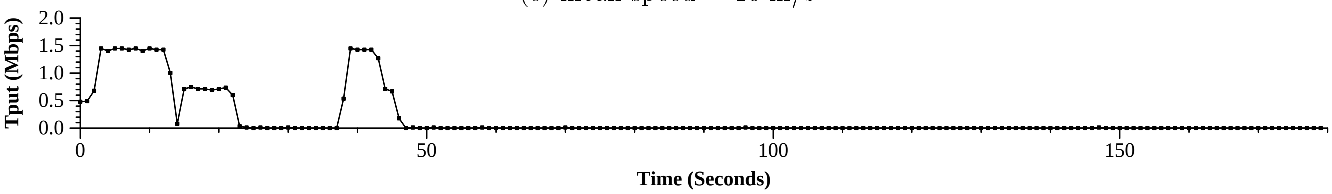
(a) profile of mobility pattern No. 1.



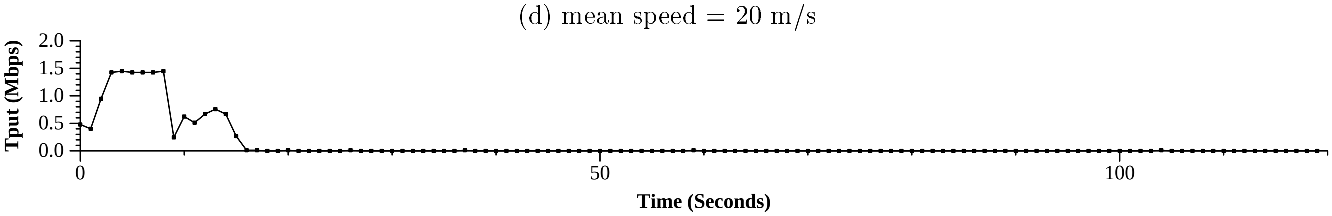
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

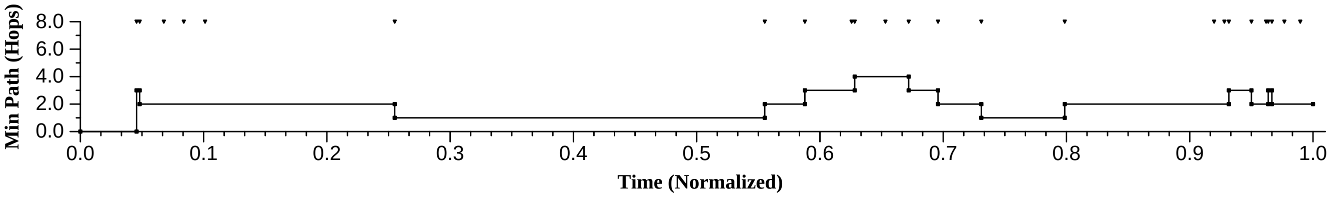


(d) mean speed = 20 m/s

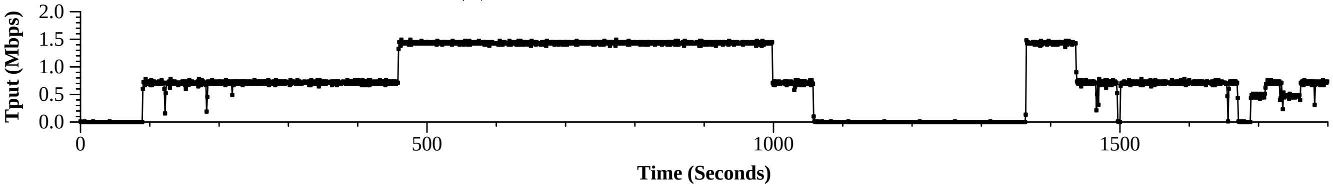


(e) mean speed = 30 m/s

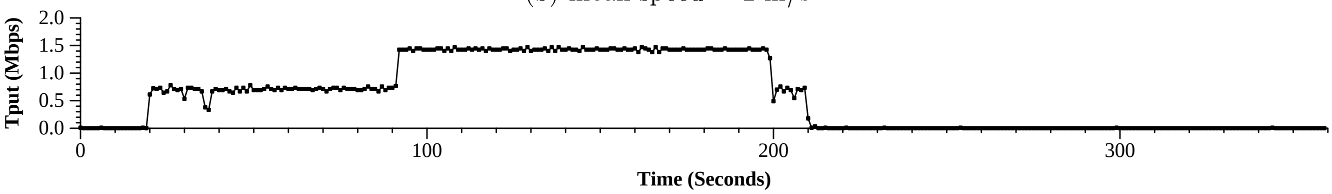
Figure 4: Profile for mobility pattern No. 1, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



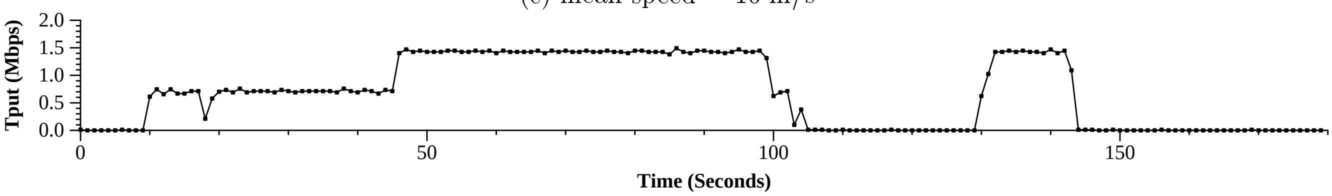
(a) profile of mobility pattern No. 2.



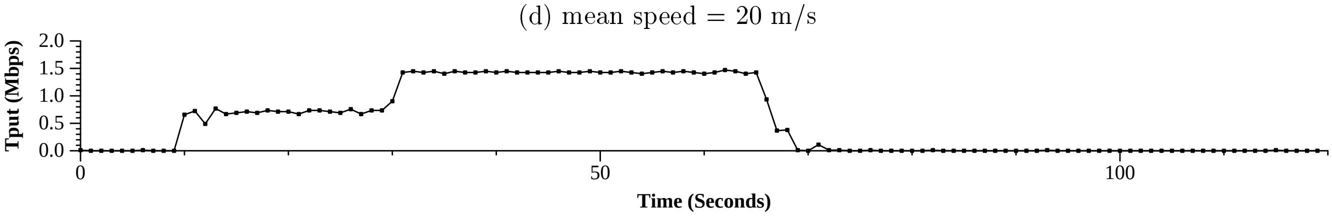
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

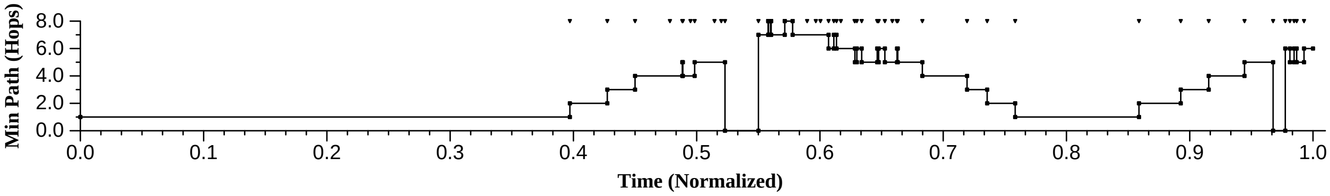


(d) mean speed = 20 m/s

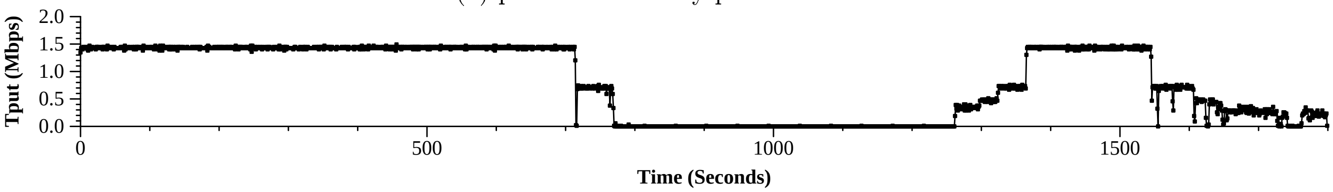


(e) mean speed = 30 m/s

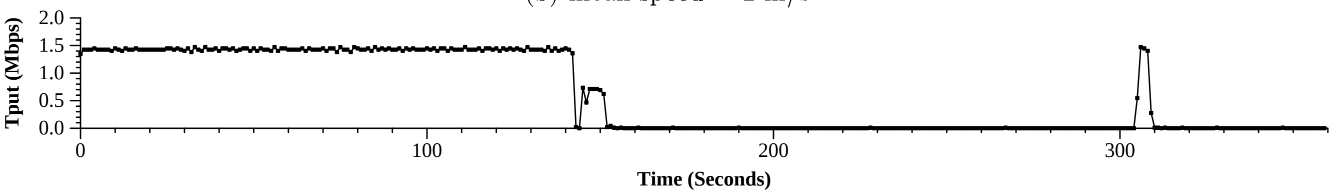
Figure 5: Profile for mobility pattern No. 2, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



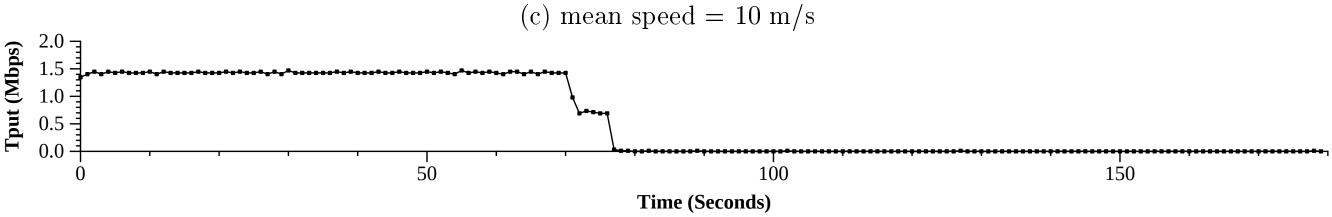
(a) profile of mobility pattern No. 3.



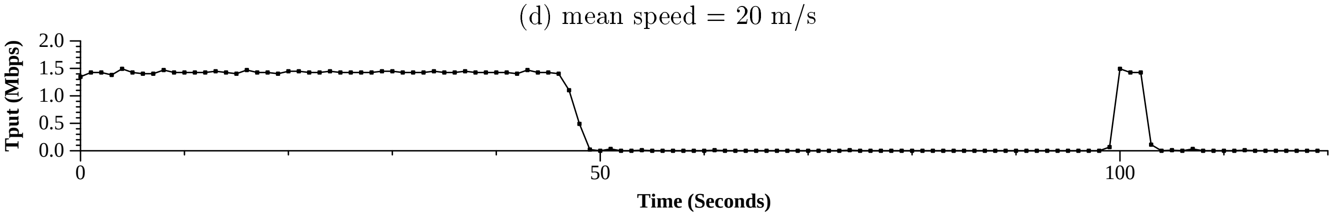
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

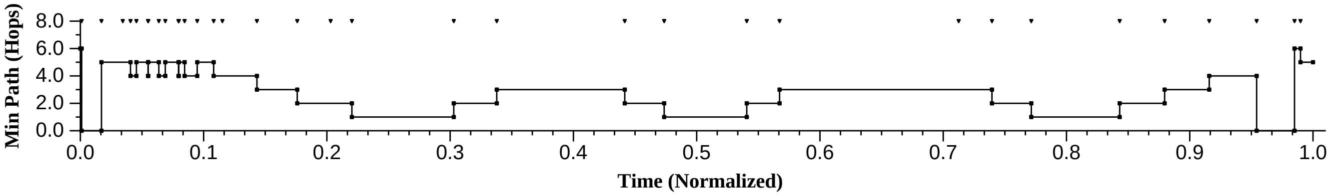


(d) mean speed = 20 m/s

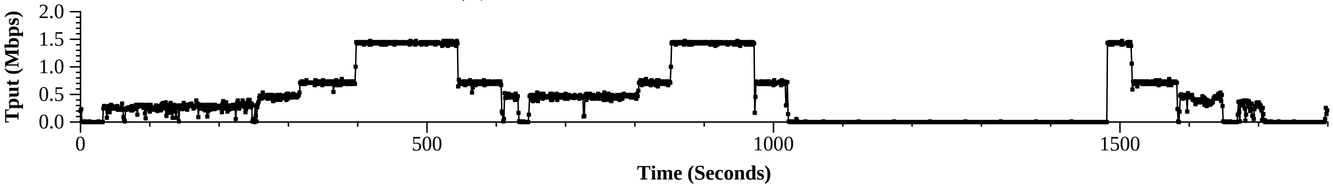


(e) mean speed = 30 m/s

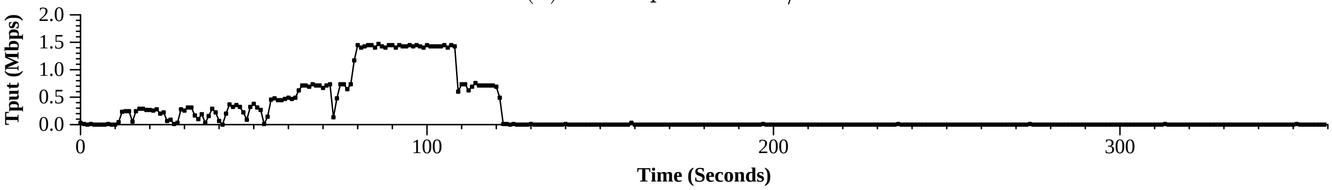
Figure 6: Profile for mobility pattern No. 3, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



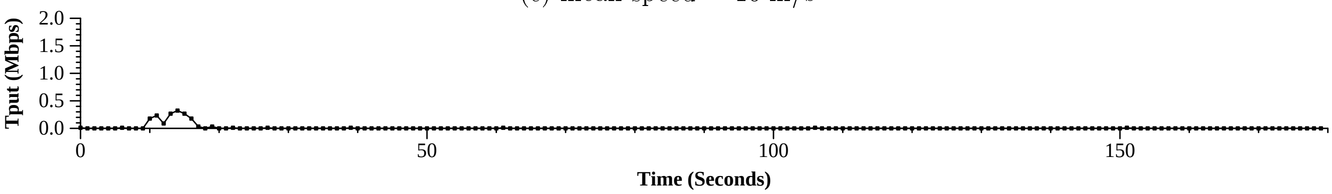
(a) profile of mobility pattern No. 4.



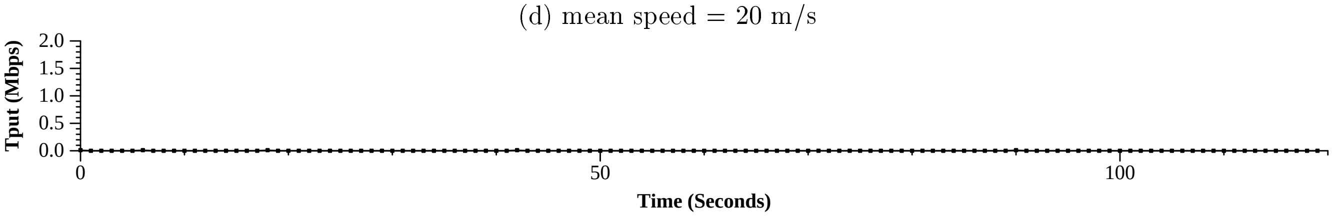
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

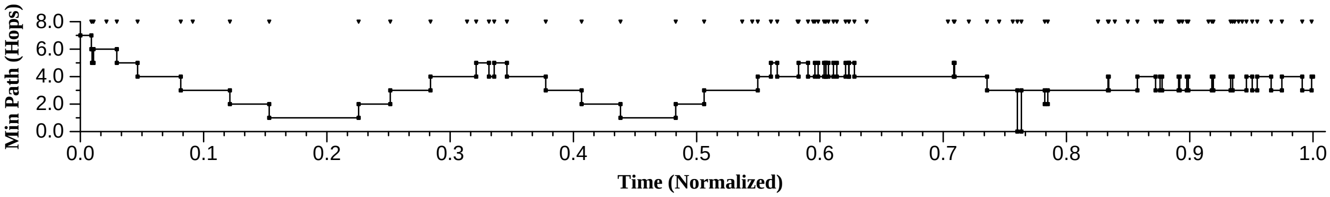


(d) mean speed = 20 m/s

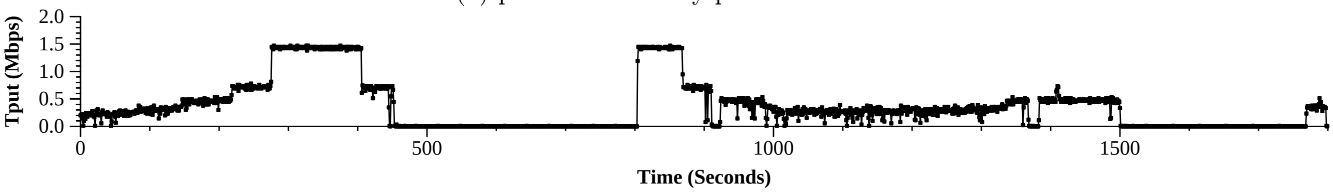


(e) mean speed = 30 m/s

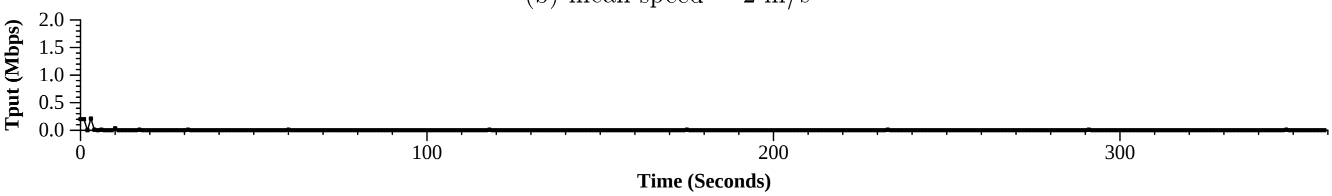
Figure 7: Profile for mobility pattern No. 4, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



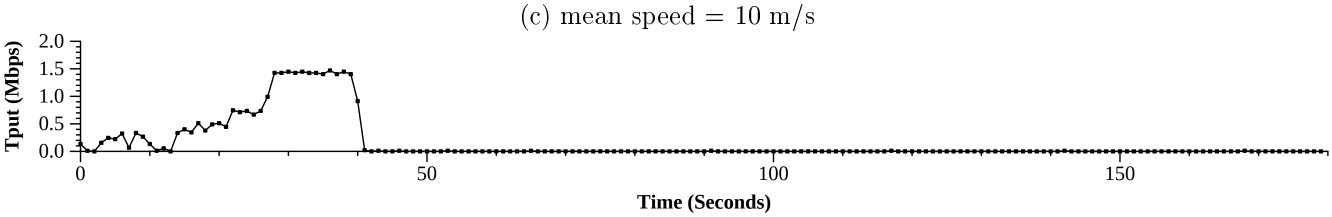
(a) profile of mobility pattern No. 5.



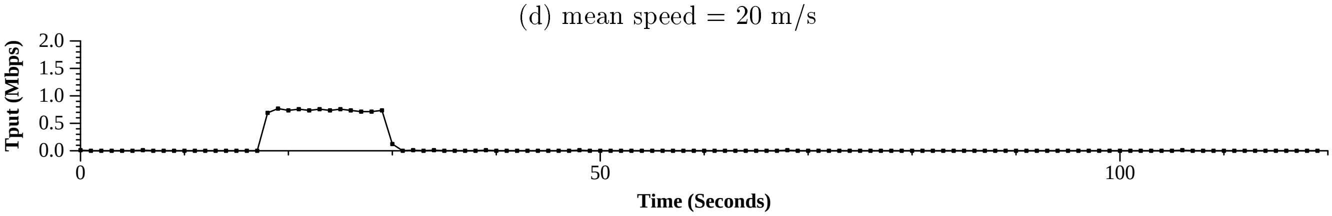
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

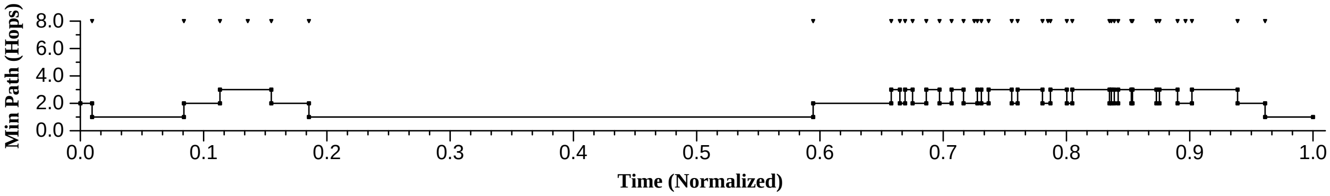


(d) mean speed = 20 m/s

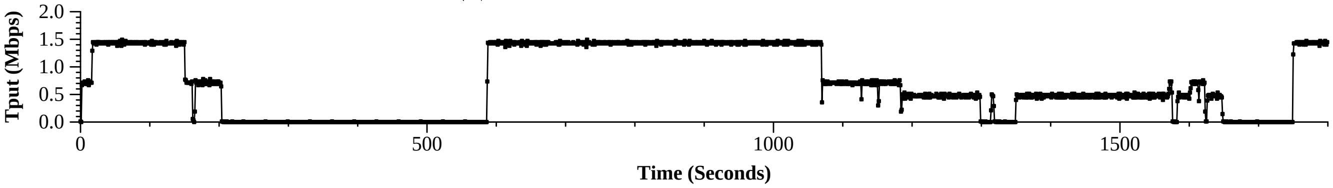


(e) mean speed = 30 m/s

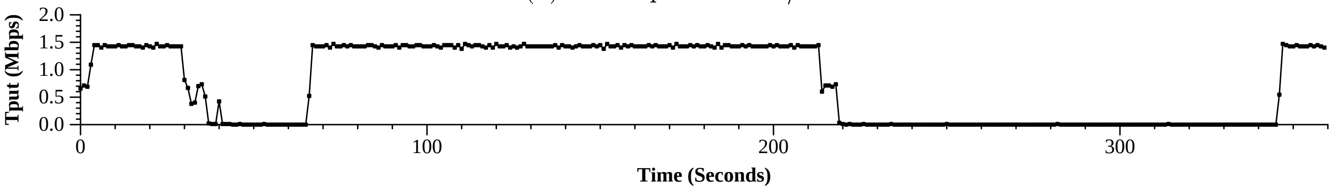
Figure 8: Profile for mobility pattern No. 5, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



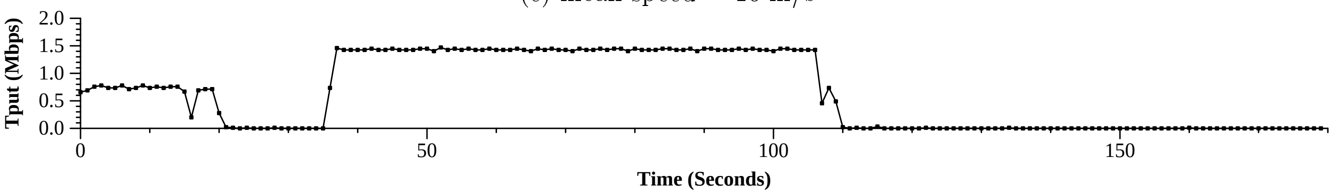
(a) profile of mobility pattern No. 6.



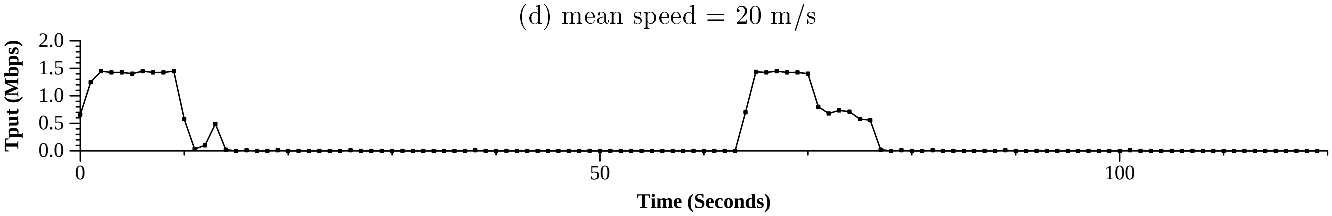
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

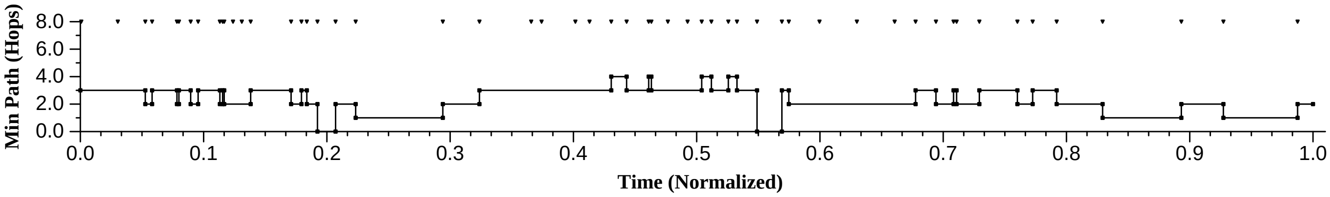


(d) mean speed = 20 m/s

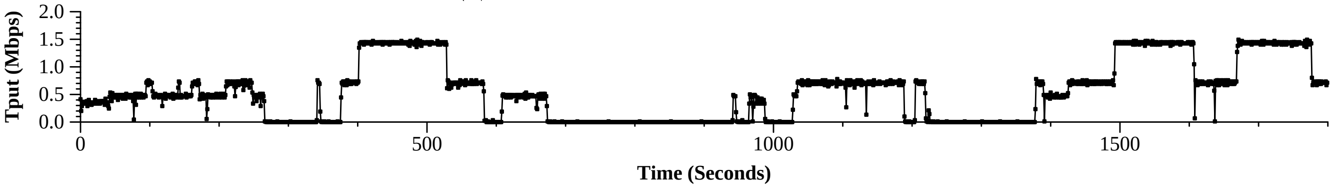


(e) mean speed = 30 m/s

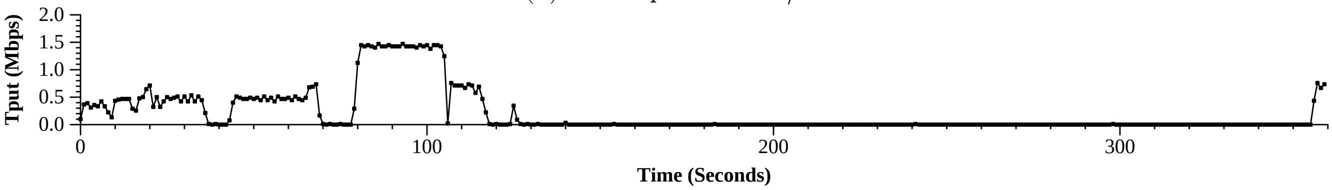
Figure 9: Profile for mobility pattern No. 6, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



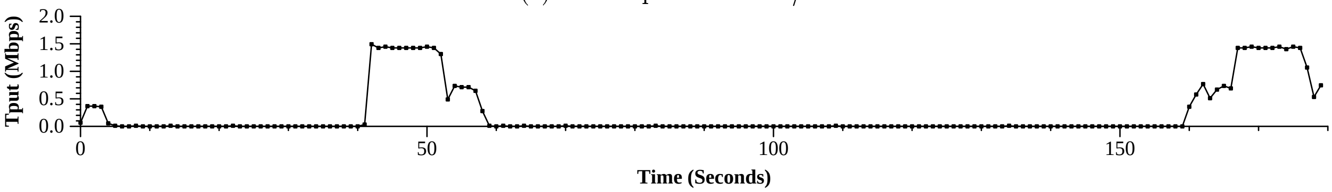
(a) profile of mobility pattern No. 7.



