TRUST: A Trigger-based Automatic Subjective Weighting Method for Network Selection

Lusheng Wang
Telecom ParisTech (E.N.S.T.)
37 Rue Dareau, 75014, Paris, France
Email: lusheng.wang@telecom-paristech.fr

David Binet
France Telecom R&D (Orange Lab)
42 Rue des Coutures, 14066, Caen, France
Email: david.binet@orange-ftgroup.com

***Abstract*—In recent years, network selection in heterogeneous wireless networks has been widely studied by combining various attributes to make a final decision. However, weighting methods in the current literature are not suitable for practical usage. Therefore, we propose in this paper a novel weighting method, named TRigger-based aUtomatic Subjective weighTing (TRUST), which can efficiently calculate the subjective weights of various network attributes based on both terminal-side and network-side subjective requirements in the network selection issue. Trigger events of the network selection procedure are specifically considered by this method, so the subjective weights are calculated based on the effects that these events bring to the selection results. The proposed method has several obvious benefits compared with the widely used eigenvector method in the literature of network selection.**

***Index Terms*—network selection; heterogeneous wireless networks; eigenvector; entropy; analytic hierarchy process.**

I. Introduction and Related Work

In the context of the present trend towards ubiquity of networks and global mobility of services, we see that network access is provided by a large diversity of technologies with coverage overlaps. In this heterogeneous wireless network (HWN) environment, the previous always connected concept becomes always best connected (ABC) which requires dynamic selection of the best network and access technology when multiple options are available simultaneously [1].

In the near future, the HWN environment could contain many networks, e.g. universal mobile telecommunications system (UMTS), world-wide interoperability for microwave access (WiMAX), wireless local area network (WLAN), Bluetooth, etc. These networks have various attributes: either static (e.g. price, bandwidth, security level, bit error rate, jitter and power consumption, etc.) or dynamic (e.g. traffic load, signal strength, handover properties, etc.). Therefore, it is quite difficult to define the *best* network in the selection procedure. In order to always select a reasonable network, it is necessary to take a large number of attributes into consideration simultaneously, which turns network selection into an issue of multiple attribute decision making (MADM) [2]–[6].

In order to combine multiple attributes together, weights should be evaluated to represent their relative importance. Generally speaking, there are two types of weighting methods: objective and subjective [7]. Entropy and variance are two common objective weighting methods, while eigenvector and weighted least square are two well known subjective weighting methods. In the literature of network selection, eigenvector method has been widely used [2]–[6]; entropy and variance has also been considered [8], [9]; while weighted least square method has not been used up to now.

In this article, we firstly model the weighting procedure and present how to use the above four common objective or subjective weighting methods for the network selection issue in section II, III and IV, respectively. Then, we analyze their inappropriateness and propose our novel subjective weighting method TRUST in section V. Finally, we compare TRUST with eigenvector method and analyze possible improvement in section VI.

II. Modeling the Weighting Issue

In the MADM-based network selection procedure, multiple attributes are usually adjusted by normalization, fuzzy logic, utility functions, etc. Then, they are combined based on their weights to obtain a total utility (or total cost) for each network. Finally, the network with the maximum utility (or minimum cost) will be chosen as the best network. Weights are calculated in two parts: network attributes are used for calculating objective weights, while subjective requirements are used for subjective weights, as shown in Fig. 1. Therefore, we model the weighting procedure as follows:

Suppose n attributes are used in a network selection scheme and the number of candidate networks is m, we have the following decision matrix:

$$\mathbf{NW} = \begin{bmatrix} a_{11} & a_{12} & . & a_{1n} \\ a_{21} & a_{22} & . & a_{2n} \\ . & . & . & . \\ a_{m1} & a_{m2} & . & a_{mn} \end{bmatrix}, \tag{1}$$

where a_{mn} represents the value of the nth attribute of the mth candidate network.

In order to combine these attributes together, it is necessary to know their relative importance in advance, so weights of these attributes should be calculated and employed in network selection schemes. There are two types of weights: objective and subjective. Objective weights are directly obtained based on the above decision matrix, which are represented as

$$\mathbf{W_O} = \begin{bmatrix} wo_1 & wo_2 & ... & wo_n \end{bmatrix}. \tag{2}$$

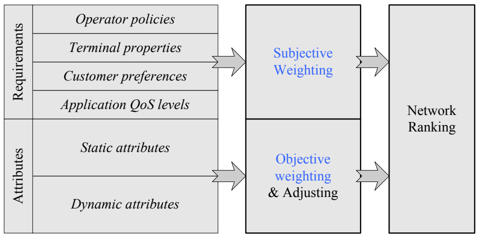

Fig. 1. MADM-based network selection.

