Radar Chart: Scanning for Satisfactory QoE in QoS Dimensions

Yu-Chun Chang[1,2], Chi-Jui Chang[2], Kuan-Ta Chen[2], and Chin-Laung Lei[1]

[1]Department of Electrical Engineering, National Taiwan University
[2]Institute of Information Science, Academia Sinica

Abstract

The ongoing convergence of QoE (Quality of Experience) and QoS (Quality of Service) studies to give a thorough understanding of the end-user has posed numerous exciting possibilities for network and multimedia researchers. However, there is not yet a proper visualization tool that is able to map the many-to-one relationship between QoS metrics and QoE, leaving researchers speechless in the cacophony of traditional two-dimensional diagrams. Though mostly employed in qualitative analysis, we found that the radar chart, with a few tweaks, surprisingly suitable for the purpose. In this article, we present our adaptation of the radar chart, and demonstrate in a Voice-over-IP context its use in single- and cross-application performance analysis, application recommendation, and network diagnosis.

Index Terms

E-Model, Mean Opinion Score (MOS), Perceptual Evaluation of Speech Quality (PESQ),
Quality of Experience (QoE), Quality of Service (QoS), Voice-over-IP (VoIP), Peak Signal-to-Noise Ratio (PSNR)

Introduction

Any network application in the development life cycle must come to a point where its targeted users, based on their *personal feelings*, make the final assessment of its enjoyability and determine its fate. The assessment, technically known as Quality of Experience (QoE) [1], is vital and somewhat curious to network and multimedia researchers who dedicate themselves into fulfilling the end-users' needs. Presently, the most prevalent method for assessing QoE is the Mean Opinion Score (MOS) [2]. In a MOS test evaluating a particular application, human-beings are asked to give category ratings of it, ranging from Bad to Excellent. The MOS for the application is then the arithmetic mean of all individual ratings. Alternatively, MOS can be estimated from objective, automated methods like PESQ [3] to avoid the personnel cost and the trouble to set up a standard testing environment.

Quality of Service (QoS), on the other hand, approaches the subject matter from a researcher or engineer's point of view. In other words, QoS is just like the black and white of a contract along with a means to determine how the application provider lives up to it. He may meet the contract's requirements but fail to impress the end-users and score a high QoE, or less frequently, the users may be content with the application (as they have found the unintended usage of it, for instance) while the provider could not keep his word. Common QoS metrics in networking include latency (delay), jitter, bandwidth, data loss rate, bit error rate, and so on. While the idea of appreciating QoE for the sake of application refinement may seem straightforward, analytically modelling QoE in QoS terms is not. As a matter of fact, we theorize QoE to be a function of an almost infinite number of QoS metrics $\{M_i | i = 1, 2, ..., n\}$ as in

$$QoE = f(M_1, M_2, M_3, ..., M_n)$$

if everything including the production level of equipments or even the user's instantaneous mood is considered. The most comprehensive visualization of such complex relationship, after we have stripped it down to $n = 3$ (delay, loss rate, and bandwidth), is perhaps a versicolor three-dimensional contour map, rendered on a computer screen with some interactive functionality for seeing through and scrolling

the cross-sections to and fro. Unfortunately, contemporary printing technology is far from being able to present such *objet d'art*. Researchers often resort to a multitude of line charts, box plots, histograms, scatter graphs, etc. to establish their arguments. It is therefore very tempting to contrive a technique that combine all the necessary information onto a single diagram.

Among known tools of planar visualization, the *radar chart* emerges as a promising candidate for displaying multivariate data. It is capable of conveying singular or plural observations, represented as polygons with one vertex for each variable. The vertices are placed on equi-angular spokes in a way to depict the respective magnitude of the variables. Radar charts are oftenest seen in the context of business management and electronic gaming. For example, an in-game character's polygon readily explains his ability level in every facet (strength, dexterity, constitution, wisdom, charisma, to name a few), and when superimposed or juxtaposed with other characters' it unleashes a telling comparison between them.

In the ensuing parts of this article, we shall first review the related work and propose an adaptation of radar chart that is suitable for mapping QoS metrics to QoE. We then explicate the technicalities of our radar chart. The contrived uses of the customized radar chart, along with its original and intricate layout, are the article's main contributions. In particular, we regard that the radar chart is helpful to carry out

1) **single-application performance analysis**, in which an application's QoE polygons at different values of MOS are superimposed and compared,
2) **cross-application performance analysis**, in which polygons of competing applications at the same MOS are superimposed and compared,
3) **application recommendation**, which requires the "meteorological" variant to suggest the application providing the best QoE given certain network conditions, and,
4) **network diagnosis**, or monitoring the transition of network conditions to pinpoint implicit connection problems.

The final section summarizes and concludes the article.

