Journal of Reliable Intelligent Environments (2018) 4:123–129
https://doi.org/10.1007/s40860-018-0064-3

ORIGINAL ARTICLE

Is the internet spatial?

Hedda R. Schmidtke[1]

Received: 29 December 2017 / Accepted: 31 May 2018 / Published online: 15 June 2018

Abstract
One of the internet's design principles is to make physical distance transparent, which has made it a motor of globalization and economic development. However, the physical world within which it operates is spatial, and reduction of actual spatial distance is a factor of high impact leveraged, e.g., in today's commercially successful content distribution networks as well as new technologies, such as edge computing. Predictable network behavior is key for reliable intelligent environments. It is therefore worthwhile to study whether and how spatial the internet actually is. We studied a small set of globally dispersed public universities on six continents, recording delays reported by the ping tool to a server in the Pacific Northwest area of the U.S. at four different times of the day (morning, noon, evening, midnight) and over both weekends and weekdays, thus retrieving more than 17,210 data points. We then determined the Pearson's correlations between distance and delay and between hop count and delay. The experiment yielded that there is an overall correlation of 0.57 (a border case strong correlation) between distance and delay and a weaker correlation of 0.4 (medium strength correlation) between hop count and delay. The results suggest that there is a physical distance effect and that this effect is stronger than the hop count effect.

Keywords Spatial networks · Internet · Round trip time

1 Introduction

When someone chooses a mirror from which to download a software they usually pick the ones nearest to them, a strategy derived from human experience in a physical world. In the physical world, longer distances mean more effort and we drive to the shop nearby rather than the one further away if both offer the same interface and products. From a network perspective, we may suggest to pick one such that traffic is with a high probability smaller, however, far away. In computer networks distance seems to disappear shrinking the world into a "global village." In today's internet, however, mirrors are increasingly important as the growing sector for content delivery networks (CDNs) suggests. Content owners, such as streaming services, send their content to CDN operators to host and distribute it to a worldwide network of servers close to the locations of customers and clients. Hofmann and Beaumont [9] define:

> The term content network refers to a communication network that deploys infrastructure components [that] interconnect with each other, creating a virtual network layered on top of an existing packet network infrastructure. [p. 20]

Search engines, map providers, streaming services, and social networks, for instance, benefit from employing content networks.

One of the advantages of CDNs, is that they provide distribution in proximity to the end-users. CDNs thus reduce global traffic, but also, one might think, that they provide an advantage by delivering information over shorter distances. This is worthwhile noticing because we think of computer networks rather as non-spatial graphs, potentially depending on hop count, but not on spatial distance. Moreover, hop count may be greatly divergent within areas, i.e., is generally not spatial in the sense of physical geographic space, but only in an abstract topological sense. Internet protocols, for instance, do not make use of a notion of physical distance and leverage a temporal rather than spatial measure: delays or roundtrip time (RTT). Electric signals travel at the speed of light, i.e., 300 km/ms. Given the Earth's circumference of about 40,000 km, a signal traveling at this speed could circle the whole planet within 134 ms. Distance thus should not play any significant role.

✉ Hedda R. Schmidtke
schmidtke@acm.org

[1] University of Oregon, Eugene, OR 97403-1251, USA

The purpose of this paper was to investigate the question of whether there is any spatial factor to internet delays, measured with RTT, besides hop count. We found this question of interest as everyday experience and the success of CDNs may suggest that there is a spatial factor but the current internet specification and network theory suggest that this should be an at most spurious effect. Our study set out to investigate this by measuring RTT from a terminal in the US to several public universities in the world and checking for the correlation between RTT and spatial distance. While we look at the literature with respect to potential explanations and assessment of the generality of the problem with respect to today's and tomorrow's networks, the focus of this article is on reporting the experimental findings.

Structure of the article We start with a short introduction of TCP (Transmission Control Protocol) and a discussion of literature mentioning spatial properties in TCP (Sect. 2). We then describe the experiment and its results and discuss the outcome (Sect. 3). We outline conclusions and potential for future work in Sect. 4.

