Infrastructure Arrangement For Application Virtualization Services

Chung-Ping Hung* and Paul S. Min†
Department of Electrical and Systems Engineering, Washington University in St. Louis
One Brookings Drive, St. Louis, MO 63130, USA
Email: *chung23@wustl.edu †psm@wustl.edu

***Abstract*—In this paper, we first briefly introduce the current development of application virtualization and the Internet usage trend. Implementing application virtualization technologies on mobile computing devices will bring both benefits and challenges. We propose a distributed server arrangement with the corresponding hand-off protocol to provide smooth and responsive user experiences for application virtualization on mobile devices. We also propose several quantitative approaches to estimate the performance improvement or impact based on the proposed configuration and protocol and evaluate the accuracies of these estimators by the Monte Carlo experiments.**

***Index Terms*—telecommunication and wireless networks, computer networks, information technology.**

I. Introduction

Over the years, application virtualization technologies have fallen into two major paradigms: one is creating a compatible runtime platform and deploying well managed application software to each client's device [1][2], and the other is executing application software on a well managed server while each client's device only deals with user inputs, such as keystrokes, and outputs, such as display updates from the server [3][4][5]. The former paradigm offers a more responsive user experience but requires more computational resources on clients' devices and efforts on dealing with the compatibility issues among miscellaneous hardware and software platforms. On the other hand, applying the latter paradigm is far less demanding of computational resources on clients' devices but induces higher response latency due to the geographical distance between the client and the server, which might ruin the user experience especially for heavily interactive applications.

In parallel to the advance of virtualization technologies, more and more users access the Internet through mobile devices such as smartphones rather than general purpose computers. We expect the computer usage pattern should change significantly in recent years and thus versatile Internet resources should be made available for mobile users. Therefore, it will be considered short-sighted to design an application virtualization service without concerning mobile users in the future. Mobile devices, however, have relatively limited computational resources concerning the dimension, weight, and power consumption. Furthermore, different operating systems and processors for mobile devices are still competing against each other so far, which makes deploying one application software running on every mobile device on market a tedious and costly mission. Consequently, implementing application virtualization services for mobile users based on the latter paradigm is a reasonable decision.

Assume we have decided to provide our virtual application service for mobile users based on the latter paradigm due to the compatibility and computational capability concerns. The conventional solution is setting up a server or a group of servers at a colocation center provided by an Internet service provider (ISP) and providing the application virtualization service through established Internet infrastructure. Though this configuration is very simple and straightforward, the long response latency could significantly prevent the clients from enjoying the service since every input must travel through a series of routers and bridges to the colocation center and the corresponding update has to traverse through the nodes backward. Each node along the route may induce congestion delay, and each link comprises the route induces propagation delay. Although we can invest in high-end routing instruments to reduce congestion delay, propagation delay, which is proportion to the geographical distance between endpoints, is inevitable, i.e., the *speed-of-light problem*.

To alleviate this issue, we propose an alternative configuration that geographically partitions the service area into multiple smaller service areas and each one has a smaller scale data center to provide the service locally. The proposed configurations should significantly reduce propagation delay since each server is geographically closer to its user. The proposed configuration, however, has to handle hand-off cases, i.e., mobile stations in use moving from one service area to another. Therefore, we also propose a hand-off protocol offering seamless user experience.

The proposed configuration comes with a price, such as inducing longer response latency during hand-off periods in addition to higher overall system complexity. Therefore, we also propose an average propagation delay estimation and comparison to figure out the condition where the proposed configuration can outperform the conventional one.

II. Proposed Configuration

Running application software on a remote server while creating an illusion that the client has full control of the software in hand is conceptually similar to the usage model of time-sharing mainframe computers in the 1960s [6]. Although the communication bandwidth between terminals and mainframe servers at that time was very low by modern standards, it

didn't affect the user experience thanks to the text-only display and short traverse distance. However, in recent application virtualization technologies which follow the same concept, such as Virtual Desktop Infrastructure (VDI) proposed by VMWare[7], much more complex and bloated content must be exchanged over much longer distances between clients and servers than their predecessors, especially for mobile users.

An infrastructure ready to offer mobile users application virtualization services includes base stations covering the whole service area, a core network connecting base stations and servers together, and a server hosting the services. A command sent by a mobile station has to travel over the wireless channel to the base station, go through the core network to the server, and then make some changes on the server. Should any update corresponding to the command be sent to the mobile station, the information has to travel all the way backward. In order to reduce the propagation delay generated by long transmission distances among the core network, we geographically deploy multiple servers among a wide area to serve their nearby mobile stations in the proposed configuration, instead of setting up a group of servers located at one data center serving all mobile stations.

In the proposed configuration, each server connects to several nearby base stations which form a local service group. The area covered by the base stations of the same local service group is defined as the local service area. Every base station should belong to one local service group in order to provide the service all over the wireless network's coverage area. When a user demands a virtual application program, the server of the local service group, based on VDI[7] paradigm, starts a virtual machine (VM) dedicated to the user and launches the application software on top of it. The mobile station only handles inputs and outputs that interact with the VM at the server.

