Science in China Series F: Information Sciences

www.scichina.com
info.scichina.com
www.springerlink.com

Analysis and case study on multi-dimensional scalability of the Internet architecture

XU Ke[†], XU MingWei, LI Qi & LIN Song

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

This paper presents the definition of multi-dimensional scalability of the Internet architecture, and puts forward a mathematical method to evaluate Internet scalability based on a variety of constraints. Then, the method is employed to study the Internet scalability problem in performance, scale and service scalability. Based on the examples, theoretical analysis and experimental simulation are conducted to address the scalability issue. The results show that the proposed definition and evaluation method of multi-dimensional Internet scalability can effectively evaluate the scalability of the Internet in every aspect, thus providing rational suggestions and methods for evaluation of the next generation Internet architecture.

Internet architecture, scalability, performance scalability, scale scalability, service scalability

1 Introduction

The Internet has already become one of the most important infrastructures supporting economic development, social progress, and technological innovation of modern society. With the growing popularity of the Internet as well as the continuous emergence of heterogeneous environments, pervasive computing, ubiquitous networking, mobile access and mass-media, etc., people's demands on scale, function, performance and so on are growing fast. In order to meet these demands, researchers have begun to pay attention to an important issue of Internet: how to design a new Internet architecture to better adapt to the development demand of future Internet? In particular, how to create metrics, such as functions, performance, cost and other aspects to provide good scalability to these continuously changing factors? One of the most central problems is the multi-dimensional scalability of Internet architecture, and the issue of how to analyze and evaluate the Internet scalability is still open.

In order to investigate and analyze the Internet architecture scalability, above all, we need to

Received June 3, 2008; accepted August 23, 2008
doi: 10.1007/s11432-008-0151-9
[†]Corresponding author (email: xuke@tsinghua.edu.cn)
Supported by the National Basic Research Program of China (973 Program) (Grant No. 2003CB314801), the National High-Tech Research & Development Program of China (863 Program) (Grant Nos. 2008AA01A326, 2006AA01Z205, 2006AA01Z209), and the National Natural Science Foundation of China (Grant No. 90704001)

understand the concept of scalability. Neuman[1] gave the definition of system scalability as follows: scalability means the increase of users, and system resources will not lead to obvious decline in system performance or increase in management complexity. When the system refers to network architecture, the architecture of a specific network that provides good scalability also means that the increase of network scale (mainly an increase in the number of nodes and their links) will not lead to the obvious decline of system performance.

There are mainly two evaluation methods in network scalability research. One is theoretical analysis, and the other is experimental simulation. The latter is comparatively simple and with it one can easily get experimental results, but this method is hard to be improved according to the experimental result or to be used to find the intrinsic problems. With a method other than the simulation method, we can find the essential features of the system through theoretical analysis and thus improve and optimize the design of architecture. However, due to the features of networks, such as diversity, complexity and stability, theoretical analysis is much more difficult to be carried out. In addition, different systems generally have different analytical methods. There is no general approach applicable to all theoretical models or analytical methods.

To solve the above problem, this paper addresses the scalability issues of Internet architecture and brings forth the definition of scalability as well as the mathematic model and analytical method. We present the definitions of single-constraint one-dimensional scalability, single-objective multi-dimensional scalability, and multi-objective multi-dimensional scalability. Aiming at different features of constraint conditions, different mathematic descriptions and analyses are given. Based on the real architecture, analyses and experimental simulations are presented. The mathematic model and analytical method proposed in this paper provide an analytical and comparative approach for different network architectures, thus laying a theoretical foundation for the scalability of Internet architecture.

The remainder of this paper is as follows: Section 2 summarizes the analysis and relevant researches on network scalability. Section 3 puts forward the relevant definitions and classifications of network scalability. Section 4 introduces a mathematic description of Internet scalability. Section 5 analyzes and evaluates the research on three aspects of Internet architecture, namely performance scalability, scale scalability, service scalability, and illustrates the researching significance of evaluation method of Internet multi-dimensional scalability for evaluating the Internet scalability through examples. Finally, conclusions and future work are drawn in section 6.

2 Related work

Currently, the simulation analysis and theoretical study of the network scalability focus on the scalability of overlay networks and ad hoc networks. In addition, many researchers analyzed and evaluated the network service system in the cost and price aspects.

2.1 Scalability evaluation of overlay network

Through the research and analysis of overlay network architecture, researchers consider that network scalability can be enhanced by revising network topology[2−6]. This conclusion shows that network topology is an important factor that can impact the network scalability. However, these studies are not conducted through quantified analysis, and a systemic investigation of the scalability evaluation is lacking. Therefore, effective evaluation and analysis for various systems could not be carried out. Grossglauser et al.[7] studied the network mobility's impact to scalability, but their work did not consider the impact of constraint factors on network scalability, such as memory capacity, link delay and so on. In fact, for real network systems, the memory capacity cannot expand infinitely, and network applications cannot tolerate the unlimited growth of the packet delay in the network.

2.2 Scalability evaluation of ad hoc networks

Several papers analyzed and evaluated the scalability and other network performance of ad hoc network routing protocol through simulations[8−10].

Although the simulation results can indicate some problems in the network architecture, the process and the results are often limited to specific applications. Thus researchers are unable to have an in-depth understanding of the constraints of protocol itself, relevant system parameters and environmental characteristics. For instance, Santivanez et al.[11] gave the definition of the network scalability by calculating the smallest network traffic load under various conditions.

Arpacioglu et al.[12] first proposed the definitions of absolute scalability, optimal scalability, as well as weak scalability of network, and made analysis and evaluation. Although the definitions are applied to ad hoc network, the basic idea can still be used in network analyses and evaluations of all kinds of networks. Yang et al.[13] concluded and summed up network types under various application needs. But the work only gave the definition of scalability in certain situations, failing to consider the network topology's own limitation to scalability.

2.3 Scalability evaluation of service networks

Billhartz et al.[14] discussed the network cost in different multicast networks. The cost model they proposed mainly concentrates on network resource demand and data transmission cost. The proposed cost model can be used to design and evaluate multi-service data transmission system. However, they still have many shortcomings. For instance, the model only pays attention to the Internet service providers, without considering the user's satisfaction degree in using network service. This model cannot effectively assess the network service system.

Chuang et al.[15] proposed a network cost based multi-cast network scalability. They found that the cost of a multicast tree is changing around the power that takes the size of multicast tree as base and 0.8 as power, and the cost of multicast can be decided according to the size of multicast group. Similarly, their work as well as Billhartz's did not consider user's satisfaction degree on different services in different network architectures. Furthermore, they did not analyze different network services deployed in different network topologies, ignoring the fact that network topology is also an important factor in evaluating network performance.

Moreover, Chalmers et al.[16] compared the efficiency benefit between unicast and multicast. Other relevant researches put forward the network cost for multi-ISP provider in the Internet[17−20], as well as the complexity of network cost model.

3 Concept and classification of Internet scalability

3.1 Concept of scalability

Before introducing the concept of scalability, we need to introduce the concept of constraint condition, the interval under constraint, evaluation metric, etc. We mainly check the factors which we can control in network architectures, such as message transmission rate of end systems, link bandwidth, the Internet scale, topology, and so on. Constraint condition can be divided into continuous constraint condition and discrete constraint condition according to whether the value is continuous or not. For example, message transmission rate and link bandwidth are continuous constraint conditions, while Internet scale and topology are discrete constraint conditions.

Definition 1. The interval under constraint refers to the changing bound of constraint condition.