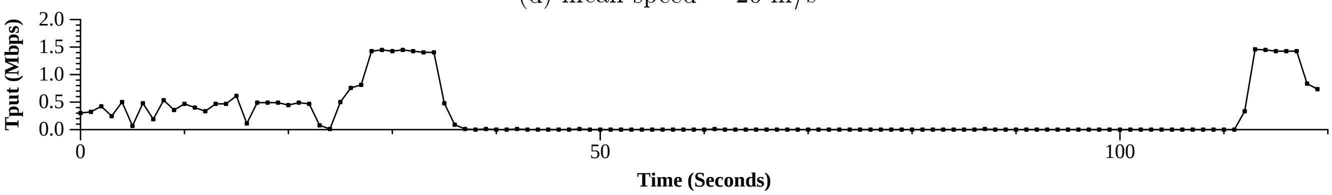
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

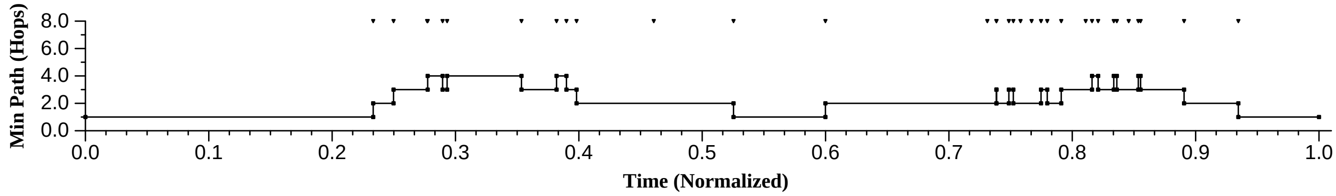


(d) mean speed = 20 m/s

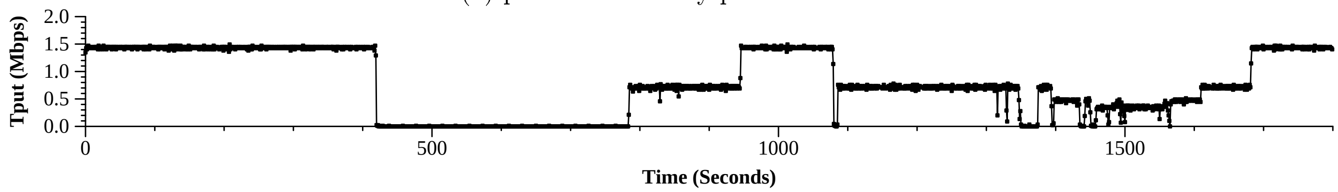


(e) mean speed = 30 m/s

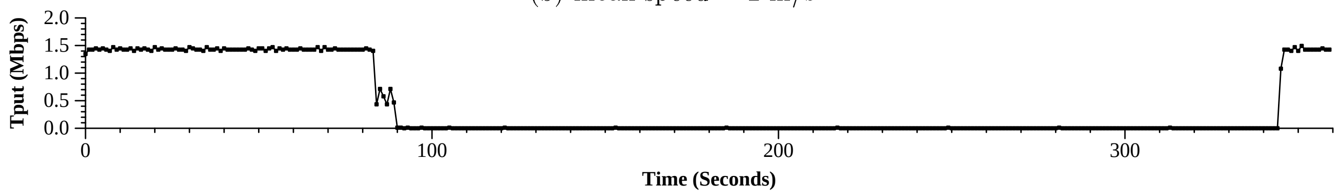
Figure 10: Profile for mobility pattern No. 7, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



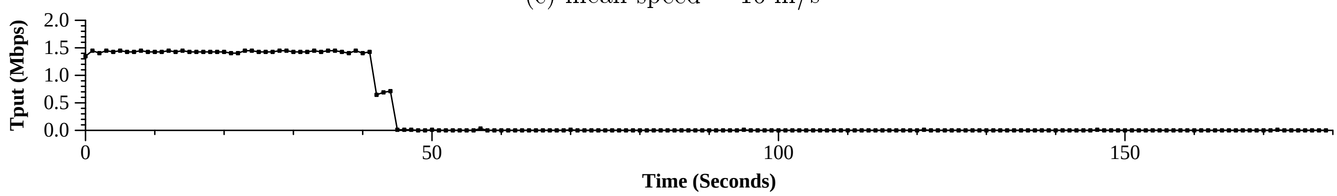
(a) profile of mobility pattern No. 8.



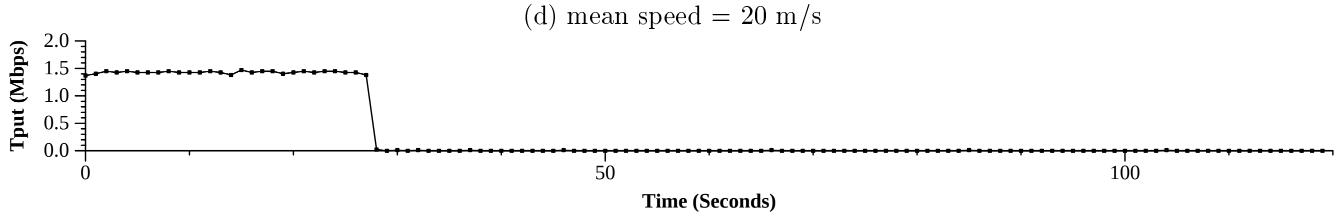
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

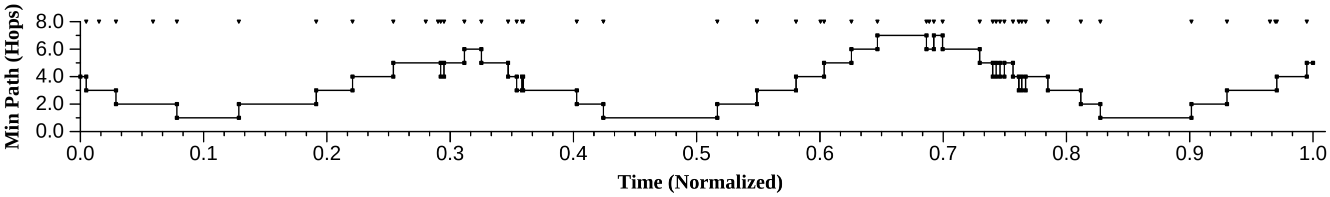


(d) mean speed = 20 m/s

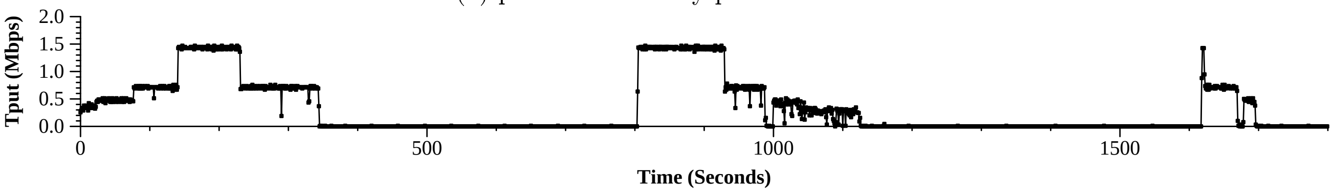


(e) mean speed = 30 m/s

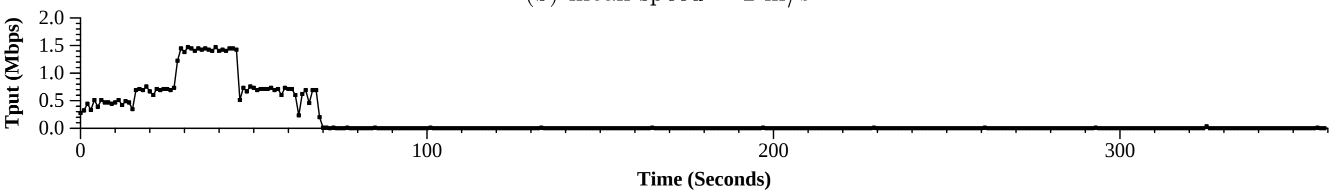
Figure 11: Profile for mobility pattern No. 8, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



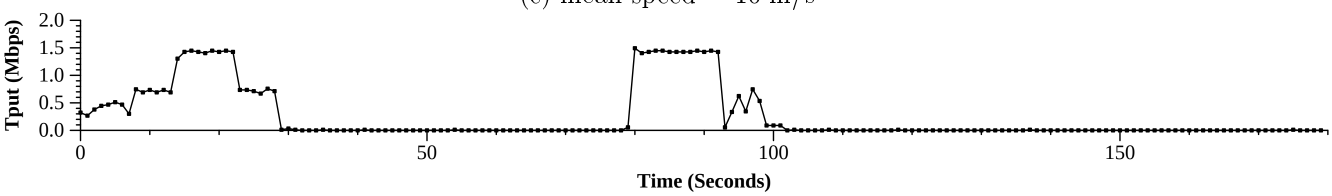
(a) profile of mobility pattern No. 9.



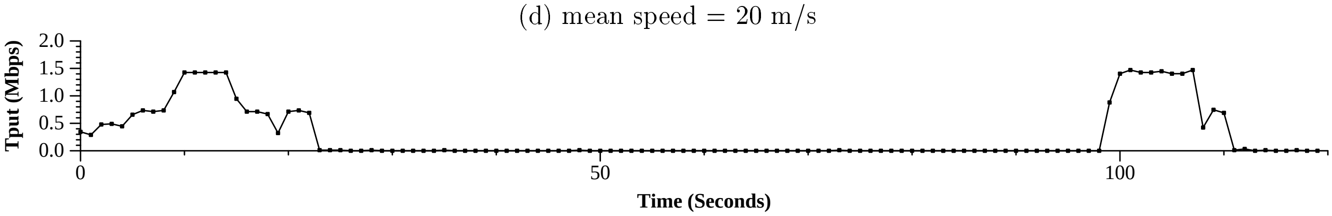
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

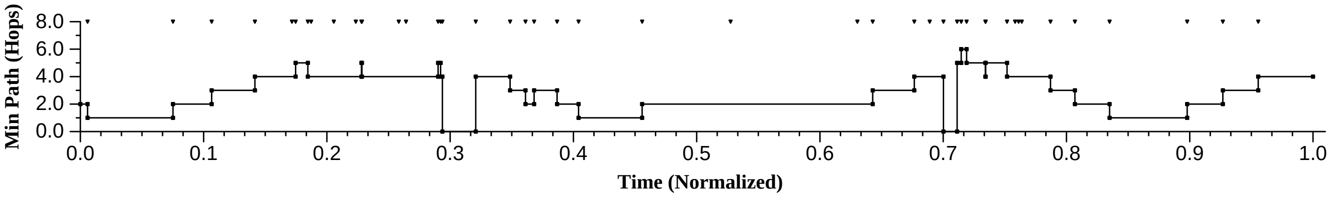


(d) mean speed = 20 m/s

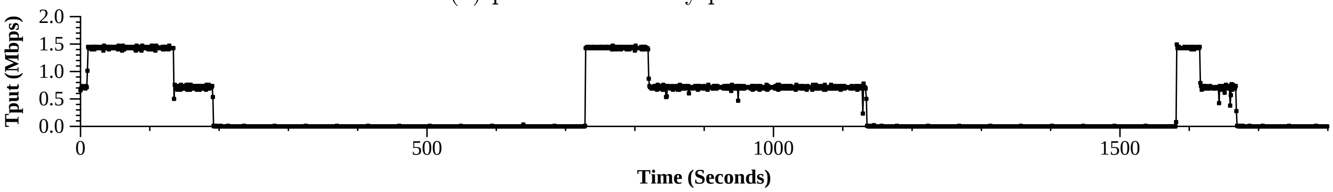


(e) mean speed = 30 m/s

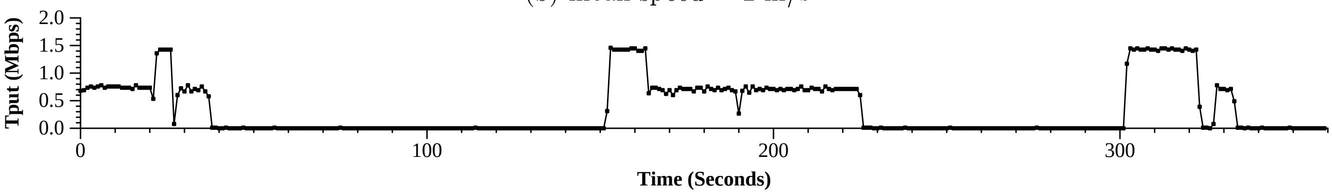
Figure 12: Profile for mobility pattern No. 9, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



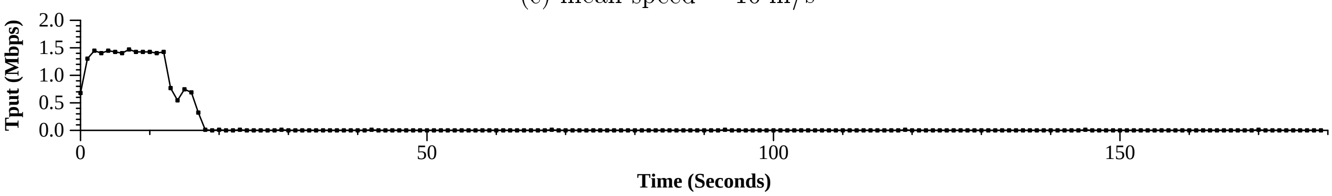
(a) profile of mobility pattern No. 10.



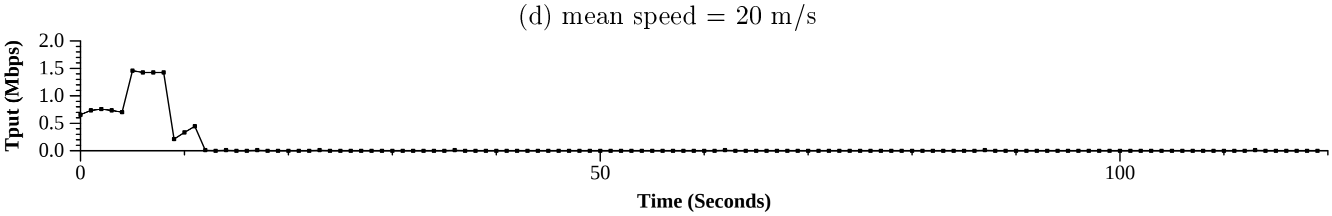
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

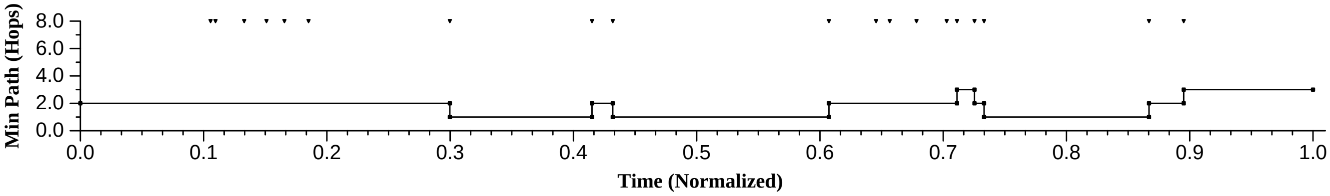


(d) mean speed = 20 m/s

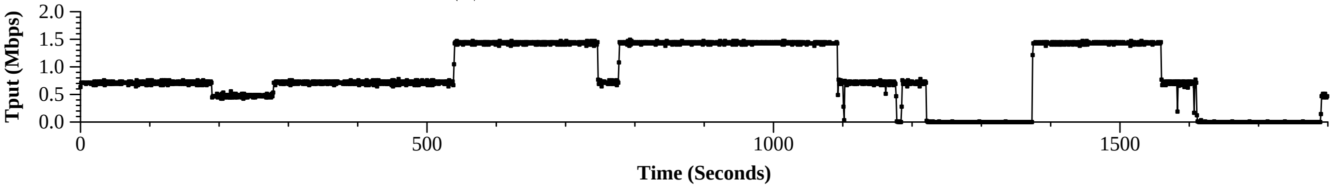


(e) mean speed = 30 m/s

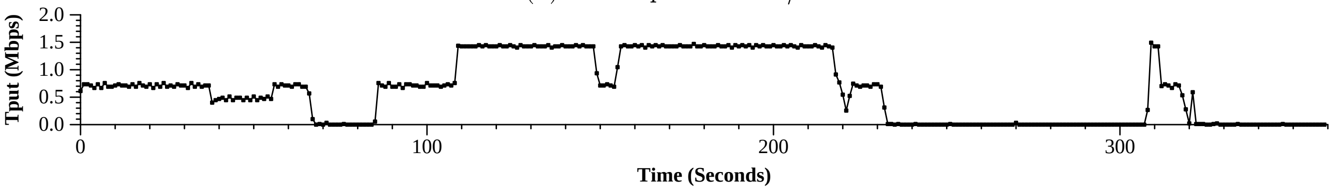
Figure 13: Profile for mobility pattern No. 10, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



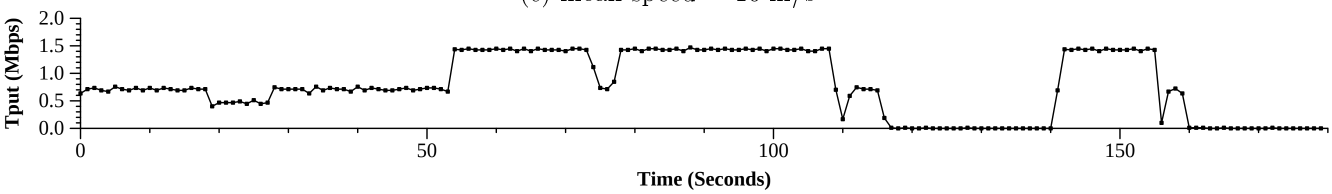
(a) profile of mobility pattern No. 11.



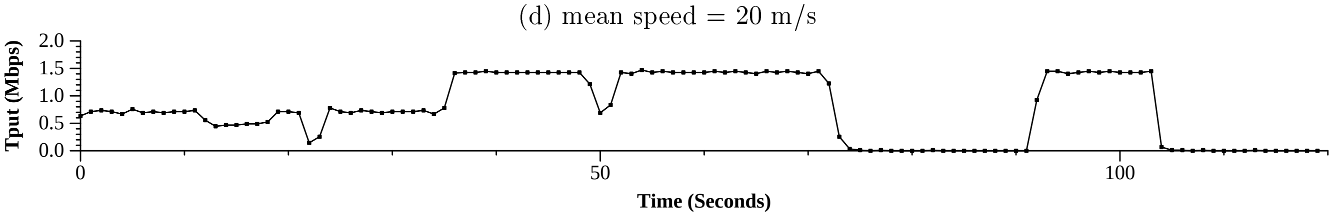
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

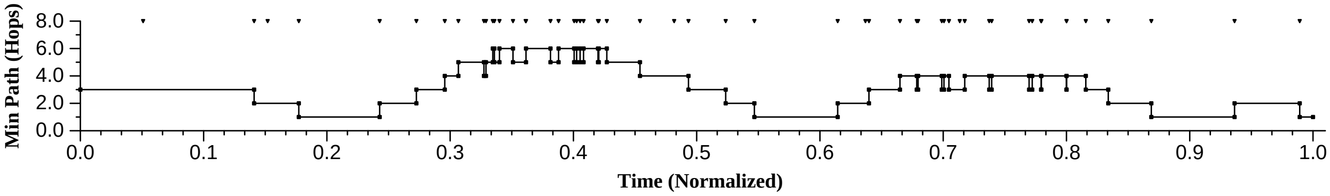


(d) mean speed = 20 m/s

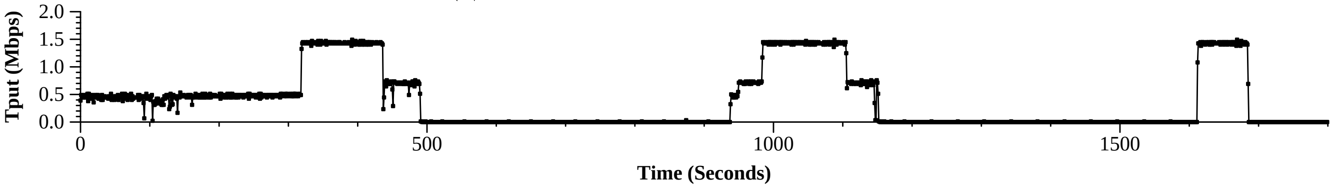


(e) mean speed = 30 m/s

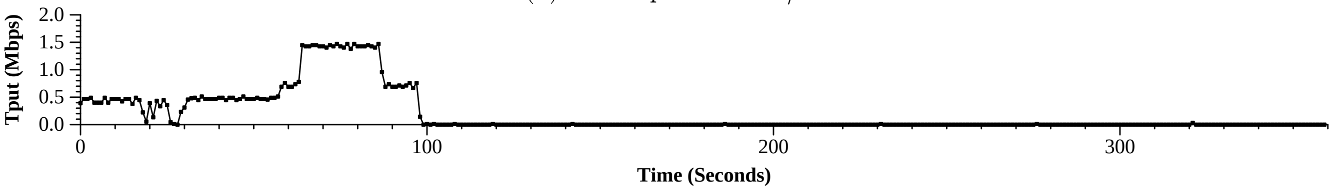
Figure 14: Profile for mobility pattern No. 11, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



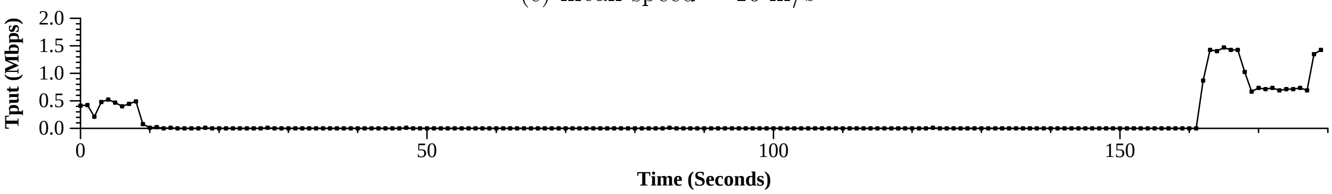
(a) profile of mobility pattern No. 12.



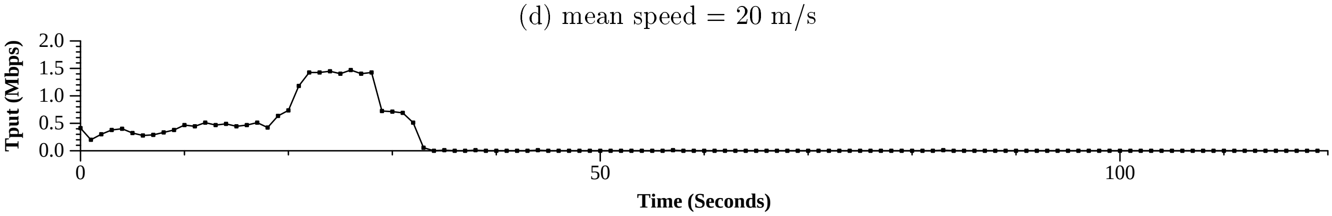
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

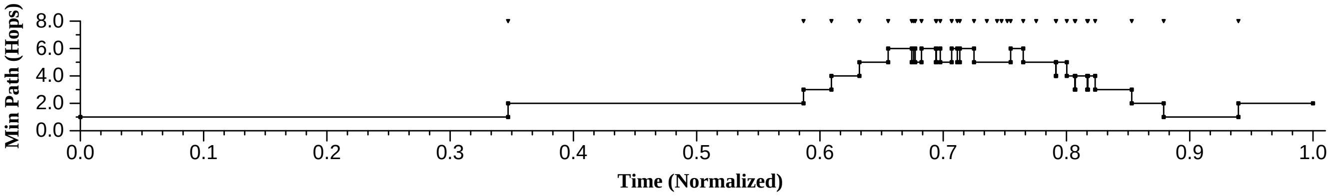


(d) mean speed = 20 m/s

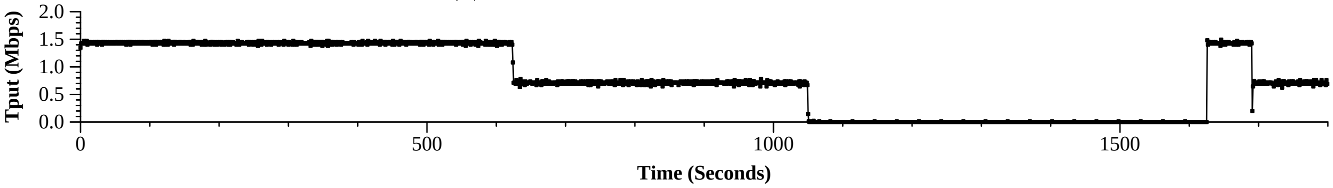


(e) mean speed = 30 m/s

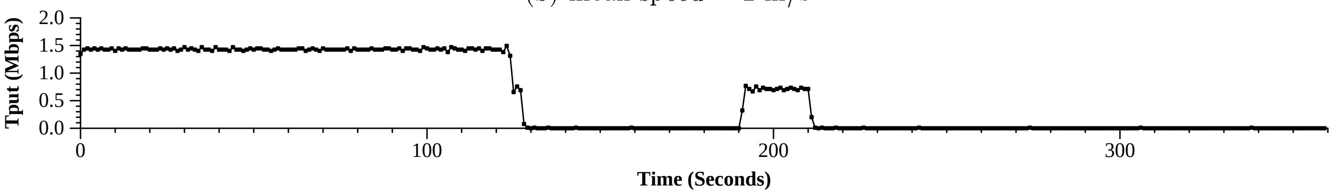
Figure 15: Profile for mobility pattern No. 12, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



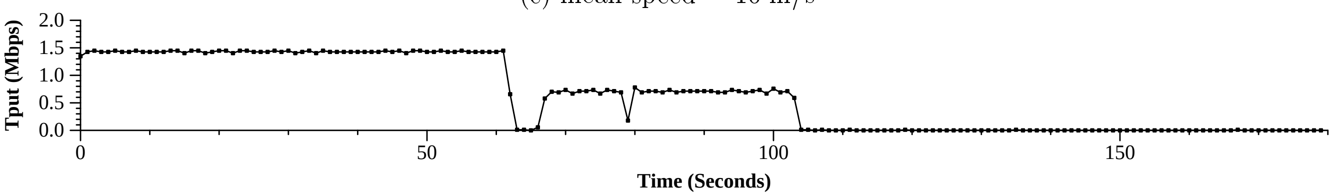
(a) profile of mobility pattern No. 13.