As long as the mobile station stays in the same local service area, the user can enjoy using application software with low response latency. If the mobile station moves from the original local service area to a nearby one, a hand-off at the VM level, which transfers the runtime environment to the server of the next local service group, is triggered. The detail of the hand-off protocol will be proposed in the next section.

III. Hand-off Protocol

The purpose of the proposed hand-off protocol is to transfer everything required to recreate a runtime environment on a remote server without interfering with user experience. The whole application software and the underlying VM may occupy a large memory space, but since the majority of it is read-only in general, we can maintain copies of the initialized runtime environment for the application software at all servers and only the differential information needs to be transferred during the hand-off. We define this minimum information required to recreate the runtime environment as the *snapshot*.

Although we can significantly reduce the transmission data volume by only sending the snapshot, it still takes a period of time before the runtime environment on the next server is ready to resume the usage. In order to provide a seamless user experience during the data exchange between servers, the next server has to record all inputs from the mobile station, relay all inputs to the previous server, and relay all output from the previous server to the mobile station, before it takes over the runtime environment. The proposed hand-off protocol is described as below:

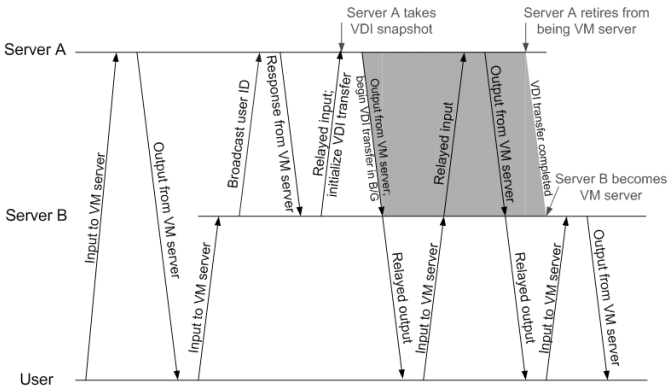

Fig. 1. Protocol timeline for a mobile station moving from Server A to Server B.

1) When a mobile station moves from Server A's to Server B's local service area and sends an input command, Server B notices a newcomer within its local service area.
2) Server B broadcasts the newcomer's identification to all geographically nearby servers.
3) Server A, which hosts the mobile station's runtime environment, i.e., its VM server, responds Server B's inquiry. Now Server B knows the newcomer's VM server is Server A.
4) Server B records and relays the user's input commands to Server A, signals Server A to transfer the runtime environment, and relays display updates from Server A to the newcomer.
5) Once Server A is signaled to transfer the runtime environment, it takes a snapshot.
6) Besides continually responding to the input commands relayed from Server B as the mobile station is still in its local service area, Server A also sends the snapshot to Server B in the background.
7) Once server B receives the complete snapshot and recreates the runtime environment from the snapshot and base data, it internally feeds the input queue, which was recorded during the transition period, to the runtime environment. Therefore, the runtime environment state on Server B is synchronous with that on Server A after the snapshot was transferred.
8) Server A completely stops serving the mobile station, the mobile station's VM server is now server B instead.

The timeline of the proposed hand-off protocol is illustrated in Figure 1.

If the mobile station turned around and reentered Server A's

local service area before the hand-off was completed, Server A can preempt the snapshot transmission and resume serving the mobile station as if the hand-off never happened. Since Server B relays all inputs to Server A while the mobile station is absent from Server A's local service area, aborting the hand-off procedure would not generate any glitch noticed by the user. This hand-off abortion mechanism can prevent unnecessary data transmission from moving VM servers back and forth if a mobile station were moving around the edge of a local service area.

IV. Performance Estimation

We define the response time as the average time interval between a user sends an input and gets an expected output update. The proposed server configuration is meant to improve the response time by reducing propagation delay along the communication route between each base station to the server which is hosting the service. Factors other than the propagation delay, such as wireless communication technologies and computational capabilities provided by servers, would affect the user experience and the quality of our service. Most of them, however, either affect both configurations equally, or can be overcome with reasonable cost.

The proposed configuration reduces the propagation delay and thus provides more responsive user experiences when uses are standing still. When a hand-off occurs, however, the user may experience longer response time waiting for the information to be exchanged between two servers before the runtime environment is successfully taken over by the new server. The smaller each local service area is, the higher occurence probability of hand-offs the user may experience. Therefore, we have to quantitatively estimate and compare the propagation delays of the conventional and the proposed server configurations.

The precise propagation delay analysis depends on a wireless service provider's core network topology and its users' moving pattern record. Instead of acquiring those field data, we focuses on the intrinsic properties of the two configurations. There are two approaches to estimate the average response time due to propagation delay; one assumes continuous service areas, the other is based on the optimal arrangement of base stations. The details are presented in the following subsections.

A. Continuous Service Area Approach

In this approach, we simplify the communication model between mobile stations and servers. Here are our assumptions:

1) The whole service area can be covered by a single server, or proximately by multiple servers, each having a regular hexagon shaped service area seamlessly tiled together as a service array.
2) A mobile station can directly communicate with the server everywhere in its (local) service area.
3) Users are uniformly distributed geographically in the beginning. Users can either move a certain distance in any direction, or stay at the same location for a while.