Related Work

The computer science literature has not seen many appearances of the radar chart. There is, however, a number of precursors in other disciplines. The authors of [4], for example, evaluated the impact of blended learning on a collegiate course with radar charts, whereas in [5] the radar chart fused plural parameters to give a verbal description of environmental comfort level. Buttock pressure distribution of patients of spinal cord injury was analyzed in [6] with a hexagonal variant to find the most comfortable wheelchair cushion and patient posture. Radar charts in [7] summarized relative concentrations of up to fourteen chemical elements in order to perform the multielement correlation analyses for medical diagnosis. Terashima et al. [8] used 5-axe cobweb charts to compare and evaluate antioxidant activities of various standard antioxidants contained in food materials and food products. In [9], Joan introduced the radar plot, a useful graphical display method for multivariate data, to help health-care researchers convince audiences of their analysis and allow them to examine data easily. Mok et al. [10] used radar charts to visualize how network quality factors, such as delay, packet loss rate, and network bandwidth, affect the QoE of streamed video clips. In addition to radar chart, we proposed a crowdsourcing platform to quantify the QoE of multimedia content based on paired comparison in [11]. We also proposed a general, cheat-proof framework that enables researchers to systematically quantify the minimum QoS needs for real-time networked multimedia services in [12].

Proposing a Radar Chart Adaptation

A radar chart consists of equi-angular spokes, or axes, each representing a distinct variable. A data point of a variable is placed on the axis so that its distance from the origin relative to the length of the axis is proportional to the magnitude of the variable relative to its maximum. Lines are drawn connecting data points on adjacent axes, thus forming the characteristic polygon for an observation. A radar chart

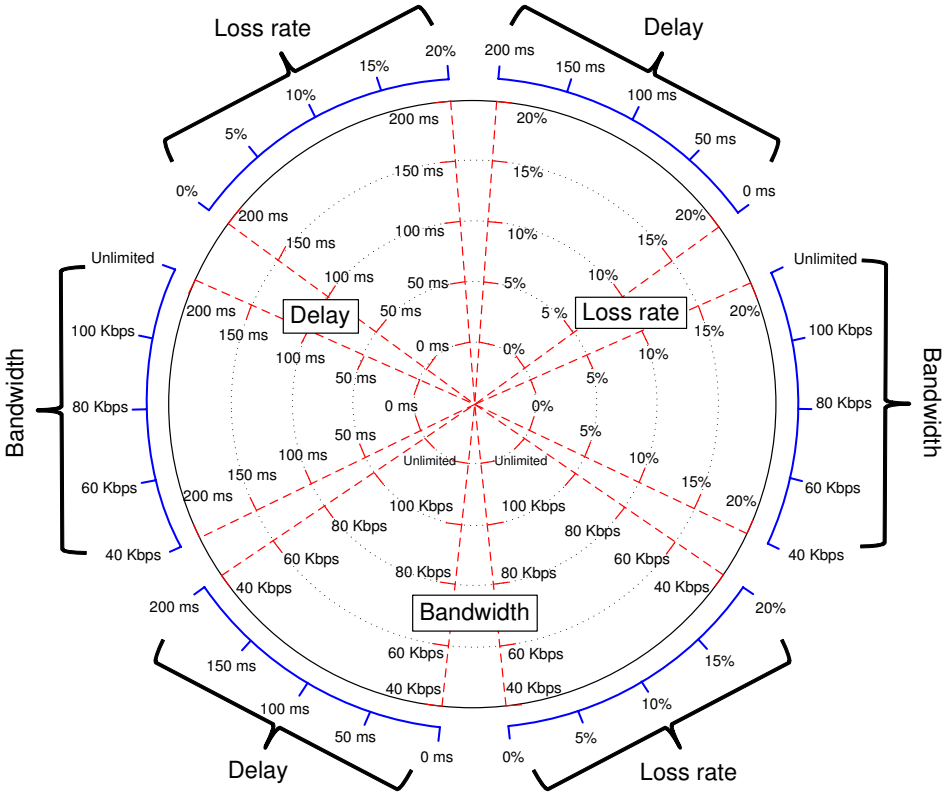


Fig. 1: The layout of our radar chart.

containing one polygon helps the researcher identify the dominant variables for a given observation. A radar chart with multiple polygons compares the relative strength and weakness of the observations.

The layout of the radar chart we use to map QoE as a function of QoS metrics is shown in Figure 1. It is divided into the three sectors of delay, loss rate, and bandwidth, each with five crossing arcs including the circumference. The arcs are the isolines for the metric represented by the sector. For example, the arcs of the bandwidth sector denote, from the inside out, unlimited bandwidth, 100 Kbps, 80 Kbps, 60 Kbps, and 40 Kbps, respectively. Each sector is further divided into two half-sectors, wherein axes intersecting the arcs. In the bandwidth sector, for example, the half neighboring the delay sector (one of the two bandwidth-delay half-sectors, the other being the half in the delay sector neighboring the bandwidth sector) is bounded by the 200 ms delay axis and the "no delay" axis, with axes for 150 ms, 100 ms, and 50 ms sitting in between. Every intersection of arc and axis on the radar chart is a network condition where the QoE of a certain application is measured. In our earlier design the centering axis of a sector always stood for the situation where all metrics other than the one represented by the sector are "perfect," but later on we decided to use in delay-loss rate half-sectors values averaged across all bandwidth settings instead of the value taken under unlimited bandwidth. A sample of the adapted radar chart is shown in Figure 2a. The area enclosed by the darkest polygon signifies network conditions where Skype is able to attain a MOS of 2.8 or above. For simplicity, we refer to the area as the polygon at (a MOS of) 2.8.