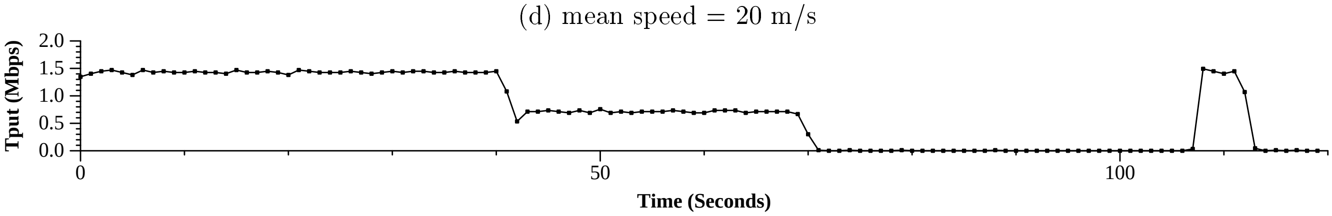
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

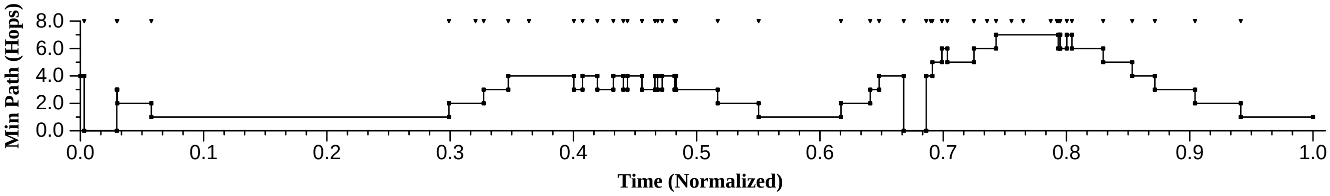


(d) mean speed = 20 m/s

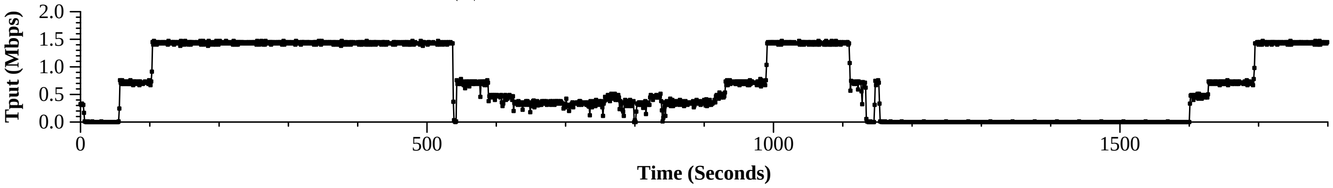


(e) mean speed = 30 m/s

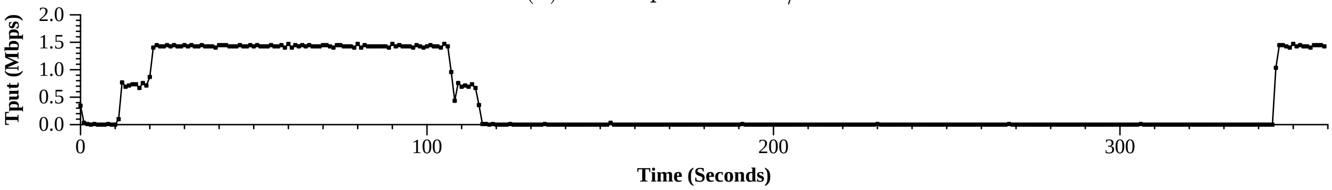
Figure 16: Profile for mobility pattern No. 13, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



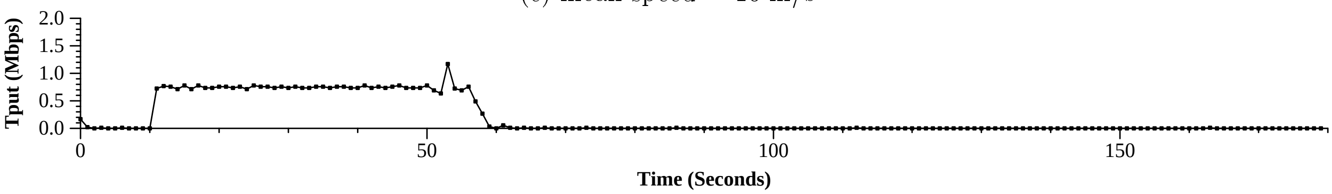
(a) profile of mobility pattern No. 14.



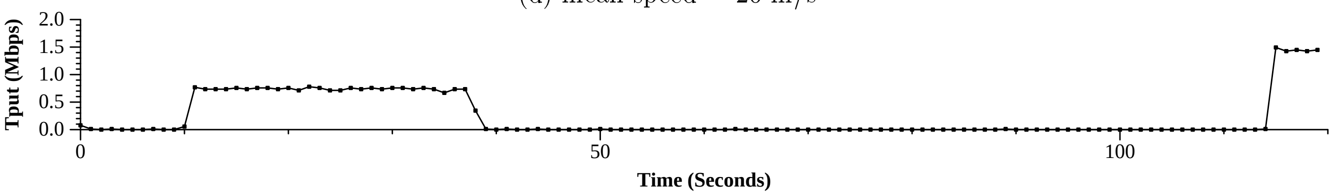
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

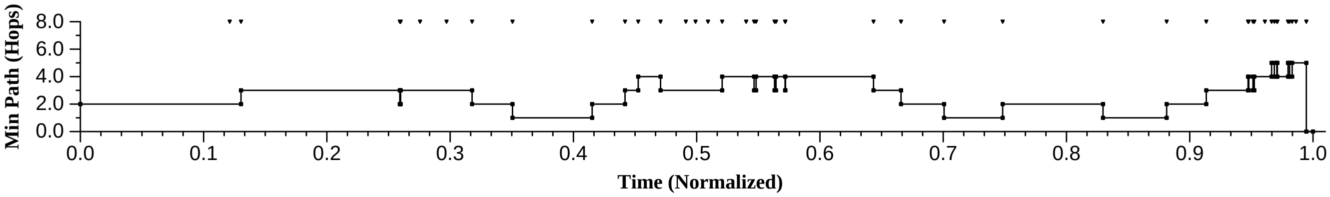


(d) mean speed = 20 m/s

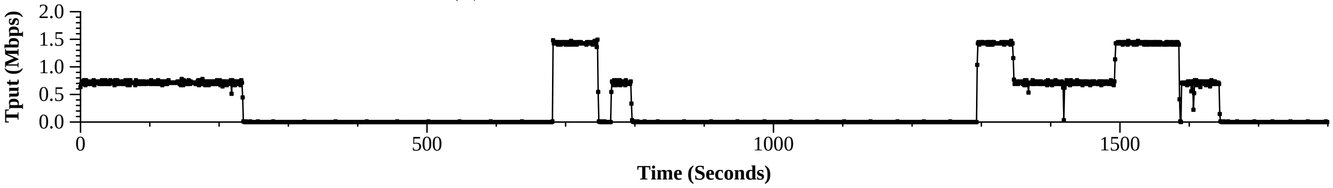


(e) mean speed = 30 m/s

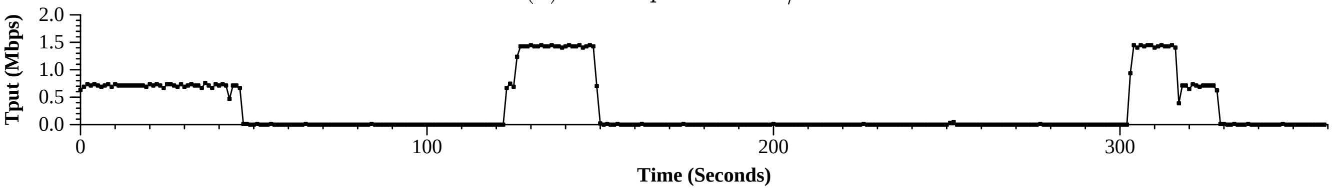
Figure 17: Profile for mobility pattern No. 14, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



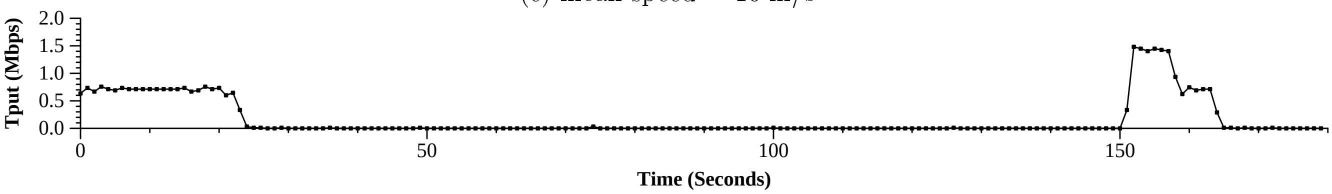
(a) profile of mobility pattern No. 15.



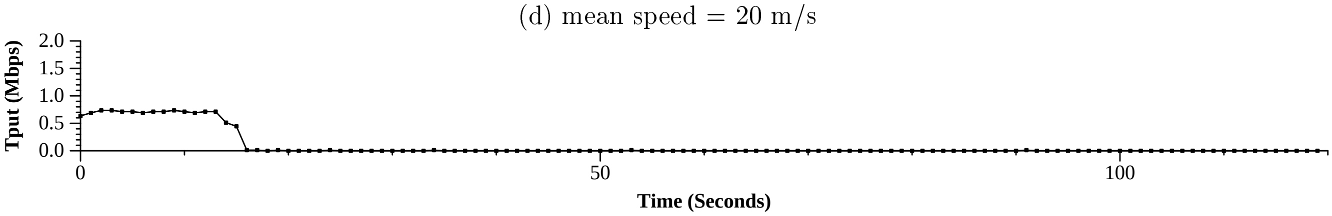
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

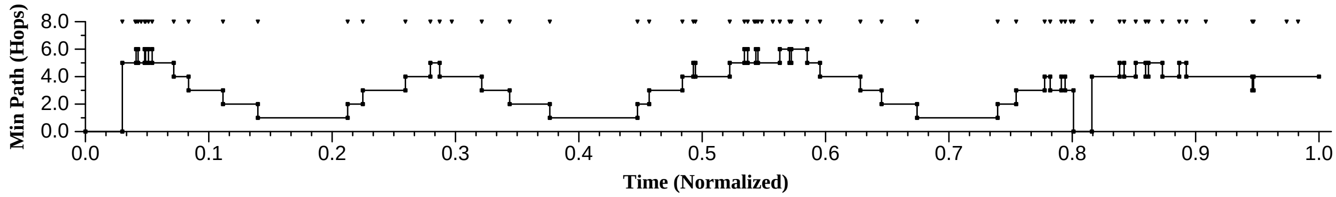


(d) mean speed = 20 m/s

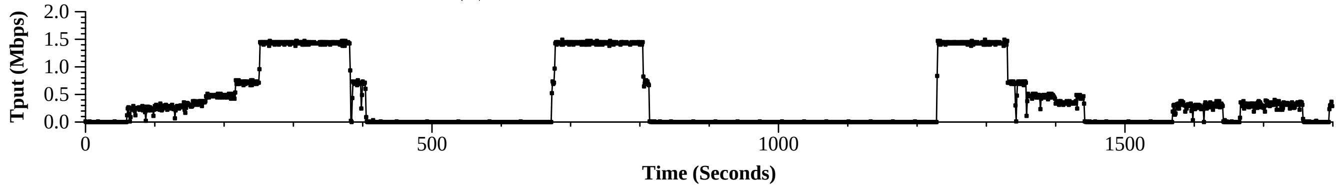


(e) mean speed = 30 m/s

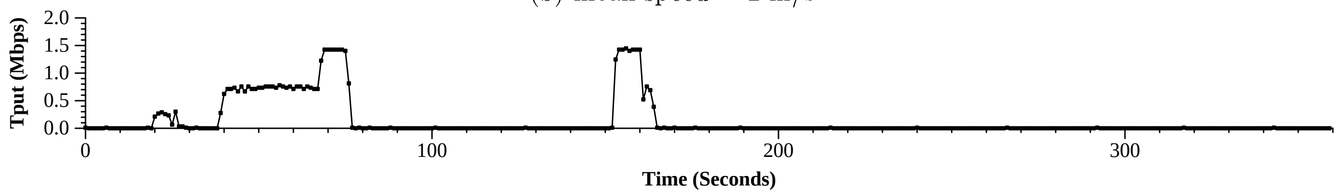
Figure 18: Profile for mobility pattern No. 15, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



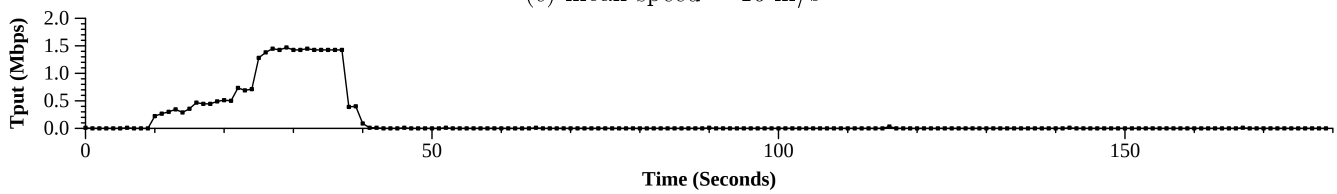
(a) profile of mobility pattern No. 16.



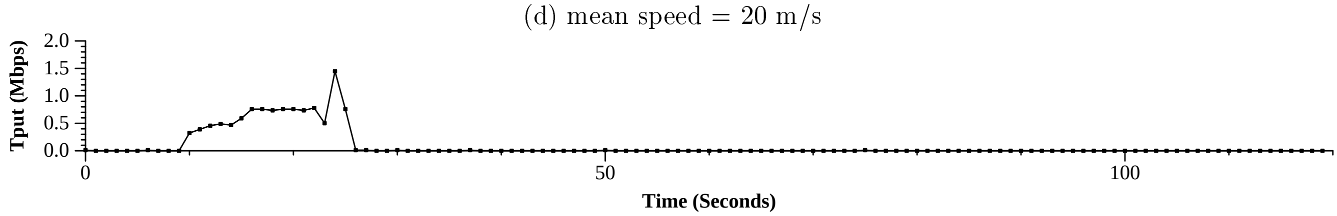
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

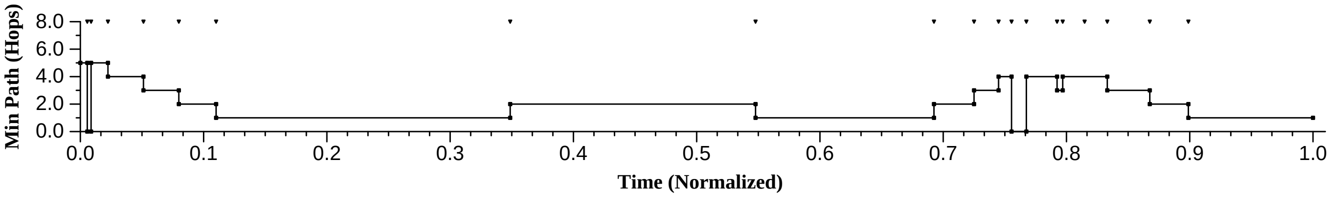


(d) mean speed = 20 m/s

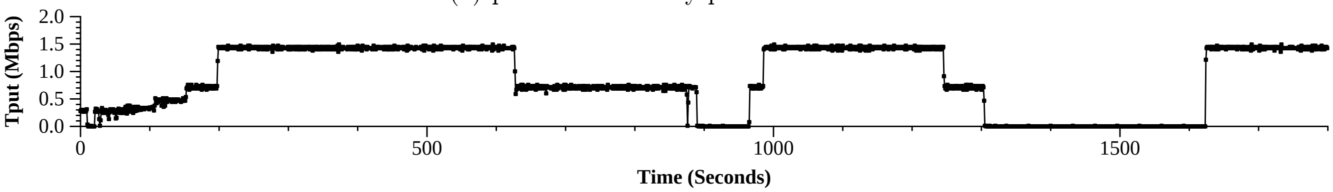


(e) mean speed = 30 m/s

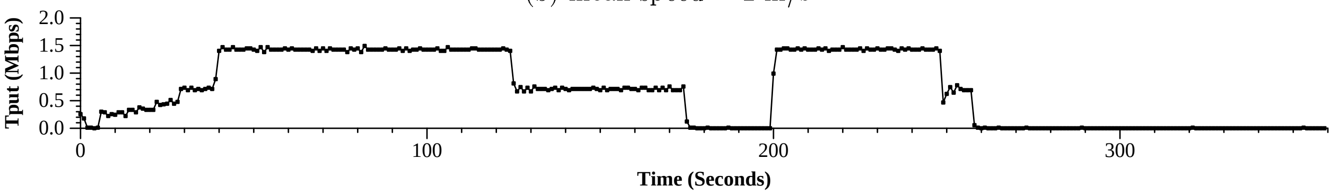
Figure 19: Profile for mobility pattern No. 16, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



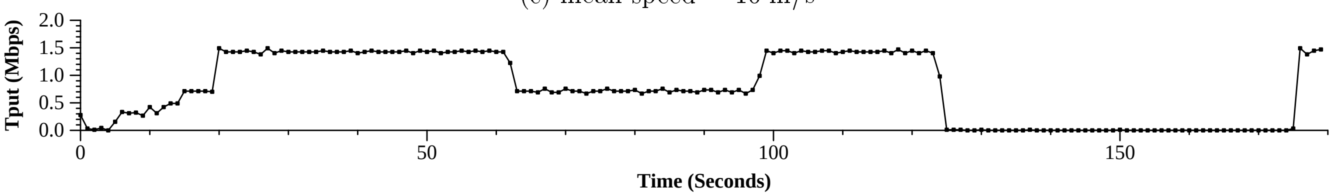
(a) profile of mobility pattern No. 17.



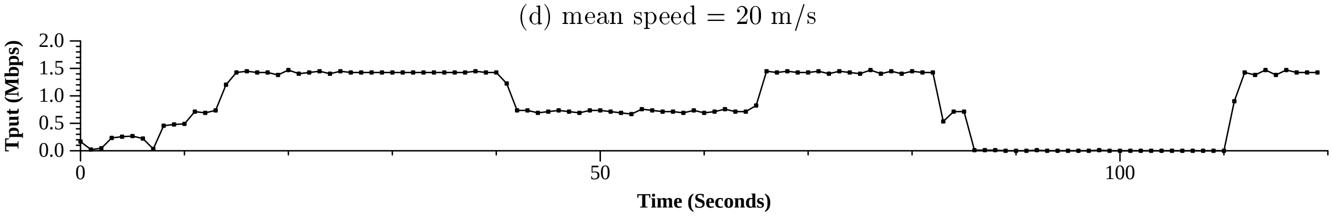
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

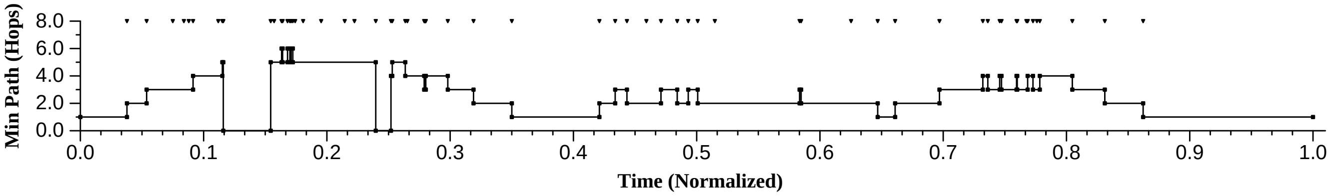


(d) mean speed = 20 m/s

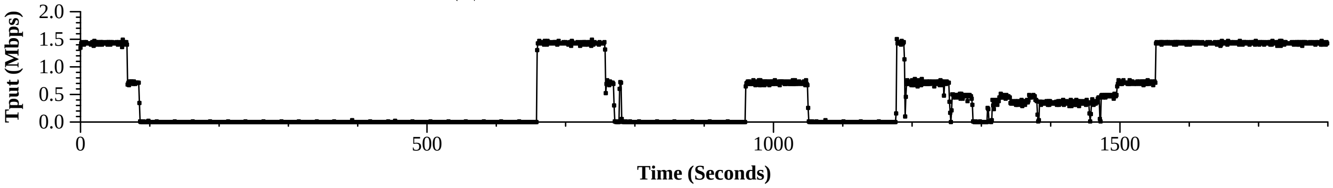


(e) mean speed = 30 m/s

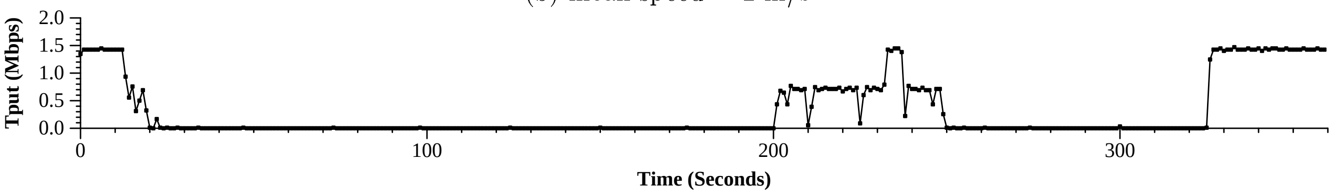
Figure 20: Profile for mobility pattern No. 17, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



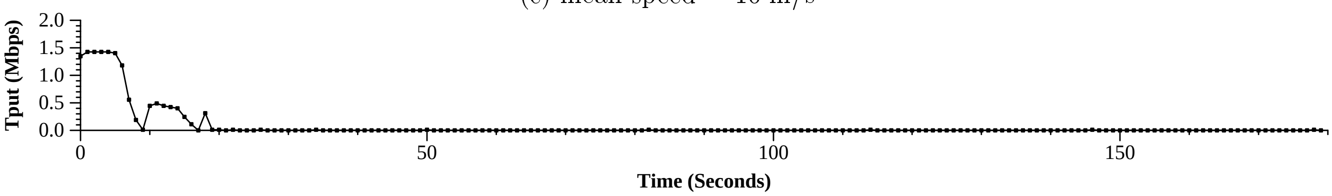
(a) profile of mobility pattern No. 18.



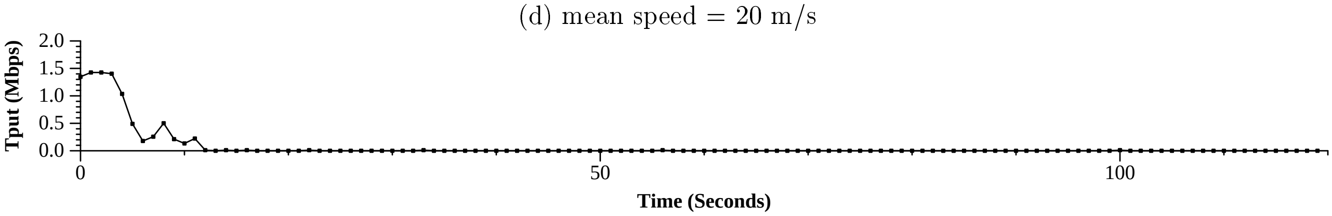
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

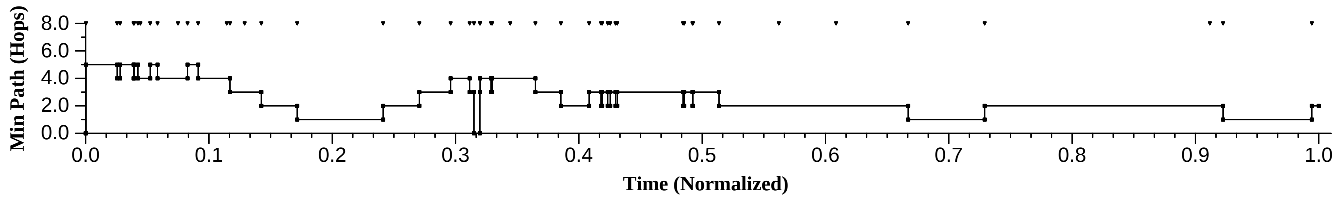


(d) mean speed = 20 m/s



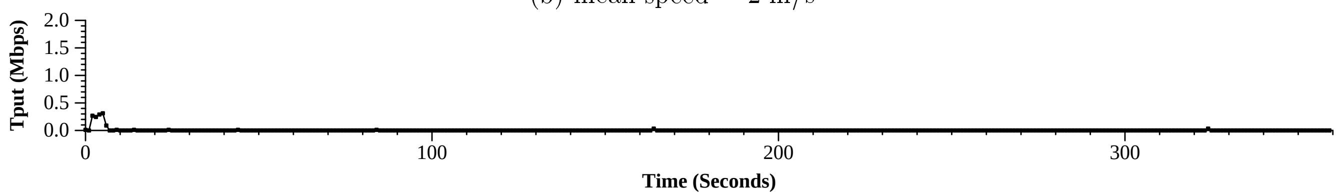
(e) mean speed = 30 m/s

Figure 21: Profile for mobility pattern No. 18, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.

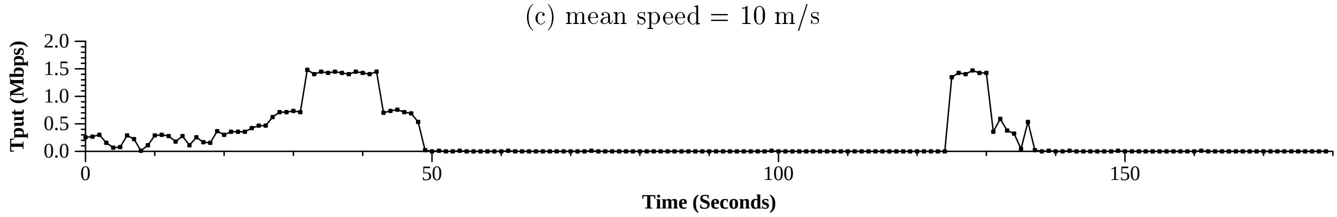


(a) profile of mobility pattern No. 19.

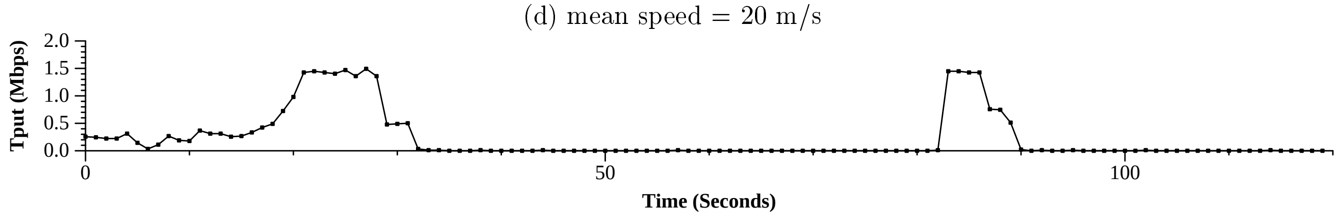
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

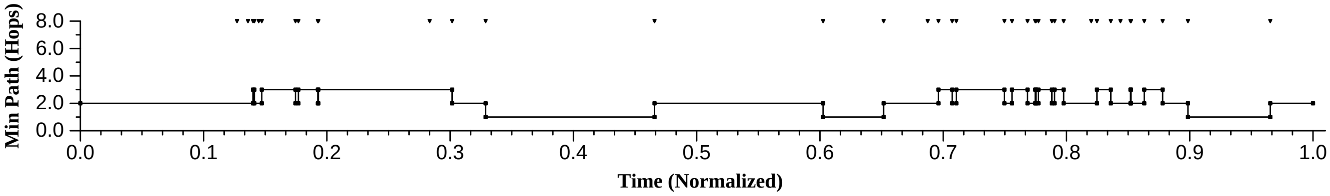


(d) mean speed = 20 m/s

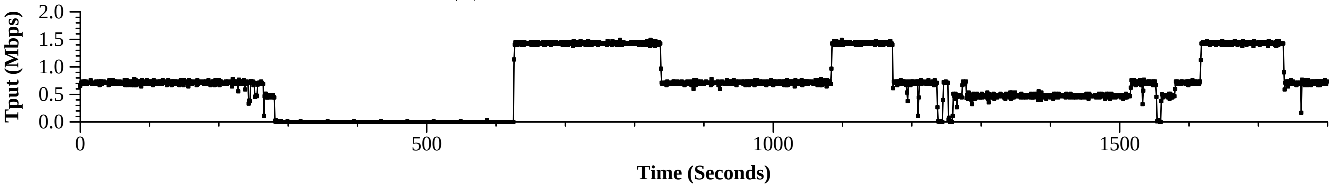


(e) mean speed = 30 m/s

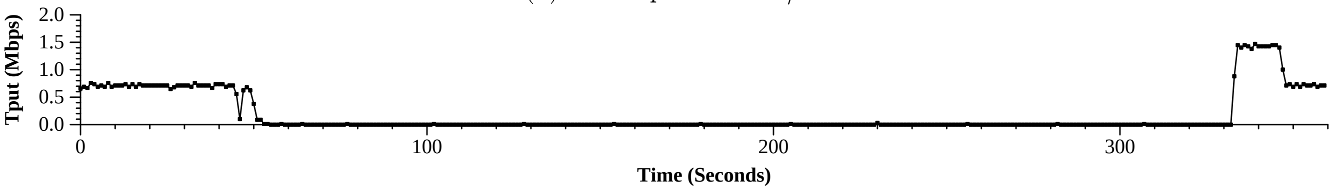
Figure 22: Profile for mobility pattern No. 19, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



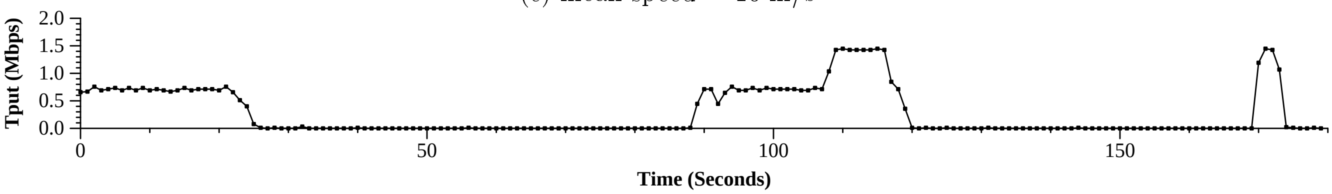
(a) profile of mobility pattern No. 20.