An interval under constraint has direct impact on the performance of scalability. We may draw the conclusion that the scalability is completely different in different intervals. Since constraint condition can be divided into continuous and discrete constraint conditions, interval under constraint can also be divided into continuous interval and discrete interval.

Definition 2. Evaluation metric refers to certain good or bad character possessed by Internet architecture we test, such as throughput, stability, network cost and so on.

Evaluation metric will change with the changing constraint condition. For example, throughput may increase, decrease or stay static with the changing of end user's sending rate. With a certain bound (bound of intervals under constraint), the throughputs of two architectures have different integrated good or bad character; that is, with a certain bound, A may be good, while in another bound, B may be good.

Definition 3. The scalability of Internet architecture refers to the character contained in the evaluating metric of Internet architecture with the change in network constraint condition in a certain bound.

The scalability of Internet architecture has four essential factors:

(1) Constraint condition. Constraint condition is the changing factor of scalability. It decides whether the scalability considers all the factors comprehensively, and it is the reflection of application bound.

(2) Interval under constraint. It decides the effective bound of scalability. The good or bad character of scalability in network architecture is only effective in constraint interval, but meaningless if it exceeds the interval. For example, if the scalability performance of two architectures in a five-year period is: A is good; while B is bad, we cannot judge which is good or bad in ten years.

(3) Evaluating metric. Evaluating metric is the target for examining the Internet architecture. And it is also one of the factors reflecting whether the scalability is comprehensive or not. The less evaluating metrics, the less comprehensive is the architecture scalability. Conversely, the more the evaluation metrics, the more comprehensive it can reflect the architecture scalability.

(4) Change law. Change law is a law for evaluating metric changes with the constraint condition. It is a direct reflection of scalability. For example, the scalability of a system whose performance linearly increases with the scale is surely better than those that logarithmically increased with the scale.

3.2 The classification of scalability

According to different purposes of our study, scalability of the architecture can have three different meanings.

3.2.1 Static scalability. Static scalability refers to the good or bad character of an architecture represented by the current value of evaluating metric. For example, between two different network architectures, if one has a larger throughput, while the other only has a small fraction of throughput, then the static scalability of the former is superior to that of the latter in the evaluating metric of throughput.

3.2.2 Dynamic scalability. Dynamic scalability refers to the fact that at a certain time (the time is not the real time, but the current location of immediate value of constraint condition), the evaluating metric changes with the changing constraint condition. For example, of the overall throughput of three architectures, if one increases linearly with end user's rate, one increases logarithmically, and another decreases reciprocally, then their dynamic scalabilities are decreasing in turn on the metric of throughput.

3.2.3 Cumulative scalability. Cumulative scalability refers to the cumulative results of evaluating function in constraint condition. It is the cumulative character of constraint condition in a certain bound, independent of good or bad character of a particular moment.

Here, we cannot consider certain scalability solely, but should integrate three scalabilities into one to evaluate systems. For example, in the static scalability, sometimes A is better than B while sometimes B is better than A. But the dynamic scalability of A is always better than B, and at the same time, the cumulative scalability of A is better than B as well. In such a case, we are inclined to choose system A as the system with better scalability. Again, the static and cumulative scalabilities of system C are all worse than that of system D in constraint interval. But at the beginning of interval, system C performs badly, and becomes better gradually later, while system D keeps good performance. Although the dynamic scalability of system C is better than that of system D, we are still inclined to choose system D. When we compare two systems X and Y, regardless of integrating these three scalabilities, we deem that we can get the comparing degree of X and Y mainly from every scalable target, and get the sum or weighted sum of them to get the comprehensive evaluation. If the weighted sum is chosen, weight can be determined by discussion of decision-maker. When future development is concerned, dynamic scalability can be emphasized; when the integrated situation of constraint interval is concerned, comprehensive scalability should be emphasized.

According to the amount difference of evaluation metrics, scalability can be divided into one-dimensional scalability and multi-dimensional scalability. And the former can be divided into single-constraint one-dimensional scalability and multi-constraint one-dimensional scalability.

(1) Single-constraint one-dimensional scalability, also called single-constraint single-objective scalability, refers to scalability that an evaluation metric is possessed with the change of one constraint condition.

(2) Multi-constraint one-dimensional scalability, also called multi-constraint single-objective scalability, refers to scalability that an evaluation metric is possessed with the change of many constraint conditions.

(3) Multi-dimensional scalability, also called multi-constraint multi-objective scalability, refers to scalability achieved by comprehensively considering many evaluation metrics changing with many constraint conditions. There are different weights among evaluation metrics. The values of weights are determined by the actual objective situation or subjective will. If the values of weights are different, though constraint condition is completely the same with evaluation metric, the results of multi-dimensional scalability of architecture may be different.

4 The evaluation model of Internet scalability

In this section, we first introduce the mathematic description of the scalability of Internet architecture, and then introduce the one-dimensional and multi-dimensional scalability evaluation models respectively.

4.1 Mathematic description

For any network architecture, constraint condition set is defined as $X=\{x_1,x_2,\cdots,x_n\}$. Because what we usually test is the system character with constraint condition in constraint intervals, we set its change bound at $X\in\Phi=\{x_i'\leqslant x_i\leqslant x_i''\}$, and define the quantified evaluation metric set as $Y=\{y_1,y_2,\cdots,y_m\}$. Then any evaluation metric y_i must be the evaluation function of X, and we have

$$y_i=f_i(X)=f_i(x_1,x_2,\cdots,x_n),\ X\in\Phi. \tag{1}$$

Here x_i can be a continuous variable or a discrete variable. When x_i is continuous, we assume that f_i is a continuous and smooth function, which will be differentiated everywhere.

4.2 Single-constraint one-dimensional scalability

Single-constraint one-dimensional scalability refers to the scalability of single-objective single-constraint, whose evaluation function is

$$y_i=f_{ij}(x_j),\ x_j\in[x_{j1},x_{j2}], \tag{2}$$

where y_i is the evaluation metric, x_j is a single-constraint condition, f_{ij} is the function of evaluation metric y_i with the changing constraint condition x_j, and $[x_{j1},x_{j2}]$ is the constraint interval. For example, when we evaluate the performance of P2P VOD (video on demand) systems, if we take network transmission rate as constraint condition, and user's waiting time as evaluating metric, then it becomes a single-constraint one-dimensional scalability issue.

According to the scalability's classification on examining objective, we define three different scalabilities independently:

4.2.1 Static scalability. Static scalability refers to the good or bad performance of architecture's current evaluation metric. Therefore it can be denoted by an evaluation function:

$$S_{ij}=y_{ij}=f_{ij}(x_j),\ x_j\in[x_{j1},x_{j2}]. \tag{3}$$

4.2.2 Dynamic scalability. Dynamic scalability refers to evaluating a function's changing character at a certain moment. We define it as the first derivative of evaluation function to constraint condition,

$$D_{ij}(x_j)=\left.\frac{dy_{ij}}{dx_j}\right|_{x_j=x_{j0}}=f_{ij}'(x_j). \tag{4}$$

That is, when $x_j=x_{j0}$, the dynamic scalability of architecture is $D_{ij}(x_{j0})$. It is the definition when the constraint condition is a continuous constraint condition. Researchers may also encounter some discrete constraint conditions, such as the number of hosts. We define its dynamic scalability as follows:

$$D_{ij}(x_{j,k})=\frac{\Delta y_{ij}}{\Delta x_j}=\frac{f_{ij}(x_{j,k+1})-f_{ij}(x_{j,k})}{x_{j,k+1}-x_{j,k}},\ \ x_{jk}\neq x_j'',$$

in which, $x_{j,k}$ denotes that the constraint condition x_j takes the kth value.