From this point on, we shall demonstrate the uses of our radar chart in the context of a sample study, which assesses the QoE of three popular VoIP applications: Skype 5.3, MSN Messenger 2011, and Google Talk 1.0. We begin with the experiment setup and QoE assessment procedure.

Technicalities of the Sample Study

To simulate VoIP calls in a controlled environment, we string up three commodity PCs to form a local area network. On one end, the "sender" machine initiates calls by feeding human voice recordings

TABLE I: Summary of traces

VoIP Application	Calls	Host	Packets	Bit Rate (mean/sd.)	Packet Rate (mean/sd.)	Packet Size (mean/sd.)
Skype	976	Sender	$12,058,392$	64 / 35 Kbps	48 / 19 pkt/sec	166 / 76 bytes
		Receiver	$11,200,088$	58 / 30 Kbps	45 / 18 pkt/sec	162 / 75 bytes
MSN Messenger	671	Sender	$6,758,092$	38 / 18 Kbps	40 / 15 pkt/sec	120 / 58 bytes
		Receiver	$6,089,949$	33 / 15 Kbps	36 / 15 pkt/sec	116 / 57 bytes
Google Talk	793	Sender	$3,632,410$	19 / 16 Kbps	18 / 11 pkt/sec	132 / 65 bytes
		Receiver	$3,395,124$	17 / 8 Kbps	17 / 4 pkt/sec	124 / 61 bytes
Overall	$2,440$		$43,134,055$	40 / 30 Kbps	35 / 19 pkt/sec	145 / 72 bytes

acquired from the Open Speech Repository[1] to one of the three pre-installed VoIP clients. The packet streams transmitted by the client, e.g., Skype, undergo various degrees of degradation when they pass through the FreeBSD host in the middle, and finally arrive at the "receiver" on the other end, decoded by the receiver's Skype instance and stored as Wavform file traces (`.WAV`). For video chat experiments, we initiate video chats with VoIP clients on the "sender" machine and play a 3-minute H.264 video clip; at the same time, the streamed video clip are stored on the "receiver" machine. We implement artificial degradation on the FreeBSD intermediate with its built-in `dummynet` facility, pairing different settings of delay (0, 50, 100, 150, and 200 ms), loss rate (0%, 5%, 10%, 15%, and 20%), and bandwidth (40, 60, 80, 100 Kbps, unlimited for VoIP experiments and 200, 400, 600, 800 Kbps, unlimited for video chat experiments). The traces thereby collected are summarized in Table I.

To estimate each audio trace file's MOS, we employ the objective QoE assessment procedure proposed in [13]:

1) Apply PESQ to the trace and its corresponding undegraded, original recording to obtain a rudimentary MOS.
2) Convert the rudimentary MOS to an R score using the formula in ITU-T G.107, Appendix I [14].
3) Compute the E-model delay impairments I_d given the delay we set in `dummynet`.
4) Subtract I_d from the R score and substitute the difference R' into

$$MOS = \begin{cases} 1 & R' < 0, \\ 1 + 0.0035R' + R'(R' - 60)(100 - R') \cdot 7 \cdot 10^{-6} & 0 < R' < 100, \\ 4.5 & R' > 100. \end{cases}$$

We use PSNR[2] instead of MOS to quantify the QoE of video chat applications because PSNR is generally adopted to represent the quality degradation of video frames due to network transmission. Like MOS, the higher PSNR indicates higher quality of experience. To compute video trace files' PSNR, we extract frames from the trace and compute the frame distortion between transmitted and original video clips in terms of PSNR. We then use the average PSNR of all the frames to represent the PSNR of video chat trace.