2 Related works

To our knowledge, there are no studies on the spatial properties of the internet per se. However, roundtrip time (RTT) has been studied in the context of transport protocol issues and spatial properties are mentioned in some of these studies. We briefly introduce how TCP operates in Sect. 2.1. Then we look at the packet reordering problem, a problem that has been reported to particularly affect long-distance connections, and solution proposals (Sect. 2.2). Finally, we move beyond the TCP/IP suite to proposals for more atypical distances (Sect. 2.3).

2.1 TCP/IP over long distances

The transport layer of the TCP/IP model is built upon the internet layer (IP). While IP connects individual hosts and routers physically linked to each other and is responsible for routing packets, TCP is responsible for providing reliable end-to-end connections upon which internet application services such as HTTP but also, for instance, transaction-based services are built, such as a travel booking tool, an educational tool, a business process management tool, or a banking application. A connectionless alternative to TCP is UDP (User Datagram Protocol), a protocol without reliability guarantees often found in real-time applications, such as VoIP or video streaming applications, where missing a packet is less critical than the delay that would be caused by retransmission.

The specific difficulties regarding RTT in TCP can best be illustrated by comparison with UDP. UDP sends individual messages and provides checksums for error detection and port numbers for identifying services. If a packet is lost, it simply disappears, and the recipient may hear a noise or brief distortion in a VoIP connection. If packets arrive in a different ordering than the one they were sent in, the recipient's UDP layer cannot detect this, but they can be discarded without much loss. TCP, on the other hand, establishes a connection before starting to send data as a sequence of packets, each identified with a sequence number, which allows the recipient's TCP layer to reassemble the original data stream. In order to do this, it uses an acknowledgement and retransmission mechanism to make sure all packets have been received. We need to look at this process in slightly more detail to understand the impact of RTT. A detailed introduction and comparison of techniques is beyond the scope of this paper and can be found in Dordal [6]; an evaluation of the different newer proposals with respect to satellite links, in Shin et al, [16].

In establishing a TCP connection, the client and server, inter alia, exchange information about their respective capabilities. A main concern at this stage is to avoid congestion on the client's side by establishing the maximum window *rwnd* of packets the receiver can process, as congestion leads to packets being dropped by the underlying IP layer. After the connection has been started, data packets are sent from the server to the client, which responds by acknowledging receipt. The receiver sends an acknowledgement (ACK) with a packet number when a new packet arrives and the receiver has received each packet up to that number. If the packet numbered n is lost or delayed and the packet with number $n + 1$ arrives, the receiver thus, sends an acknowledgement for sequence number $n - 1$ again. If packet n now arrives, $n + 1$ can be acknowledged (cumulative acknowledgements). If the same packet, e.g., $n - 1$, is acknowledged three times, the sender assumes the packet is lost. The original TCP specification [14] of 1981 lets the sender restart sending all packets from n again, even if other packets have arrived. In contrast, the 2009 TCP [2] then resends n, and continues from where it was (fast retransmit). The sender also keeps a timer for each packet. If the sender does not receive an acknowledgement before a timeout is reached, the not acknowledged packet is assumed lost and retransmitted.

During the transmission phase, the sender is responsible for avoiding congestion on the network. It, therefore, tries to adjust its sending rate to the conditions of the network. The only information available to TCP is roundtrip time (RTT). Packet loss is inferred using this mechanism. The timeout window is a conservative estimate of RTT. The basic assumption is that delays and packet loss are caused by congestion in the network and can, therefore, be addressed by reducing the packet rate. Several updates to TCP [2] focus on congestion recognition and avoidance.