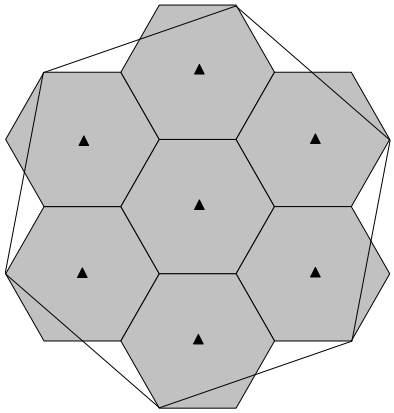

Fig. 2. Service area of 7-server configuration compares with of single-server one.

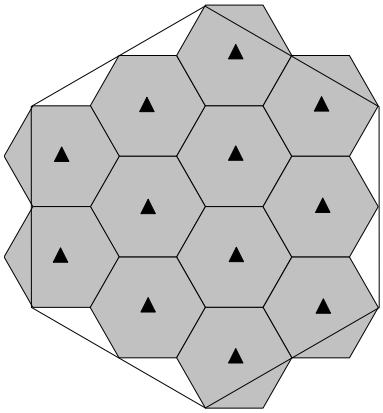

Fig. 3. Service area of 12-server configuration compares with of single-server one.

4) The propagation delay of each link is proportional to its length.
5) Each server's allocation is geographically optimized, that is, each server is located at the center of its (local) service area to reduce the average propagation delay. The traverse time in our case is defined by:

$$T_{traverse} = 2\cdot(1-P_{HO})\cdot(T_r+T_l)+2\cdot P_{HO}\cdot T_{HO} \quad (1)$$

where P_{HO} is the probability of transactions which either trigger hand-offs or occur during each hand-off, T_r is the radio propagation delay, T_l is the line propagation delay, and T_{HO} is the prolonged traverse time during each hand-off according to the proposed protocol.

To simplify the problem, we only compare the following three configurations covering the same amount of area.

A A single server covering a regular hexagon service area of edge length L.

B 7 servers, each covering a regular hexagon service area of edge length $\sqrt{7}L$, as shown in Figure 2.

C 12 servers, each covering a regular hexagon service area of edge length $\sqrt{12}L$, as shown in Figure 3.

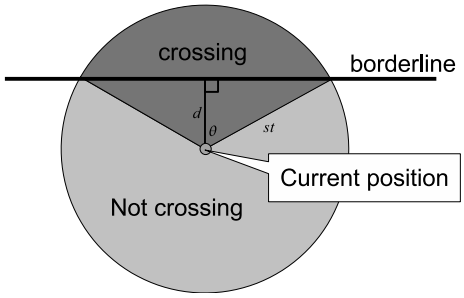

Fig. 4. For a moving user close to the borderline who can freely choose his direction, the probability of crossing the borderline in the next time instance is $\frac{2\theta}{2\pi}$

1) Average Transmission Distance: We can calculate the distance from an arbitrary point within each hexagon-shaped service area to the center of the area, where the optimal server is. Since we assume that our users' locations are uniformly distributed geographically in our service area, the average transmission distance for each user in terms of the edge length of the service area is:

$$R_{avg}(L) = \left\{ \frac{1}{3} + \frac{\ln(3)}{4} \right\} \cdot L \approx 0.60799L \tag{2}$$

2) Probability of Transactions Relevant to Hand-offs: Several factors would affect the probability of transactions relevant to hand-offs. The users can either stay in the same place or use the service on the move. Therefore, the probability of a user on the move P_M and the average moving speed $s/\Delta t$ are two factors in this subject. To make it possible to trigger a hand-off in the following time instance Δt, the user has to be on the move and within s from the current service area's borderline.

Furthermore, the probability of a user satisfying these two requirements actually crossing the borderline and thus triggers a hand-off depends on how close to the borderline he is as shown in Figure 4.

Therefore, the probability of a user who is located d from the borderline with speed $s/\Delta t$ actually crossing the borderline in the next time instance Δt is given by:

$$P_{cross}(d,s) = \begin{cases} \frac{1}{\pi} \cdot \cos^{-1}\left(\frac{d}{s}\right) \cdots 0 \le d \le s \\ 0 \qquad\qquad \cdots \text{otherwise} \end{cases} \tag{3}$$

However, since the service areas are hexagon-shaped, the borderline within the moving range is not always a straight line. As shown in Figure 5, the above equation does not apply to the users on the move located in the *singular area*, i.e., the area near vertices. It is so complex to estimate exact P_{cross} at singular area, such that we only calculate the range of P_{cross} instead.

We define $\hat{P}_{cross}(d,s)$ as the probability of a user at the sigular area crossing the borderline. Intuitively, the upper bound of $\hat{P}_{cross}(d,s)$ is $2/3$, in case of the user starting at the corner, while the lower bound is $P_{cross}(d,s)$. The singular area would not be a problem in our estimation if L is relatively larger than s.