From Figure 1(a), we can see that the geometric significance of single-constraint one-dimensional dynamic scalability is the slope of x_j on the constraint point x_{j0} or $x_{j,k}$.

4.2.3 Cumulative scalability. Cumulative scalability reflects the cumulative character of evaluation function in the constraint interval. We define the single-constraint one-dimensional cumulative scalability of continuous condition as

$$C_{ij}=\int_{x'_j}^{x''_j} f_{ij}(x_j)dx_j, \tag{5}$$

while the one-dimensional cumulative scalability of discrete constraint condition is defined as

$$C_{ij}=\sum_{x_{j,k}} f_{ij}(x_{j,k})(x_{j,k+1}-x_{j,k}),\ x_{jk}\neq x''_j.$$

From Figure 1(b), we can see that the geometric significance of single-constraint one-dimensional scalability is the sum of evaluation function $x_j=x'_j$ and the area surrounded by $x_j=x''_j$ and the abscissa axis.

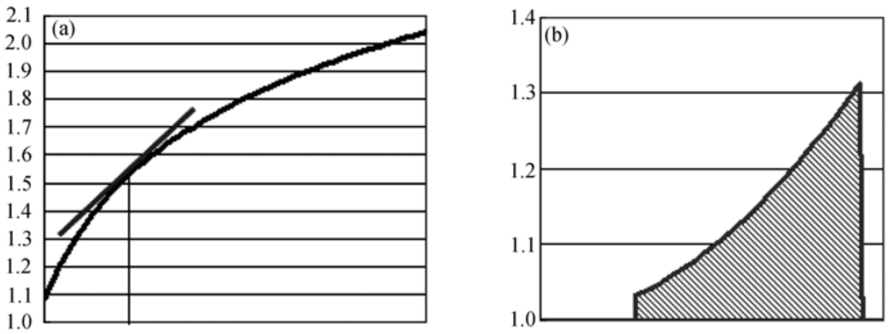

Figure 1 The geometry significance of single-constraint one-dimensional scalability. (a) Dynamic scalability; (b) cumulative scalability.

4.3 Multi-constraint one-dimensional scalability

The scalability of multi-constraint one-dimensional refers to the single-objective multi-constraint scalability, whose evaluation function is

$$y_i=f_i(X)=f_i(x_1,x_2,\cdots,x_n),\ X\in\Phi, \tag{6}$$

in which, y_i is the evaluation metric, X is the vector of multi-constraint condition, $x_1,x_2,\cdots,x_n$ are each of the constraint conditions, f_i is the function that evaluation metric y_i changes with constraint condition X, and Φ is the constraint intervals. In the above example of evaluating the performance of P2P VOD systems, if we not only consider the impact of user access rate to the evaluation metric of user's waiting time, but also take the three constraint conditions of user's access rate, delay and delay jitter into consideration, we need a multi-constraint one-dimensional scalability model to study this issue.

According to scalability's classification on evaluation metric, we define three types of scalabilities respectively.

4.3.1 Static scalability. Static scalability refers to the good or bad character of system's current evaluation metric. Therefore it can be denoted by an evaluation function

$$S_i=y_i=f_i(X),\ X\in\Phi. \tag{7}$$

4.3.2 Dynamic scalability. We first consider the situation in which constraint conditions are all of continuous constraint. The definition of multi-constraint dynamic scalability is more complicated than single-constraint scalability, for the relativity of each constraint condition should be considered.

We assume that the constraint conditions are independent; that is, any value of x_i has nothing to do with x_j. Then its dynamic scalability can be defined as

$$D_i(X_0)=\left.\frac{\partial^n y_i}{\partial x_1 \partial x_2 \cdots \partial x_n}\right|_{X=X_0}=\left.\frac{\partial^n f_i(X)}{\partial x_1 \partial x_2 \cdots \partial x_n}\right|_{X=X_0}. \tag{8}$$

On the other hand, we assume that all x_i are related with each other, and are both function of another variable t. Then we have

$$x_j=z_j(t),$$
$$y_i=f_i(X)=f_i(x_1,x_2,\ldots,x_n)=f_i(z_1(t),\ldots,z_n(t))=f_{it}(t).$$

In this case, dynamic scalability can be simplified into single-constraint one-dimensional scalability

$$D_i(X_0)=\left.\frac{df_{it}}{dt}\right|_{t=t_0}=f_{it}'(t_0),$$

of which

$$X_0=(x_{10},x_{20},\ldots,x_{n0})=(z_1(t_0),z_2(t_0),\ldots,z_n(t_0)).$$

Generally, among the constraint conditions, if the first $t(1)$ ones are related with t_1, next $t(2)$ ones are related with t_2, …, $t(l)$ ones are related with t_l, and $r-q+1$ ones have nothing to do with each other lastly, then let $(x_{i,1},x_{i,2},\ldots,x_{i,t(i)})$ be the ones related with t_i, $x_{i,j}=z_{ij}(t_i)$, $x_q,x_{q+1},\cdots,x_r$ have nothing to do with any other x, of which, $i=1,2,\cdots,l,\ j=1,2,\cdots,t(i)$. We have

$$(r-q+1)+\sum_{i=1}^{l} t(i)=n.$$

Then, $f_i(X)$ can be written as

$$f_i(X)=f_{it}(t_1,t_2,\ldots,t_l,x_q,x_{q+1},\ldots,x_r).$$

So the dynamic scalability can be defined as

$$D_i\left(X_0\right)=\left.\frac{\partial^{(l+r-q+1)} f_{it}}{\partial t_1 \partial t_2 \cdots \partial t_l \cdot \partial x_q \cdots \partial x_r}\right|_{t_i=t_{i0},x_i=x_{i0}}=f_{it}^{(l+r-q+1)}\left(t_{10},t_{20},\cdots,t_{l0},x_{q0},\cdots,x_{r0}\right),$$

in which, when $t_i=t_{i0}\left(i=1,2,\cdots,l\right)$, $x_i=x_{i0}\left(i=q,q+1,\cdots,r\right)$, $X=X_0$.

When constraint conditions are all of discrete constraints, dynamic scalability can be defined as

$$D_i(X_0)=\frac{f_{it}(t_{1,k_1+1},t_{2,k_2+1},\cdots,t_{l,k_l+1},x_{q,k_q+1},\cdots,x_{r,k_r+1})-f_{ij}(t_{1,k_1},t_{2,k_2},\cdots,t_{l,k_l},x_{q,k_q},\cdots,x_{r,k_r})}{(t_{1,k_1+1}-x_{1,k_1})(t_{2,k_2+1}-x_{2,k_2})\cdots(x_{r,k_r+1}-x_{r,k_r})},$$

in which, when $t_i=t_{i,k_i}\left(i=1,2,\ldots,l\right)$, $x_i=x_{i,k_i}\left(i=q,q+1,\ldots,r\right)$, $X=X_0$.

When the constraint conditions contain both continuous ones and discrete ones, the method of thinking is the same with the situation where the constraint conditions are all continuous. Here we do not illustrate it again.