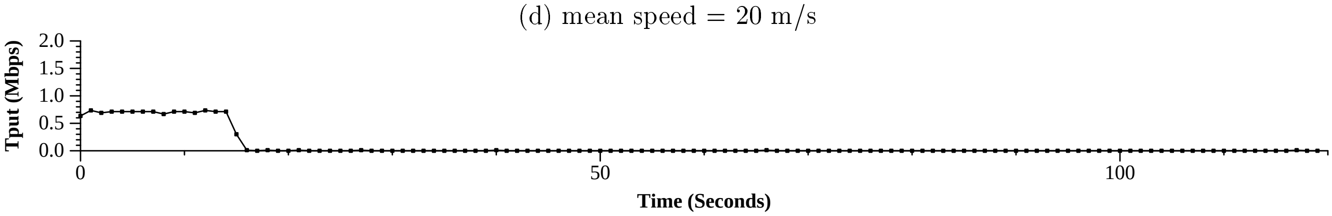
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

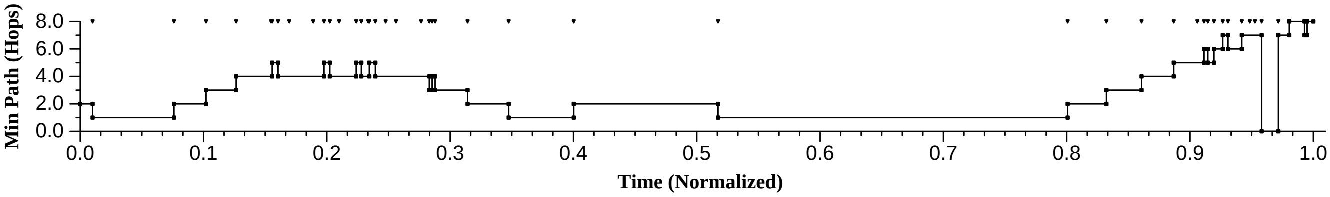


(d) mean speed = 20 m/s

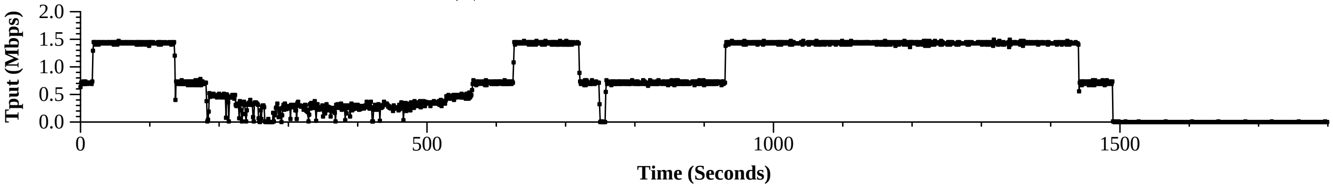


(e) mean speed = 30 m/s

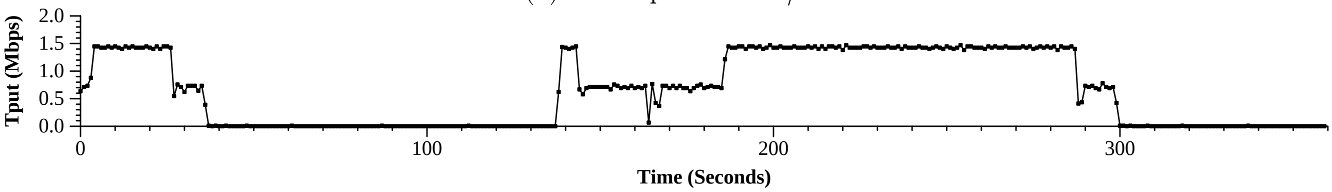
Figure 23: Profile for mobility pattern No. 20, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



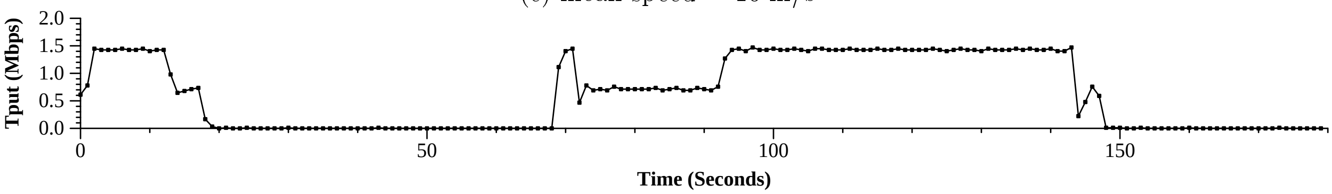
(a) profile of mobility pattern No. 21.



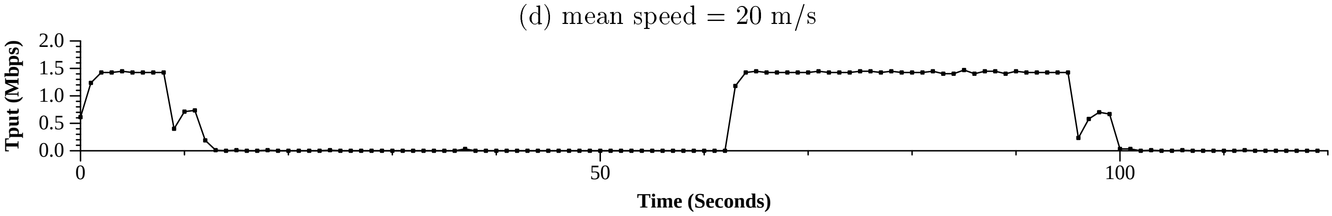
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

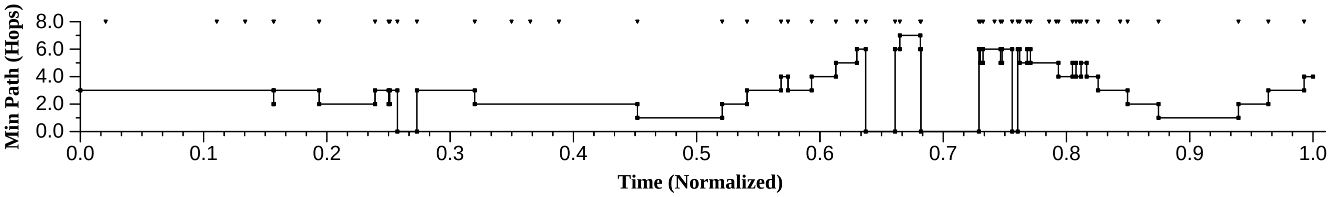


(d) mean speed = 20 m/s

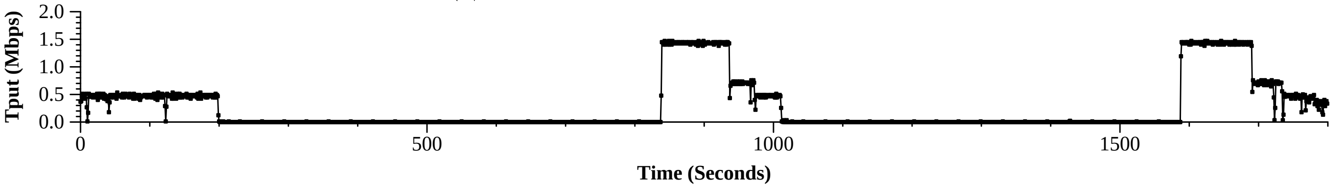


(e) mean speed = 30 m/s

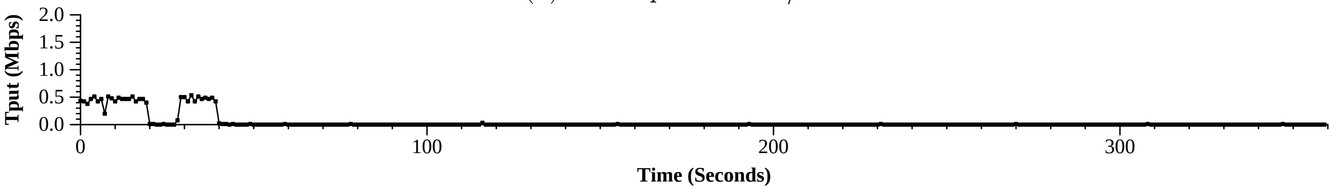
Figure 24: Profile for mobility pattern No. 21, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



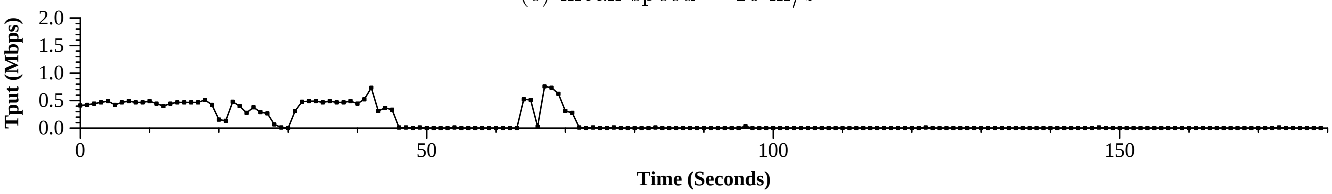
(a) profile of mobility pattern No. 22.



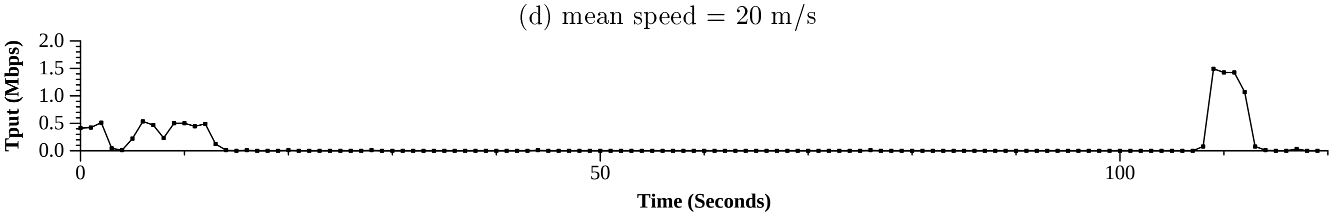
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

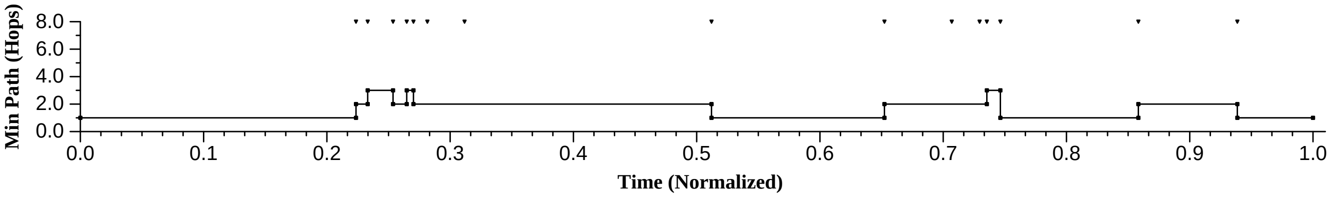


(d) mean speed = 20 m/s

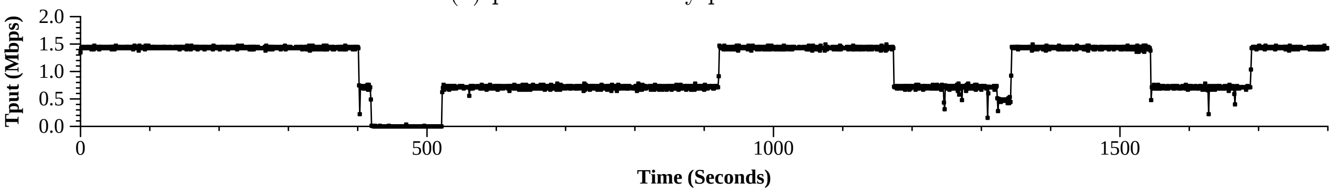


(e) mean speed = 30 m/s

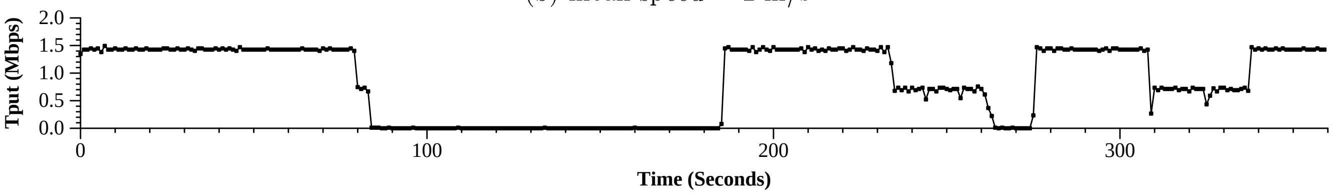
Figure 25: Profile for mobility pattern No. 22, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



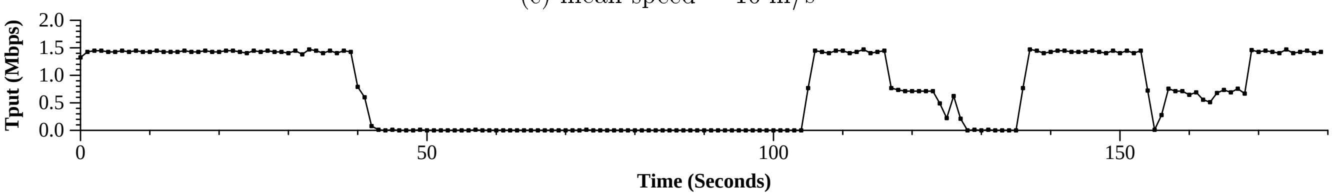
(a) profile of mobility pattern No. 23.



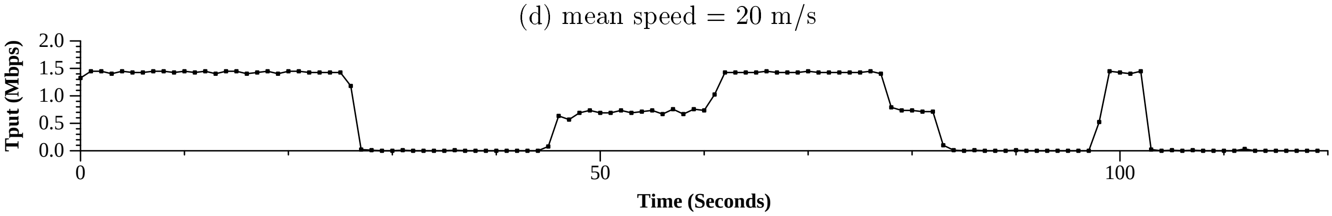
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

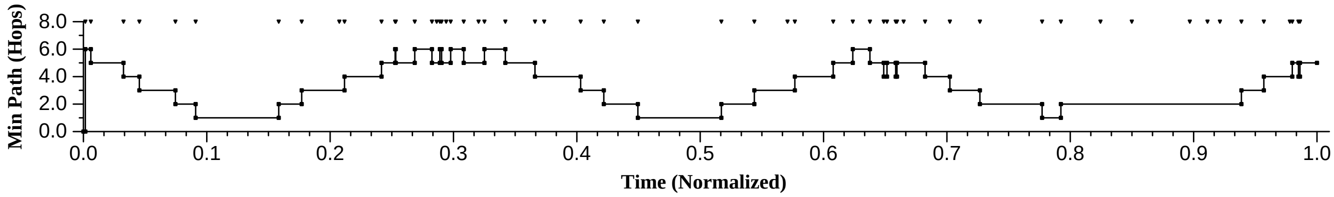


(d) mean speed = 20 m/s

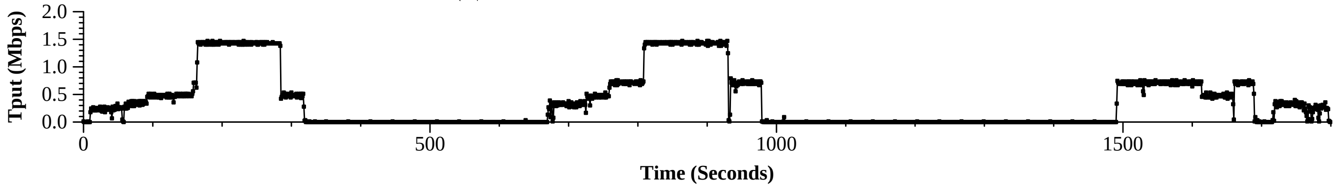


(e) mean speed = 30 m/s

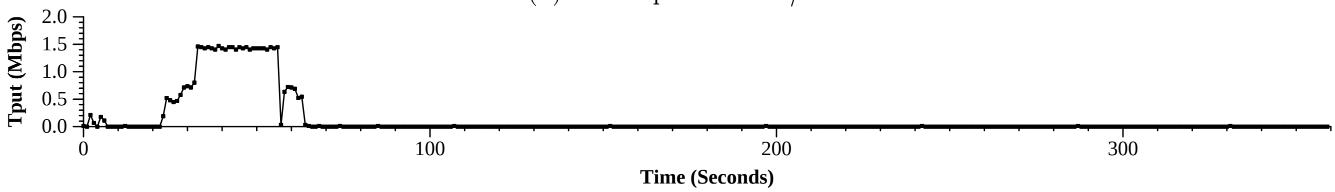
Figure 26: Profile for mobility pattern No. 23, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



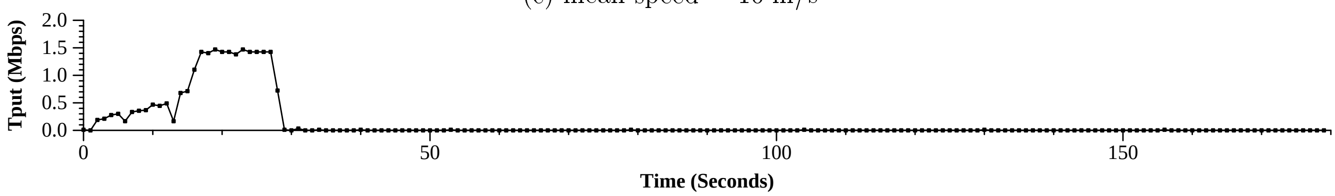
(a) profile of mobility pattern No. 24.



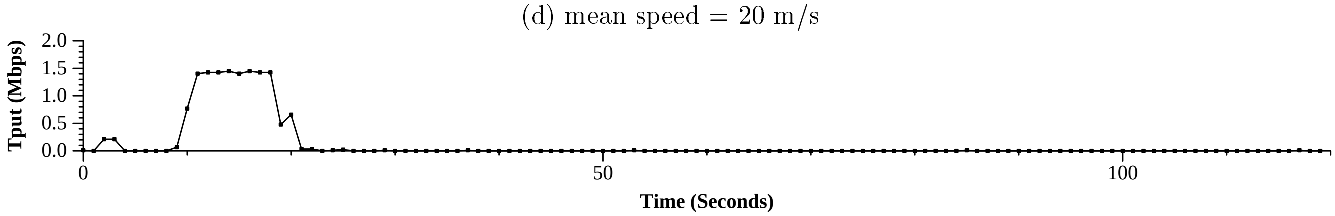
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

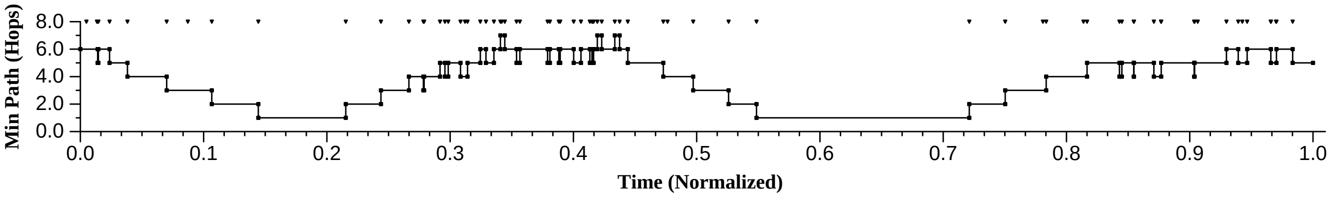


(d) mean speed = 20 m/s

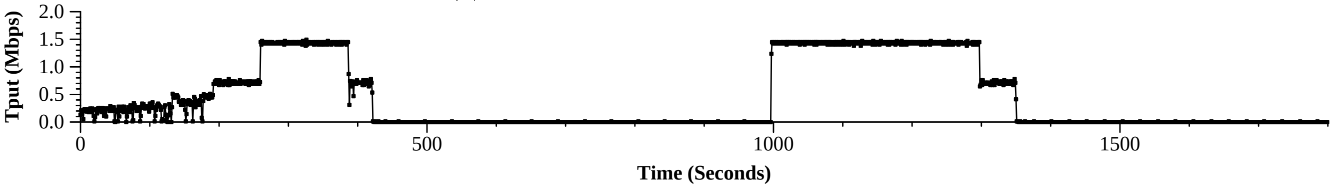


(e) mean speed = 30 m/s

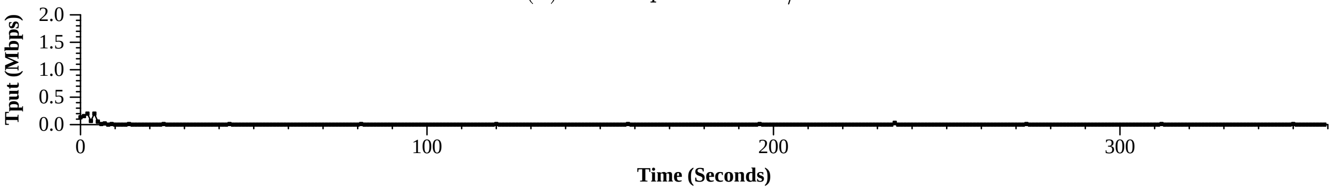
Figure 27: Profile for mobility pattern No. 24, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



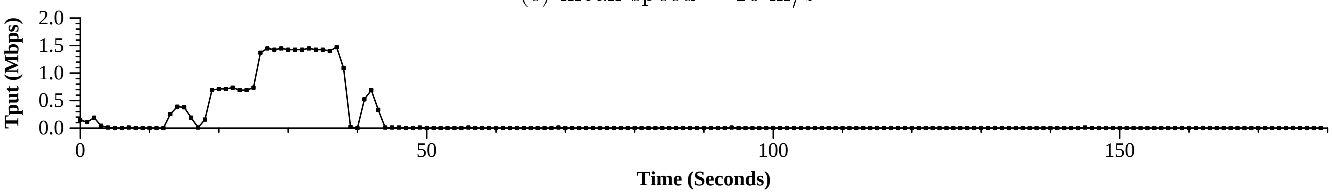
(a) profile of mobility pattern No. 25.



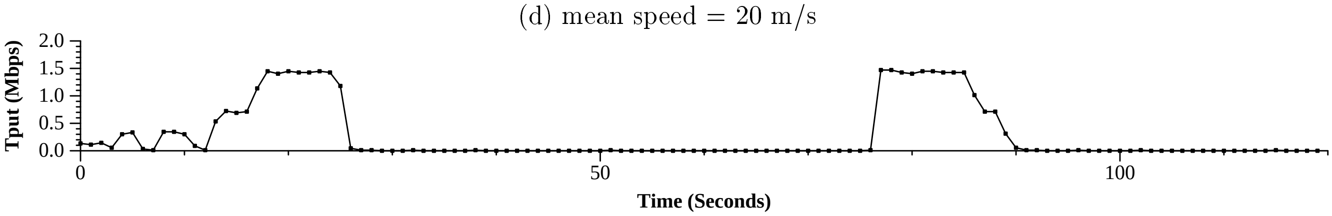
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

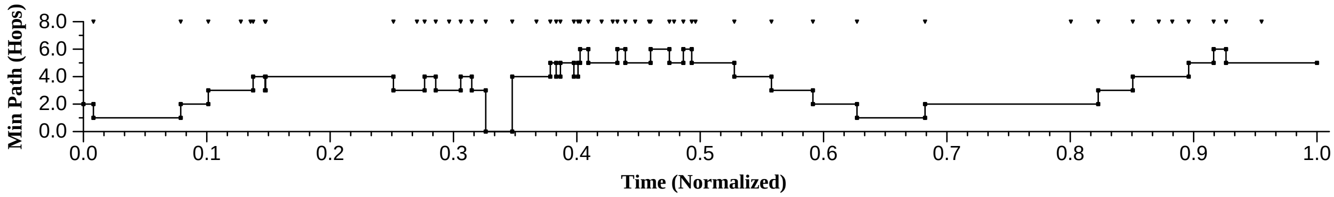


(d) mean speed = 20 m/s

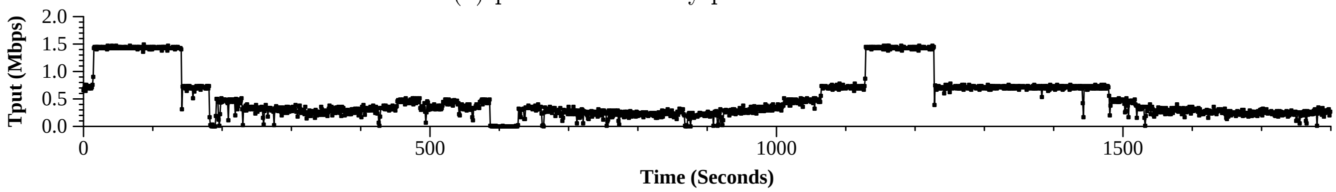


(e) mean speed = 30 m/s

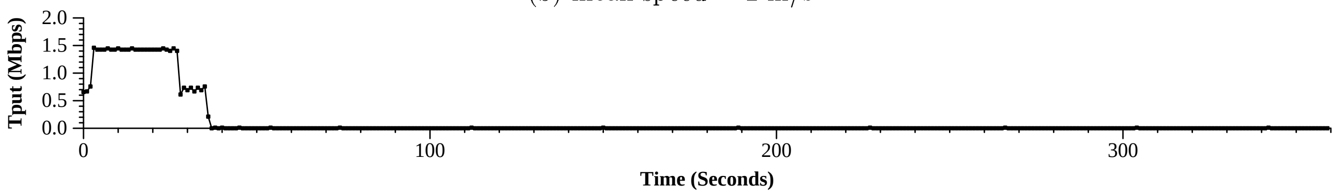
Figure 28: Profile for mobility pattern No. 25, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



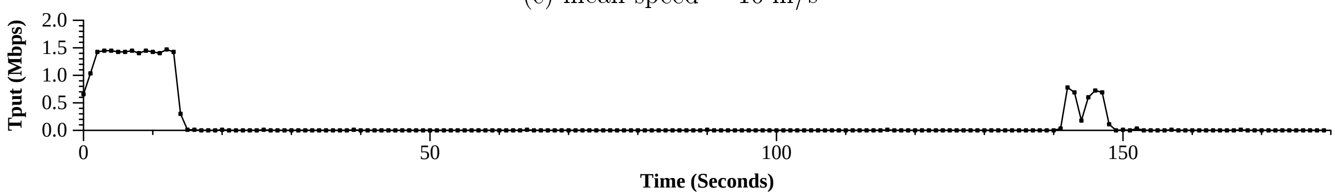
(a) profile of mobility pattern No. 26.