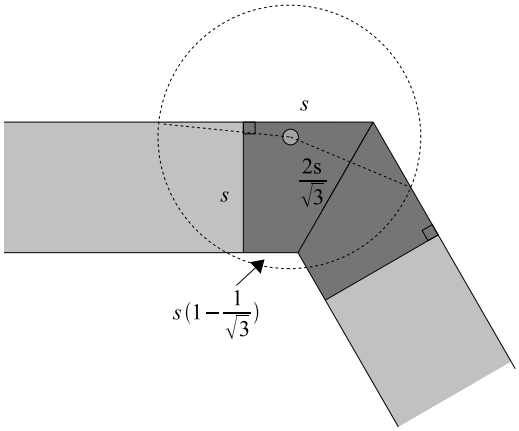

Fig. 5. Users at the singular area (dark area) have higher P_{cross}; the above equation can only apply in the normal areas (light areas).

For each hexagon-shaped service area, the probability for an arbitrary moving user crossing the borderline and triggering a hand-off is:

$$\begin{aligned} \bar{P}_{cross}(L,s,n) =& \frac{2}{3\sqrt{3}L^2} \int_0^s \{n(L-2s) \cdot P_{cross}(d,s)\}\, dd \\ &+ ns^2 \left(2 - \frac{1}{\sqrt{3}}\right) \cdot \frac{2\bar{\hat{P}}_{cross}}{3\sqrt{3}L^2} \\ =& \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} + \frac{2ns^2(2\sqrt{3}-1) \cdot \bar{\hat{P}}_{cross}}{9L^2} \end{aligned} \tag{4}$$

where $\bar{\hat{P}}_{cross}$ is the average probability of a user at the singular area crossing the borderline, and n is the number of edges which border another service area.

Since $\bar{\hat{P}}_{cross} \le 2/3$,

$$\begin{aligned} \bar{P}_{cross}(L,s,n) \le& \frac{2ns(L-2s)}{3\pi\sqrt{3}L^2} + \frac{2ns^2(2\sqrt{3}-1) \cdot 2/3}{9L^2} \\ =& \frac{2\sqrt{3}n}{9\pi} \cdot \left(\frac{s}{L}\right) \\ &+ \frac{4(2\sqrt{3}\pi - 2\pi - 3\sqrt{3})n}{27\pi} \cdot \left(\frac{s}{L}\right)^2 \end{aligned} \tag{5}$$

For $s \ll L$,

$$\bar{P}_{cross}(L,s,n) \approx \frac{2\sqrt{3}n}{9\pi} \cdot \left(\frac{s}{L}\right) \tag{6}$$

In the proposed design, the hand-off procedure takes a period of time while the user can continue sending requests. The average number of requests during the hand-off period N_{HO}, as a result of the time interval between two consecutive user inputs T_u, the total amount of data which are transferred for each hand-off D_{sync}, the transmission bandwidth provided by the link between the two adjacent servers BW_s, and the transmission latency of the link T_{ls}, all affect the fraction of transactions relevant to hand-offs. The equation is given by:

$$N_{HO} = \frac{\frac{D_{sync}}{BW_s} + T_{ls}}{T_u} \tag{7}$$

Once a user triggers a hand-off, the following N_{HO} requests are categorized as hand-off related transactions. Therefore, the average probability of transactions relevant to hand-off P_{HO} is given by the following equation:

$$P_{HO}=P_M \cdot \bar{P}_{cross} \cdot (1+N_{HO}) \approx \frac{P_M \cdot 2\sqrt{3} \cdot E(n)}{9\pi} \cdot \left(\frac{s}{L}\right) \cdot \left\{1+\frac{\frac{D_{sync}}{BW_s}+T_{ls}}{T_u}\right\} \quad (8)$$

where $E(n)$ is the average number of edges which border another service area. It is 0, $24/7$, and 4 for Configuration A, B, and C, respectively.

We can roughly conclude that P_{HO} can be increased by a higher user mobility, a larger volume of the data required for the synchronization, and a longer transmission latency between the servers. On the other hand, it will be reduced by a wider service area, a higher bandwidth between the servers, and a slower user input speed. However, the transmission latency between the two adjacent servers is proportional to the service range. We will see how the service range affects the average response time in the following context.

3) Average Response Time Comparison of The Three Configurations: In Configuration A, there is only one server thus no hand-off mechanism. The average traverse time of configuration A is quite straightforward:

$$T^A_{traverse} = 2 \cdot (T_r + T^A_l) \quad (9)$$

Now we have to consider hand-offs in Configuration B. Its average traverse time is:

$$\begin{aligned} T^B_{traverse} &= 2 \cdot (1-P^B_{HO}) \cdot (T_r + T^B_l) + 2 \cdot P^B_{HO} \cdot T^B_{HO} \\ &= 2 \cdot (1-P^B_{HO}) \cdot (T_r + \frac{T_l}{\sqrt{7}}) + 2 \cdot P^B_{HO} \cdot (T_r + T_{lmax} + T_{ls}) \\ &= 2\left\{T_r + \frac{T_l}{\sqrt{7}} - \frac{P^B_{HO}T_l}{\sqrt{7}} + \frac{\left\{\frac{1}{2}+\frac{3\ln(3)}{4}+\sqrt{3}\right\}P^B_{HO}T_l}{\sqrt{7}\left(\frac{1}{3}+\frac{\ln(3)}{4}\right)}\right\} \\ &= 2\left\{T_r + \frac{T_l}{\sqrt{7}}\left\{1+P^B_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\}\right\}\right\} \end{aligned} \quad (10)$$

where T_{lmax} is the propagation delay between the mobile station and the new server during hand-offs, which is $\left\{\frac{1}{2\sqrt{3}}+\frac{3\ln(3)}{8\sqrt{3}}\right\}T_{ls}$, since we assume that the mobile stations are still located around the borderline at the time.