Dissimilarly, subjective weights are usually obtained based on the subjective feelings of decision maker (DM). In the network selection issue, the DM is not exactly the customer. Subjective information should actually include customer preferences, terminal properties, application QoS requirements, operator policies, and so on. Based on these information, subjective weights can be obtained, which are represented as

$$\mathbf{W_S} = \begin{bmatrix} ws_1 & ws_2 & \ldots & ws_n \end{bmatrix}. \tag{3}$$

The final weights are obtained by combining the objective and subjective weights, as follows:

$$\mathbf{W} = \frac{\mathbf{W_O} \cdot \mathbf{W_S^{\Delta}}}{\mathbf{W_O} \cdot \mathbf{W_S^{T}}} = \begin{bmatrix} w_1 & w_2 & \ldots & w_n \end{bmatrix}, \tag{4}$$

where w_j is the combined weight of attribute j, given by

$$w_j = \frac{ws_j \cdot wo_j}{\sum_{i=1}^{m}(ws_i \cdot wo_i)}, \tag{5}$$

and $\mathbf{W_S^{\Delta}}$ is a diagonal matrix, given by

$$\mathbf{W_S^{\Delta}} = \begin{bmatrix} ws_1 & 0 & . & 0 \\ 0 & ws_2 & . & . \\ . & . & . & 0 \\ 0 & . & 0 & ws_n \end{bmatrix}, \tag{6}$$

In the following two sections, we will discuss detailedly on the methods to obtain the objective weights $\mathbf{W_O}$ and the subjective weights $\mathbf{W_S}$.

III. Objective Weighting Methods

A. Entropy method

To obtain objective weights, entropy is a widely used weighting method in various research and industrial fields. In information theory, entropy is a criterion for the amount of uncertainty represented by a discrete probability distribution which agrees that a broad distribution represents more uncertainty than does a sharply peaked one [7]. This measure of uncertainty is given by Shannon as

$$E = S\begin{bmatrix} p_1 & p_2 & \ldots & p_m \end{bmatrix} = -k\sum_{i=1}^{m}[p_i \ln(p_i)], \tag{7}$$

where p_i are normalized values with $\sum_{i=1}^{m} p_i = 1$ and k is a positive constant.

When all p_i are equal to each other, the entropy E reaches its maximum value, which represents the information denoted by $\begin{bmatrix} p_1 & p_2 & \ldots & p_m \end{bmatrix}$ is minimum.

For calculating the objective weights in the network selection procedure, the decision matrix $\mathbf{NW}$ is firstly normalized as follows

$$\mathbf{NW_{NORM}} = \begin{bmatrix} x_{11} & x_{12} & . & x_{1n} \\ x_{21} & x_{22} & . & x_{2n} \\ . & . & . & . \\ x_{m1} & x_{m2} & . & x_{mn} \end{bmatrix}, \tag{8}$$

where $x_{ij} = a_{ij}/\sum_{i=1}^{m} a_{ij}$.

The entropy of each attribute is calculated as

$$E_j = S\begin{bmatrix} x_{1j} & x_{2j} & \ldots & x_{mj} \end{bmatrix} = -k\sum_{i=1}^{m}[x_{ij}\ln(x_{ij})], \tag{9}$$

where $k = 1/\ln m$ to guarantee that $0 \leq E_j \leq 1$.

The degree of diversification D_j of the information provided by the outcomes of attribute j can be defined as

$$D_j = 1 - E_j. \tag{10}$$

Finally, the objective weight of attribute j can be obtained by normalization as

$$wo_j = \frac{D_j}{\sum_{i=1}^{n} D_j}. \tag{11}$$

B. Variance method

Another objective weighting method is the variance method which decides the weights of attributes according to their variations among all the candidate alternatives [8], given by

$$D_j = V\begin{bmatrix} x_{1j} & x_{2j} & \ldots & x_{mj} \end{bmatrix} = \sqrt{\frac{\sum_{i=1}^{m}(x_{ij} - \bar{x}_j)^2}{m\bar{x}_j}}, \tag{12}$$

where $\bar{x}_j$ represents the mean value of the jth attribute. The objective weights of the jth attribute can be finally obtained by the same normalization as the entropy method.