4.3.3 Cumulative scalability. According to the fact that the relativities of constraint condition differentiate with each other, we consider two situations. Suppose that all the constraint conditions are independent. Then the cumulative scalability can be defined as

$$C_i = \int_{X\in\Phi} f_i(X)dX = \int_{x_{11}}^{x_{12}}\int_{x_{21}}^{x_{22}}\cdots\int_{x_{n1}}^{x_{n2}} f_i(x_1,x_2,\ldots,x_n)dx_1dx_2\ldots dx_n. \quad (9)$$

From the definition we can see that, it is the volume of the cylinder surface surrounded by function $f_i(X)$ in the bound of $X\in\Phi$ (Figure 2(a)). When all the constraint conditions related with *t*, we define its cumulative scalability as

$$S_i = \int_l f(X)dl,$$

where l is the formula of curve $x_j = z_j(t)$, whose geometric significance is to carry out the first form curve integral to the curve (Figure 2(b)). Because

$$dl = \sqrt{dx_1^2 + dx_2^2 + \cdots + dx_n^2} = \sqrt{\sum_{j=1}^{n} z_j'(t)^2}\,dt,$$

we have

$$C_i = \int_{t'}^{t''} f_i(X)\sqrt{\sum_{j=1}^{n} z_j'(t)^2}\,dt.$$

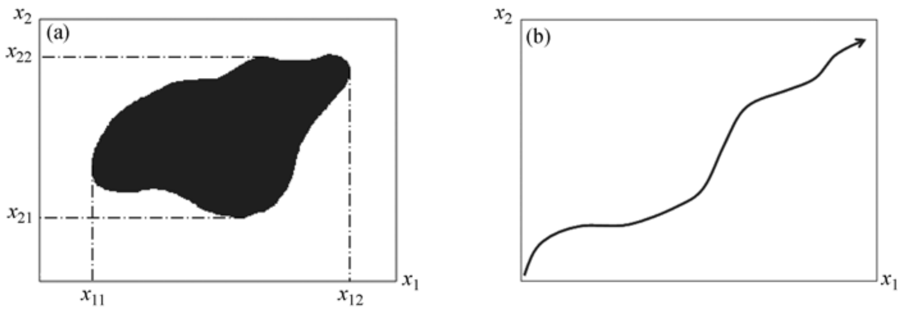

Figure 2 The geometry significance of multi-constraint one-dimensional scalability. (a) Cumulative volume; (b) cumulative curve.

When the constraint conditions are all discrete conditions and are all non-related with each other, the cumulative scalability can be defined as

$$C_i = \sum_{x_1}\sum_{x_2}\cdots\sum_{x_n} f_i(X)\Delta x_1\Delta x_2\cdots\Delta x_n.$$

When the constraint conditions are all discrete conditions and all relate with *t*, the cumulative scalability can be defined as

$$C_i = \sum_t f_i(X)\Delta t.$$

For more general case, when the constraint conditions contain both discrete conditions and constraint conditions, and have relative and independent situations between each other, the method of thinking is the same as the above analysis. Here we do not illustrate it again.

4.4 Multi-dimensional scalability

On the basis of multi-constraint one-dimensional scalability, we consider the multi-dimensional scalability, which is the scalability under many evaluation metrics. The above example of evaluating the performance of P2P VOD systems further demonstrates that we not only consider the evaluation of user waiting time, but also at the same time use two evaluation metrics of both user's waiting time and server load. Then we need multi-dimensional scalability model to study this issue.

The methods for solving multi-objective optimization problems mainly include linear weighted sum method, main-objective method, layered sequences method, ideal point method and so on[21], each having its own applicable situation. As for the issues we pose, adopting linear weighted sum solution is more workable, which introduces weighted parameters at the time of comprehensive consideration. We define the multi-dimensional evaluation function or static scalability of Internet architecture as

$$S = k_1 \cdot S_1 + k_2 \cdot S_2 + \cdots + k_m \cdot S_m, \tag{10}$$

in which, S_i represents the function of multi-constraint one-dimensional static scalability, and $k_1, k_2, \cdots, k_m$ are weight of m evaluation metrics respectively. They can be decided by objective factors, evaluator's own preference or policy-making. Different weight setting will get different results. According to the above definition, we can give dynamic scalability and cumulative scalability of them on point X_0.

5 Case analysis

We have introduced the basic idea of Internet architecture scalability as well as evaluation method of multi-dimensional scalability. Based on specific application examples, this section analyzes and evaluates performance scalability, scale scalability and service scalability aspects of Internet architecture, and illustrates how to use the proposed evaluation theory and method to evaluate the scalability of certain elements in the Internet.

5.1 Performance scalability

Performance scalability refers to the character that after the increase of network resources capacities such as links capacities, network performance and end-to-end performance can increase accordingly. Below we use scalability evaluation method to analyze and compare the performance scalability of content delivery network (CDN) and peer-to-peer (P2P). CDN and P2P all concentrate on the issue as how to distribute large amounts of content to a large number of users, so performance scalability becomes the most crucial issue. The two performance factors, link bandwidth and hard disk capacity, have great impact on CDN and P2P, so we focus on these two factors and take them as constraint conditions. Suppose that the file is divided into pieces, all with the same size g, we take expectation value of time downloading a piece with size g as evaluation metric. We will study the performance scalability of two constraint conditions as link bandwidth and hard disk capacity on the evaluation metric of the two systems. This is a multi-constraint one-dimensional scalability issue.

For convenience, we assume that network topology consists of a core network and some edge networks connected to it. Hosts are all in edge networks. Suppose that the current load demand changes slowly. Pieces that are currently requested frequently are the same as before, and therefore are more likely to be cached. We suppose that there are resource pieces of n types, and the number of current downloaders is $m_1 \geqslant m_2 \geqslant \cdots \geqslant m_n$. Let $M = \sum_{i=1}^{n} m_i$, and suppose the distribution of m_i meets $m_i = m_1 \lambda^{i-1}$. Then n can be infinitely large; therefore $M = m_1 / (1-\lambda)$.

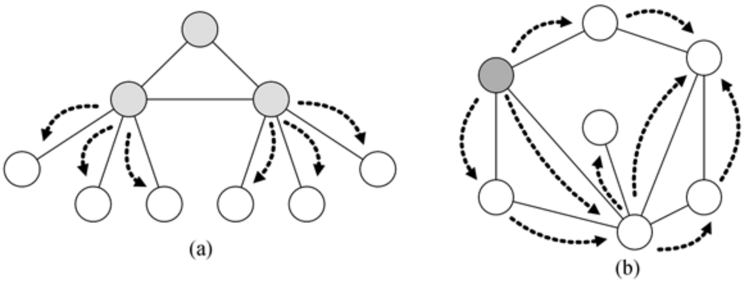

Figure 3 CDN network (a) and P2P network (b).

5.1.1 Modeling and analyzing CDNs. CDN networks (Figure 3(a)) put CDN nodes close to user, caching content requested for future demand from other users. CDN nodes are one or several common hosts in network topology. We assume that there are h CDN nodes in all, the overall hard disk capacity is G_C, and the hard disk capacity of each node is G_C/h.

Suppose that the number of users who are downloading files from a node is S_A, among which, there are S_H users getting the file directly from the CDN node, and other S_N users are requested for pieces that are not found in the cache. CDN node needs to request files from content provider for them.

Suppose that each CDN node's bandwidth to be used for downloading is y; the bandwidth of CDN node downloads a file from ICP is y_2; when CDN nodes are not congested, the bandwidth the users downloaded from CDN is z, which is also the least link bandwidth along the path between CDN and users. Considering the characteristics of CDN, and z is generally large, we can suppose that users directly share the available bandwidth y of CDN server.