Therefore, if we expect that Configuration B would outperform Configuration A, i.e., $T^B_{traverse} < T^A_{traverse}$, we can estimate the upper bound of P^B_{HO} as below:

$$1+P^B_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\} < \sqrt{7}$$

$$P^B_{HO} < \frac{(\sqrt{7}-1)(4+3\ln(3))}{12\sqrt{3}+2+6\ln(3)} \approx 0.4087356087 \quad (11)$$

This constrain is generally considered very slack.

Similarly, the average traverse time for Configuration C is:

$$T^C_{traverse} = 2\left\{T_r + \frac{T_l}{\sqrt{12}}\left\{1+P^B_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\}\right\}\right\} \quad (12)$$

And the upper bound of P^C_{HO} to outperform Configuration A is:

$$1+P^C_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\} < \sqrt{12}$$

$$P^C_{HO} < \frac{(\sqrt{12}-1)(4+3\ln(3))}{12\sqrt{3}+2+6\ln(3)} \approx 0.6119795056 \quad (13)$$

Furthermore, to outperform Configuration B given the same BW_s, the criteria are estimated below:

$$\begin{aligned} &\frac{T_l}{\sqrt{12}}\left\{1+P^C_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\}\right\} \\ &< \frac{T_l}{\sqrt{7}}\left\{1+P^B_{HO}\left\{\frac{12\sqrt{3}+2+6\ln(3)}{4+3\ln(3)}\right\}\right\} \\ &\Rightarrow P_M \cdot \left(\frac{s}{L}\right) \cdot \left\{1+\frac{1}{T_u}\left\{\frac{D_{sync}}{BW_s}-0.65T^B_{ls}\right\}\right\} < 0.3168 \end{aligned} \quad (14)$$

Therefore, depend on P_M, $\frac{s}{L}$, T_u, $\frac{D_{sync}}{BW_s}$, and T_{ls} of the baseline configuration, shrinking service areas by deploying more servers might reduce the average response time.

B. Optimal Arranged Base Stations Approach

In this approach, the service area is covered by a group of base stations, each connected to a server. Unlike the continuous service area approach which assumes each service area is a perfect regular hexagon, in this model the service areas are shaped by overlapping disks, each covered by a base station with omni-directional antenna. Consequently, each (local) service area is similar to a regular hexagon but with some "ripples" around the edges, which make it very difficult to estimate the hand-off probability. We can, however, proximately estimate it in certain conditions.

Here are the assumptions, which are slightly different from those of the other approach:

1) The whole service area is covered by minimum number of base stations with omni-directional antennae. In other words, base stations are located at unit points of a two-dimensional Synergetics coordinates [8].
2) We can either connect all base stations to one server, or separate base stations into several groups and connect them to the server of each group. The optimal service area of each group is approximately a regular hexagon.
3) Users are uniformly distributed geographically in the beginning. Users can either move a certain distance in any direction, or stay at the same location for a while.
4) The propagation delay of each link is proportional to its length.

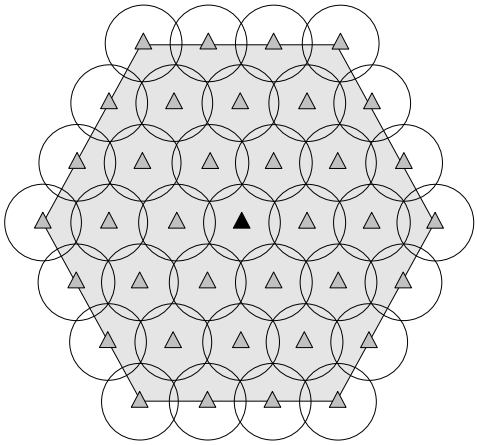

Fig. 6. Service area of single-server configuration with $m = 3$.

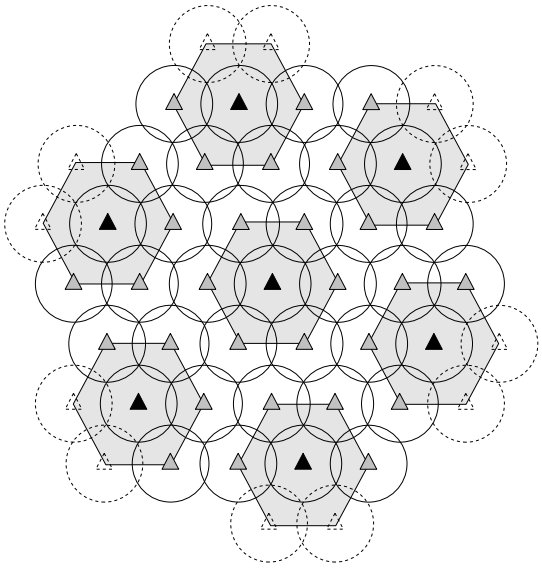

Fig. 7. Service areas of 7-server configuration, each with $m = 1$, covering the same area.