IV. Subjective Weighting Methods

To obtain subjective weights, pair-wise comparison between each pair of attributes is usually performed by the DM. By doing this, an n-by-n square matrix is obtained as

$$\mathbf{PW} = \begin{bmatrix} b_{11} & b_{12} & . & b_{1n} \\ b_{21} & b_{22} & . & b_{2n} \\ . & . & . & . \\ b_{m1} & b_{m2} & . & b_{mn} \end{bmatrix}, \tag{13}$$

where b_{ij} represents the comparison between attribute i and j, given as $b_{ij} = ws_i/ws_j$ for precise subjective estimation. Therefore, in the above pair-wise comparison matrix, we have $b_{ij} = 1/b_{ji}$ and $b_{ii} = 1$ for all $i, j \in 1, 2, ..., n$.

Based on the above matrix, there are two commonly used methods for calculating the subjective weights: eigenvector method and weighted least square method.

A. Eigenvector method

In ideal case, the pair-wise comparison matrix of (13) can be written as

$$\mathbf{PW} = \begin{bmatrix} \frac{ws_1}{ws_1} & \frac{ws_1}{ws_2} & . & \frac{ws_1}{ws_n} \\ \frac{ws_2}{ws_1} & \frac{ws_2}{ws_2} & . & \frac{ws_2}{ws_n} \\ . & . & . & . \\ \frac{ws_n}{ws_1} & \frac{ws_n}{ws_2} & . & \frac{ws_n}{ws_n} \end{bmatrix}. \tag{14}$$

Multiplying this matrix with the vector of subjective weights $\mathbf{W_S} = \begin{bmatrix} ws_1 & ws_2 & ... & ws_n \end{bmatrix}$ yields

$$\mathbf{PW} \cdot \mathbf{W_S} = n \cdot \mathbf{W_S}. \tag{15}$$

The above derivation is based on the assumption of $b_{ij} = ws_i/ws_j$ for all $i, j \in 1, 2, ..., n$. Actually, the precise values of b_{ij} are unknown and must be estimated. In other words, the judgments by the DM are subjective and cannot be completely accurate to satisfy the above derivation. Since small perturbations in the coefficients imply small perturbations in the eigenvalues, we can define a matrix $\mathbf{PW}^*$ as the DMs estimate of matrix $\mathbf{PW}$ (surely with small perturbations). Then, a vector of subjective weights $\mathbf{W_S^*}$ can be calculated as the eigenvector of matrix $\mathbf{PW}^*$ corresponding to its largest eigenvalue λ_{MAX}. The eigenvector can be obtained by solving the following system of linear equations:

$$(\mathbf{PW}^* - \lambda_{MAX}\mathbf{I}) \cdot \mathbf{W_S^*} = 0, \tag{16}$$

where $\mathbf{I}$ is an identity matrix.

Combining with the eigenvector method, analytic hierarchy process (AHP) is usually employed, which is defined as a procedure to divide a complex problem into a number of deciding criteria and sub-criteria and integrate their relative importance to find the optimal solution [5]. Based on AHP, attributes in the network selection issue are structured as a decision hierarchy. Then, eigenvector of each sub-layer is calculated to represent the weights of attributes in this sub-layer. In the end, weights of different sub-layers are synthesized as the final weights. An example of using AHP to calculate weights of QoS-related attributes can be found in [5].

B. Weighted least square method

Another method to obtain the subjective weights is the weighted least square method, but this method has not been well used in the studies of the network selection issue.

The concept of this method is to solve the following constrained optimization problem:

$$\begin{aligned} \min z = \sum_{i=1}^{n}\sum_{j=1}^{n}(b_{ij}^* ws_j - ws_i)^2 \\ s.t. \sum_{i=1}^{n} ws_i = 1, \end{aligned} \tag{17}$$

where b_{ij}^* represents the value in the estimated matrix PW^*.

The weights in the above model can be obtained by solving a system of linear equations as follows:

$$\begin{aligned} \sum_{i=1}^{n}(b_{ik}^* ws_k - ws_i)b_{ik}^* - \sum_{j=1}^{n}(b_{kj}^* ws_j - ws_k) + \eta = 0, \\ \text{and} \sum_{i=1}^{n} ws_i = 1, k = 1, 2, ..., n, \end{aligned} \tag{18}$$

where η represents the Lagrangian mutiplier.