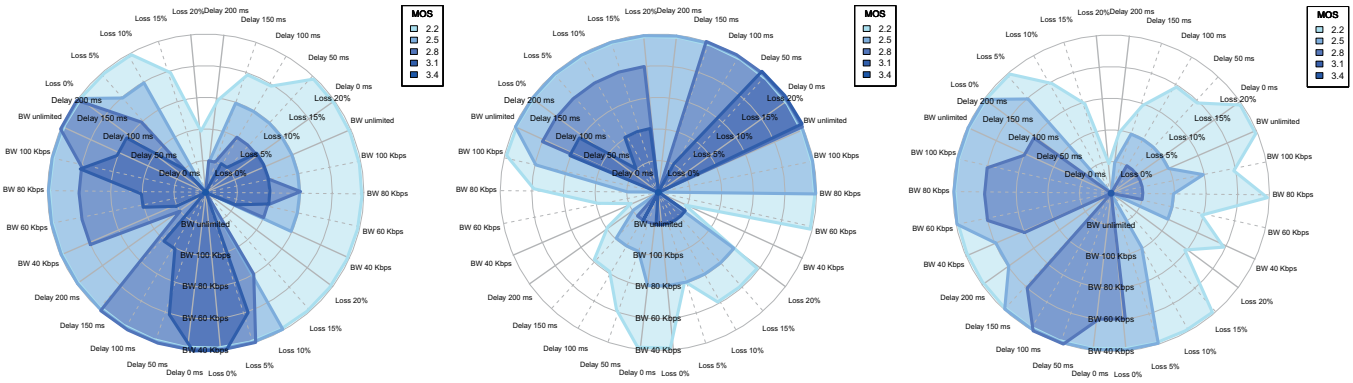
The following three sections are dedicated to the versatility of our radar chart. We first show a taste of how the visualization works, for example, in single- and cross-application performance analysis. We then débuts the "meteorological" radar chart variant and its accompanying "climate" triangle to establish a methodology for application recommendation. Finally, we attest that climate triangles alone can be used to give an diagnosis of long-term network problems.

Application Performance Analysis

In this section we demonstrate how to use the radar chart for application performance analysis. There are at least two ways. One can superimpose polygons of the same application at increasing levels of

[1] http://www.voiptroubleshooter.com/open_speech/

[2] http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio



(a) (b) (c)

Fig. 2: Single-application analysis, superimposing polygons at different values of MOS of (a) Skype, (b) MSN Messenger, and (c) Google Talk.

MOS onto the chart to see how the application deals with more and more stringent quality requirements. Alternatively, polygons of different applications at the same MOS can be plotted on a single chart to identify the applications' respective strength and weakness.

Single-Application Analysis

We start with the characteristic radar chart of Skype, Figure 2a. The polygons shrink rather steadily until the MOS reaches 3.4, the maximal QoE Skype is able to provide even under a perfect network condition where there is unlimited bandwidth, zero loss rate, and no delay. A closer look at the chart reveals that the areas involving loss rate shrink most rapidly, followed by mild contractions of delay parts along the bandwidth axes and arcs (compare the polygons at 2.5 and 3.4 in the bandwidth-delay half-sectors), indicating that the software is most susceptible to packet drops as compared to other factors.

The polygons in MSN Messenger's characteristic radar chart, Figure 2b, reveals obviously different performance than those in Skype and Google Talk. It indicates that MSN Messenger has a niche in lossy situations with network delay. Google Talk performs similarly as Skype, according to Figure 2c. The chart shows that Google Talk cannot offer acceptable QoE under most of lossy situations but for two small regions: 1). bandwidth ≥ 40 Kbps and loss rate $\leq 5\%$, and 2). delay ≤ 150 ms and loss rate $\leq 5\%$.

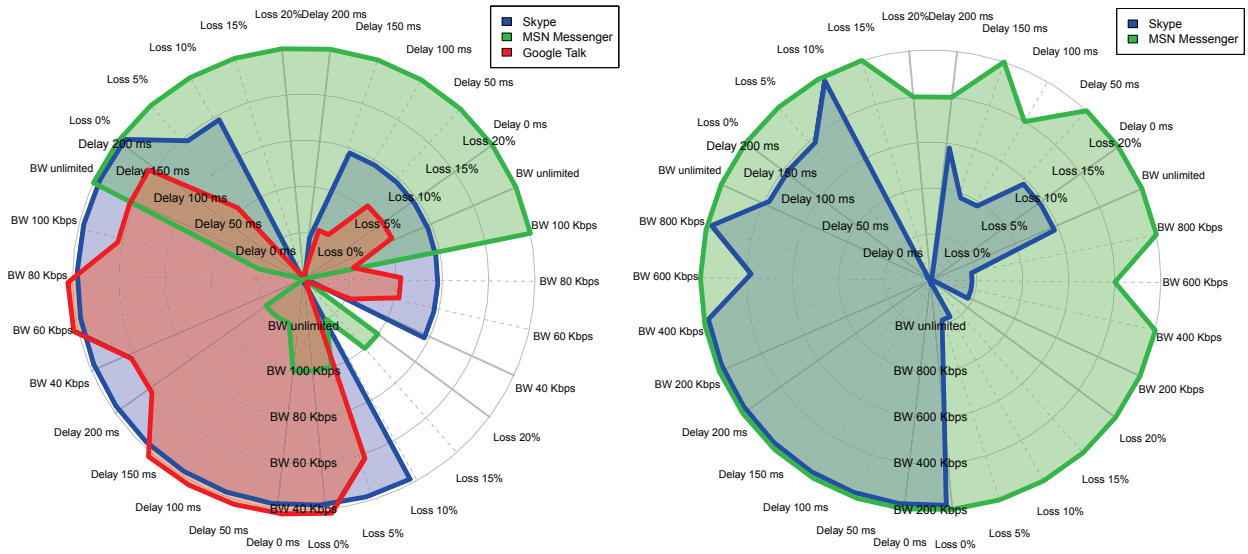
Generally, by skimming through the radar charts of the three applications, we can quickly capture the fact that in general Skype performs better than MSN Messenger while they both outshine Google Talk. The darkest polygons at 3.1 are most informative in this case, where that of Google Talk has been reduced to the origin and that of Skype spans larger part of the chart.