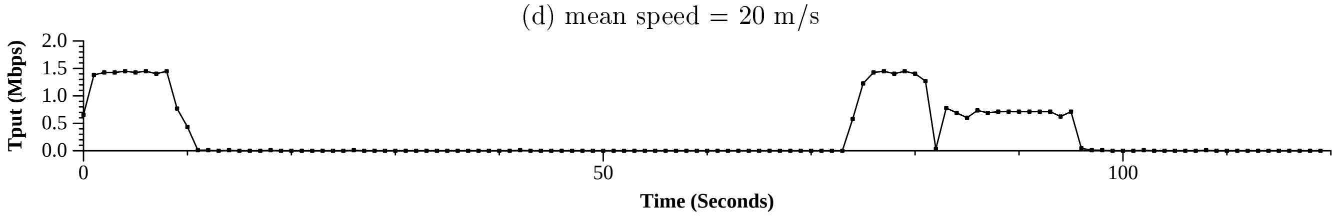
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

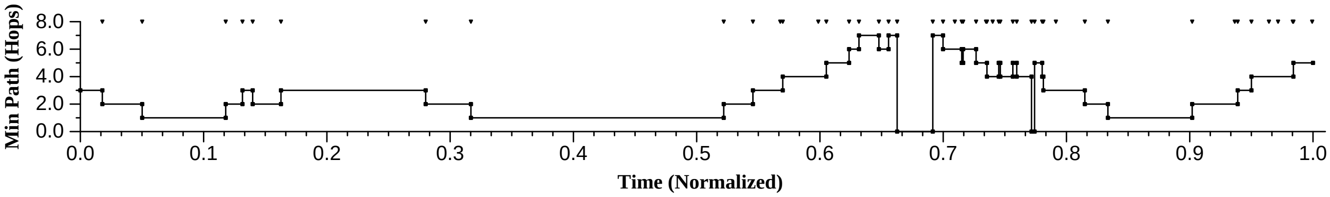


(d) mean speed = 20 m/s

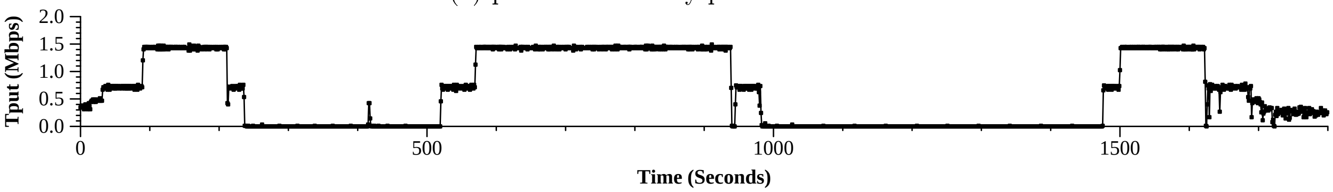


(e) mean speed = 30 m/s

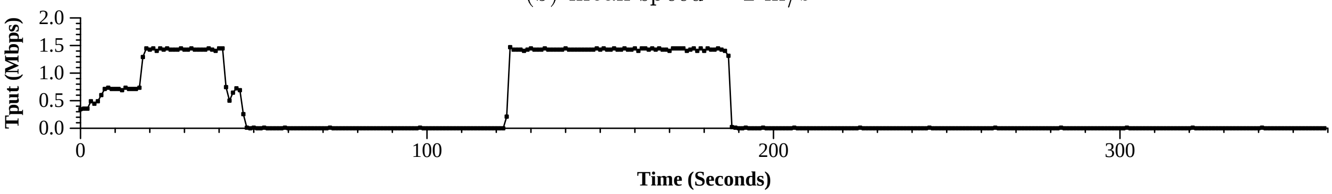
Figure 29: Profile for mobility pattern No. 26, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



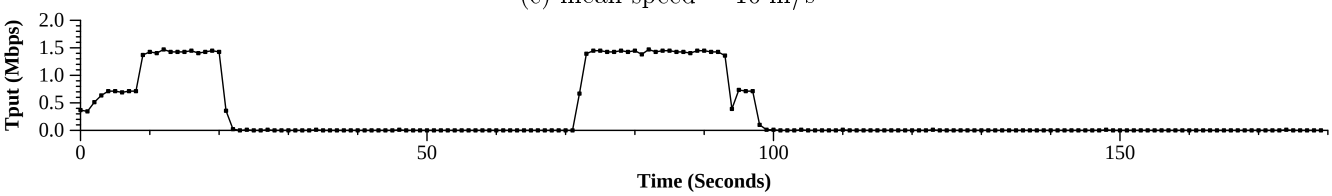
(a) profile of mobility pattern No. 27.



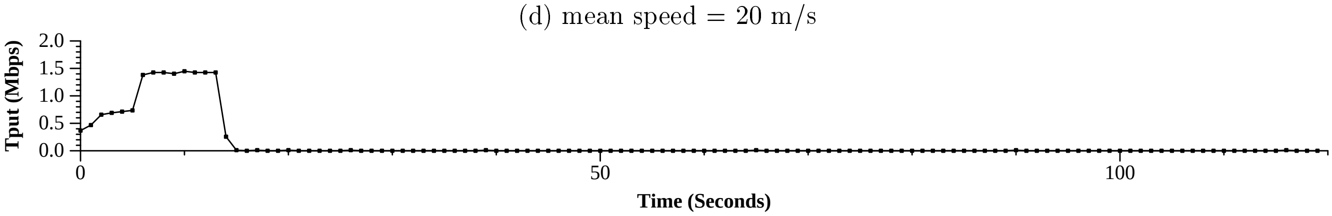
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

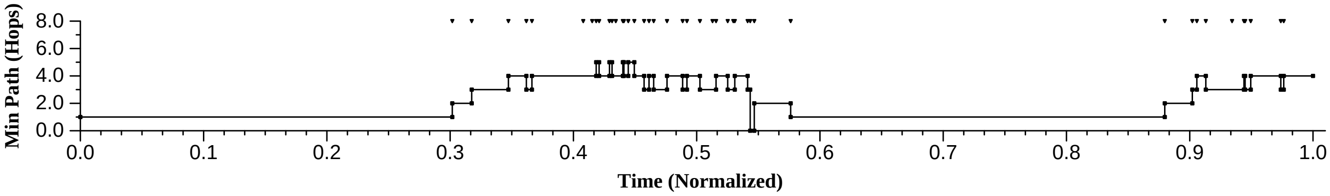


(d) mean speed = 20 m/s

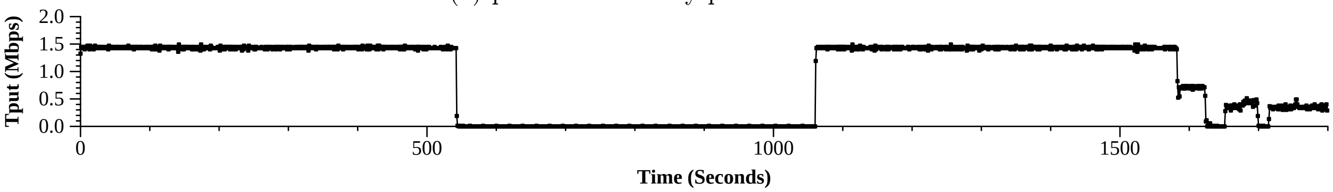


(e) mean speed = 30 m/s

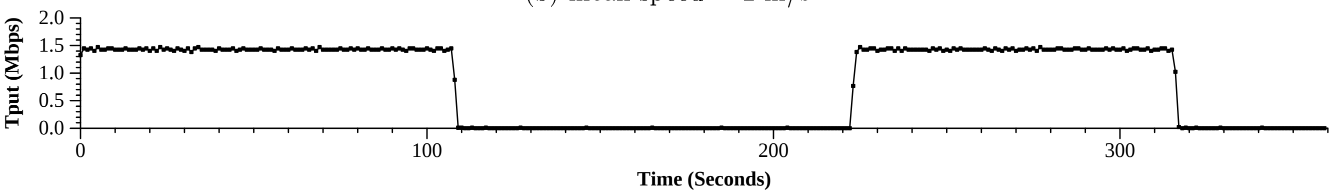
Figure 30: Profile for mobility pattern No. 27, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



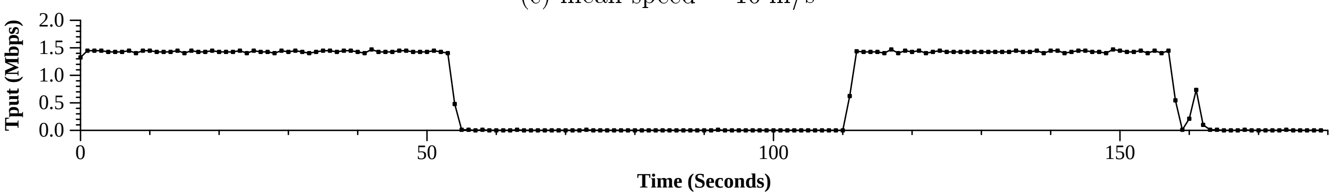
(a) profile of mobility pattern No. 28.



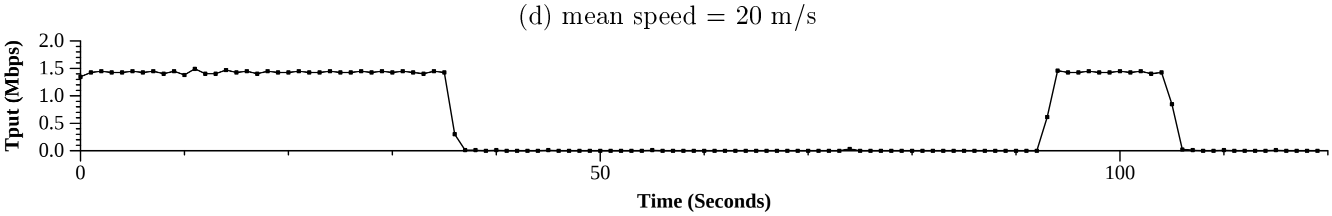
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

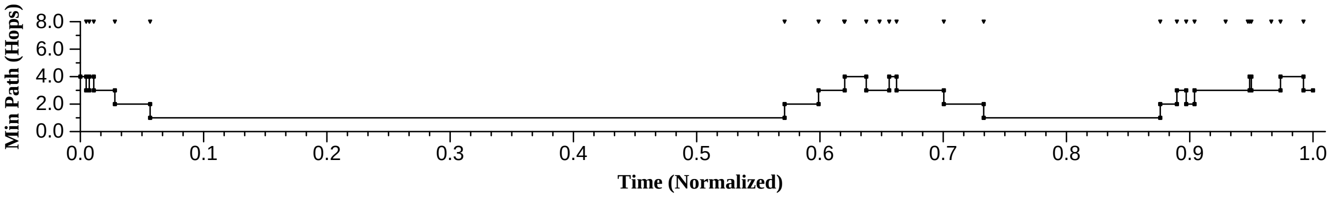


(d) mean speed = 20 m/s

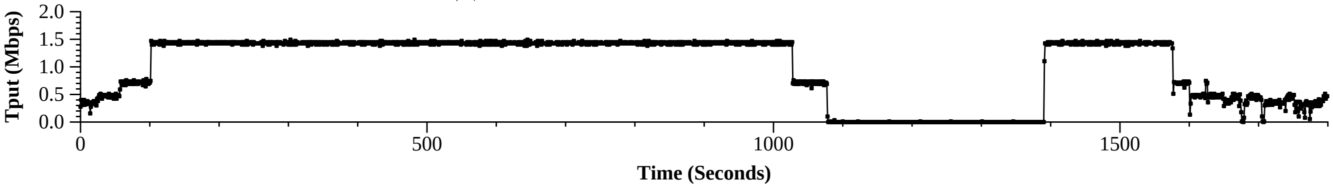


(e) mean speed = 30 m/s

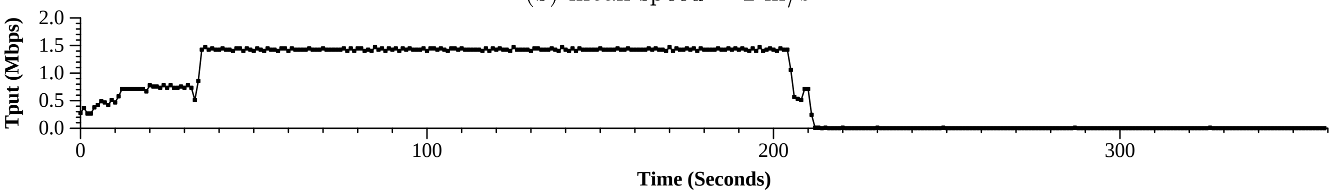
Figure 31: Profile for mobility pattern No. 28, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



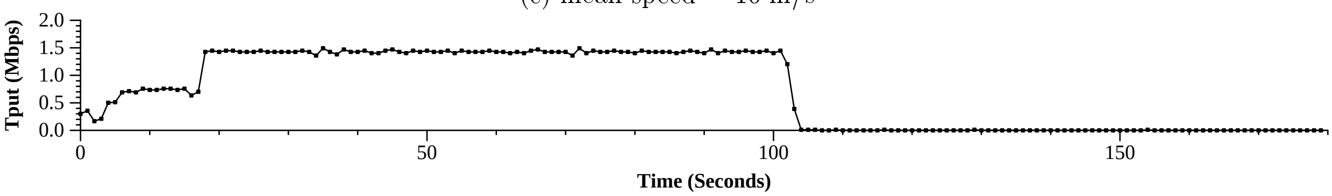
(a) profile of mobility pattern No. 29.



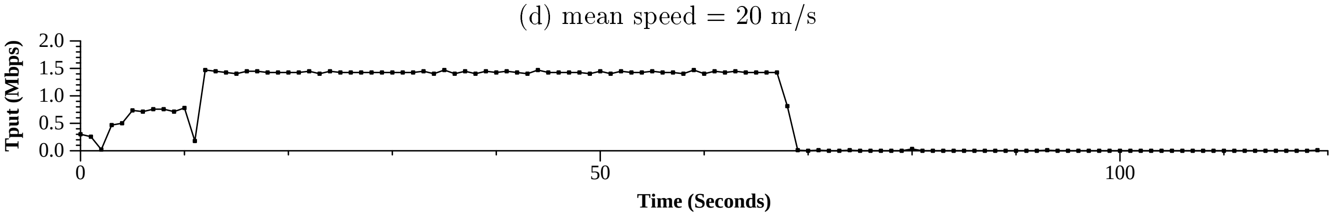
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

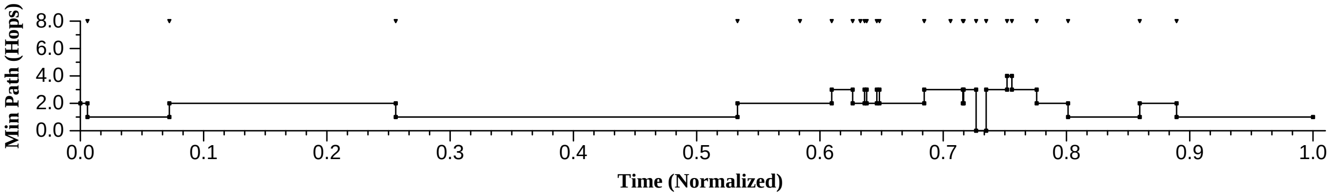


(d) mean speed = 20 m/s

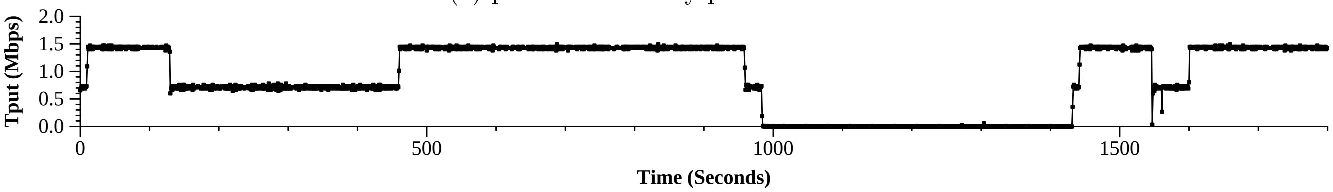


(e) mean speed = 30 m/s

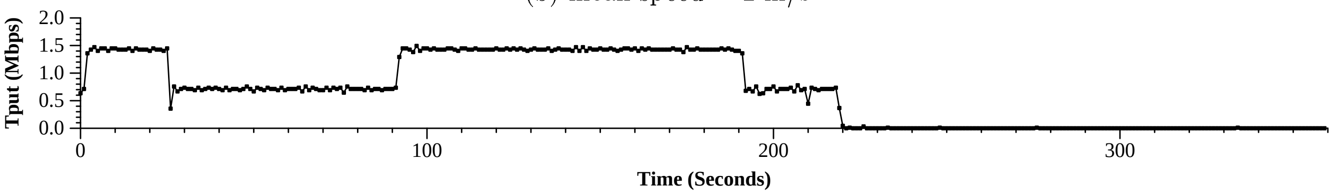
Figure 32: Profile for mobility pattern No. 29, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



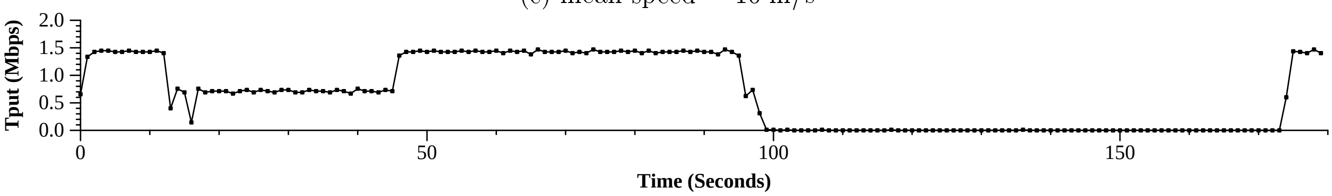
(a) profile of mobility pattern No. 30.



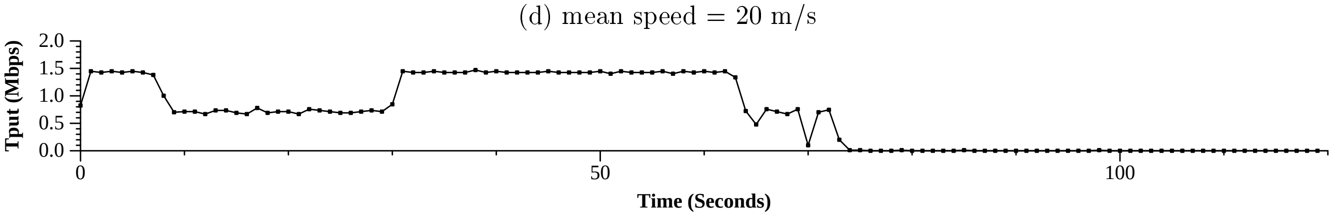
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

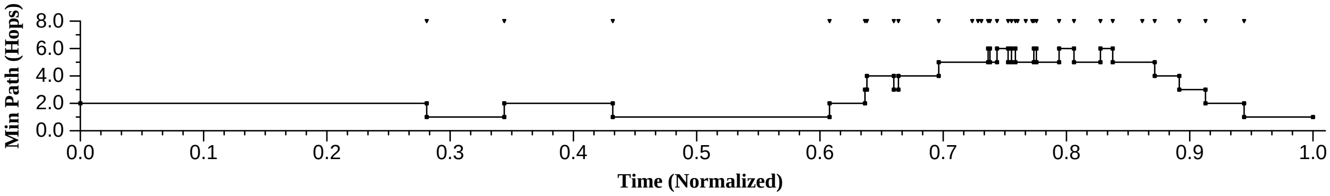


(d) mean speed = 20 m/s

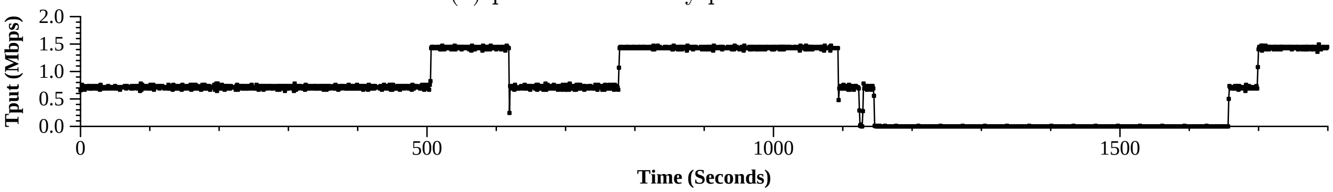


(e) mean speed = 30 m/s

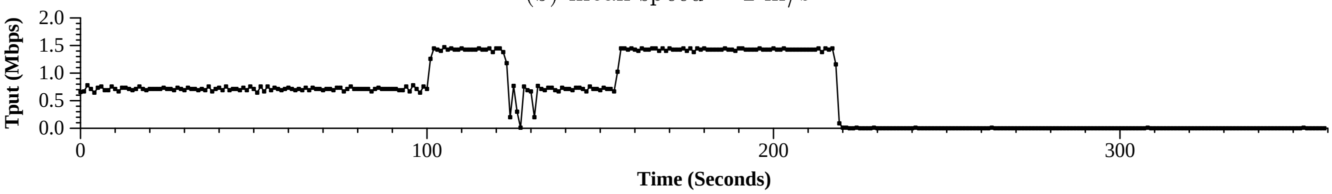
Figure 33: Profile for mobility pattern No. 30, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



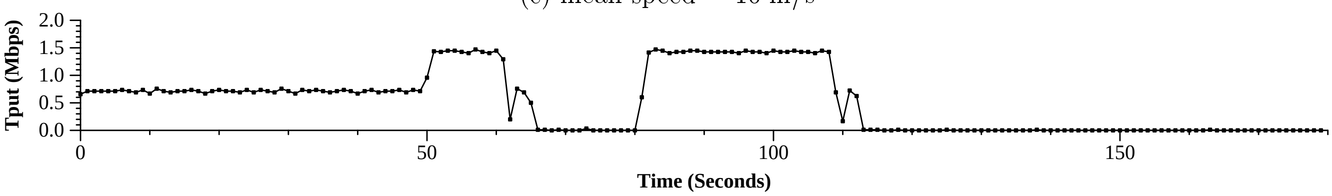
(a) profile of mobility pattern No. 31.



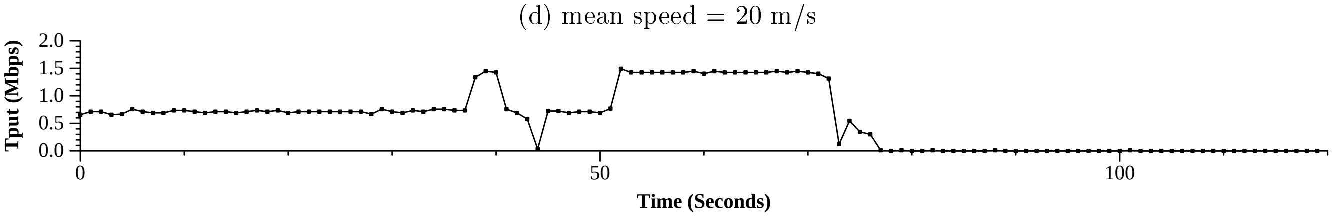
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

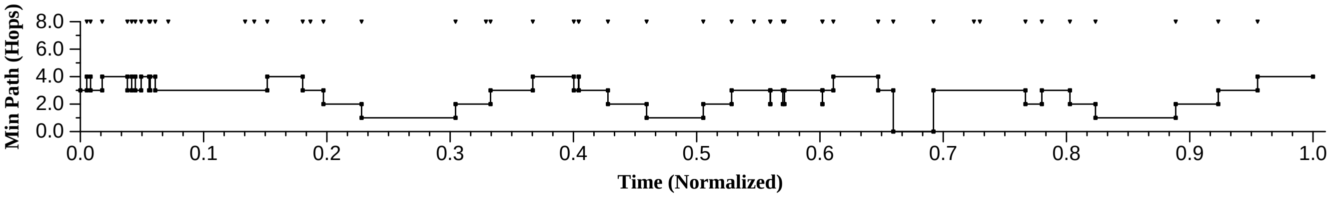


(d) mean speed = 20 m/s

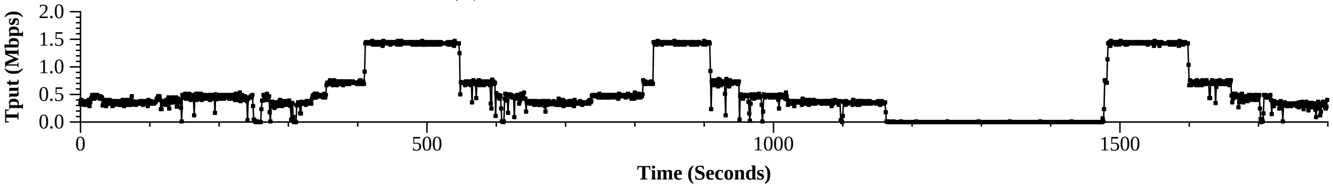


(e) mean speed = 30 m/s

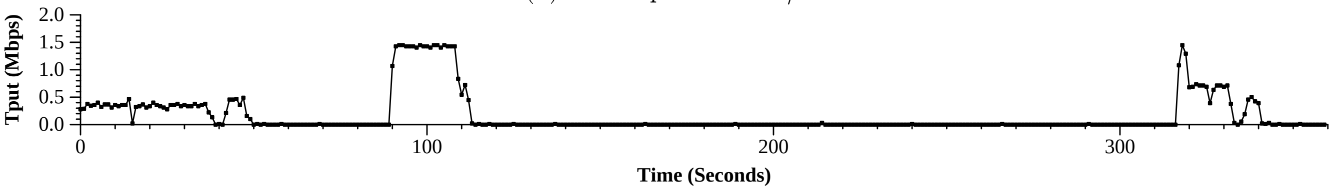
Figure 34: Profile for mobility pattern No. 31, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



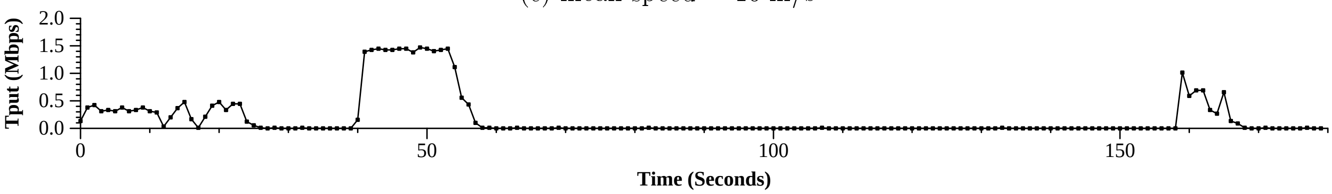
(a) profile of mobility pattern No. 32.