V. Proposition

A. Inappropriateness of the above methods for weighting the attributes in the network selection issue

For weighting the attributes in the network selection issue, various factors should be considered. Entropy method can be used to obtain the objective weights which denote the relative differences of candidate networks respecting to various attributes. However, it is not enough to use only objective weights for representing these attributes relative importance because most of the factors for deciding the weights in this issue are subjective information which requires subjective weighting methods.

The two subjective weighting methods described above both use pair-wise comparison matrices, but only eigenvector method with AHP is commonly used in the literature of network selection. We also have the same preference in our research because the calculation of the eigenvector is relatively simpler than solving an $n+1$ system of linear equations in the weighted least square method, where n representing the number of attributes is usually large.

Unfortunately, the eigenvector method has an obvious problem while being used in the network selection procedure. As we know, pair-wise comparison matrices in this method are given by the DM based on his subjective feelings on the attributes, and the DM is usually human beings in most decision making processes. However, for the network selection issue, an automatic method is required because customers usually do not have the basic knowledge to construct the pair-wise comparison matrix. Moreover, the matrix changes in different situations (e.g. different applications and terminal properties), so customers do not want to be involved in the complicated pair-wise comparison process for each situation, even though they know how to do it. To sum up, when designing a network selection scheme, we should not suppose the customers to be the DMs who could provide pair-wise comparison matrix to calculate subjective weights. Furthermore, mobile terminals cannot be DMs either, because machines do not have subjective feelings on the attributes. In a word, evaluating the subjective weights in the network selection procedure is a tough work.

One possible approach to solve this problem is to do this work by the designer of the network selection scheme and to store the matrices for all scenarios in a ROM of the mobile terminal in advance. Imagining a mobile terminal that stores matrices of all the scenarios in its ROM, numerous factors should be considered, e.g. application QoS levels, customer preferences, terminal properties, operator policies, dynamic network attributes, etc.

All the above factors contain two or more options (for example, terminal velocity could be divided into moveless, low speed, mid speed, high speed, etc.), so there are actually thousands of scenarios that a network selection scheme should consider. Furthermore, if the terminal is a multi-mode one which can only use one access at one time, the scheme should synthetically consider all the on-going applications to make the final decision, which leads to much more scenarios to consider. Therefore, we can see that it is not efficient to store in advance the pair-wise comparison matrices of all the scenarios into a ROM of the terminal by the scheme designer.

In order to evaluate the subjective weights in an efficient and automatic manner, we propose in the following sub-section a novel weighting method, named TRUST, which can be considered as an extended usage of the eigenvector (plus AHP) method, but only specific for the network selection issue.

B. TRUST: a novel weighting method for network selection

Once the network selection procedure is triggered, a subjective weighting method will perform to calculate the subjective weights. The widely employed eigenvector method in research is actually not suitable for this task in practice, as we explained in the above sub-section. We emphasize here that the subjective weighting method for the network selection issue should satisfy at least the following conditions:

1) subjective weights should be automatically obtained by the mobile terminal (or a network-side entity), not by customer's pair-wise comparisons for each scenario;

2) the procedure should be efficient and fast for obtaining appropriate weights in different scenarios.

As we know, the network selection procedure is usually triggered by the following events:

- new application comes or previous application ends;
- terminal property (e.g. velocity) obviously changes;
- customer or operator changes his preference;
- certain dynamic network-side attribute (e.g. traffic load) obviously changes, etc.

Now that the above events can trigger the network selection procedure, we wonder what kind of effect certain event actually brings into the network selection procedure. Table I at the end of this article shows the relationship between trigger events and changes of subjective weights. We can see that the content of this table is generally fixed, such as streaming applications require large bandwidth; conversational applications require low jitter; high-speed terminals require good handover capability. Therefore, in order to obtain the subjective weights $\mathbf{W_S}$, we just need to know the on-going events and their relative importance, then we will be able to evaluate $\mathbf{W_S}$ based on Table I.

Therefore, we define a k-by-n matrix $\mathbf{EA}$ as follows to represent the right part of Table I, where k is the total number of events and n is the number of attributes considered by the scheme:

$$\mathbf{EA} = \begin{bmatrix} c_{11} & c_{12} & . & c_{1n} \\ c_{21} & c_{22} & . & c_{2n} \\ . & . & . & . \\ c_{k1} & c_{k2} & . & c_{kn} \end{bmatrix}, \tag{19}$$

where c_{ij} represents the effect of the ith event on the jth attribute, and the value of c_{ij} is either TRUE (1) or FALSE (0).