Cross-Application Analysis

Figure 3a and Figure 3b juxtapose the radar charts which incorporate VoIP clients' polygons respectively at MOS 2.6 with VoIP and PSNR 17 with video chat. We have discovered in the figure that:

- With a MOS requirement of 2.6 for VoIP analysis in Figure 3a, both Skype and Google Talk cannot offer acceptable QoE under most of lossy situations. It is also observed that the bitrate of Google Talk are fixed at around 17 Kbps, way below our bandwidth limits. The lack of bandwidth adaptation mechanism is clearly the cause of its vulnerability. In contrast to Skype and Google Talk, MSN Messenger perform better under most of lossy situations with network delay.
- With PSNR 17 for video chat analysis in Figure 3b[3], MSN Messenger outperforms Skype *in every way* especially in loss-delay and loss-bandwidth areas. It is that MSN Messenger has better encoding and transmission mechanisms to cope with lossy situations for higher QoE on video chatting.

[3]Google Talk does not support the functionality of video chat.



(a) (b)

Fig. 3: Cross-application analysis, superimposing polygons of the VoIP clients at (a) MOS 2.6 with VoIP and (b) PSNR 17 with video chat.

Just as we have done in this section, the preferred way to read the radar chart is to first grasp the "big picture," or the general shape of the polygons, before digging into the axes and arcs for numerical information. The radar chart is concomitantly concise and comprehensive. It is capable of being an intuitive indicator of application performance in terms of bandwidth, delay, and loss rate, and an ample organizer of multidimensional data points, which, in this case, accommodates 75 non-repetitive measurements encompassing four coordinates.

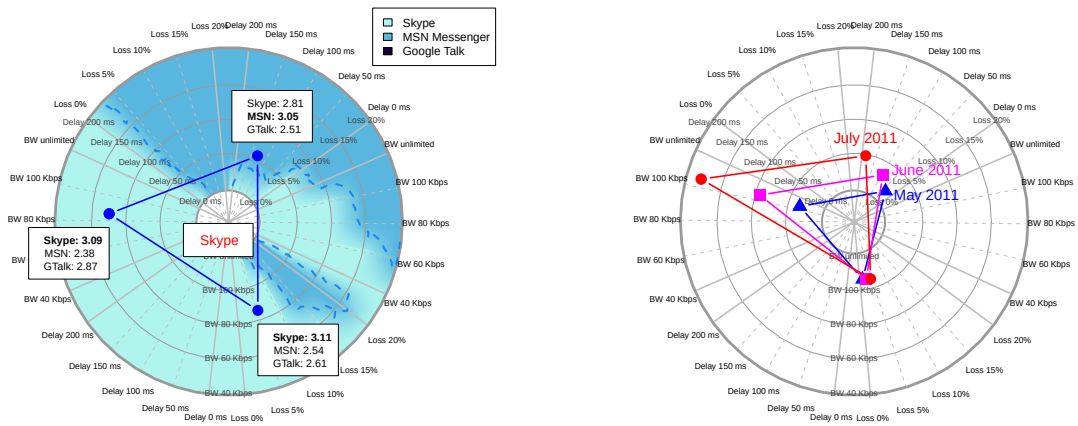
Application Recommendation

In this section we formulate the methodology of using radar charts for application recommendation. We first introduce the graphical tools involved, then validate the mathematics behind them and the results derived with an alternative method and different network condition sets.

Meteorological Radar Chart

Oftentimes a VoIP user has to make decisions on which application suits him best. We address the want for a recommendation in terms of *perceived voice quality* with the "meteorological" radar chart (for its resemblance to the radar echo sketch used in precipitation forecasts) as in Figure 4a. (The triangle and its associated notes will be explained shortly.) It is so constructed that every VoIP application occupies the region(s) where it can provide the best QoE among the three. Skype is recommended in most cases as it dominates most of the chart, while MSN Messenger has a niche in lossy situations.