5) Each server's allocation is geographically optimized, that is, each server is located in the center of its (local) service area to reduce average propagation delay. The traverse time in our service is defined by (1) as well.

Again, we only compare the following two configurations covering the same area.

A $(3m^2 + 3m + 1)$ base stations are placed like a regular hexagon, where m is the number of the base stations' intervals along one of the hexagon's edges. Each interval is $\sqrt{3}R$ long, where R is the effective communication range of each base station. An example is illustrated in Figure 6.

B 7 servers, each connected to $(3\lceil \frac{m}{3} \rceil^2 + 3\lceil \frac{m}{3} \rceil + 1)$ base stations as a local service area. The base stations in each local service area are placed like a regular hexagon with $\lceil \frac{m}{3} \rceil$ intervals along one of its edge, as shown in Figure 7.

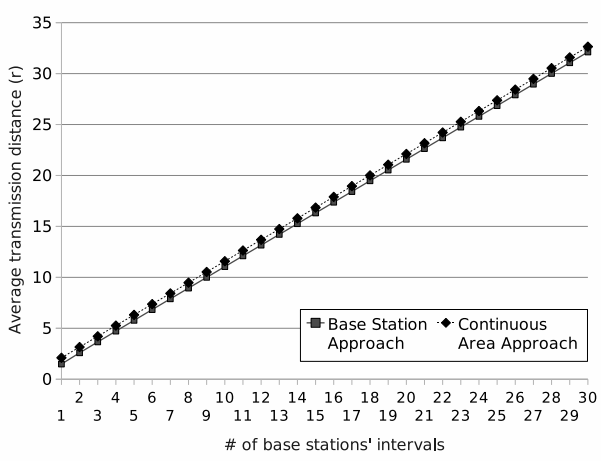

Fig. 8. Comparison of average transmission distances of different approaches covering approximately equal service area.

1) Average Transmission Distance: The average transmission distance in this approach is the discrete version of the continuous service area's counterpart. However, it is very difficult to represent in terms of m, as shown below:

$$\frac{3r\sum_{t=0}^{m-1}\sum_{k=1}^{m-t}\sqrt{3(2k+t)^2+9t^2}}{3m^2+3m+1} \tag{15}$$

Fortunately, we find out that the average transmission distance in this approach is approximately linear and gets closer to its continuous counterpart as m increases according to the computer calculation, as shown in Figure 8.

In other words, we can estimate the average transmission distance by either (15), or the continuous counterpart (2) with comparable parameters. In the later sections, we will use the latter one to focus on the quantitative relationships between the parameters and the performance rather than the exact value.

2) Probability of Transactions Relevant to Hand-offs: Due to the irregular shape of each local service area, it is difficult to estimate the exact probability of an arbitrary user around the border moving out of the service area by equations. However, if users' moving distances in each time instance are relatively short compared to a base station's effective communication range, the perimeter of each local service area at any point is near a straight line from a user's point of view.

Therefore, we can borrow the results from the continuous counterpart (4) to estimate the probability of a user crossing the borderline. The average probability of a user crossing the borderline along an arbitrary line which is perpendicular to the assumed straight borderline is:

$$\bar{\delta P}_{cross}(s) = \int_0^s \left\{ \frac{1}{\pi} \cos^{-1}\left(\frac{d}{s}\right) \right\} dd = \frac{s}{\pi} \tag{16}$$

The perimeter of the service area has to be recalculated as $6(m-1)$ one-third arcs and 6 half circles of radius R:

$$6L_{edge} = 6(m-1)\cdot\left(\frac{2\pi R}{3}\right) + 6\cdot\left(\frac{2\pi R}{2}\right) = 2\pi R(2m+1) \tag{17}$$

For a local service area with n edges which border another one, the length of the borderline eligible to invoke hand-offs is:

$$nL_{edge} = \frac{n\pi R(2m+1)}{3} \tag{18}$$

And we recalculate the service area as well. The area is basically a hexagon with some "decorations" around the perimeter:

$$\begin{aligned} A=&\frac{3\sqrt{3}}{2}\left(\sqrt{3}mR+\frac{R}{\sqrt{3}}\right)^2 \\ &+6\left\{\frac{\pi R^2}{2}-\frac{R^2}{\sqrt{3}}+(m-1)\left(\frac{\pi R^2}{2}-\frac{\sqrt{3}R^2}{4}\right)\right\} \\ =&R^2\left\{\frac{9\sqrt{3}m^2}{2}+\left(2\pi+\frac{3\sqrt{3}}{2}\right)m+\pi\right\} \end{aligned} \tag{19}$$

By accumulating the $\delta\bar{P}_{cross}$ along the perimeter and averaging with total area, the probability of a user crosses the borderline for mobile stations located in the service area for $s \ll R$ is:

$$\bar{P}_{cross} = \frac{n(2m+1)s}{3R\left\{\frac{9\sqrt{3}m^2}{2}+\left(2\pi+\frac{3\sqrt{3}}{2}\right)m+\pi\right\}} \tag{20}$$

The average number of requests during the hand-off period N_{HO} is the same in both approaches. P_{HO} is given by the following equation:

$$\begin{aligned} P_{HO} &= P_M \cdot \bar{P}_{cross} \cdot (1+N_{HO}) \\ &= \frac{P_M \cdot E(n) \cdot (2m+1)s}{3R\left\{\frac{9\sqrt{3}m^2}{2}+\left(2\pi+\frac{3\sqrt{3}}{2}\right)m+\pi\right\}} \cdot \left\{1+\frac{\frac{D_{sync}}{BW_s}+T_{ls}}{T_u}\right\} \end{aligned} \tag{21}$$

for $s \ll R$, where $E(n)$ is the average number of edges bordering another local service area as well, which is 0 and $24/7$ in Configuration A and B, respectively.