The on-going events can be obtained by checking all the events one by one in the table. Since there are only a few dozens of events in the table, this process can actually be completed very quickly. By doing so, we obtain a diagonal matrix TF with k non-negative integer as follows:

$$\mathbf{TF} = \begin{bmatrix} tf_{11} & 0 & . & 0 \\ 0 & tf_{22} & . & 0 \\ . & . & . & . \\ 0 & 0 & . & tf_{kk} \end{bmatrix}, \tag{20}$$

where element tf_{ii} in this matrix represents whether the event i is currently true or not. For applications, this integer represents how many of this type of application are on-going at the moment of network selection.

The relative importance of these events is complicated to be obtained, but it is much easier than calculating the weights of attributes because weights of these events do not change frequently. For example, an operator of a telecommunication network with mainly voice services feels that conversational application has higher importance than others; 'mid-speed' is less important than 'high-speed', although both of the two events require good handover capability; whether load balancing is more important than customer's preferences is decided by the operator's policy, etc. Therefore, the weights of these events will be calculated in advance and sent to the mobile terminal when the terminal is initiated by the customer. We use $\mathbf{W_E}$ to represent the weights of all the events in table I, given by

$$\mathbf{W_E} = \begin{bmatrix} we_1 & we_2 & ... & we_n \end{bmatrix}. \tag{21}$$

Eigenvector method (plus AHP) is suggested for calculating $\mathbf{W_E}$ by the scheme designer or the operator in advance. A hierarchy of several trigger events is formed in Fig. 2. There are two levels in the hierarchy: for the upper level, weights are obtained as

$$\mathbf{W_E1} = \begin{bmatrix} we1_1 & we1_2 & ... & we1_{k1} \end{bmatrix}, \tag{22}$$

and for the bottom level, weights of each group are obtained as

$$\mathbf{W_E2i} = \begin{bmatrix} we2_{i,1} & we2_{i,2} & ... & we2_{i,k2} \end{bmatrix}, \tag{23}$$

where i represents the ith group. The synthesized weights of the jth events in the ith group can be obtained as

$$we_{i,j} = we1_i \cdot we2_{i,j}. \tag{24}$$

Based on the preparation above, the procedure of TRUST is carried out as follows:

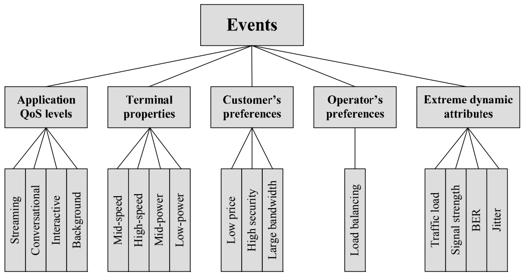

Fig. 2. Hierarchy of trigger events.

- In the mobile terminal, table I is stored. Suppose n network attributes and k events are considered for network selection, this table forms a k-by-n matrix $\mathbf{EA}$;
- Since the relative importance of these events does not change frequently, their weights $\mathbf{W_E}$ can be calculated and transmitted to the terminal when it is initiated by the customer;
- When the network selection procedure is triggered, all the k events are checked to obtain a diagonal matrix $\mathbf{TF}$ with k non-negative integer as explained above;
- Finally, the subjective weights are obtained by

$$\mathbf{W_S} = \left[\begin{array}{cccc} ws_1 & ws_2 & ... & ws_n \end{array} \right] = \mathbf{W_E} \cdot \mathbf{TF} \cdot \mathbf{EA}, \tag{25}$$

where the subjective weight of the jth attribute is

$$ws_j = \sum_{i=1}^{k} we_i \cdot tf_{ii} \cdot c_{ij}. \tag{26}$$

VI. Comparisons with Eigenvector Method

In this section, we compare the proposed method TRUST with the widely used eigenvector method in extensive scenarios. The weights of the eigenvector method are obtained by subjective pair-wise comparison matrices formed by the authors, while the weights of TRUST are calculated automatically by a simulation program using Matlab. 9 attributes are considered in the following comparisons, i.e. price (PR), bandwidth (BD), security (SC), power consumption (PC), bit error rate (BER), jitter (JT), traffic load (TL), handover properties (HO) and signal strength (SS).