For a more personalized recommendation, we need for every candidate application a characteristic radar chart that shows the maximum attainable MOS at each data point within. A particular user's instantaneous or averaged network condition, in terms of latency, packet loss, and bandwidth, corresponds to six points on each of these radar charts. The points are found by relating two of the metrics and ignoring the other one. Due to the apparent redundancy, on each chart only three heterologous points are necessary to represent the condition, thus forming a "climate" triangle like the one in Figure 4a. Because of the discrete nature of our radar chart design, we propose an appropriate method to compute the MOS of any VoIP application based on the climate triangle under such a network condition. We start the method MOS



(a) (b)

Fig. 4: (a) The meteorological radar chart with a climate triangle. (b) Transition of the climate triangle.

computation from the vertices of this climate triangle. As shown in Figure 5a, we derive the attainable MOS at triangular vertex N by bilinearly interpolating the four surrounding data points:

$$\begin{aligned} MOS_N &= MOS_A * (1-x)(1-y) + \\ &\quad MOS_B * x(1-y) + \\ &\quad MOS_C * (1-x)y + \\ &\quad MOS_D * xy, \end{aligned}$$

in the square-like form of edge length 1 in which x and y stand for relative distances of x- and y-direction respectively between the vertex N and the surrounding data points A. (Note the discrete nature of our radar chart design.) On each chart, the arithmetic mean of the vertices' MOS is computed as the best QoE the pertinent application can offer. Clearly the application producing the highest MOS for its triangle is recommended. To give a numerical example, the climate triangle in Figure 4a denotes a network condition where delay is 125 ms, loss rate 5.5%, and bandwidth 85 Kbps. Averaging the vertices' MOS on each application's characteristic radar chart yields 3.01, 2.66, and 2.65 for Skype, MSN Messenger, and Google Talk respectively. It is then determined that Skype is the most appropriate choice under such a network condition.

Validation of the Method

Bilinear interpolation, despite its seamless integration with the radar chart, is an out-of-dimension, projected approximator. To validate the performance of this radar chart/bilinear interpolation approach (RC/BI), we give here the formula of trilinear interpolation[4] for computing *theoretically more exact* MOS values of three-dimensional network conditions. As shown in Figure 5b, the maximum attainable MOS at network condition M is calculated by trilinearly interpolating the eight vertices of smallest enclosing

[4]http://en.wikipedia.org/wiki/Trilinear_interpolation/

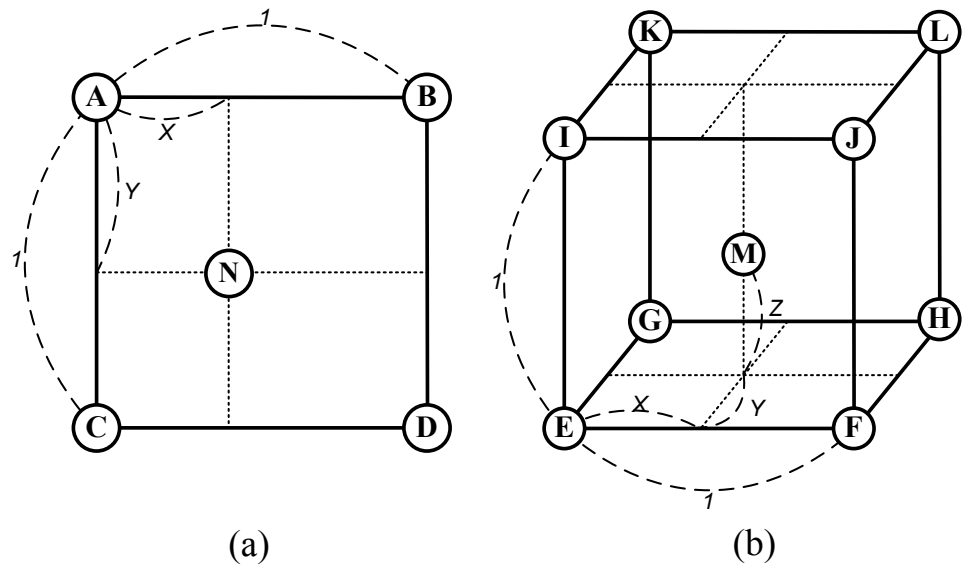


Fig. 5: (a) Climate triangle vertex N with its four surrounding data points. (b) Trilinear interpolation.

cube in which the length of edge is 1:

$$\begin{aligned} MOS_M &= MOS_E * (1-x)(1-y)(1-z) + \\ & MOS_F * x(1-y)(1-z) + \\ & MOS_G * (1-x)y(1-z) + \\ & MOS_H * xy(1-z) + \\ & MOS_I * (1-x)(1-y)z + \\ & MOS_J * x(1-y)z + \\ & MOS_K * (1-x)yz + \\ & MOS_L * xyz, \end{aligned}$$

where x, y and z represent relative distances of x-, y-, and z-direction respectively between the point M and the vertex E. Comparing the MOS_M of various applications should indicate which one is the most fitting under M.

To compare the RC/BI approach with trilinear interpolation, we feed them with both simulated and real-world network conditions. The simulated ones, known as uniform network conditions, are those uniformly distributed within the range of our `dummynet` settings. We randomly generated $1,000$ uniform network conditions, and processed them with both methods. Under $1,000$ uniform network conditions, while RC/BI suggests Skype with 70% probability in contrast to the trilinear method's 55%, RC/BI suggests MSN Messenger with 28% in contrast to the trilinear method's 42%. Both agree that Google Talk should almost never be used. The difference of MOS computed with bilinear and trilinear interpolation falls mostly in the range of 0 to 0.3.