The congestion window (*cwnd*) controls the packet rate. The “slow start” algorithm was introduced as an algorithm for beginning “transmission into a network with unknown conditions.” It starts from a very small *cwnd* and successively increases the rate by one with each ACK. This doubles the *cwnd* after each RTT. This doubling of packet rate stops when the initially set maximum window size of the receiver is reached. When the slow start threshold *ssthresh* is reached the increase is slowed to a linear increase of packets per RTT. When a packet is lost, congestion avoidance is initiated. Different TCP implementations use different congestion avoidance algorithms. In the simplest version (TCP Tahoe) the lost packet is sent again (fast retransmit), the *cwnd* is set to its very small initial size and *ssthresh* is halved. Another version (TCP Reno and NewReno) differs in that it sets the *cwnd* not to the minimal size but to the halved *ssthresh*, essentially stopping the slow start phase and moving directly to the linear growth phase. Other proposals (e.g., Vegas, FAST, Westwood, Veno, and Illinois) improve on Reno’s greedy loss-based strategy using a delay-based strategy and are able to speed up a high-speed connection faster, but also assume that RTT depends solely on congestion. TCP Veno targets noisy wireless connections. High-speed connections are better supported, in particular, by Highspeed TCP and TCP CUBIC, which aim for a more aggressive *cwnd* increase.

That distance could play a significant role in wired networks seems counterintuitive, given that electric signals travel at the speed of light. However, “TCP extensions for long-delay paths” have been a topic since the 1980s [11] when addressing satellite links.

TCP Hybla [5] is a TCP implementation that targets specifically satellite connections. Interestingly, it achieves its improvements by replacing RTT measurements for estimates of minimum RTT with a reference value chosen to be suitable for satellite links, i.e., a fixed value depending on the satellites’ orbit, that is, a distance. The geostationary networking satellite orbits are in 35,786 km above the equator, and satellite links suffer from about 600 ms delays [16].

2.2 Packet reordering

A generally high variability of RTT has been experimentally confirmed by Aikat et al [1]. This has a large impact on TCP as it bases its strategy only on measurements of RTT. High variability leads to reordering of packets. As would be expected, the variability for small RTT is smaller, while intercontinental travel employs submarine cables and is thus more prone to meet bottlenecks (Fig. 1).

The fact that reordering is not pathological behavior but an issue of TCP in particular over long-haul connections is well known [4] and was studied experimentally, in particular with respect to high-speed networks [7] with renewed interest more recently. High speed networks face the same issues as long-distance networks. The problem in both cases is the number of packets in transit in relation to reordering capability. With a high-speed connection this number is high even over short distances; likewise, with lower speeds and longer distances. The number of packets in transit for a connection can be summarized as: $\tau = \rho * \text{RTT}$, where ρ is the packet send rate in packets per second. In both cases, τ is high. In the case of high speed networks, because ρ is large. In case of long distance networks, because RTT is large.

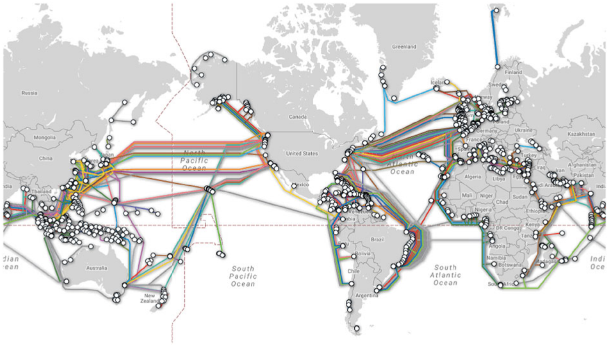

Fig. 1 Current and planned submarine cables [18]

An important aspect is the availability of alternative paths. Today’s internet makes considerable use of path diversity to increase throughput, and transmission speeds are steadily increasing letting the problem surface over increasingly shorter distances [3]. From a spatiotemporal point of view, the measure τ reflects that high-speed connections generate a similar effect as spatial distance. With a short RTT, however, control messages can be exchanged relatively quickly between sender and receiver to restart a connection. A long-distance connection does not have this option. Additionally, restarting a connection after a timeout, happens faster with a smaller RTT.