As for the cache-hit S_H users, they share y with the other S_N users, so their download time is gS_A / y. As for the other S_N users, their download time is $gS_A / y + gS_N / y_2$, so the expectation of overall download time is

$$t_C = \frac{gS_A}{y} + \frac{gS_N^2}{S_A y_2} = S_A \left[\frac{g}{y} + \frac{g}{y_2} \left(1 - \frac{S_H}{S_A} \right)^2 \right].$$

Here $S_A = M / h$. The hard disk capacity of every CDN node is G_C/h, so we may have G_C/gh pieces. Considering the stability state, that is, suppose the current download demand changes slowly, the cache of CDN nodes is the G_C/gh, whose piece has the most downloaders. Therefore,

$$\frac{S_H}{S_A} = \frac{m_1 + m_2 + \cdots + m_{G_C/gh}}{m_1 + m_2 + \cdots m_n} = 1 - \lambda^{G_C/gh},$$

$$t_C = \frac{Mg}{h} \left(\frac{1}{y} + \frac{\lambda^{2G_C/gh}}{y_2} \right).$$

5.1.2 Modeling and analyzing P2P systems. For convenience, the searching process of P2P system (Figure 3(b)) is not considered. Suppose that the number of P2P user is A, which does not exceed the number of users of the whole network. We assume that the A users all have the same bandwidth x (sending bandwidth and receiving bandwidth are both x), and the same-sized hard-disk space, which is not large and allocated for P2P service, is G_P/A, where G_P is the sum of hard-disk space of all users. Users can download from two types of users at the same time: 1) users who still preserve the file in its own cache after downloading; 2) users who are still downloading the file with a larger downloaded percentage. Here we suppose that the files users downloaded are reserved in the disk for others as long as his cache is not exhausted. After the cache is exhausted, the exceeding part will be deleted.

All the A users share hard-disk space G_P. Suppose in the resource pieces of n types, the ith piece's copy number for reserving is distributed by the proportion of m_i. That is, there are $G_P m_i/M$ spaces used for reserving the downloaded ith piece, which are only to be download by others.

Suppose that the downloaded proportion of users who are downloading is distributed averagely. A user downloading the ith piece is considered. Suppose that his downloading percentage is p. Then he can download from $m_i(1-p)$ users who are downloading. Besides, from our assumptions, the space of $G_P m_i/M$ is used to reserve the downloaded ith piece, so we have $G_P m_i/M$ in all. Therefore we have totally $m_i(1-p)+G_P m_i/Mg$ users which can be downloaded. Although the rates of these pieces are different, the overall downloading rate should be equal to the expectation value of every downloading rate multiplying the number of downloading sources. In the whole download process, the average downloading rate equals $m_i(1-p)+G_P m_i/Mg$ multiplying the average rate downloaded from each place.

Below we will calculate the average rate of each download. The sending bandwidth of every user is x. In addition to this downloading request, the other $M-1$ downloading per time per user can be considered as distributing to the number of total users A averagely, so we can deem that there are $(M-1)/A+1$ users who download from here. Thereby, assume that the downloading rate is $xA/(M-1+A)$. Then the expected value of downloading the ith piece is

$$t_{pi} = g\bigg/\left[m_i\left(\frac{1}{2}+\frac{G_p}{Mg}\right)\frac{xA}{M-1+A}\right],$$

the rate at which the user downloads the ith piece is m_i/M, and $M(1-\lambda)\lambda^{n-1}=m_1\lambda^{n-1}=m_n=1$. We make summation and get the expectation value of P2P's downloading time

$$t_p=\sum_{i=1}^{n}t_{pi}\frac{m_i}{M}=\frac{g\dfrac{\ln(M(1-\lambda))}{-\ln(\lambda)}}{M\left(\dfrac{1}{2}+\dfrac{G_p}{Mg}\right)\dfrac{xA}{M-1+A}}.$$

Now, we get the evaluation function of CDN and P2P's performance scalabilities in accordance with eq. (7), under two constraint conditions, namely network bandwidth and hard-disk capacity, and use eqs. (8) and (9) to analyze them. The detailed comparing process is not illustrated here. The main results are on the dynamic scalability of network bandwidth. P2P is better than CDN. On the hard disk capacity constraint condition, the performance of CDN is better than that of P2P only when caching rate increases to such a degree that it can almost preserve all the files. This condition is hard to be met in practice. Therefore, generally speaking, in performance scalability aspect, P2P is better than CDN.

At the same time, simulation experiments are made to address these questions. The results are shown on Figure 4.

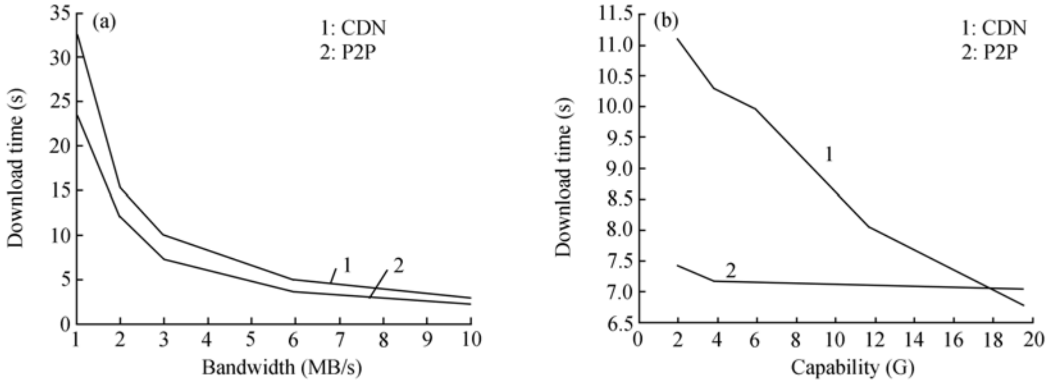

Figure 4 The performance scalability of CDN and P2P. (a) The comparison of bandwidth scalability; (b) the comparison of hard-disk capacity scalability.

We may see that the results are in accordance with the analytical result. The average downloading time of P2P in most cases is less than that of CDN. The performance scalability of P2P is good on two constraint conditions of downloading time metric. From the analyzing process, we can see that the scalability evaluation method can effectively evaluate the performance scalability of network architecture.

5.2 Scale scalability

Scale scalability refers to the network performance (such as bandwidth utilization and resource utilization of network core equipment) and the end-to-end performance can have the corresponding increase with the increasing number of network nodes and links. ROFL (routing on flat labels)[22] is a message transmission architecture taking a non-semantic label for routing. We are naturally concerned with the amount of information on message transmission efficiency and memory capacity of the router, especially the comparison between ROFL and the current architecture TCP/IP under the two performance metrics when the network scale is growing. In our analyses, network topology is simplified into a tree structure. The constraint conditions are the number of topology layers, the sub-net number of each layer, and the number of host accessed to the last layer. There are two evaluation metrics: the number of hops needed by every message transmission (which can reflect the time that transmission needs), and the space needed by every router on storing routing information.

In order to simplify the issue, we assume that the Internet topology is an L-layer K-branch tree:

As shown in Figure 5, suppose the routing transmission equipment of network constitutes an L-layer network, whose top is the first layer, and the last core network layer is the Lth layer. Each layer represents a domain, and each domain has K sub-domains. Every domain of last layer has M hosts connected to it. So the host number of the whole system is Mk^{L-1}. Suppose that the delay of message transmitted from one domain to its sub-domains or father domain is D, ignoring the routing delay as well as the host accessing delay of the domain.