3) Average Response Time Comparison of The Two Configurations: The average traverse time of Configuration A $T^A_{traverse}$ is still $2(T_r + T^A_l)$. The average traverse time of Configuration B is equal to its continuous counterpart (10) as well.

If we expect that Configuration B would bring a shorter average response time over Configuration A, the upper bound of P^B_{HO} is unchanged:

$$P^B_{HO} < \frac{(\sqrt{7}-1)(4+3\ln(3))}{12\sqrt{3}+2+6\ln(3)} \approx 0.4087356087$$

Therefore, the constraints for P_M, m, s, R, $\frac{D_{sync}}{BW_s}$, T_{ls}, and T_u are represented in the equation below:

$$\begin{aligned} &\frac{P_M \cdot (2m+1)s}{R\left\{\frac{9\sqrt{3}m^2}{2}+\left(2\pi+\frac{3\sqrt{3}}{2}\right)m+\pi\right\}} \cdot \left\{1+\frac{\frac{D_{sync}}{BW_s}+T_{ls}}{T_u}\right\} \\ &< 0.3576436576 \end{aligned} \tag{22}$$

The result is similar to its continuous counterpart since $(R \cdot m)$ is proportional to the edge length L.

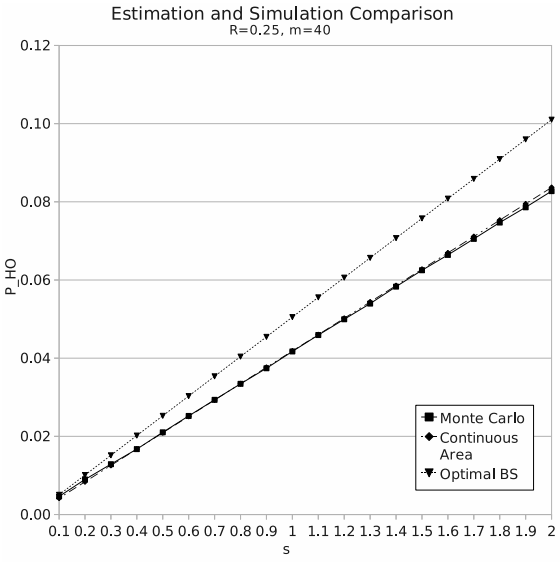

Fig. 9. Comparison of estimated and simulated P_{HO} with $R = 0.25$ and $m = 40$.

V. Simulation Results

To verify the estimations of P_{cross}, we use the Monte Carlo method by running a simulation program which sets up base stations of given R at optimal locations, randomly puts a large number of mobile stations, moves them away from their original location a fixed distance in any direction, and measures the number of the mobile stations escaping from the service area.

To compare the errors of the two different approaches, we set two environments with short R and large m, and long R with small m, and adjustable s. In the former environment, we set $R = 0.25$, $m = 40$, s varies from 0.1 to 2.0 with 0.01 steps, and place 10^7 mobile stations. The P_{HO} derived by the estimators and measured in the simulation are compared in Figure 9.

As we can see, the continuous service area approach is a better estimator since the shape of the service area is very close to a perfect regular hexagon in this environment. Furthermore, we compare the error rate of both estimators and compare them in Figure 10.

We can see in this series of simulations, the optimal arranged base stations approach only works well with very low s. However, when we set $R = 2.0$ and $m = 5$ and run the same simulations, it becomes a different story as shown in Figure 11.

Since the base stations are far less dense than in the previous setting, the "ripples" around the service area get larger and distort the shape away from a perfect regular hexagon. As we can see in Figure 11, the optimal arranged base stations approach is a very accurate P_{HO} estimator for $s \leq 0.5$ ($s \leq R/4$), and the continuous service area approach gets more and more accurate P_{HO} in response to increasing user mobility.

By comparing the estimation errors of both approaches in Figure 12, we can see the accuracies of the two estimators

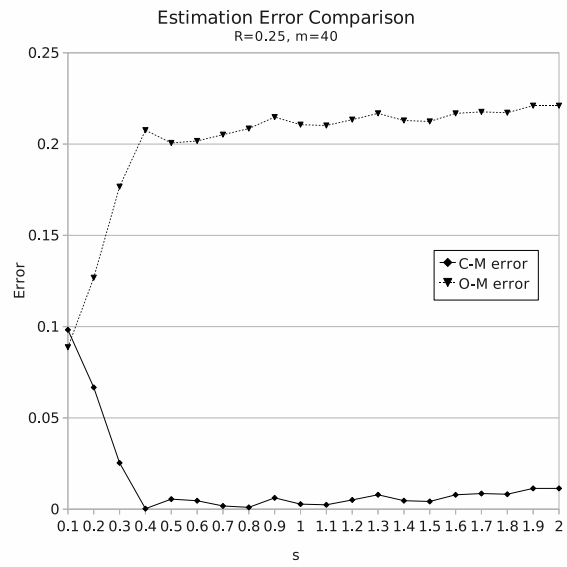

Fig. 10. Comparison of estimation errors with $R = 0.25$ and $m = 40$.