The consequences of packet reordering for TCP are that it “makes inefficient use of available link bandwidth,” and it “loses self-clocking and becomes highly bursty” [4]. Increasing reordering capability by selective acknowledgements (TCP SACK) has been shown to be an effective measure [7] and is supported by some of the newer protocols. However, relaxing the fixed three packet boundary comes at the price of increased complexity.

Parallel paths are the main reason for reordering [4], but also a way to improve packet rate. Eliminating parallel paths is a theoretical solution [8], but comes at the expense of decreasing maximum utilization of resources and thus efficiency [4], which is not desirable especially in the case of long-distance connections.

2.3 Approaches for atypical distances

Several approaches for providing reliable connection based on TCP have been cited in the above discussion. The current method to address the problem of long-haul connections is to establish mirrors close to where the clients are. A protocol that specifically supports this approach is the Stream Control Transmission Protocol [SCTP, 17]. SCTP provides multihoming support and establishes parallel channels. It uses path selection and monitoring to improve on the reordering issue. It also leverages using larger packet sizes (Ethernet jumbo frames).

Using spatial properties for network operation has been proposed in other domains of networking. Georouting approaches [13] are employed for ad hoc networks, such as geosensor networks.

The problems of long-haul transmissions have been traditionally studied in the area of delay-tolerant (DTN) or intermittently connected networks (ICN) [12], including interplanetary communications, MANETs, or communication to underdeveloped regions. ICNs are currently operating on distinct protocols employing so-called bundles [15].

3 Experiment

The goal of our experiment was to determine the correlation between RTT and distance on its own and as a secondary concern compare it to the correlation between hop count and RTT. We, therefore, did a detailed study of a widely dispersed network of servers at public universities. We chose public universities, because private companies including universities often already employ CDNs, which would disable our ability to study distance. Our rationale in selecting specific universities was to have at least two from each continent. However, in both Africa and South America we could only find one university each that fulfilled the necessary criteria and participated over the whole duration. We tried to equally select servers in both developed and developing regions on each continent.[1] However, we could not find a server among the low income countries in Africa that would have fulfilled our criteria.

3.1 Experimental setup

We designed an experiment using the widely available tools `ping` and `traceroute` to retrieve RTT and hop count and used a Haversine distance implementation[2] to approximate geodistances. Through a web search of lists of universities represented on Wikipedia[3], we selected 50 public universities worldwide. We checked all servers for the following minimum criteria:

- The servers had to be enabled to respond to queries. We tested this using ping.
- They should not be using a CDN so as to not distort localization. We determined this together with the hop count by using traceroute and checking the final responding server's geolocation with the IPLocation service.[4]
- They should have a less than 1% overall packet drop rate in an initial test. A high packet drop rate could indicate a distinct problem or misconfiguration on the path or at the server. After initial ping probes, we removed all servers that did not meet this criterion.

All servers passing the initial vetting process were kept in the pool over the whole duration. No servers were removed after this. Overall, only 11 of the selected university servers fulfilled all requirements. Table 1 shows the locations of the servers together with their distances to our server.

All in all, 16 experiments were performed. Eight experiments were executed on a weekend day (at the origin), the remaining eight during weekdays. Experiments also were varied by the times of the day (at the origin): four were executed at midnight, four at 6am, four at noon, and four at 6pm.

In each experiment, two separate bursts of 50 ping probes were sent to each server. The returned RTT values were recorded and stored. We thus all in all sent $16 \times 11 \times 100 = 17{,}600$ pings, and received 17,210 responses. There were only two anomalies. The server in Rio de Janeiro temporarily dropped out of the study and was unreachable. The second anomaly was that the university in Cape Town, maybe intentionally, as it serves a continent where the internet is still under development, replied to nearly all pings with duplicate responses. As it was the only university in Africa that passed our criteria, removing it would have removed Africa from the study. We thus only removed the duplicates.