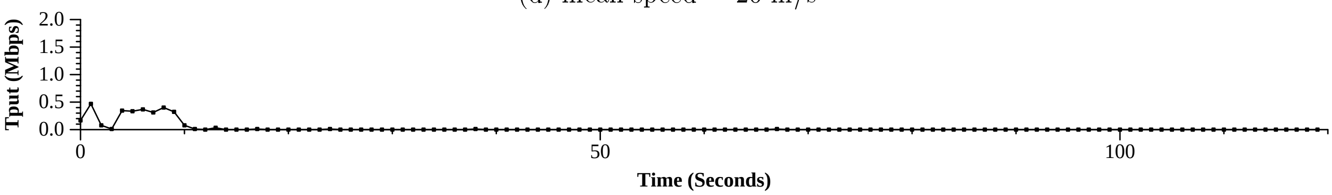
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

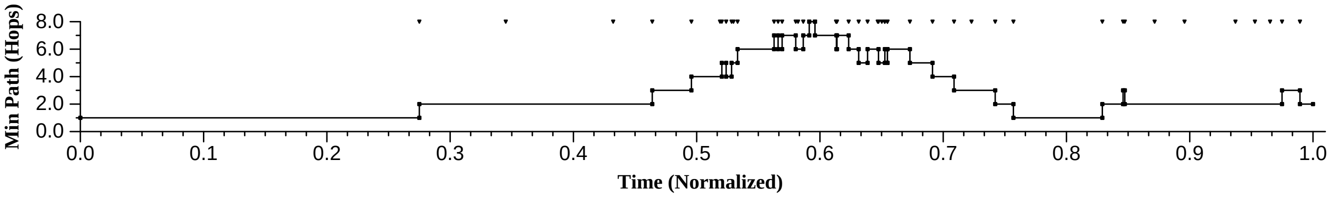


(d) mean speed = 20 m/s

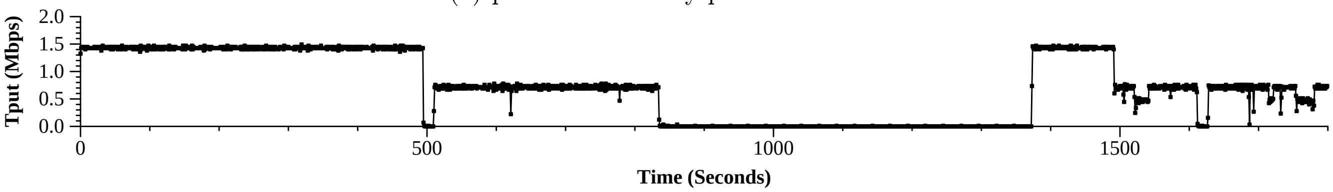


(e) mean speed = 30 m/s

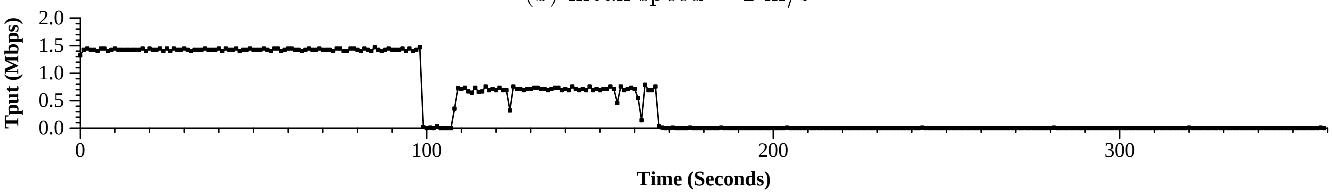
Figure 35: Profile for mobility pattern No. 32, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



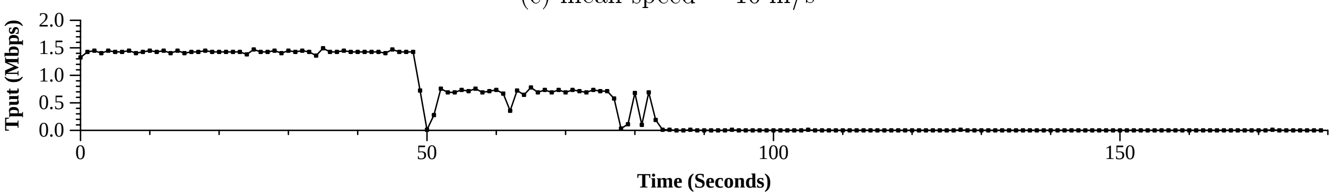
(a) profile of mobility pattern No. 33.



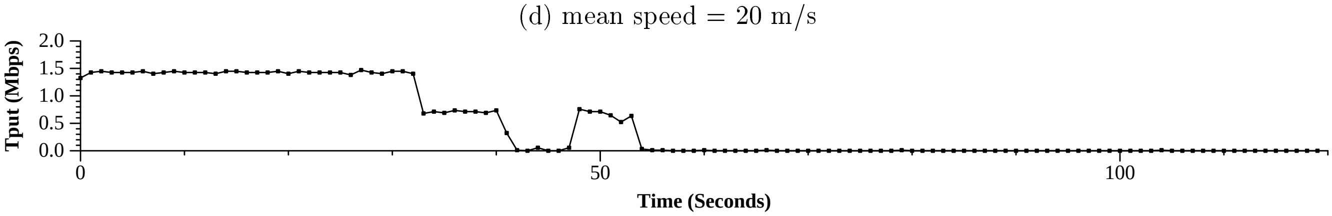
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

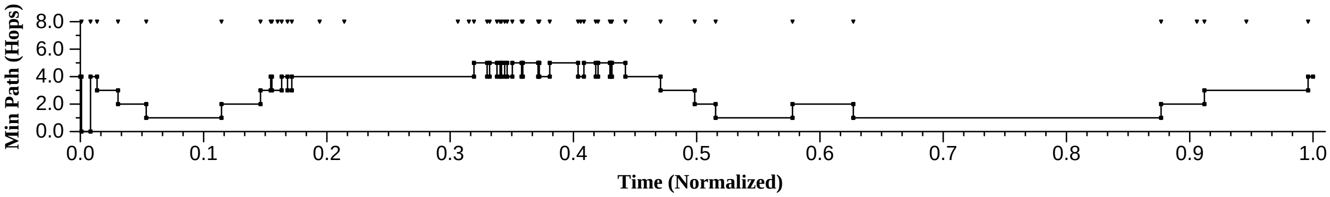


(d) mean speed = 20 m/s

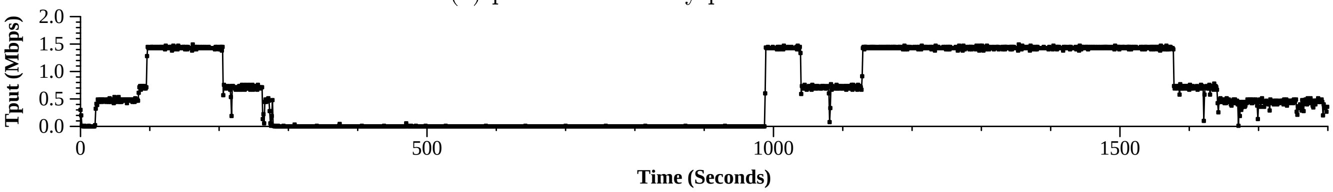


(e) mean speed = 30 m/s

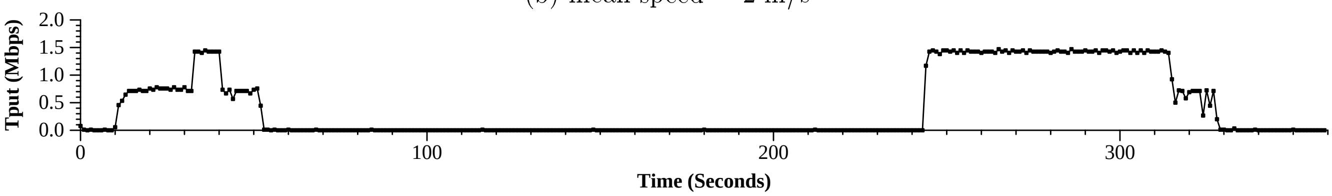
Figure 36: Profile for mobility pattern No. 33, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



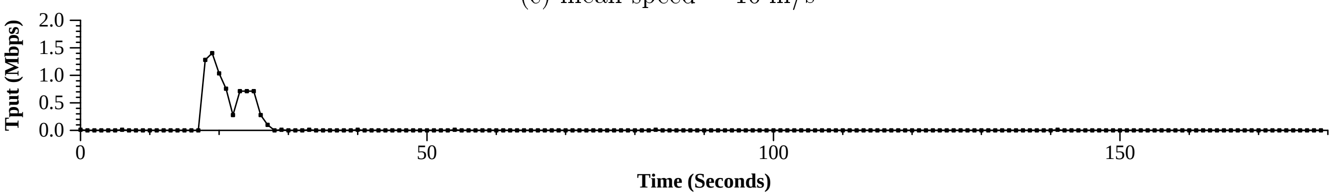
(a) profile of mobility pattern No. 34.



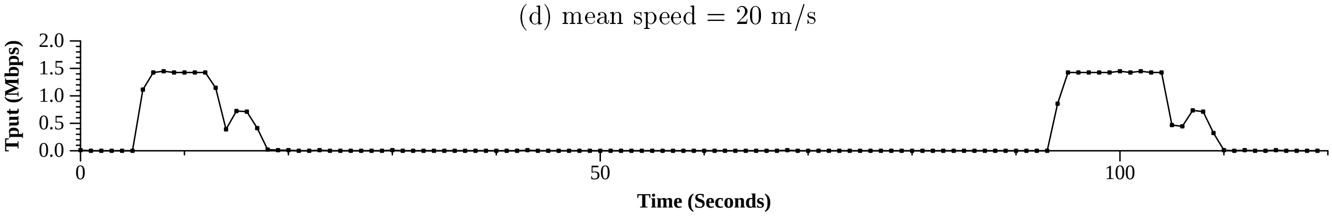
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

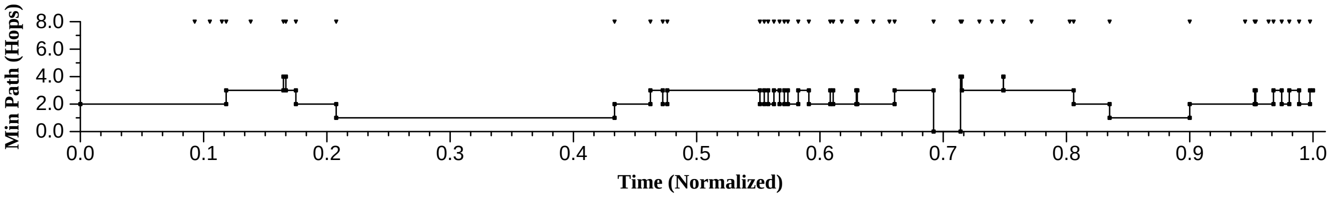


(d) mean speed = 20 m/s

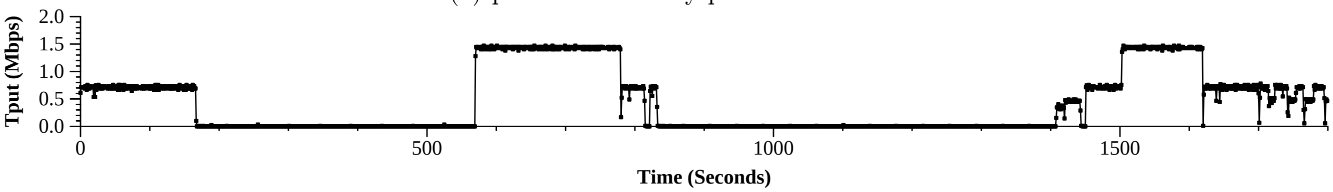


(e) mean speed = 30 m/s

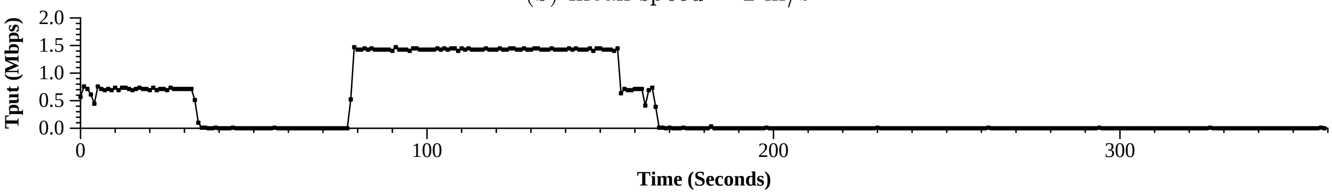
Figure 37: Profile for mobility pattern No. 34, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



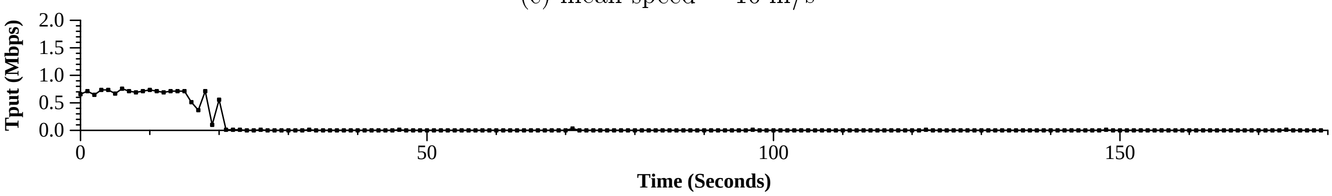
(a) profile of mobility pattern No. 35.



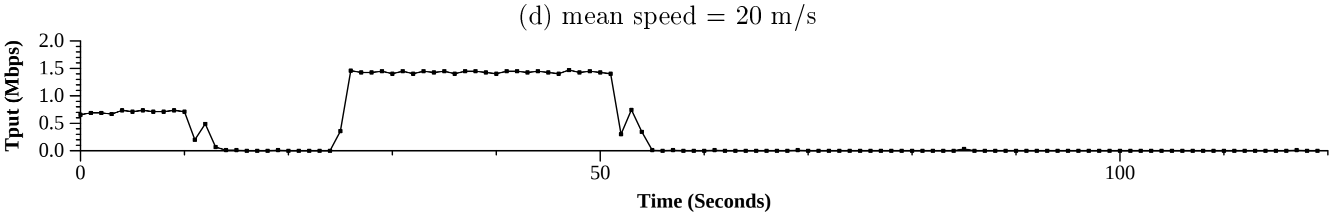
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

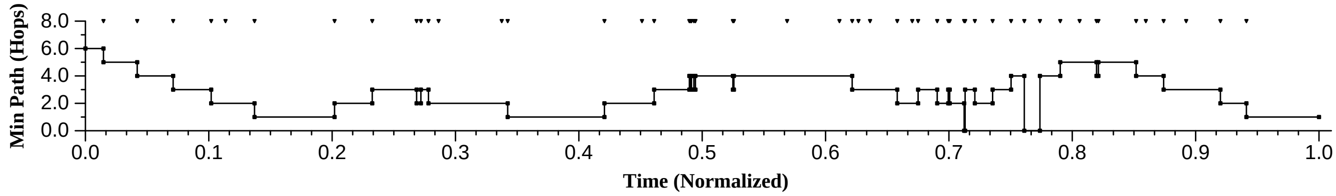


(d) mean speed = 20 m/s

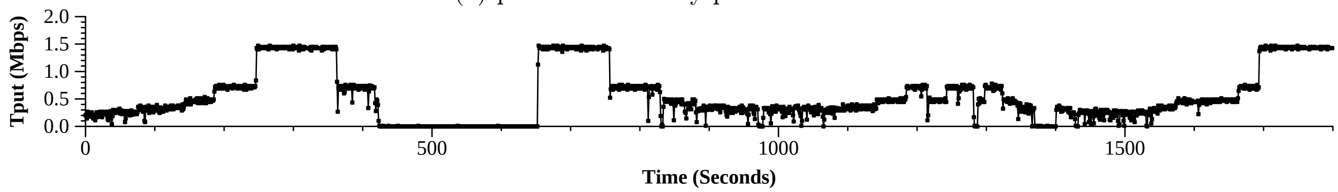


(e) mean speed = 30 m/s

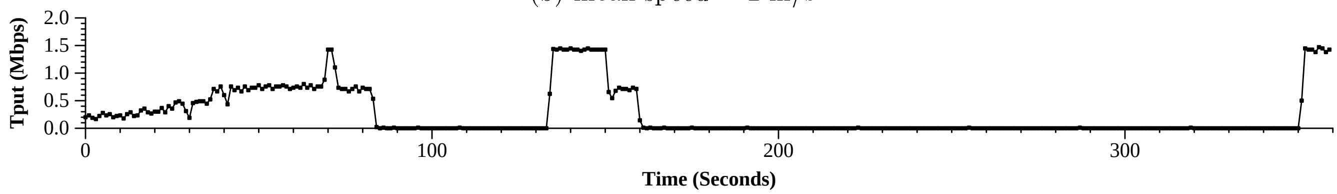
Figure 38: Profile for mobility pattern No. 35, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



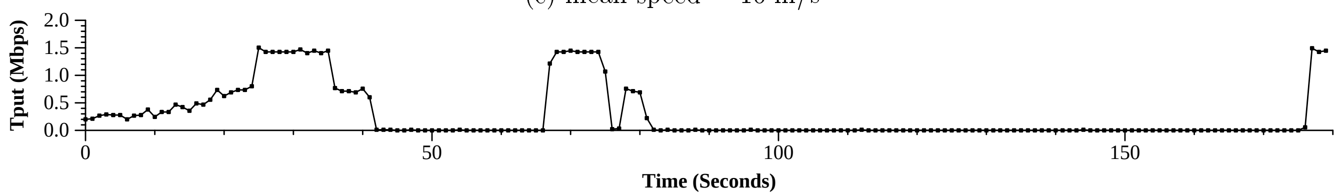
(a) profile of mobility pattern No. 36.



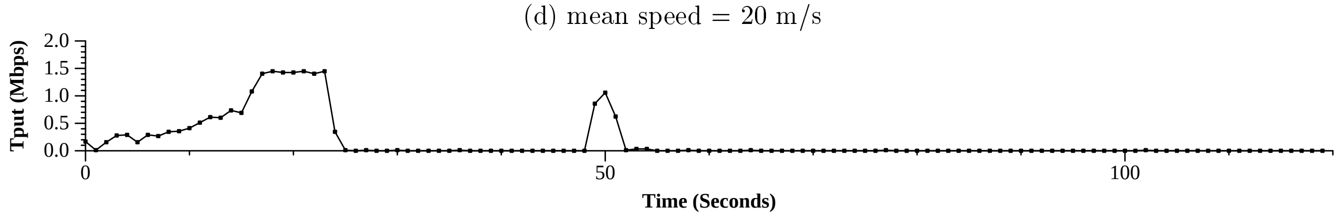
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

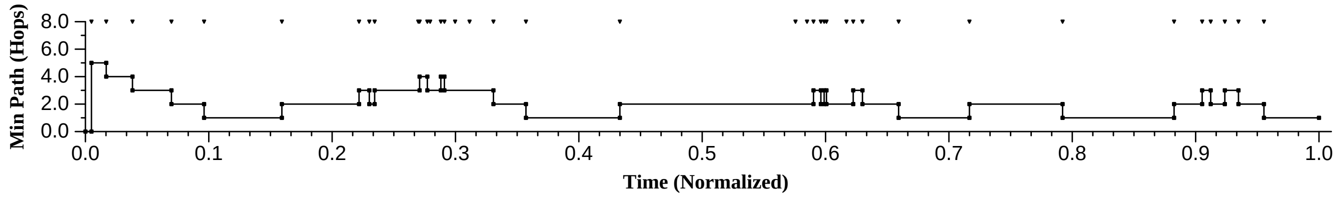


(d) mean speed = 20 m/s

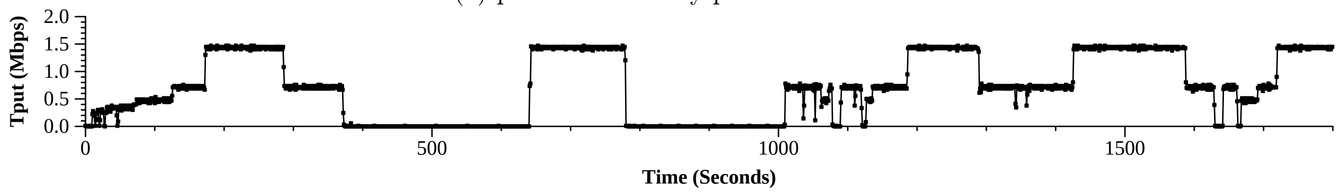


(e) mean speed = 30 m/s

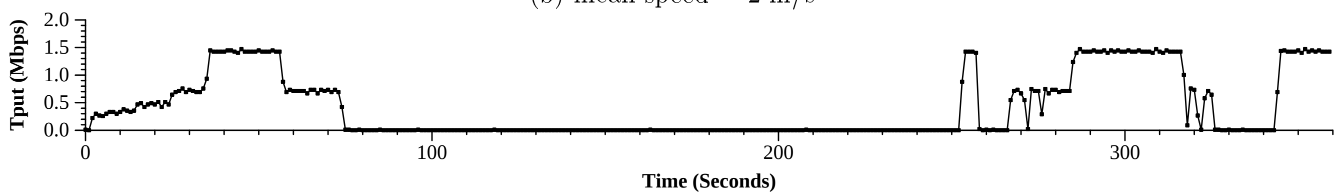
Figure 39: Profile for mobility pattern No. 36, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



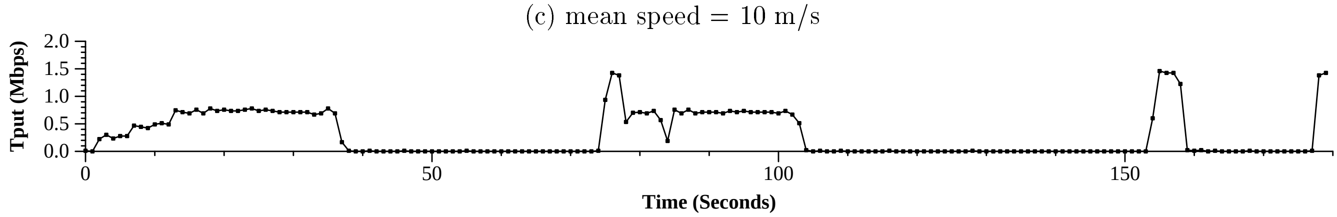
(a) profile of mobility pattern No. 37.



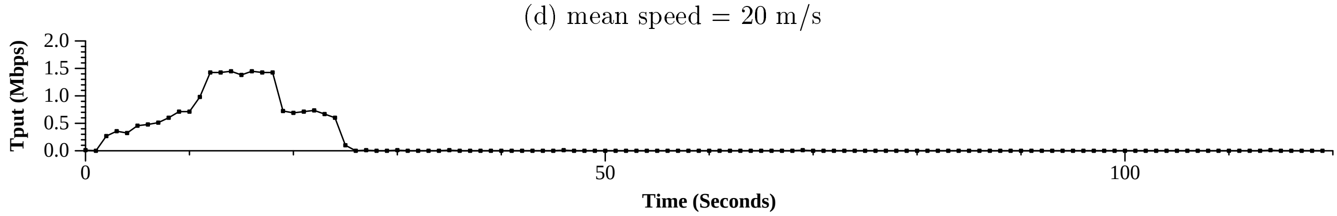
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

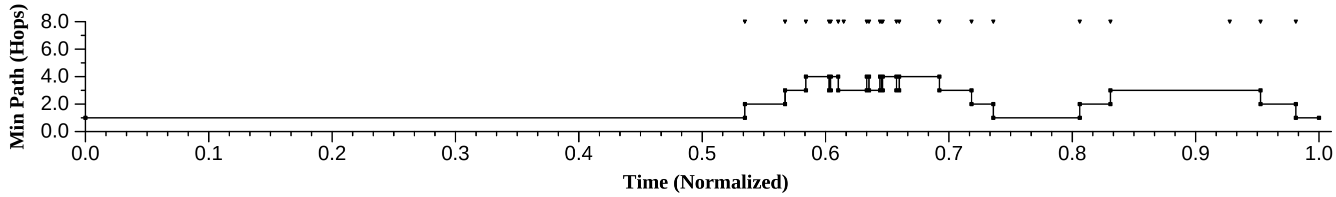


(d) mean speed = 20 m/s

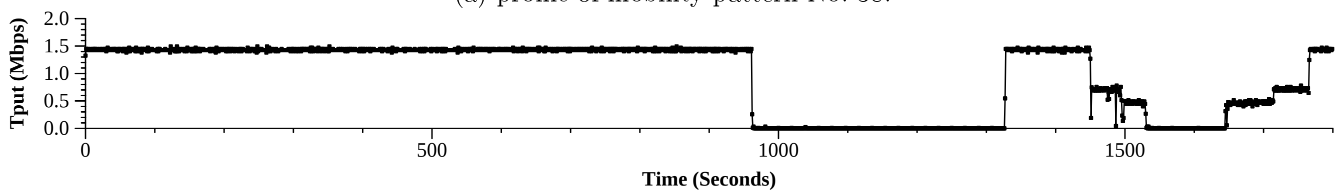


(e) mean speed = 30 m/s

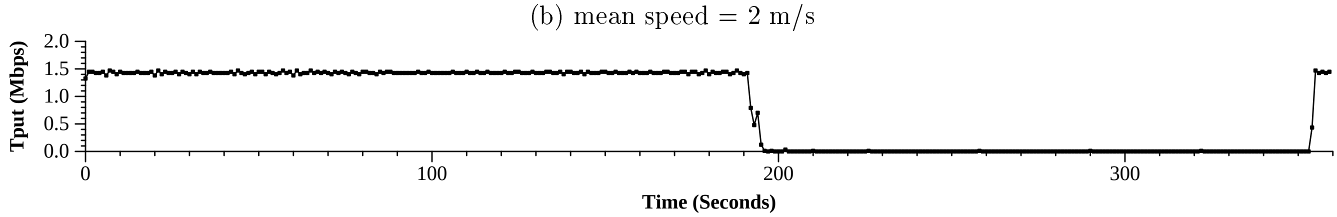
Figure 40: Profile for mobility pattern No. 37, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



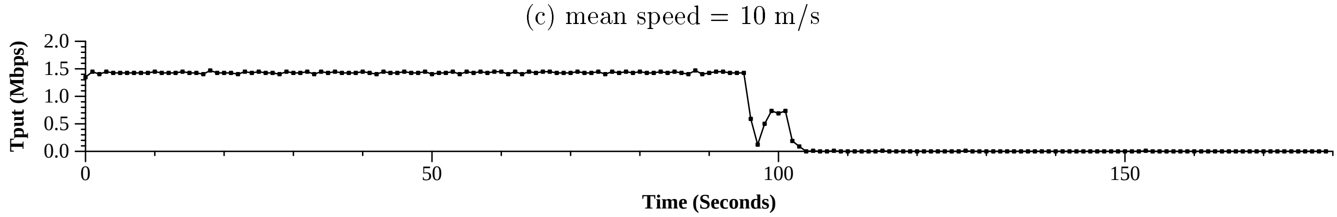
(a) profile of mobility pattern No. 38.