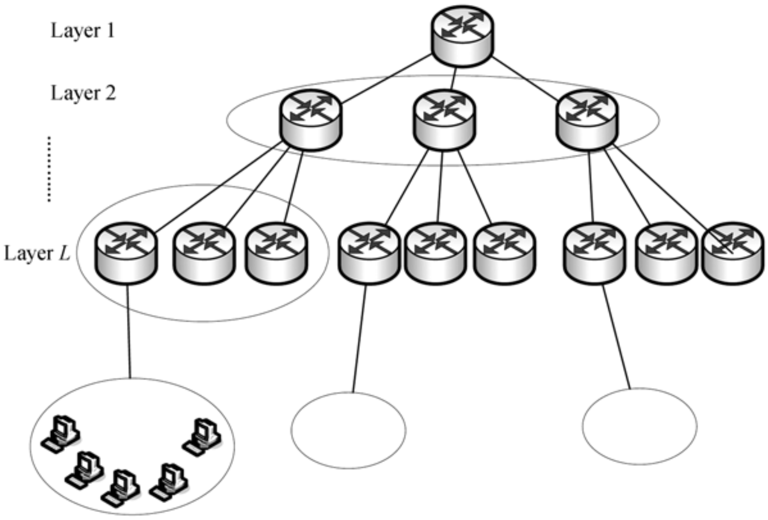

Figure 5 Network topology assumed by theory analysis.

Below we analyze the multi-constraint multi-dimensional scalability issue with two routing architectures, taking the end-to-end transmission delay and space cost needed as evaluation metrics, and taking three scale parameters of network topology as constraint conditions.

5.2.1 TCP/IP routing transmission architecture. Assume that the mathematical expected value of the transmission hop count between any two hosts is s. Then, for $1 \leqslant n \leqslant L$, the probability that any two hosts need $2n$ steps to reach equals

$$\left(\frac{1}{k}\right)^{l-n}\frac{k-1}{k}\frac{Mk^{l-1}}{Mk^{l-1}-1},$$

therefore

$$s=\sum_{n=1}^{L}2n\left(\frac{1}{k}\right)^{L-n}\frac{k-1}{k}=\left[2L-\frac{2k^{L+1}-2k}{k^{L+1}(k-1)}\right]\frac{Mk^{l-1}}{Mk^{l-1}-1}.$$

Considering $k^{L+1}(k-1)$ and $k^{L-1}M$ are both much greater than 1, we can conclude that

$$s\approx 2L-\frac{2}{k-1}. \tag{11}$$

Therefore, the time evaluation function of TCP/IP transmission architecture is

$$y_{t1}=1\bigg/\left(2L-\frac{2}{k-1}\right).$$

(Here as the shorter time means better performance, we take the reciprocal of (11)).

Because all the routers need to preserve the routing table, we define the reciprocal of the length of routing table item of every machine as space evaluation function. Suppose that IP address can be merged very well, and every domain can be denoted by a routing table item. In the network topology we consider, for every domain has k sub-domains and 1 father domain, its space evaluation function is $y_{s1}=1/(k+1)$.

In this way, we can use eq. (10) to linearly-weighted sum these two evaluation metrics, and get the scalability evaluation function of TCP/IP routing transmission architecture

$$S_1 = \alpha y_{t1} + \beta y_{s1} = \frac{\alpha}{2L - \frac{2}{k-1}} + \frac{\beta}{k+1},$$

of which, α and β are two weight parameters.

5.2.2 The analysis of ROFL routing transmission. We assume that the mathematical expected value of transmission hops between any two hosts is s, and consider the nearest ancestor node shared by these two hosts. Let the distance between host and the node be n, $1 \leqslant n \leqslant L$. Then due to the character of ROFL, the routing between these two hosts cannot exceed the shared ancestor node. However due to the character of DHT and the result of layered DHT, the execrated value of packet transmission time is $\log_2(Mk^{n-1})+1$, and will not exceed $\log_2(Mk^{n-1})+n$. And because the number of transmission hops will not exceed $2n$, as above, the probability of node distance being n is

$$\left(\frac{1}{k}\right)^{l-n} \frac{k-1}{k} \frac{Mk^{l-1}}{Mk^{l-1}-1}.$$

Considering both $k^{L+1}(k-1)$ and $k^{L-1}M$ are much greater than 1, we can get an upper bound.

$$s_0 \leqslant \log_2\left(\frac{M}{k}\right)\left(2L - \frac{2}{k-1}\right) + 2\left(\log_2 k + 1\right)\left(L^2 - \frac{2L-1}{k-1} + \frac{2}{(k-1)^2}\right). \tag{12}$$

Our estimation for the upper bound is greater than the real result. We adopt a fitting method to estimate the average delay of ROFL. According to the analytical and experimental result of (12), we can assume $s_0 = f_1 \log_2 k + f_2 \log_2 M + f_3$. Through fitting, we can get

$$s_0 = (0.476L^2 + 0.422L - 0.880)\log_2 k + (0.0885L + 0.846)\log_2 M + (0.604L - 1.088). \tag{13}$$

So the time evaluation function of ROFL can be denoted by $y_{t2} = 1/(f_1 \log_2 k + f_2 \log_2 M + f_3)$, where $f_1 = 0.476L^2 + 0.422L - 0.880$, $f_2 = 0.0885L + 0.846$, $f_3 = 0.604L - 1.088$.

Similarly, as for the ROFL transmission architecture, every router needs to preserve its subsequence of each layer for every host, then $y_{s2} = 1/ML$. Similarly, using eq. (10) we can get the scalability evaluation function of ROFL routing architecture.

$$S_2 = \alpha y_{t2} + \beta y_{s2} = \frac{\alpha}{f_1 \log_2 k + f_2 \log_2 M + f_3} + \frac{\beta}{ML}.$$

The two weight parameters of α and β can be selected according to the practical need. But actually, from the analyses of (11) and (13), we may know that the transmission scale scalability of TCP/IP's is better than that of ROFL's. Because $ML \geqslant M > k+1$, the scalability on routing reserving space of TCP/IP's is better than that of ROFL's. So no matter how the parameters α and β are selected, the static scale scalability of TCP/IP's is better than that of ROFL's. By eqs. (8) and (9), the TCP/IP's dynamic scalabilities on k, M, and L are better than that of ROFL's, so is the cumulative scalability.

Figure 6 gives the experimental comparison result of the two architectures' on the number of transmission hops. We can see that the number of transmission hops of TCP/IP's (line 1) is much less than that of ROFL's (line 2).

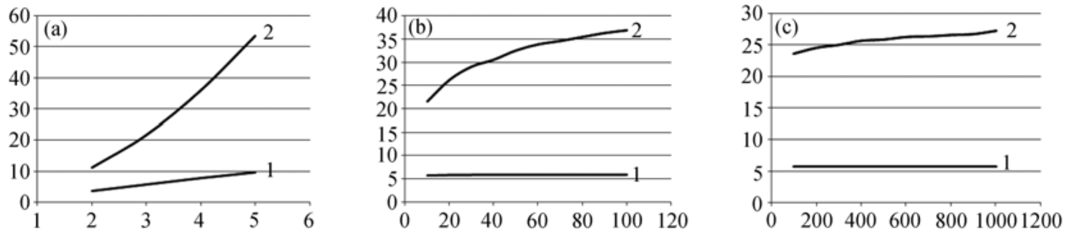

Figure 6 Comparison of average hop count between TCP/IP and ROFL. (a) Hop count change on L; (b) Hop count change on K; (c) Hop count change on M. 1, TCP/IP; 2, ROFL.