3.2 Results

We determined the Pearson correlation coefficient between distance and delay, and also between hop count and delay. The result over the whole point set was a correlation of $0.57 > 0.5$, that is, a boundary case strong correlation, for distance. We obtained $0.4 < 0.5$, a weaker correlation of medium strength, for hop count. To understand better whether time influences the result we also analyzed the different categories of time slices and locations. Since an earlier

[1] For this classification, we relied mainly on the IMF classification [10], into advanced economies (A) and emerging market and developing economies (E): South Korea (A), Vietnam (E), New Zealand (A), Australia (A), South Africa (E, here A), Germany (A), Romania (E), USA (A), Mexico (E), Brazil (E). Because of its special status as the internet hub for a large part of Africa, we classified South Africa within the advanced category for our purposes.

[2] https://nathanrooy.github.io/posts/2016-09-07/haversine-with-python/

[3] We used the prefix search at: https://en.wikipedia.org/.

[4] https://www.iplocation.net

Table 1 Experimental Setup: cities of servers queried with number of pings returned not counting duplicates (n), GMT timezone (TZ), distance from the client in Oregon (timezone GMT -8) in km, hop count (HC), minimum delay (Min), average delay (Avg), maximum delay (Max), and standard deviation of delays (StD) over all n pings sent to that location

City	n	TZ	Dist	HC	Min	Avg	Max	StD
Daejeon/Seoul	1591	9.0	8531	22	176	198	586	61
Hanoi	1594	7.0	11234	21	284	400	915	106
Christchurch	1598	13.0	11636	13	189	198	514	31
Adelaide	1588	10.5	13221	23	192	255	4937	199
Cape Town	1599	2.0	16539	32	348	465	937	107
Hamburg	1600	1.0	8280	12	199	221	559	59
Bucharest	1588	2.0	9715	14	208	314	766	104
Gainesville FL	1600	−5.0	3913	19	106	112	410	22
Mexico City	1600	−6.0	3527	19	94	100	404	19
Rio de Janeiro	1292	−2.0	10994	27	220	337	798	106
Manaus	1560	−4.0	8150	24	220	340	812	110

version of this research was rejected with the comment that time had not been accounted for, we studied the detailed behavior for different times, taking into account both time at the fixed origin in Oregon and time at the various destinations. To do this, we partitioned the day into four slices: night $= [0:00, 6:00)$, morning $= [6:00, 12:00)$, afternoon $= [12:00, 18:00)$, evening $= [18:00, 24:00)$ to classify the responses by time at the destination.

Table 2 shows the results of the analysis. Over all countries, we can see that the correlation between distance and delay (column: CD) $>$ 0.5 for every entry, except during the evening on weekdays at the destination (0.39) and during weekend nights at the origin (0.43). When we look at developed countries, it also drops below 0.5 over weekday evenings at the destination (0.35) and over nights during the weekend at the origin (0.39). With the exception of these four cases all other eight analyzed cases have a value larger than 0.5, the maximum being during weekday afternoons at the destinations in developed countries (0.74), with the same time slot at destinations over all countries being the second highest value (0.72). A value of 0.7 is reached also in developing countries when looking at time at the origin in the weekend morning and weekday night slots. In the developing countries, CD varies between 0.59 and 0.7 for time at the origin and between 0.57 and 0.69 for time at the destination, that is, the correlation CD is strong at all times for developing countries.

The correlation between hop count and delay (CH) is in every case a smaller correlation as visualized by the last column in each country group: Dif $=$ CD $-$ CH. It shows by how much the correlation between distance and delay exceeds the correlation between hop count and delay. It is positive for each location type and time, relative to origin as well as relative to destination. The difference between CD and CH is particularly large for developing countries ($0.36 \leq$ Dif ≤ 0.59) and small for developed countries ($0.07 \leq$ Dif ≤ 0.14).

3.3 Discussion

The data support the intuition that physical distance is a predictor for network behavior. The two servers in Mexico and Florida, for instance, have similar values in Table 1, although one is in a developing country and one is in a developed country. Likewise, Hamburg and Daejeon/Seoul have similar distance and similar values in Table 1.