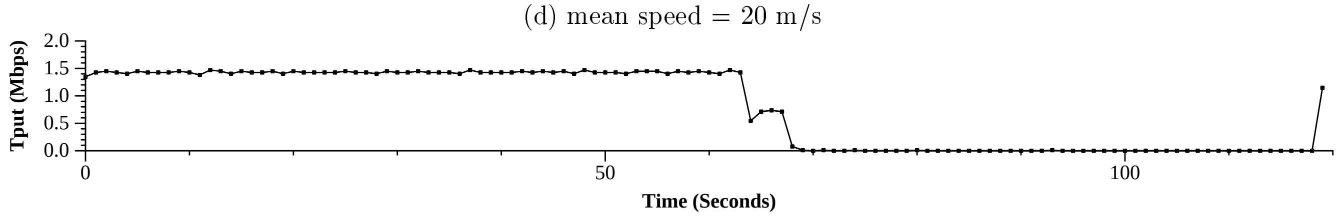
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

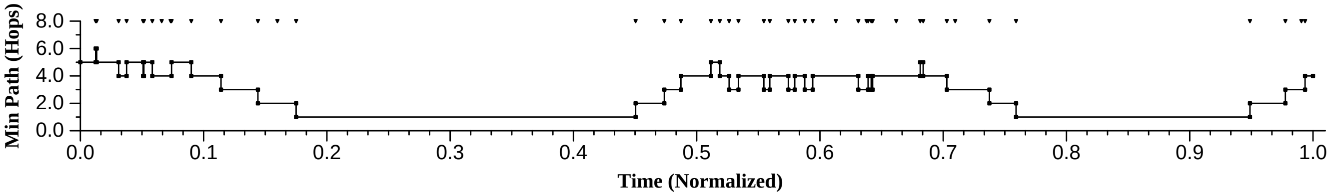


(d) mean speed = 20 m/s

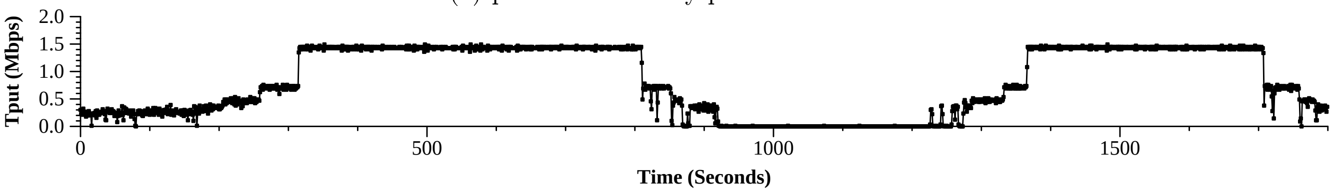


(e) mean speed = 30 m/s

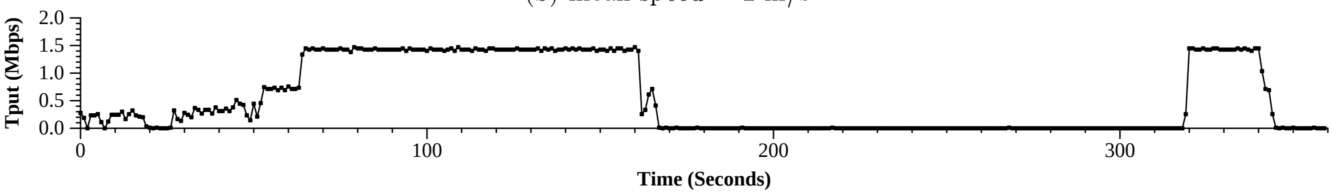
Figure 41: Profile for mobility pattern No. 38, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



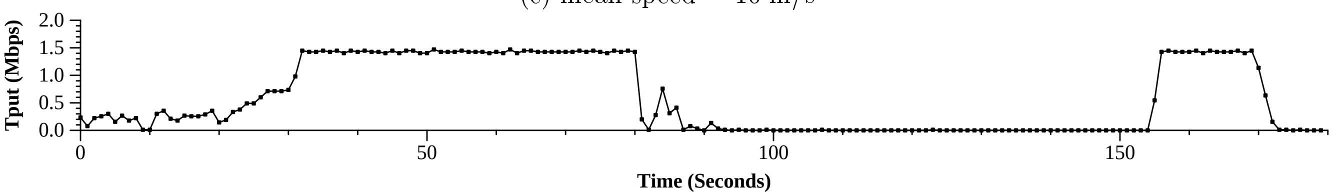
(a) profile of mobility pattern No. 39.



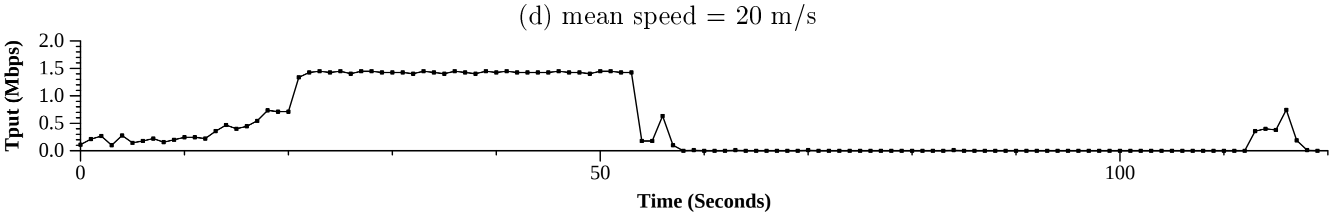
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

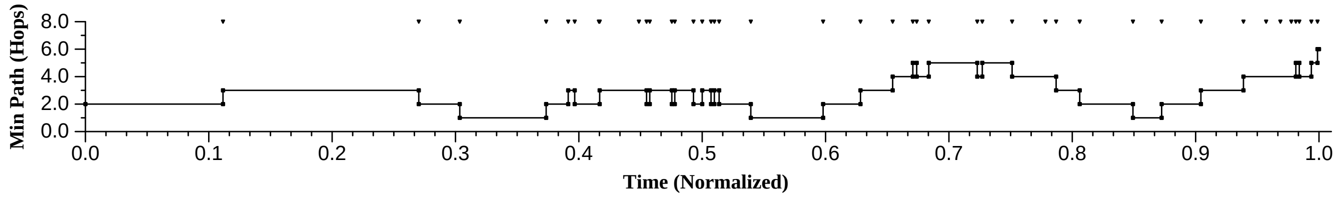


(d) mean speed = 20 m/s

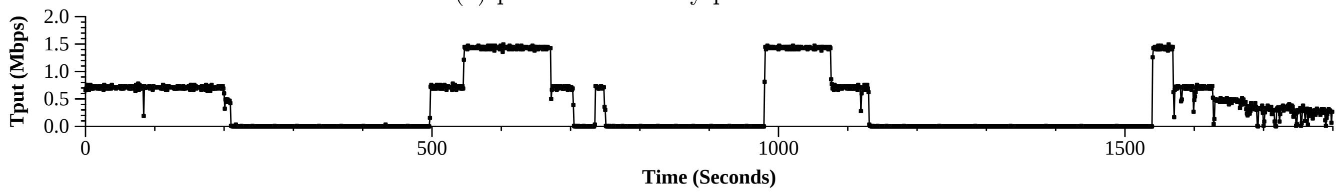


(e) mean speed = 30 m/s

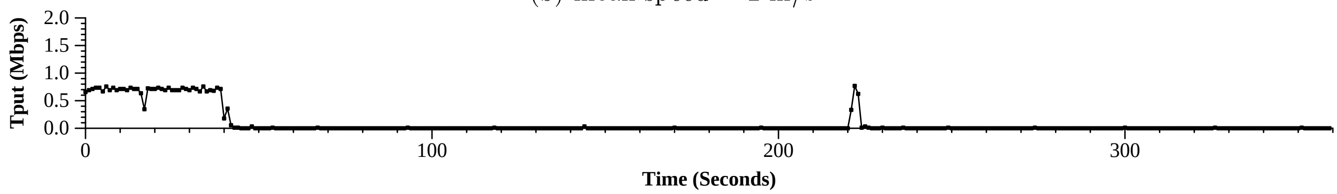
Figure 42: Profile for mobility pattern No. 39, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



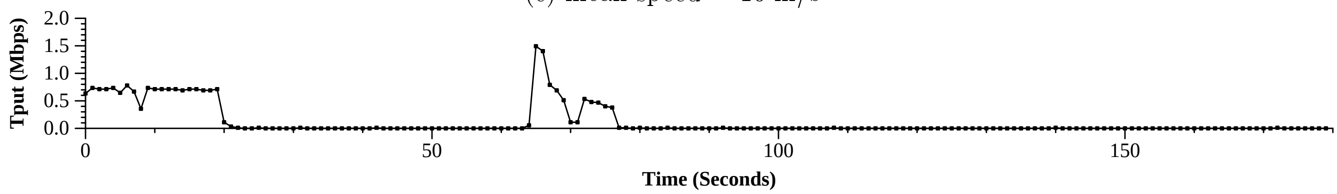
(a) profile of mobility pattern No. 40.



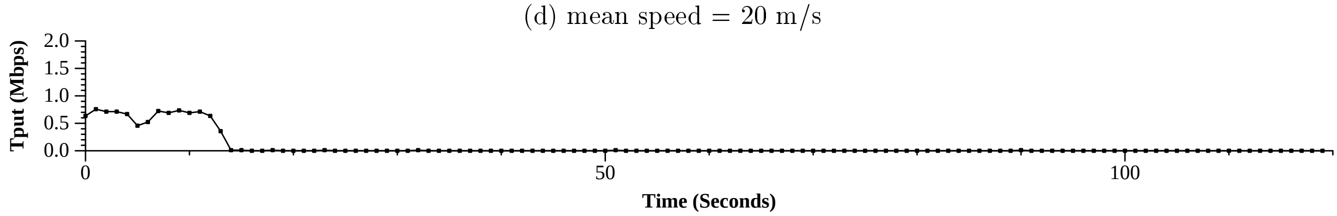
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

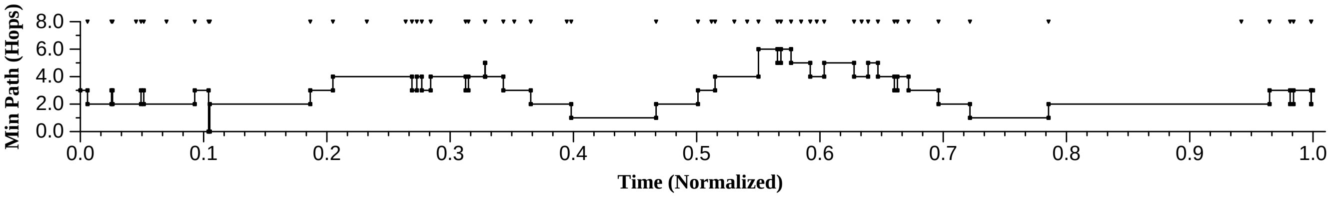


(d) mean speed = 20 m/s

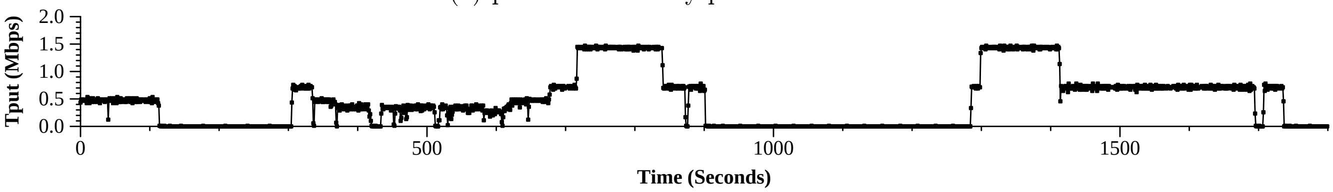


(e) mean speed = 30 m/s

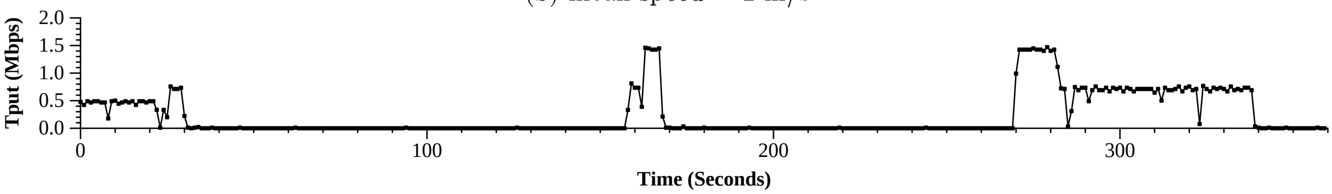
Figure 43: Profile for mobility pattern No. 40, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



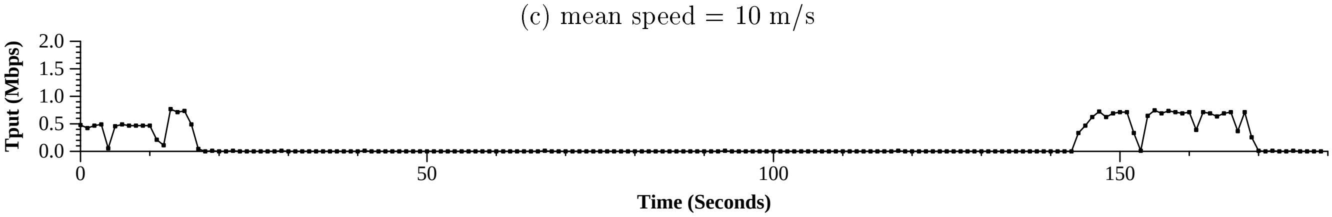
(a) profile of mobility pattern No. 41.



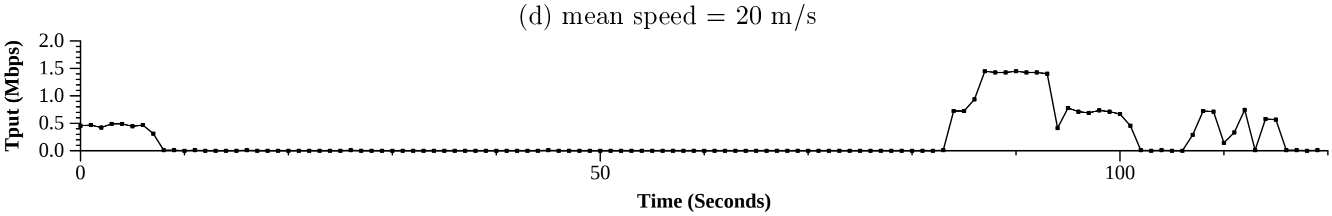
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

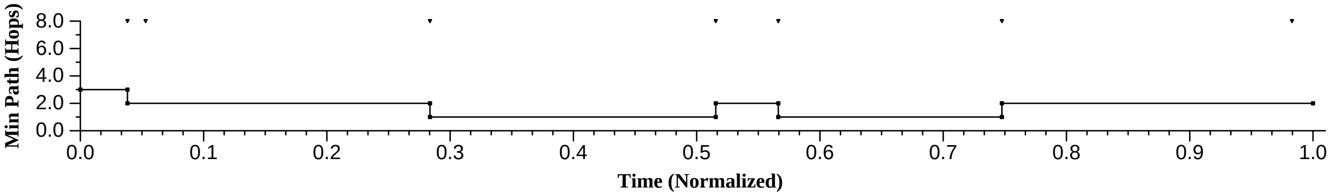


(d) mean speed = 20 m/s

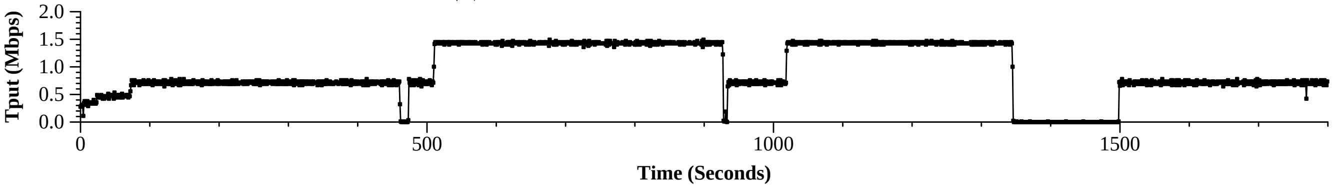


(e) mean speed = 30 m/s

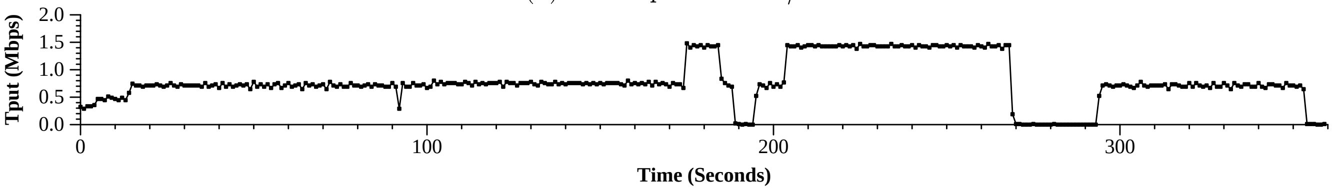
Figure 44: Profile for mobility pattern No. 41, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



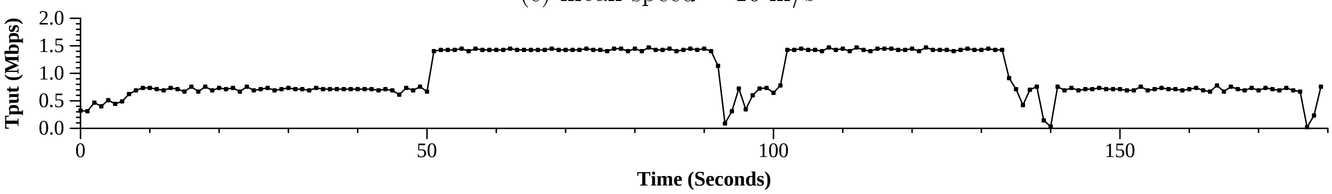
(a) profile of mobility pattern No. 42.



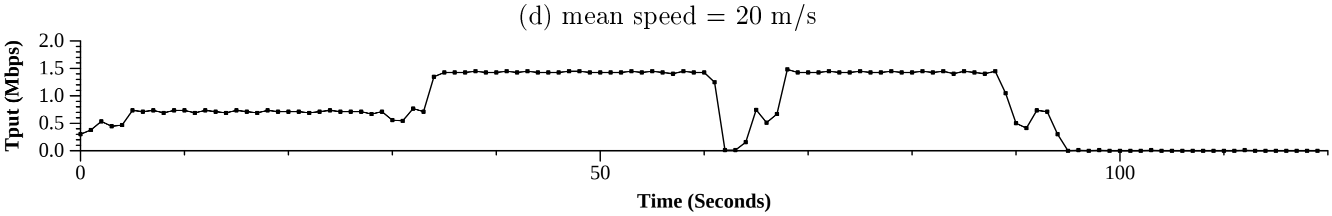
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

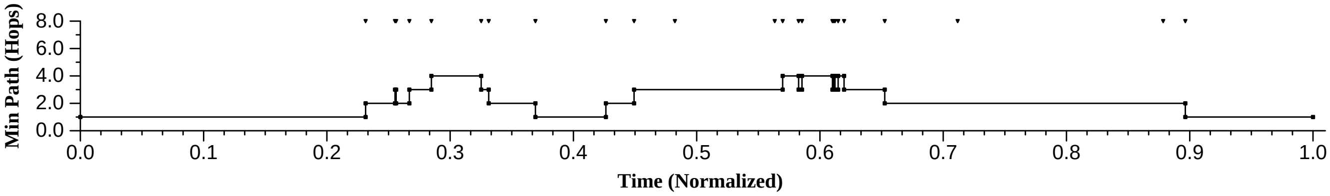


(d) mean speed = 20 m/s



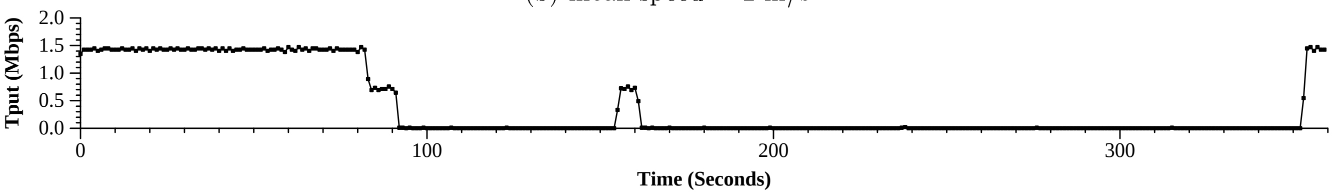
(e) mean speed = 30 m/s

Figure 45: Profile for mobility pattern No. 42, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.

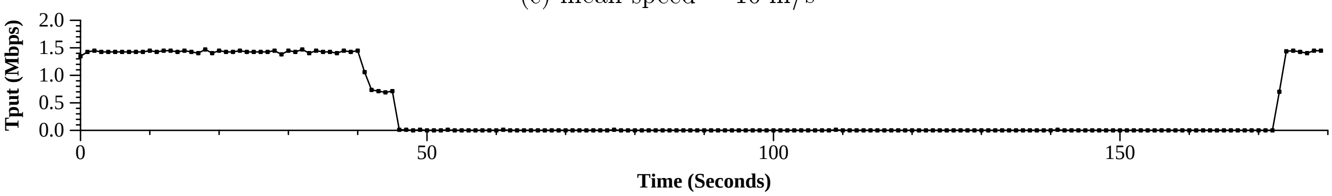


(a) profile of mobility pattern No. 43.

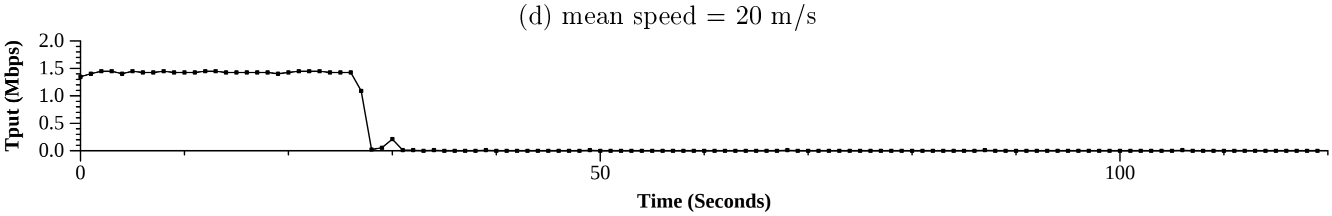
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

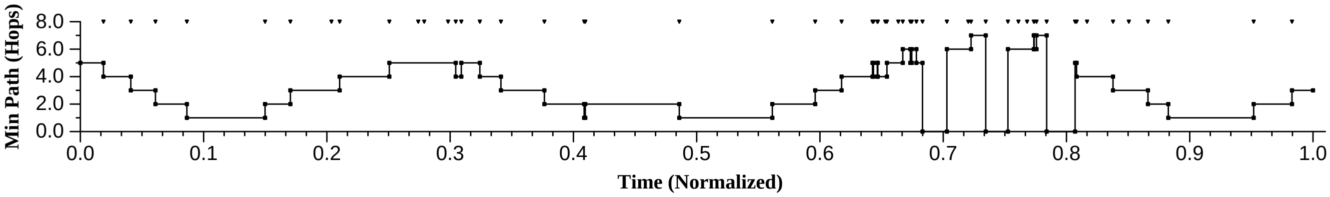


(d) mean speed = 20 m/s

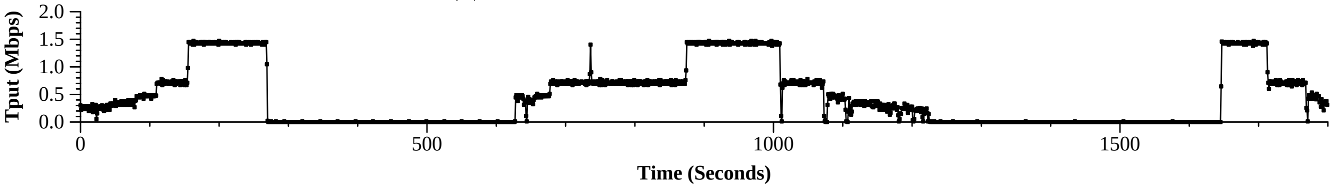


(e) mean speed = 30 m/s

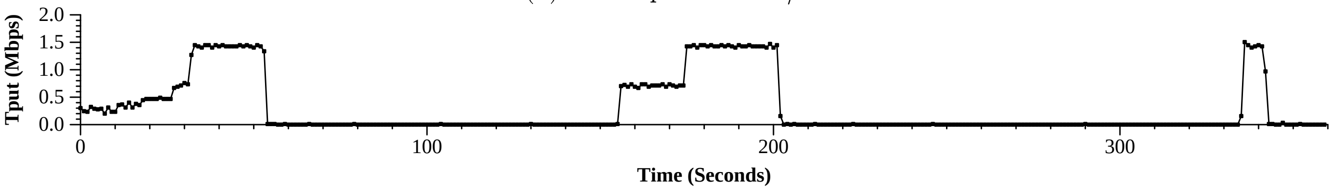
Figure 46: Profile for mobility pattern No. 43, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



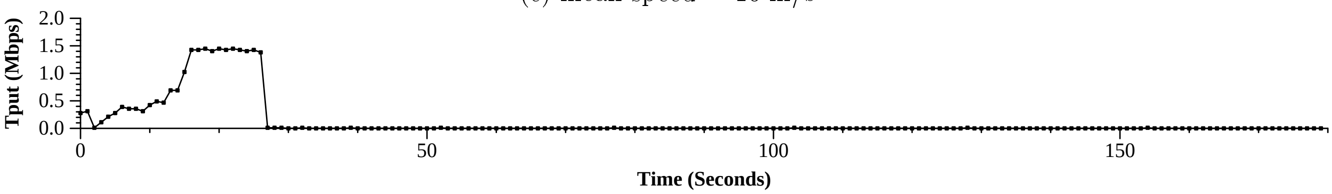
(a) profile of mobility pattern No. 44.