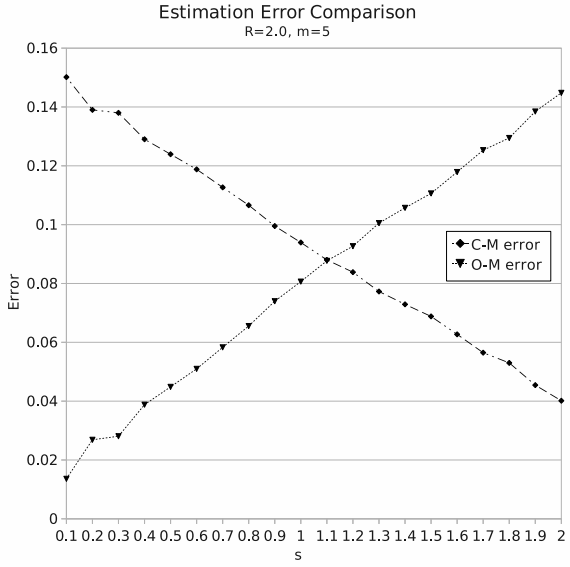

Fig. 12. Comparison of estimation errors with $R = 2.0$ and $m = 5$.

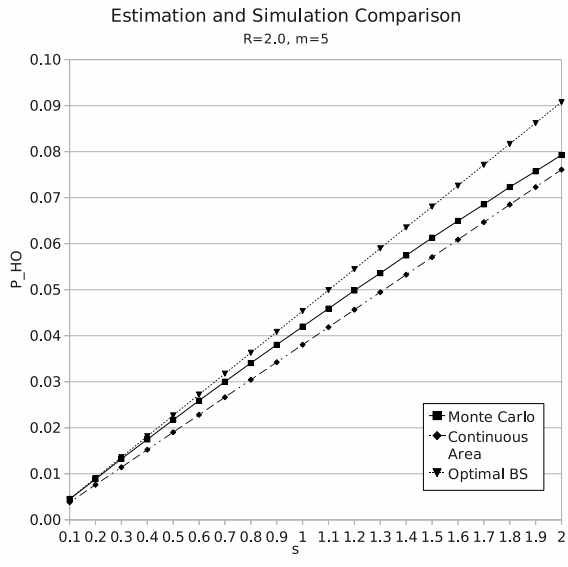

Fig. 11. Comparison of estimated and simulated P_{HO} with $R = 2.0$ and $m = 5$.

significantly depend on user mobility.

VI. Conclusion

In this paper, we have proposed a geographically distributed server arrangement and a hand-off protocol for application virtualization services for mobile users. We have also proposed several analysis and estimations in different conditions and configuration in order to evaluate the impact and the benefit of utilizing the proposed hand-off protocol. And we verify the estimators of probability of a user crossing the borderline by the Monte Carlo experiments and evaluate the accuracies and limitations of both approaches in the last part.

After going through the quantitative approaches to compare different server-user configurations, we find out the factors which should be taken into consideration when a service provider plans to launch virtual application servives or even virtual desktop services on mobile devices. If they analyze the user behaviors and the application's runtime properties and conclude that their users rarely move, or only move at a low speed, or interact infrequently, or the data volume required to recreate the runtime environment is relatively small, it is more likely to improve the performance by geographically deploying more servers to cover the whole service area and implement the proposed hand-off protocol. On the other hand, should one or more factors induce a very high hand-off count or overhead, the conventional single server configuration would be preferred.

References

[1] Sunwook Kim et al., *On-demand Software Streaming System for Embedded System*, WiCOM 2006 International Conference on Wireless Communications, Networking and Mobile Computing, 22-24 Sept. 2006, pp. 1-4.

[2] EMA Report: *AppStream: Transforming On-Premise Software for SaaS Delivery - without Reengineering*

[3] Joeng Kim; Baratto, R.A.; Nieh, J., *An Application Streaming Service for Mobile Handheld Devices*, 2006. SCC '06. IEEE International Conference on Services Computing, Sept. 2006, pp. 323-326.

[4] *VMware ThinApp Agentless Application Virtualization Overview.*

[5] Ana Fernandez Vilas et al., *Providing Web Services over DVB-H: Mobile Web Services*, IEEE Transactions on Consumer Electronics, Vol. 53, No, 2, May 2007, pp. 644-652.

[6] L. P. Deutch and B. W. Lampson, SDS 930 Time-sharing System Preliminary Reference Manual, Doc. 30.10.10, Project Genie, Univ. Cal. at Berkeley, April 1965.

[7] VMWare, *Virtual Desktop Infrastructure*

[8] Fuller, R. Buckminster, *Synergetics: explorations in the geometry of thinking*, Macmillan Publishing Company, 1975.