For real-world network conditions, we turn to the PingER project[5], which monitors end-to-end performance of Internet links of over 700 sites in over 160 countries. The project primarily used `ping`, the ICMP echo request, to measure round-trip time and packet loss of a link, and applied [15] to derive TCP throughput which can be considered as available bandwidth in empirical network conditions. The PingER data comprise $4,459$ links, whose delay ranges from 1 to 600 ms, loss rate ranges from 0% to 15%, and bandwidth ranges from 7 to $100,000$ Kbps. As in the case of their uniform counterpart, we sampled $1,000$ network conditions from the data and again processed them with the two methods. The probability of

[5]http://www-iepm.slac.stanford.edu/pinger/

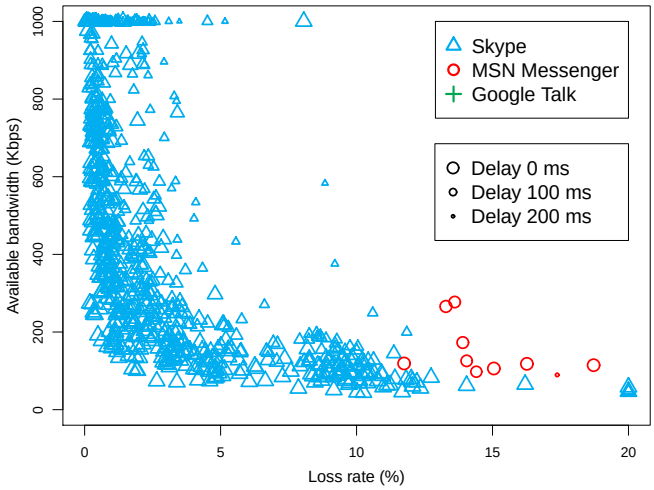


Fig. 6: Application recommendation results corresponding to different values of delay, loss rate, and bandwidth from empirical network conditions.

suggesting Skype by the RC/BI and the trilinear methods is 95% and 90%, respectively, and the difference of MOS is chiefly below 0.5. The result of MSN by both methods is 5% and 10%, respectively, and the difference of MOS is in the rage of 0.2 to 0.4. As the same as uniform network conditions, both methods agree that Google Talk should never be used under empirical network conditions.

While the recommendation results by the RC/BI approach are not significant aberration from those by trilinear interpolation, we notice that the radar chart/bilinear interpolation method is more biased towards the overall better-performing Skype under uniform network conditions, and that the distribution under empirical network condition is more concentrated than the distribution under uniform network condition. We argue that the first observation is not alarming, and even welcomed from an end-user's point of view, because frequently switching between VoIP clients just to get better conversation quality *for the moment* is simply not an option. On the other hand, to justify the over-90% recommendation ratio of Skype under empirical network conditions, we look to Figure 6 and find that 96% of the conditions have more than 80 Kbps of bandwidth available, consistent with our previous quantitative discovery by radar chart in Section "Single-Application Analysis" that MSN Messenger has a niche in lower-bandwidth, lossy situations.

Network Diagnosis

The final use of the radar chart arrives when a user notices but cannot quite pinpoint a *long-term* connection problem jeopardizing the QoE for his applications, leading to frustrations when he tries to solve it or communicate it to the ISP. By plotting the transition of the climate triangle over time on an empty radar chart, he is able to track his network condition and figure out where the problem is. To illustrate the procedure of diagnosis, let us assume that the month-averaged climate triangles in Figure 4b belongs to a user who has battled a gradual deterioration in his online gaming experience since May 2011. In each triangle, three vertexes are delay-bandwidth vertex, loss-delay vertex, and loss-bandwidth vertex, respectively. While his bandwidth-loss vertex (bottommost) remains mostly at the same position, the delay-bandwidth vertex (top left) moved conspicuously along the 100 Kbps-bandwidth axis from less than 50 ms of delay to closing in on 200 ms. In addition to the conclusion that he suffers from a growing delay, the loss-delay vertex (top right) suggests that there is a minor increase in the occurrence of packet loss as well. By so doing, users will notice the degradation of their network and conduct necessary actions such as contacting ISPs or changing his wireless APs based on the information from radar chart.

Conclusion

For the adapted radar chart presented in this article, we have proved its versatility and reliability through a sample study of VoIP clients. Specifically, we have demonstrated its usage in application performance

analysis, application recommendation, and network diagnosis. By plotting the polygons against varying QoE levels or across applications of the same nature, one may infer, both qualitatively and quantitatively, the characteristics and the relative strength and weakness of a particular application. We can also construct meteorological radar charts and climate triangles to recommend a user which application best fits his network configuration, or give diagnosis should there be any anomaly in his network connection.