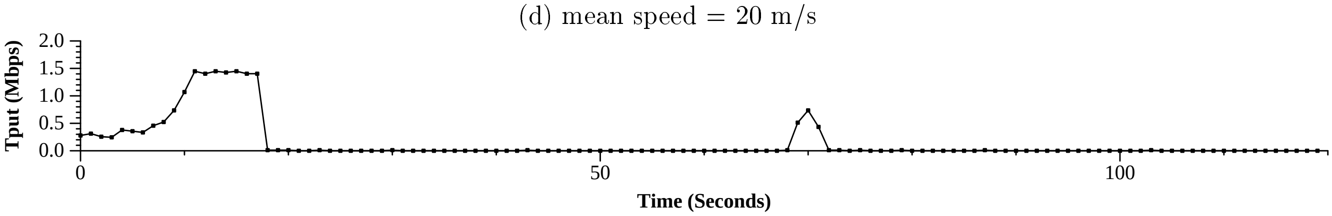
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

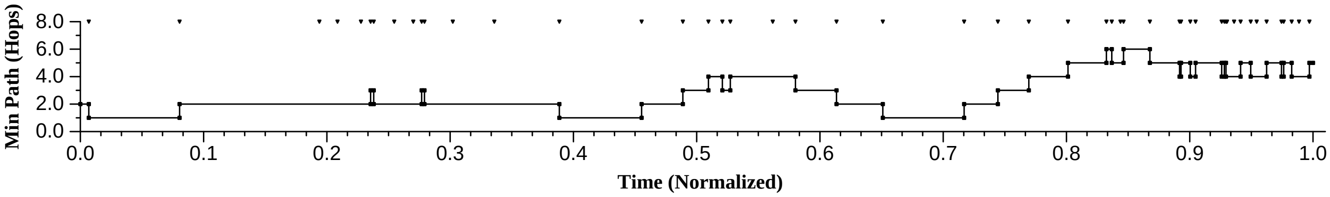


(d) mean speed = 20 m/s

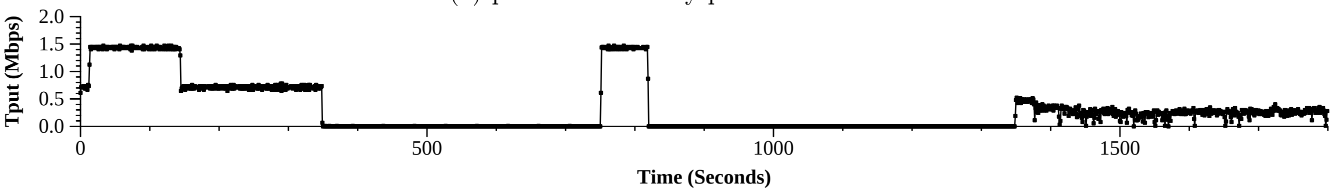


(e) mean speed = 30 m/s

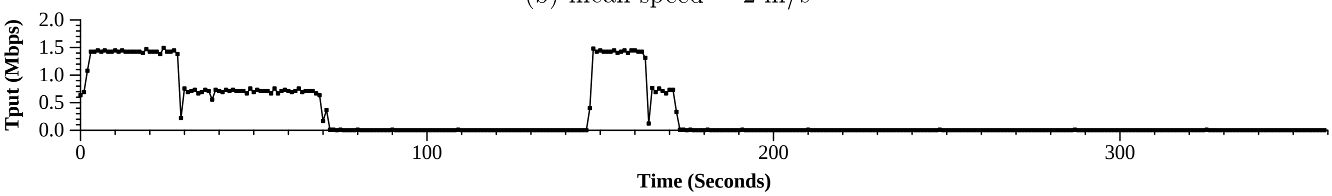
Figure 47: Profile for mobility pattern No. 44, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



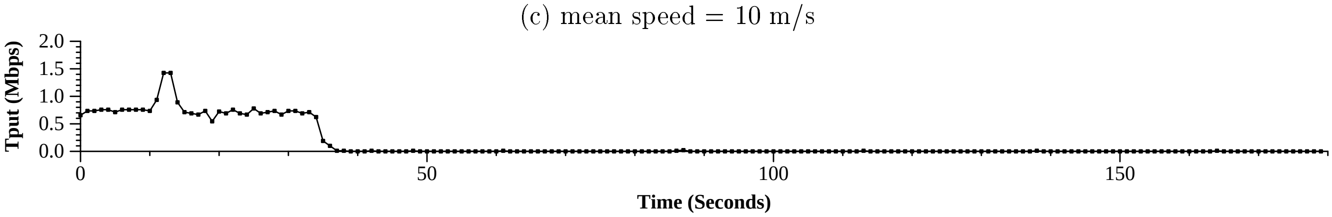
(a) profile of mobility pattern No. 45.



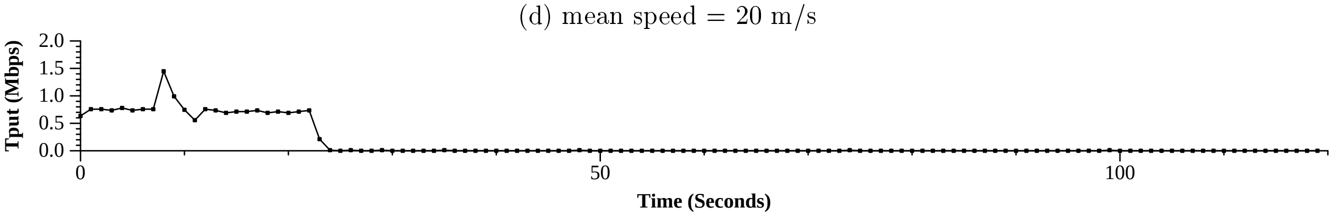
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

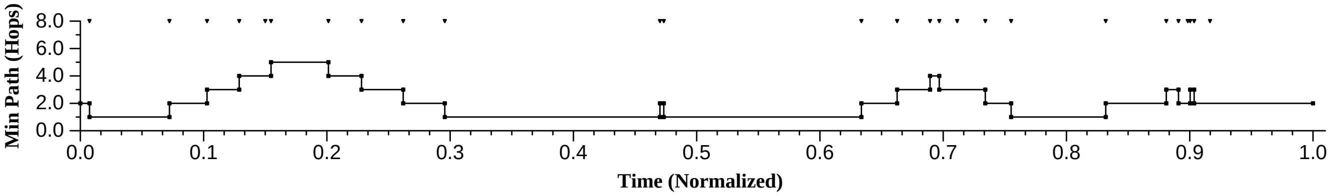


(d) mean speed = 20 m/s

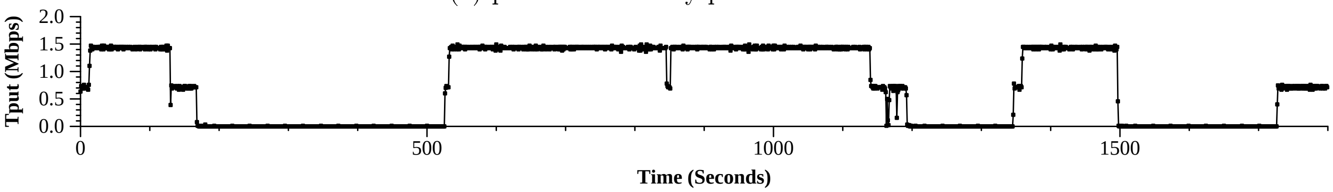


(e) mean speed = 30 m/s

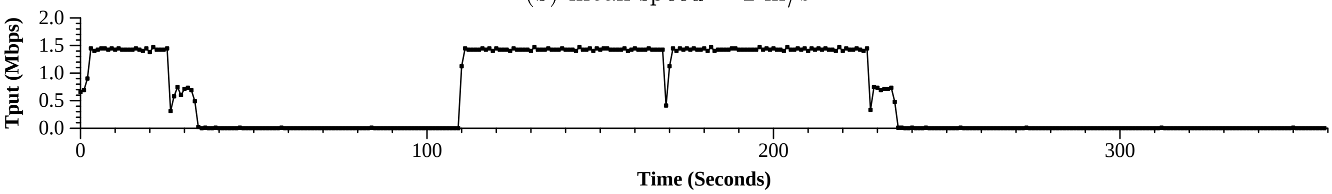
Figure 48: Profile for mobility pattern No. 45, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



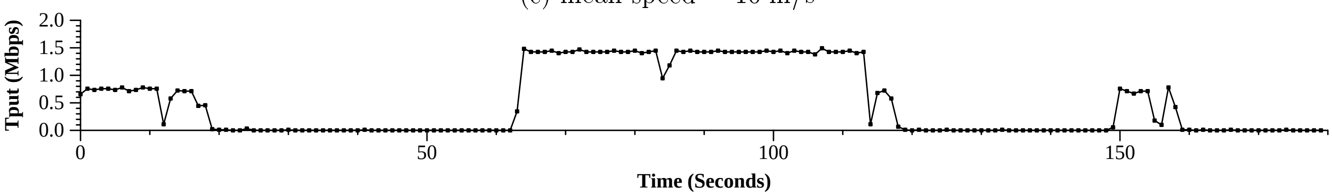
(a) profile of mobility pattern No. 46.



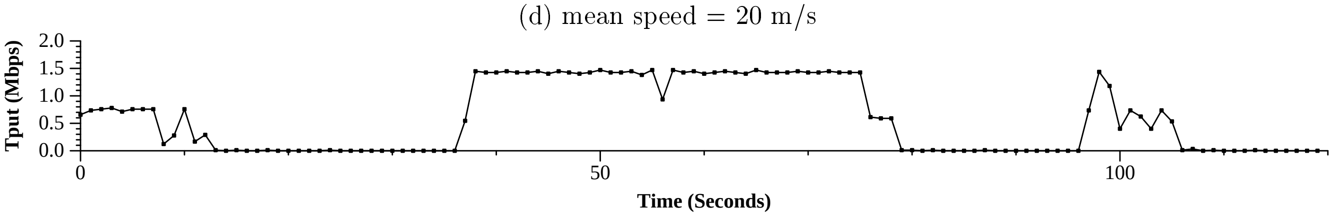
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

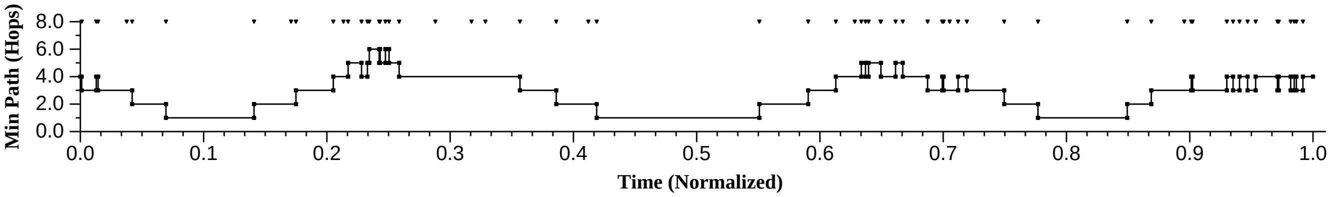


(d) mean speed = 20 m/s

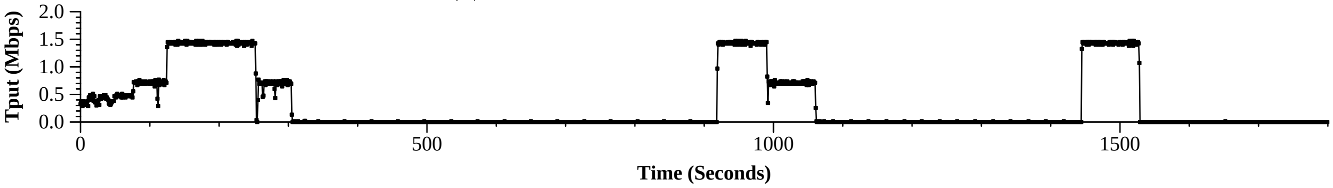


(e) mean speed = 30 m/s

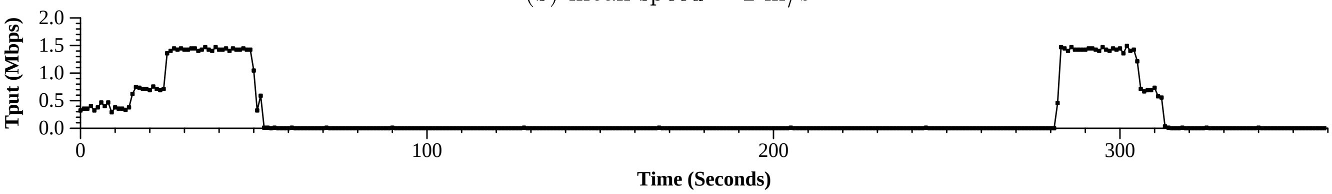
Figure 49: Profile for mobility pattern No. 46, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



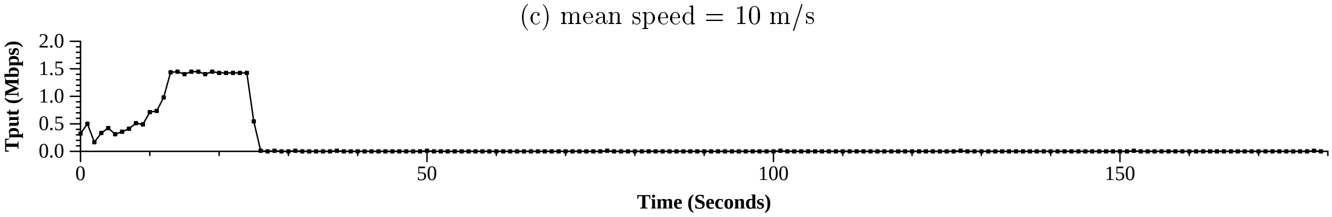
(a) profile of mobility pattern No. 47.



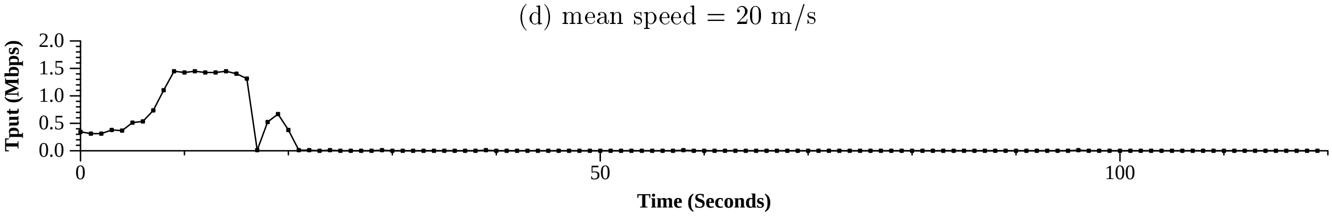
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

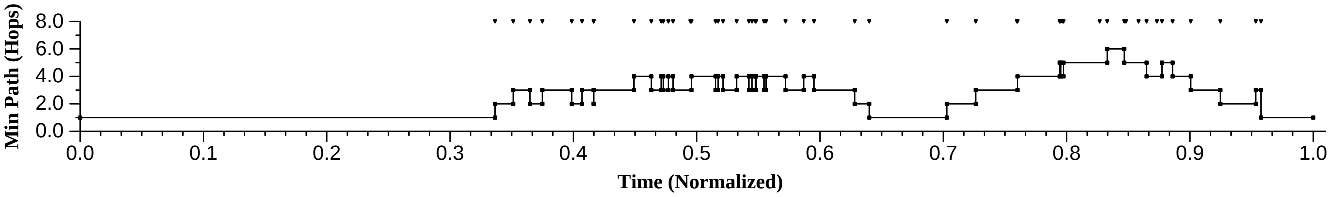


(d) mean speed = 20 m/s

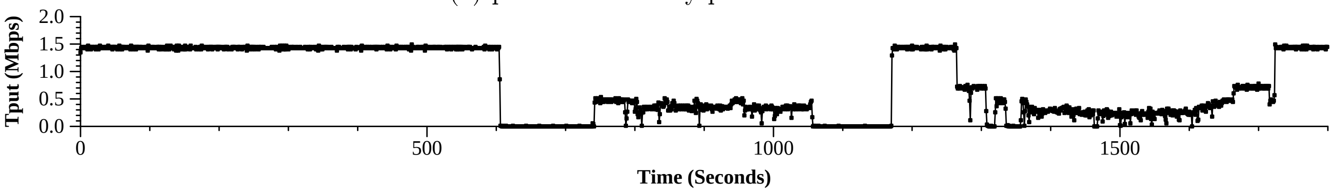


(e) mean speed = 30 m/s

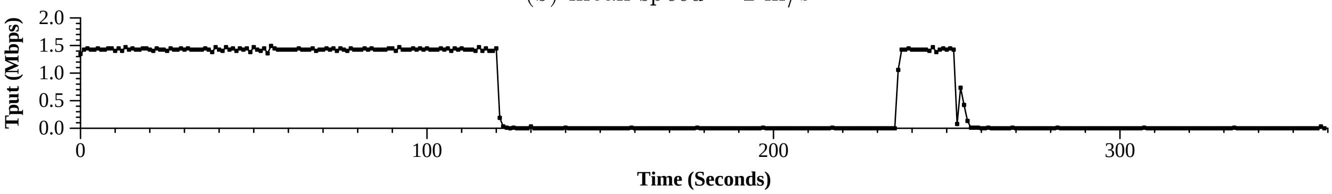
Figure 50: Profile for mobility pattern No. 47, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



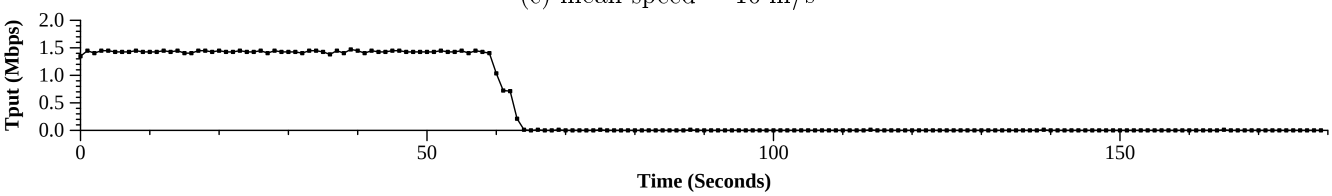
(a) profile of mobility pattern No. 48.



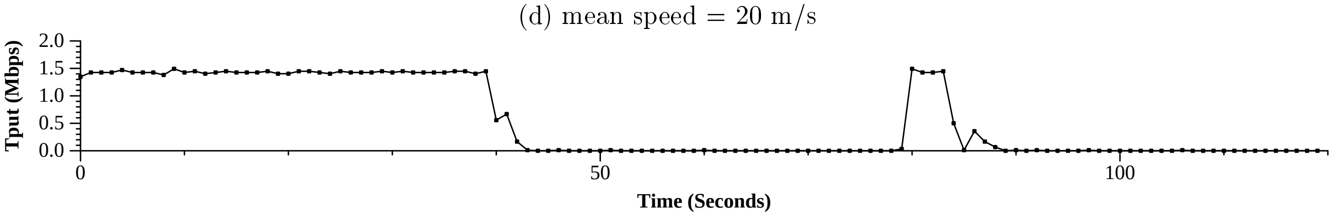
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

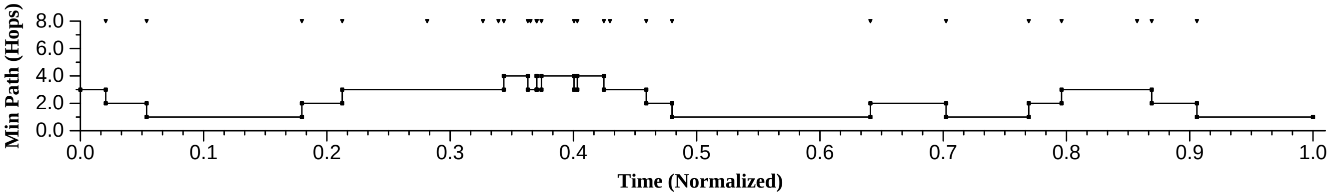


(d) mean speed = 20 m/s

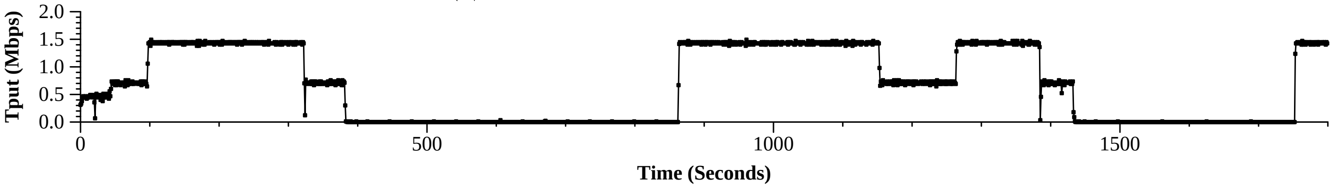


(e) mean speed = 30 m/s

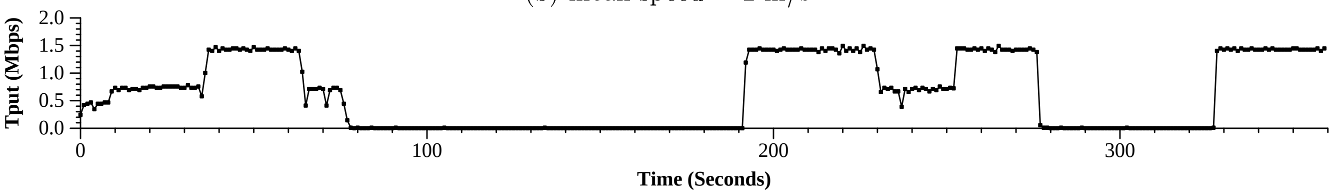
Figure 51: Profile for mobility pattern No. 48, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.



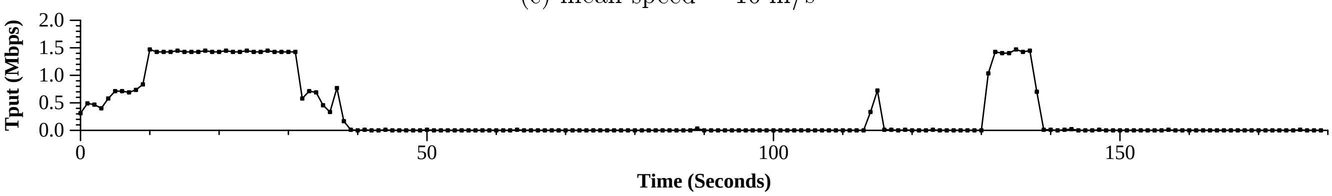
(a) profile of mobility pattern No. 49.



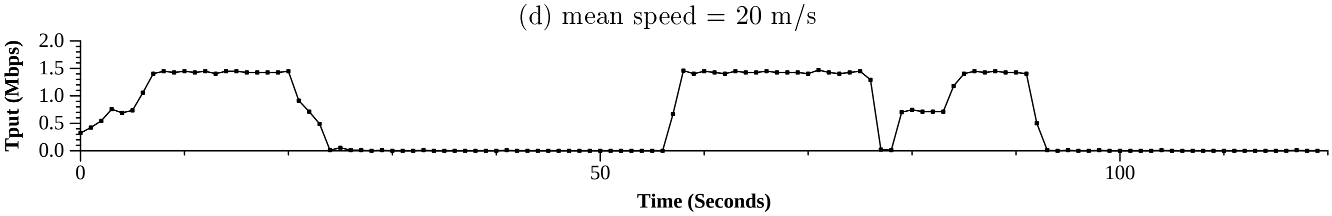
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s

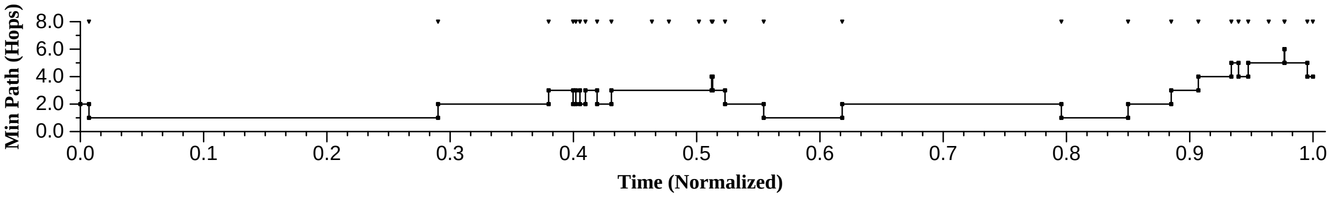


(d) mean speed = 20 m/s

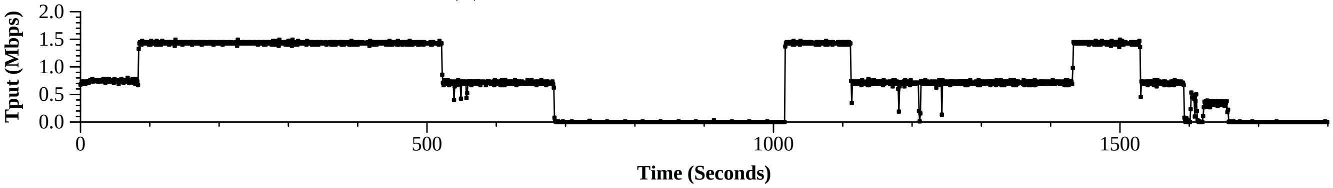


(e) mean speed = 30 m/s

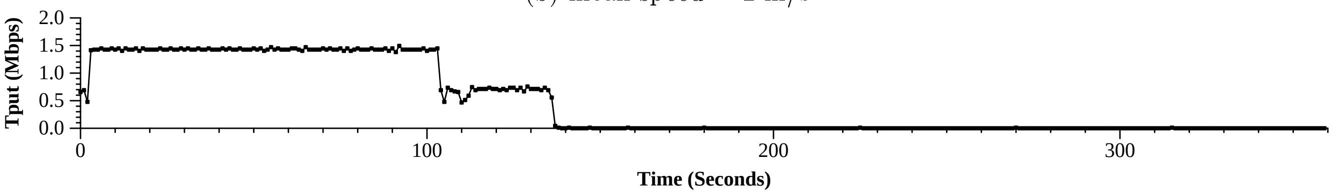
Figure 52: Profile for mobility pattern No. 49, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b)-(e)$ show the observed throughput for the connection, averaged over 1 second intervals.



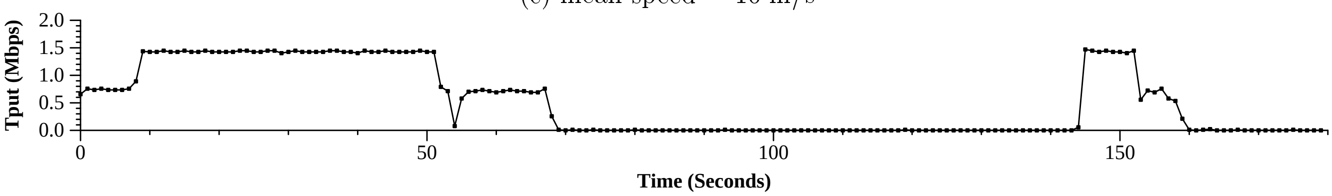
(a) profile of mobility pattern No. 50.



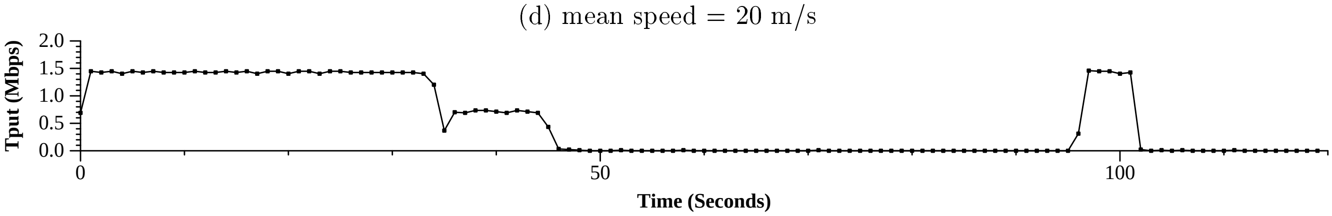
(b) mean speed = 2 m/s



(c) mean speed = 10 m/s



(d) mean speed = 20 m/s



(e) mean speed = 30 m/s

Figure 53: Profile for mobility pattern No. 50, and corresponding TCP-Reno performance. The ticks at the top of (a) denote changes on the minimum length path between the TCP sender and receiver. The curves in $(b) - (e)$ show the observed throughput for the connection, averaged over 1 second intervals.