According to the theory analysis of scalability and the experimental result shown in Figure 6, we can reach the conclusion that the scalability of TCP/IP's is much better than that of ROFL's in the two aspects of transmission delay and consumed hardware reserving space.

5.3 Service scalability

This section uses scalability theory analysis to evaluate service scalabilities of different service network architectures. Firstly, we analyze the scalability of single service model with a tree architecture, circle architecture, and star architecture, and find that tree architecture possesses the optimal service scalability according to our analyses. Secondly, we analyze the service scalability under multi-service model, and find that under tree architecture, service model can acquire better service utility than the multi-service model.

We try to evaluate service scalability in different types of networks. Because one-to-one (corresponding uncast application) and one-to-many (corresponding multicast applications) are two typical service models of current networks, in this paper, we focus on the analysis of two types of communication models. In addition, we consider different network utilities of different communication models in different network architectures. We deploy service in different network architectures, such as the tree architecture (Figure 7(a)), circle architecture (Figure 7(b)), and star architecture (Figure 7(c)).

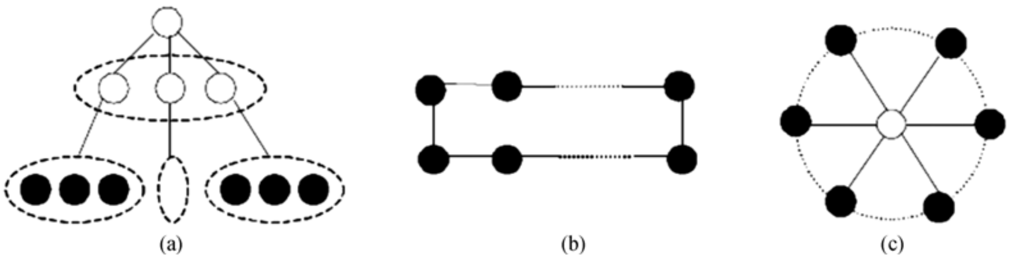

Figure 7 Different network architectures. (a) Tree architecture; (b) circle architecture; (c) star architecture.

The network service utility refers to network income acquired after certain network being deployed, the evaluation function is

$$\sum_{i=1}^{n} U_i = \sum_{i=1}^{n} \left(F(s_i) - \alpha \sum_{j=1}^{N} R_j(s_i) - \beta \sum_{l=1}^{C} L_l(s_i) \right), \tag{14}$$

where $F(s_i)$ denotes income acquired by users using service, $R_j(s_i)$ and $L_l(s_i)$ respectively refer to the processing cost of middle node after network being deployed and link cost (the total sum of these two prices are the overall network processing cost after the network service being deployed). The three functions use the service throughput s_i as the parameter. α and β are respectively used to map the overall processing cost to average processing cost of every node and the route cost to average route cost.

We first evaluate the service utility of the tree architecture. Suppose that users access service with an average distributed probability, and only visit one service in multi-service network. In the tree architecture, we study the complete tree with L-height and M branches. The tree contains $N_n = (m^L - 1)/(m-1)$ non-end nodes, $N_t = (m^{L+1} - 1)/(m-1)$ end user nodes, $L = \log_m(N_t(m-1)+1)-1$. And then we can sum up the service utility of tree architecture by using one-to-one and one-to-many model. It should be noted that in one-to-many model, non-end node will have extra maintenance cost. According to the conclusion of ref. [23], we know that the maintenance cost of one-to-many model will not exceed the 5% of overall communication cost. In our analytical model, we need to consider the extra maintenance cost of this part, and take the maximal value of maintenance cost as 0.05. From the above analysis we can find that the service utility of tree architecture can be calculated as follows. (The calculation of one-to-one and one-to-many communication models are shown as eqs. (15) and (16)).

$$\sum_{i=1}^{n} U_i = \frac{m^L - 1}{m-1} - \frac{\sum_{i=1}^{L-1}\left(2Lm^L - (m+1)\dfrac{m^L-1}{m-1}\right)}{\dfrac{m^L-1}{m-1}} - \frac{\left[\sum_{i=1}^{L-1} 2i(m-1)m^{i-1}\right] + mL}{\dfrac{m^L-1}{m-1}}, \quad (15)$$

$$\sum_{i=1}^{n} U_i = \frac{m^L - 1}{m-1} - 1.05 - \frac{\left[\sum_{i=1}^{L-1} 2i(m-1)m^{i-1}\right] + mL}{\dfrac{m^L-1}{m-1}}. \quad (16)$$

In the same way, we can also calculate the service utility in circle architecture and star architecture. Because the calculation of these utilities is simpler than that of the trees, we will not list them one by one. Below we show that one network includes 27 terminal nodes, including 1 service node, 26 client-end nodes. The service model will be a complete 3-branch tree whose height is 2 in the tree architecture. With eqs. (14) and (15), we can get the network cost and service utility of the tree architecture (Figure 8(a) and (b)). Similarly, from (14), we can get the network processing cost and service utility of circle-architecture's as well as star architecture's in different service models. In addition, in order to compare the service utilities under different service models, we compare and calculate the service utility proportion of different architectures', that is, to calculate the average network service utility acquired by each user (Figure 8(c)).

As shown in Figure 8(a), the network cost is the lowest in tree architecture, and is the highest in the star architecture. So we can make calculation and find that the service utility in tree architecture is the best, and is the worst in the star architecture (Figure 8(b)). What needs to be pointed out is that the circle architecture possesses a similar service utility value to that of the tree architecture, but the utility change bound of which is comparatively big with the increase of service rate. So the circle architecture cannot acquire the same service utility as the tree architecture. Figure 8(c) shows the network service rate in different network architectures. From the figure we can see that the worst service rate of the tree architecture is about 60%, and the best service rate is 73%. So the average service rate of the tree architecture is 73%, but the worst service utility rate of two architectures are both less than 60%.