Due to the larger number of experiments, however, the more detailed analysis of Table 2 allows us to further study how the different parameters, location, time, and hop count, influence RTT at a given time. When looking at the correlations for the smaller time slices in Table 2, the influence of how busy the network is at a given time influences the correlation considerably with correlation values increasing as measured RTT would be expected to be more homogenous within a given time slice.

The observation that the difference between the influence of hop count between developing and developed regions is large suggests a crucial difference between the different networks. A detailed interpretation, however, is not possible from the conducted experiments. However, this difference is worth noting as such differences may considerably impact the ability of developing countries to participate in the global economy.

One may object that the comparison between distance and hop count would be unfair as, depending on the chosen route, hop count could considerably vary. What this only emphasizes is, however, that hop count is an unreliable predictor, while distance is not. A statistical database of average hop count distance between any two network locations, thus,

Table 2 Experimental Results: number of pings (n), correlation between distance and delay (CD), correlation between hop count and delay (CH), and difference between the two correlations (Dif = CD − CH), distinguished by type of time of day, type day of of week, and for both time classified by the origin or by the destination of pings

Loc	Day	Time	All countries				Developing countries				Developed countries			
			n	CD	CH	Dif	n	CD	CH	Dif	n	CD	CH	Dif
Origin	Weekend	Night	2200	0.43	0.30	0.13	1000	0.61	0.16	0.45	1200	0.39	0.32	0.07
		Morning	2180	0.59	0.44	0.15	980	0.70	0.18	0.52	1200	0.67	0.59	0.08
		Afternoon	2190	0.60	0.43	0.17	991	0.59	0.20	0.39	1199	0.69	0.55	0.14
		Evening	2083	0.64	0.39	0.25	890	0.61	0.09	0.52	1193	0.67	0.52	0.15
	Weekday	Night	2086	0.63	0.46	0.17	893	0.70	0.19	0.51	1193	0.68	0.60	0.08
		Morning	2089	0.60	0.46	0.14	892	0.68	0.20	0.48	1197	0.68	0.61	0.07
		Afternoon	2196	0.57	0.41	0.16	996	0.58	0.22	0.36	1200	0.65	0.51	0.14
		Evening	2186	0.63	0.45	0.18	992	0.67	0.19	0.48	1194	0.68	0.56	0.12
Destination	Weekend	Night	2096	0.58	0.38	0.20	896	0.57	0.10	0.47	1200	0.65	0.53	0.12
		Morning	2185	0.59	0.44	0.15	986	0.60	0.20	0.40	1199	0.68	0.56	0.12
		Afternoon	2190	0.62	0.44	0.18	991	0.69	0.17	0.52	1199	0.68	0.59	0.09
		Evening	2191	0.60	0.40	0.20	991	0.61	0.19	0.42	1200	0.67	0.53	0.14
	Weekday	Night	2189	0.59	0.45	0.14	993	0.63	0.20	0.43	1196	0.67	0.57	0.10
		Morning	2091	0.58	0.45	0.13	893	0.69	0.20	0.49	1198	0.66	0.59	0.07
		Afternoon	2078	0.72	0.47	0.25	894	0.69	0.10	0.59	1184	0.74	0.62	0.12
		Evening	2190	0.39	0.29	0.10	990	0.60	0.24	0.36	1200	0.35	0.28	0.07

would not only be considerably more costly to create, but may even be not as useful as a simple distance calculation.

4 Conclusions

We reported a global experiment in which we sent 17,600 pings to 11 servers on six continents and collected 17,210 responses. Our study is unique in its reproducibility as it only uses publicly available internet tools. Large-scale studies reporting on variability in RTT as depending on distance, such as [4], were performed at internet service provider (ISP) facilities, and are thus not easily reproducible. The study is also unique in that it probes the internet with a geographic rather than technical perspective as a complex ecosystem.