Table I is just a simple example of $\mathbf{EA}$ which is used in this paper to calculate the weights by TRUST. If this matrix is designed carefully, the weights obtained by TRUST can be much closer to the weights by the eigenvector method. Here we use a simple example of $\mathbf{EA}$ in order to show not only the consistency but also the difference of the two methods.

Table II shows weights of the 9 attributes obtained by the two methods in 24 scenarios. We can see that the main trends of these weights are the same for the two methods in various scenarios.

In order to further evaluate our proposed method TRUST, we calculate the *mean of difference* and the *correlation* of weights obtained by the two methods, as shown in Fig. 3. We can see that weights of some attributes in some scenarios are obvious different when calculated by the two methods, e.g. the weight of bandwidth for streaming applications, the weight of jitter for conversational applications and the weight of biter error rate for interactive applications. Meanwhile, we can see that the correlations of the first two attributes are poor but that of the third attribute is good. To further explain this problem, we draw specifically the three attributes' weights in three subfigures, and find that distribution of the third attributes weights is linear, which is the reason that their correlation approaches 1.

Actually, the above analysis finds out two unexpected phenomena of the TRUST method. One phenomenon is that the important attribute usually obtains a larger weight by the TRUST, which is because the unimportant attributes' weights are considered as 0s by this method. By contrast, weights of these attributes by the eigenvector method are usually tiny values. The adjustment is quite easy if the designer feels the unimportant attributes should be still somewhat considered. One possible solution is to insert a *virtual event* into Table I, which has a tiny weight (e.g. 0.02) but requires to consider all the attributes.

Another phenomenon is that, sometimes, the difference of weights by the two methods is obvious but the correlation is good, such as the bit error rate of the interactive application in Fig. 3. This regular difference between the two methods is actually caused by the $\mathbf{EA}$ matrix. For example, interactive applications require large weights on both security and bit error rate and we suppose security is more important than bit error rate, but the $\mathbf{EA}$ matrix of TRUST in Table II cannot take the two attributes relative importance into account. One simple solution is to separate one event into two *semi-events* which correspond to the two important attributes.

By doing the above two adjustments, the obtained weights by TRUST can be quite close to the weights by the eigenvector method, but TRUST does not involve in the complicated pairwise comparisons as the eigenvector method.

VII. Conclusions

This article proposed a novel weighting method, named TRUST, which can efficiently calculate the subjective weights of various attributes based on both terminal-side and network-side subjective requirements in the network selection issue. TRUST has obvious benefits compared with the widely used eigenvector method in the literature of network selection. It can automatically calculate the subjective weights by the terminal in various scenarios and this process is very quick. We compared the weights obtained by TRUST with those by the eigenvector method, and found that the resulted weights of the two methods can be quite close to each other.

VIII. Acknowledgement

This research is supported by France Telecom R & D.

References

[1] E. Gustafsson and A. Jonsson, "Always best connected," *IEEE Wireless Commun.*, vol. 10, no. 1, pp. 49 - 55, Feb. 2003.