Nevertheless, we realize that the design is not without drawbacks. Readability could be an issue for laymen or neophytes of the field, as they may need more time to comprehend a radar chart than traditional curve-line graphs. Redundancy of information, the necessary evil of our current adaptation, is also something we definitely want to get rid of. In the future, we wish to propose an even more advanced radar chart which accommodates the abundance of QoS factors while maintaining clarity and succinctness of information. In all, we regard the original visualization tool described in this article as an initiative to a broader and deeper exploration in the multidimensional realm of QoE and QoS.

Acknowledgment

This work was supported in part by the National Science Council under the grant NSC100-2628-E-001-002-MY3.

References

[1] R. Jain, "Quality of Experience," *IEEE Multimedia*, vol. 11, no. 1, pp. 96–97, Jan.-Mar. 2004.

[2] ITU-R Recommendation P.800, "Methods for subjective determination of transmission quality," Aug. 1996.

[3] ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Feb. 2001.

[4] A. Harding, D. Kaczynski, and L. Wood, "Evaluation of blended learning: Analysis of qualitative data," in *Proceedings of the Blended Learning in Science Teaching and Learning*, Sept. 2005, pp. 56–62.

[5] X. Li, W. Hong, J. Wang, J. Song, and J. Kang, "Research on the radar chart theory applied to the indoor environmental comfort level evaluation," in *The Sixth World Congress on Intelligent Control and Automation 2006*, vol. 1, 2006, pp. 5214–5217.

[6] Y. Tanimoto, H. Takechi, H. Nagahata, and H. Yamamoto, "Pressure measurement of air cushions for patients," in *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, vol. 2, 1999, pp. 1245–1250.

[7] T. Hasegawa, K. Inagaki, and H. Haraguchi, "Multielement correlation analysis of major-to-trace elements in human blood serum for medical diagnosis as studied by ICP-AES and ICP-MS," in *Proceedings of IUPAC International Congress on Analytical Sciences 2001*, vol. 17, 2001, pp. 979–982.

[8] M. Terashima, R. Watanabe, M. Ueki, and S. Matsumura, "Comprehensive evaluation of antioxidant activity for various substances with 5-axe cobweb chart," *Food Chemistry*, vol. 120, no. 1, pp. 150–155, 2010.

[9] M. J. Saary, "Radar plots: a useful way for presenting multivariate health care data," *Journal of Clinical Epidemiology*, vol. 61, no. 4, pp. 311–317, 2008.

[10] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of http video streaming," in *Proceedings of IFIP/IEEE IM (pre-conf.)*, 2011, pp. 485–492.

[11] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of Euphoria: A crowdsourcing platform for QoE assessment," *IEEE Network*, vol. 24, no. 2, pp. 28–35, Mar. 2010.

[12] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quantifying QoS requirements of network services: A cheat-proof framework," in *Proceedings of ACM Multimedia Systems 2011*, Feb. 2011, pp. 81–92.

[13] L. Ding and R. Goubran, "Assessment of effects of packet loss on speech quality in VoIP," in *Proceedings of IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications*, Sept. 2003, pp. 49–54.

[14] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning," Mar. 2005.

[15] P. Jitendra, F. Victor, F. T. Donald, and F. K. James, "Modeling TCP throughput: A simple model and its empirical validation," in *Proceedings of SIGCOMM 1998*, vol. 28, Oct. 1998, pp. 303–314.

BIOGRAPHIES

YU-CHUN CHANG (congo@fractal.ee.ntu.edu.tw) received his B.S. degree in computer science from National Taiwan University in 2005. He is currently a Ph.D. candidate in the Department of Electrical Engineering at National Taiwan University and the research assistant at the Institute of Information Science of Academia Sinica. His research interests include Internet measurement, QoE management, quality of service, and cloud gaming.

CHI-JUI CHANG (changcj@umich.edu) received his B.S. in electrical engineering in 2006 and M.Sc. in computer science in 2008, both from National Taiwan University. He is currently with the Multimedia Networking and Systems Laboratory of the Institute of Information Science, Academia Sinica, researching in QoE management, online privacy, multimedia networking, distributed systems, and social/human computing.

KUAN-TA CHEN (ktchen@iis.sinica.edu.tw) is an associate research fellow at the Institute of Information Science and the Research Center for Information Technology Innovation (joint appointment) of Academia Sinica. He received his Ph.D. in electrical engineering from National Taiwan University in 2006, and his B.S. and M.S. in computer science from National Tsing Hua University in 1998 and 2000, respectively. His research interests include QoE management, crowdsourcing, network security, and online games.

CHIN-LAUNG LEI (lei@cc.ee.ntu.edu.tw) received his B.S. degree in electrical engineering from National Taiwan University in 1980, and his Ph.D. degree in computer science from the University of Texas at Austin in 1986. He is currently a professor in the Department of Electrical Engineering, National Taiwan University, and the vice president of the Chinese Cryptology and Information Security Association. His current research interests include network security, cloud computing, and multimedia QoE management.