We studied the question whether the internet is a spatial network. To our knowledge, this question has not been asked before, although there are a number of studies reporting a distance effect and a number of technologies dedicated to handling specific atypical constellations. We tested how strongly correlated distance and RTT are and found a strong correlation for seven of the eight time slices by type of day in the week studied, and for both developed and developing regions. Moreover, the correlation was in all cases stronger for distance than for hop count. In developing countries the correlation was at least 0.57 for all days at all times. Given this result, we conclude that our research question has been affirmed.

While we did not make recommendations or try to give a deeper explanation beyond what has been stated in the literature, we can conclude that, given that distance can be calculated easily, a more flexible, spatiotemporally distance-adaptive protocol could be developed with relatively small overhead. Such a spatiotemporally adaptive protocol would not only serve currently underserved regions and satellite connections at the long range, but equally also high-speed connections at the short range.

Future work should also study the surprising difference between the influence of hop count for servers in developing and developed countries. The internet plays an increasingly important role in developing countries as everywhere. For developing countries, however, it connects local businesses to potential customers and presents the necessary access route to creating local employment and revenues on a global market, a vital factor in economic development and thus regional stability.

Acknowledgements The author is grateful to the anonymous reviewers of this article for helpful comments.

References

1. Aikat J, Kaur J, Smith FD, Jeffay K (2003) Variability in TCP round-trip times. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. ACM, New York, pp 279–284
2. Allman M, Paxson V (2009) RFC 5681: TCP congestion control. Technical report. IETF. https://tools.ietf.org/html/rfc5681. Accessed 26 Oct 2017
3. Bellardo J, Savage S (2002) Measuring packet reordering. In: Proceedings of the 2nd ACM SIGCOMM workshop on internet measurement. ACM, New York, pp 97–105
4. Bennett JC, Partridge C, Shectman N (1999) Packet reordering is not pathological network behavior. IEEE/ACM Trans Netw (TON) 7(6):789–798
5. Caini C, Firrincieli R (2004) TCP Hybla: a TCP enhancement for heterogeneous networks. Int J Sat Commun Netw 22(5):547–566
6. Dordal PL (2014) An introduction to computer networks. Loyola University Chicago, Chicago
7. Feng J, Ouyang Z, Xu L, Ramamurthy B (2009) Packet reordering in high-speed networks and its impact on high-speed TCP variants. Comput Commun 32(1):62–68
8. Gao C, Ling Z, Yuan Y (2014) Packet reordering analysis for concurrent multipath transfer. Int J Commun Syst 27(12):4510–4526
9. Hofmann M, Beaumont LR (2005) Content networking: architecture, protocols, and practice. Elsevier, Amsterdam
10. International Monetary Fund (2018) World economic outlook: cyclical upswing, structural change. Technical report. International Monetary Fund, Washington, DC
11. Jacobson V, Braden R (1988) RFC 1072: TCP extensions for long-delay paths. Technical report. IETF. https://tools.ietf.org/html/rfc1072. Accessed 26 Oct 2017
12. Khabbaz MJ, Assi CM, Fawaz WF (2012) Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges. IEEE Commun Surv Tutor 14(2):607–640
13. Maihofer C (2004) A survey of geocast routing protocols. IEEE Commun Surv Tutor 6(2):32–42
14. Postel J (1981) RFC 793: transmission control protocol. Technical report. IETF. https://tools.ietf.org/html/rfc793. Accessed 26 Oct 2017
15. Scott K, Burleigh S (2007) RFC 5050: bundle protocol specification. Technical report. IETF. https://tools.ietf.org/html/rfc5050. Accessed 26 Oct 2017
16. Shin M, Park M, Oh D, Kim B, Lee J (2014) Adaptive logarithmic increase congestion control algorithm for satellite networks. TIIS 8(8):2796–2813
17. Stewart R (2007) RFC 4960: Stream control transmission protocol. Technical report. IETF. https://tools.ietf.org/html/rfc4960. Accessed 26 Oct 2017
18. TeleGeography (2017) Submarine cable map. http://www.submarinecablemap.com/. Accessed 14 June 2017