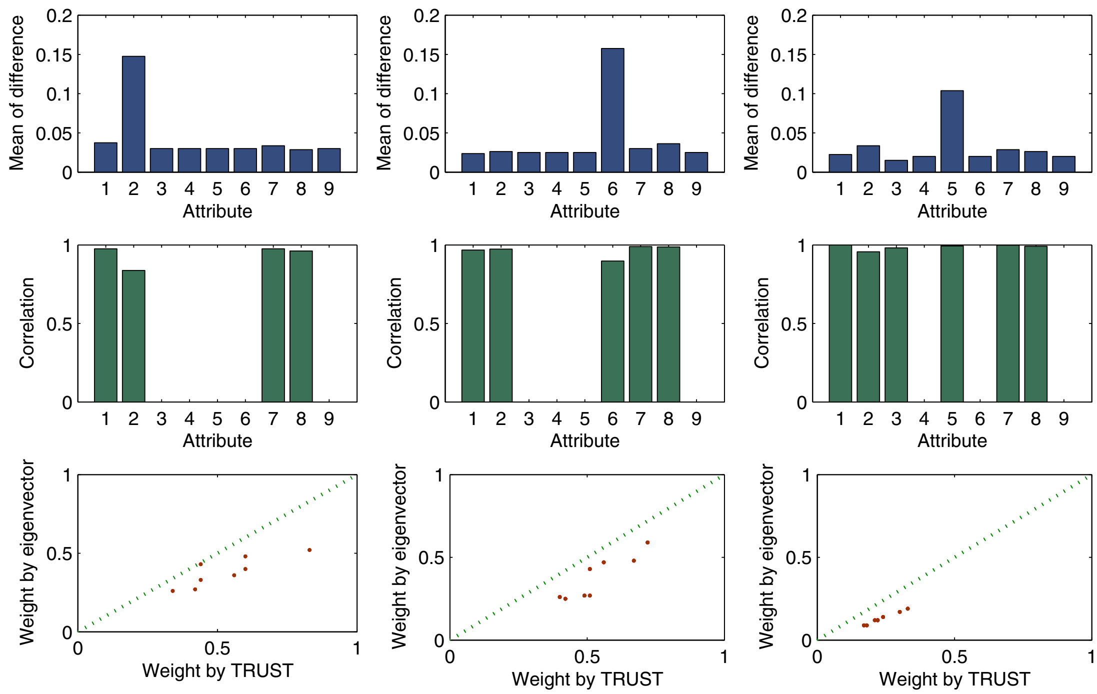

Fig. 3. Comparison results between TRUST and Eigenvector method (from left to right: Streaming, Conversational and Interactive).

TABLE I
RELATIONSHIP BETWEEN TRIGGER EVENTS AND SUBJECTIVE WEIGHTS OF ATTRIBUTES.

Events & Weights				Attributes									
Layer-1		Layer-2		PR	BD	SC	PC	BER	JT	TL	HO	SS	...
Application QoS levels	0.37	Streaming	0.12		•								
		Conversational	0.16						•				
		Interactive	0.06			•		•					
		Background	0.03										
Terminal properties	0.21	Mid speed	0.03								•		
		High speed	0.12								•		
		Mid power	0.01				•						
		Low power	0.05				•						
Customer preferences	0.13	Low price	0.05	•									
		High security	0.05			•							
		Large bandwidth	0.03		•								
Operator preference	0.07	Load balancing	0.07							•			
Extreme dynamic attributes	0.22	Traffic load	0.07							•			
		Single strength	0.07									•	
		BER	0.03					•					
		Jitter	0.05						•				

[2] E. Stenens-Navarro and V. W. S. Wong, "Comparison between vertical handoff decision algorithms for heterogeneous wireless networks," in *proc. IEEE VTC 2006-Spring*, pp. 947 - 951.
[3] M. Kassar, B. Kervella and G. Pujolle, "An overview of vertical handover decision strategies in heterogeneous wireless networks," *Comput. Commun. (2008)*, Doi:10. 1016/j.comcom.2008.01.044.
[4] W. Zhang, "Handover decision using fuzzy MADM in heterogeneous wireless networks," in *proc. IEEE WCNC 2004*, pp. 653 - 658.
[5] Q. Y. Song and A. Jamalipour, "Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques," *IEEE Wireless Commun.*, vol. 12, no. 3, pp. 42 - 48, June 2005.
[6] F. Bari and V. C. M. Leung, "Automated network selection in a heterogeneous wireless network environment," *IEEE Network*, vol. 21, no. 1, pp. 34 - 40, Jan. - Feb. 2007.
[7] Ching-Lai Hwang and Kwangsun Yoon, "Multiple attribute decision making methods and applications," *Springer-Verlag*, 1981.
[8] Weiwei Jiang, Hongyan Cui, Qiangjun Yan, Xiaojuan Wang and Jianya

TABLE II
COMPARISONS BETWEEN TRUST AND EIGENVECTOR METHODS IN EXTENSIVE SCENARIOS.