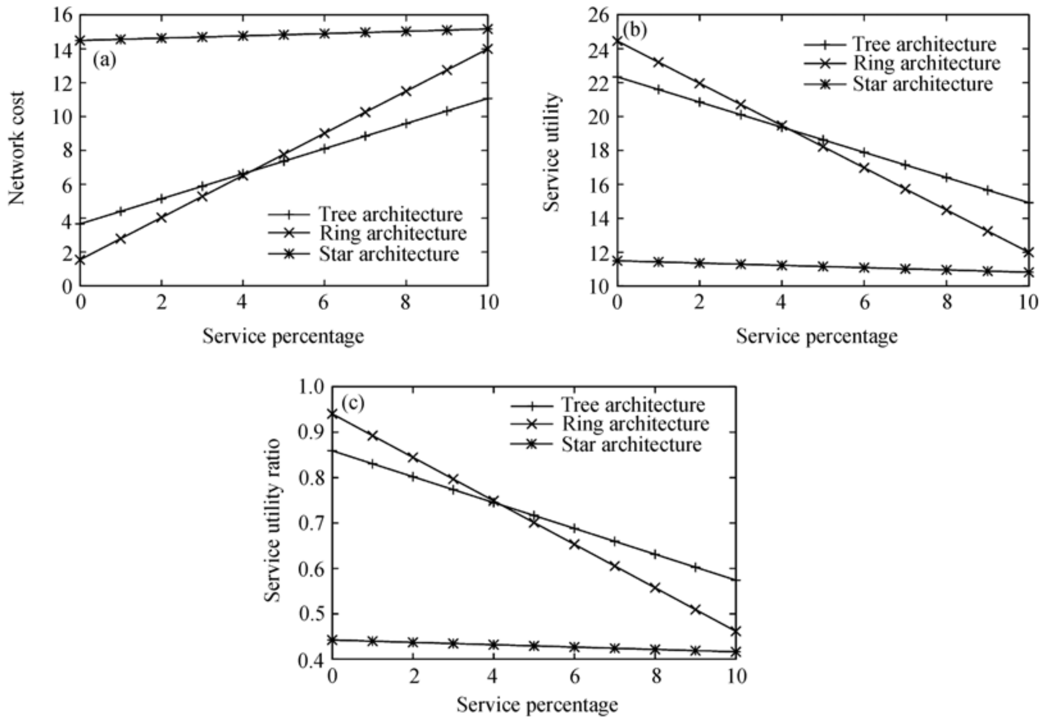

Figure 8 The service utility of one-to-one service model. (a) Network cost in different network architectures; (b) service utility in different network architectures; (c) service utility ratio in different network architectures.

5.4 Case summary

Among the three cases mentioned above, case 1 discusses the performance scalabilities of content distribution architectures of CDN and P2P, which is the multi-constraint one-dimensional scalability issue taking download time as the evaluation metric on the two constraint conditions of network bandwidth and hard disk capacity. In the analyzing process of the cases, we mainly adopt the analytical method of scalability described in section 3. After establishing the relation of evaluation function and constraint condition, we use the mathematic model established for multi-constraint one-dimensional scalability issue in section 4 to analyze the static scalability, dynamic scalability and cumulative scalability.

Case 2 discusses the scale scalabilities of two transmission mechanisms of TCP/IP and ROFL, which is multi-constraint multi-dimensional scalability issue on three constraint conditions of network layer number, sub-domain number on each layer, and the number of last layer user network hosts, taking routing transmission time and space that router needs as evaluation metrics. The difference from the first case is that we mainly adopt the mathematic model established for multi-constraint multi-dimensional scalability issue, solving the multi-objective issue by linearly weighted summing the two evaluation metrics.

Case 3 discusses the service scalability of different service modes in different architectures, taking network topology and service deployment proportion as constraint conditions, and taking network service utility as evaluation metric. It is similar to the first case in terms of adopting scalability theory.

We deem that scalability analyzing theory gives us instructions in the following aspects: in issue searching aspect, its classification and analyses on performance scalability, scale scalability, and service scalability. Other important network scalabilities can instruct us to find problems to be solved in all kinds of scalabilities; in issue confirming aspect, scalability theory's definition and analyses on constraint condition, constraint section, evaluation metric and change rule. Change rule can also instruct us to confirm the four basic elements of the issue we study, in issue resolving aspect. Scalability theory adopting the easy to in-depth method establishes a suite of mathematic model, which can be used to solve the issue of multi-constraint multi-dimensional case, gives sensible scalability definition to discrete constraint condition, and analyzes relevant constraint conditions as well. These mathematical models provide important mathematic tools for solving the scalability issue.

6 Conclusion and future work

We propose the concept of multi-dimensional scalability of Internet architecture, and establish the mathematical analytical theory and evaluation method of multi-dimensional scalability in Internet architecture. As case study analysis, we investigate and evaluate the scalability issue in the three aspects of performance, scale and service. Theoretic analysis and experimental simulation are conducted by combining them with real cases in network architectures. The experiment shows that our proposed scalability analytical method has certain instruction significance.

As future work, we will further investigate multi-dimensional scalability in performance, scale and service aspects based on our analytical method; in performance scalability aspect, we will further investigate the Internet demand on performance scalability on the basis of the analytical result of the performance scalability in CDN, P2P and other architectures. In scale scalability aspect, we need to further analyze the performance of ROFL architecture, and at the same time we need to consider the mobility and other factors in the evaluation metric. In service scalability aspect, we will further investigate more accurate cost and utility functions to analyze the service scalability of complicated network architecture.

1 Neuman B C. Scale in Distributed systems. In: Casavant T, Singhal M, eds. Readings in Distributed Computing System. Los Alamitos, CA: IEEE Computer Society Press, 1994. 463－489

2 Lv Q, Ratnasamy S, Shenker S. Can Heterogeneity Make Gnutella Scalable? IPTPS 2002

3 Ratnasamy S, Francis P, Handley M, et al. A scalable content-addressable network. In: Proceedings of SIGCOMM, 2001

4 Rowstron A, Druschel P. Storage management and caching in PAST, a largescale, persistent peer-to-peer storage utility. In: Proceedings of the Eighteenth SOSP, ACM. 2001

5 Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proceedings of SIGCOMM, 2001

6 Zhao B, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. UCB Technical Report, 2001

7 Grossglauser M, Tse D. Mobility increases the capacity of ad-hoc wireless networks. In: Proceedings of IEEE Infocom'2001, Anchorage, Alaska, April 2001

8 Broch J, Maltz D, Johnson D, et al. A performance comparison of multihop wireless ad hoc network routing protocols. In: Proceedings of MOBICOM'98, Dallas, TX, October 1998

9 Park V D, Corson S. A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing. In: Proceedings of IEEE Symposium on Computers and Communications ISCC'98, Athens, Greece, June 1998

10 Perkins C E, Royer E M, Das S R, et al. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. IEEE Person Commun Mag, 2001, 8(1): 16－28
11 Santivanez C A, McDonald B, Stavrakakis I, et al. On the scalability of ad hoc routing protocols. IEEE INFOCOM'02, New York, June 2002
12 Arpacioglu O, Small T, Haas Z J. Notes on scalability of wireless ad hoc networks. Internet draft, work in progress, 2003
13 Yang L, Conner S, Guo X, et al. Common wireless ad hoc network usage scenarios. Internet draft, work in progress, 2003
14 Billhartz T, Cain J, Farrey-Goudreau E, et al. Performance and resource cost comparisons for the CBT and PIM multicast routing protocols. IEEE J Select Areas Commun, 1997, 15(3): 304－315
15 Chuang J, Sirbu M. Pricing multicast communication: A cost-based approach. Telecommun Syst, 2001, 17(3): 281－297
16 Chalmers R, Almeroth K. Modeling the branching characteristics and efficiency gains in global multicast trees. In: IEEE INFOCOM, 2001. 449－458
17 He L, Walrand J. Pricing differentiated internet services. IEEE INFOCOM, 2005. 195－204.
18 Lee S, Jiang J, Chiu D, et al. Interaction of isps: Distributed resource allocation and revenue maximization. IEEE Trans Parallel Distrib Syst, to appear.
19 Shakkottai S, Srikant R. Economics of network pricing with multiple ISPs. In: IEEE INFOCOM, 2005. 184－194
20 He L, Walrand J. Pricing and revenue sharing strategies for internet service providers. IEEE INFOCOM, 2005. 942－951
21 Tang H W, He M F. An Introduction to Mathematic Models (in Chinese). Beijing: Higher Education Press, 2001. 84－86
22 Caesar M, Condie T, Kannan J, et al. ROFL: Routing on flat labels. SIGCOMM, 2006
23 Adler M, Rubenstein D. Pricing multicasting in more flexible network models. ACM Trans Algorithms, 2005, 1(1): 48－73