QoS level	Terminal property	Customer preference	Dynamic attribute	Weighting method	PR	BD	SC	PC	BER	JT	TR	HO	SS
Streaming	Mid Speed	Low PR	Some networks	TRUST	0.19	0.44					0.26	0.11	
			with high traffic	Eigenvector	0.17	0.43	0.02	0.02	0.02	0.02	0.20	0.10	0.02
			All networks	TRUST	0.25	0.60						0.15	
			with low traffic	Eigenvector	0.20	0.48	0.03	0.03	0.03	0.03	0.03	0.14	0.03
		Large BD	Some networks	TRUST		0.60					0.28	0.12	
			with high traffic	Eigenvector	0.03	0.40	0.03	0.03	0.03	0.03	0.28	0.14	0.03
			All networks	TRUST		0.83						0.17	
			with high traffic	Eigenvector	0.04	0.52	0.04	0.04	0.04	0.04	0.04	0.20	0.04
	High speed	Low PR	Some networks	TRUST	0.14	0.34					0.18	0.34	
			with high traffic	Eigenvector	0.10	0.26	0.02	0.02	0.02	0.02	0.16	0.38	0.02
			All networks	TRUST	0.16	0.42						0.42	
			with low traffic	Eigenvector	0.11	0.27	0.03	0.03	0.03	0.03	0.03	0.44	0.03
		Large BD	Some networks	TRUST		0.44					0.21	0.35	
			with high traffic	Eigenvector	0.03	0.33	0.03	0.03	0.03	0.03	0.16	0.33	0.03
			All networks	TRUST		0.56						0.44	
			with low traffic	Eigenvector	0.04	0.36	0.04	0.04	0.04	0.04	0.04	0.36	0.04
Conversational	Mid Speed	Low PR	Some networks	TRUST	0.16					0.51	0.23	0.10	
			with high traffic	Eigenvector	0.17	0.02	0.02	0.02	0.02	0.43	0.20	0.10	0.02
			All networks	TRUST	0.21					0.67		0.12	
			with low traffic	Eigenvector	0.20	0.03	0.03	0.03	0.03	0.48	0.03	0.14	0.03
		Large BD	Some networks	TRUST		0.10				0.56	0.24	0.10	
			with high traffic	Eigenvector	0.02	0.11	0.02	0.02	0.02	0.47	0.21	0.11	0.02
			All networks	TRUST		0.14				0.72		0.14	
			with high traffic	Eigenvector	0.03	0.18	0.03	0.03	0.03	0.59	0.03	0.15	0.03
	High speed	Low PR	Some networks	TRUST	0.13					0.40	0.17	0.30	
			with high traffic	Eigenvector	0.10	0.02	0.02	0.02	0.02	0.26	0.16	0.38	0.02
			All networks	TRUST	0.15					0.49		0.36	
			with low traffic	Eigenvector	0.11	0.03	0.03	0.03	0.03	0.27	0.03	0.44	0.03
		Large BD	Some networks	TRUST		0.08				0.42	0.18	0.32	
			with high traffic	Eigenvector	0.02	0.13	0.02	0.02	0.02	0.25	0.13	0.39	0.02
			All networks	TRUST		0.10				0.51		0.39	
			with low traffic	Eigenvector	0.03	0.11	0.03	0.03	0.03	0.27	0.03	0.41	0.03
Interactive	Mid Speed	Low PR	Some networks	TRUST	0.19		0.22		0.22		0.26	0.11	
			with high traffic	Eigenvector	0.22	0.02	0.22	0.02	0.12	0.02	0.22	0.14	0.02
			All networks	TRUST	0.25		0.30		0.30			0.15	
			with low traffic	Eigenvector	0.28	0.02	0.28	0.02	0.17	0.02	0.02	0.17	0.02
		Large BD	Some networks	TRUST		0.12	0.24		0.24		0.28	0.12	
			with high traffic	Eigenvector	0.02	0.14	0.24	0.02	0.14	0.02	0.24	0.16	0.02
			All networks	TRUST		0.17	0.33		0.33			0.17	
			with high traffic	Eigenvector	0.02	0.19	0.31	0.02	0.19	0.02	0.02	0.21	0.02
	High speed	Low PR	Some networks	TRUST	0.14		0.17		0.17		0.19	0.33	
			with high traffic	Eigenvector	0.16	0.02	0.16	0.02	0.09	0.02	0.16	0.35	0.02
			All networks	TRUST	0.17		0.21		0.21			0.41	
			with low traffic	Eigenvector	0.19	0.02	0.19	0.02	0.12	0.02	0.02	0.40	0.02
		Large BD	Some networks	TRUST		0.09	0.18		0.18		0.20	0.35	
			with high traffic	Eigenvector	0.02	0.16	0.16	0.02	0.09	0.02	0.16	0.35	0.02
			All networks	TRUST		0.11	0.22		0.22			0.45	
			with low traffic	Eigenvector	0.02	0.19	0.19	0.02	0.12	0.02	0.02	0.40	0.02

Chen, "A novel application-oriented dynamic network selection in an integrated UMTS and WiMAX environment," in *proc. ChinaCom 2008*, pp. 158 - 161.

[9] B. Bakmaz, Z. Bojkovic and M. Bakmaz, "Network selection algorithm for heterogeneous wireless environment," in *proc. IEEE PIMRC 2007*, pp. 1 - 4.