Advances in Social Science, Education and Humanities Research, volume 290
2nd International Conference on Economic Development and Education Management (ICEDEM 2018)

A Cost-effective Mobile Edge Computing Model

Lin Qing
Intelligent Science and Information Engineering College
Xi'an Peihua University
Xi'an 710125, China
530610232@qq.com

Huang Yulei
Intelligent Science and Information Engineering College
Xi'an Peihua University
Xi'an 710125, China
1397328804@qq.com

***Abstract*—With the explosive growth of smart devices and the advent of many new applications, data traffic over Internet has grown dramatically. Due to high latency and heavy burden on the backhaul links, traditional centralized network architecture cannot accommodate such challenges. Therefore, new architectures bring network functions to the edge of network, such as small-cell base stations and access point. Mobile Edge Computing (MEC) has been recognized as a promising technique to provide caching, computation and communication functionalities at the edge of cellular networks. In this paper, we first give an overview of mobile edge networks, including architecture and features. Next, a distributed and local-first computing (DLFC) model based on MEC is proposed, including distributed computing model and transmission model. Subsequently, the comparison between DLFC and centralized cloud computing (CCC) model is given and analyzed service delay and performance. Finally, the conclusion of this paper and future direction is presented as well.**

Keywords—Mobile edge computing; Mobile network; Cost-effective; Service latency

I. INTRODUCTION

With the development of Internet, data traffic has tremendously grown over the past few years, particularly video content delivery, which is expected to further grow in the next few years [1]. This is mainly driven by the explosive increase of intelligent devices whose number will increase to about 50 billion by 2020 [2]. Traditional centralized approaches for coping with this growth cannot satisfy the requirements of emerging interactive applications due to the long-distance communication. Moreover, the resource-constrained mobile device impose restrictions on the run of data-intensive or resource-hungry applications that are booming and prevalent, such as argument reality, online gaming, video conferencing and 3D modeling. At the same time, user's demands for high data rate and low latency of mobile network become more and more strict. These have put forward challenges of Internet, e.g., how to cache and distribute the large-scale contents to supply latency-sensitive, location-aware service for mobile users. Hence, a new computing paradigm is in great demand.

In the next five years, the more powerful processing capability, storage and other advance features of dedicated AI chips will be added to a wider range of edge devices. The diversity of this embedded Internet of Things world, coupled with assets such as industrial systems, will have a long life cycle, which will pose significant challenges to management.

In the long run, as 5G matures, the ever-expanding edge computing environment will have more reliable communication technologies back to centralized services. 5G offers lower latency, higher bandwidth, and a sharp increase in the number of nodes per square kilometer (edge endpoints), the last point being very important to the edge.

Recently, mobile edge computing (MEC) has emerged as a promising paradigm to handle the exponentially increasing data traffic and to alleviate the communication burden, and it is characterized by location-aware, latency-sensitive and mobile-support [3]. Edges are endpoint devices that people use or endpoint devices embedded around us (e.g., base stations, access points, etc.). Edges can satisfy the demand of computation, communication and caching capability. Edge computing describes a computing topology where information processing and content collection and delivery are closer to these endpoints. It tries to keep traffic and handle localization, with the goal of reducing traffic and reducing latency [4]. Because the MEC servers are closer to the end users, the data transmission rate is high and data pre-processing can be conducted locally. According to the demands of end users, network resource for computation, communication and storage are automatically allocated. At the same time, according to the dynamic changes of network load, resource distribution and service requirement should keep be consistent in order to avoid the degradation of serve quality or resource waste. Because the MEC servers are usually deployed with BSs and access points, it is easy for terminal devices (such as smart phones, wearable devices, vehicles, etc.) to access the edge computing service anytime and anywhere. Due to its distinct caching features, content can be cached to MEC servers from original content servers, so that the communication cost for accessing the content can be greatly reduced.

Many research efforts have been dedicated to mobile edge computing based on those advantages. MEC has two major development trends:

1) Popular Content Caching: Edges are used to build the content caching and delivery framework, which can alleviate the burden of core networks and reduce the service latency experienced by end users. In [5], in order to manage 5G network, an end-to-end cache-enabled heterogeneous network based on the combination of content delivery strategies and MEC was proposed, and its performances were analyzed from the aspect of coverage, throughput and energy efficiency. In [6], computation results of tasks were cached in a multi-user cache-assisted MEC system. The work formulated the average total energy minimization problem that is subject to the caching and

Xi'an Peihua university scientific research project in 2018 (PHKT18054).

deadline constraints to allocate the BS's storage resource for caching computation results. Edge caching can be used as a specialized cache for recognition applications. Cachier is proposed to minimize the latency through adaptively balancing load between the edge and cloud and leveraging spatiotemporal locality of requests and network conditions, offline analysis of applications and online estimates of network conditions [7].

2) Computation Offloading: Edges provides computation capability, which not only support computation offloading from terminal users for saving the battery of terminal devices. In order to jointly tackle these issues in wireless cellular networks with mobile edge computing, we formulate computation offloading decision, resource allocation and content caching strategy as an optimization problem, considering the total revenue of the network. Furthermore, we transform the original problem into a convex problem and then decompose it in order to solve it in a distributed and efficient way [8]. In this work we study the feasibility of both mobile computation offloading and mobile software/data backups in real-life scenarios. In our study we assume an architecture where each real device is associated to a software clone on the cloud [9].

In this paper, we first exploit the architecture of mobile edge networks, including its components, functions, features different from other networks, which consists of users layer, MEC layer and cloud layer. Then, in order to further investigate MEC's working mechanism, we propose the distributed and local-first computing (DLFC) model, and theoretically analyze its computation and communication models. Finally, compared with the centralized computing paradigm, the performance in term of service latency is analyzed. Finally, numerical simulations are presented to show the effectiveness of the DLFC over alternative cloud computing model.

The rest of this paper is organized as follows. Section II introduces the architecture of mobile edge networks. In Section III, the computation model and transmission model of distributed and local-first computing model was presented. Performance comparison analysis are given in Section V, followed by the conclusion in Section VI.

II. MOBILE EDGE NETWORKS

A. Architecture of Mobile Edge Networks

As depicted in Fig.1, mobile edge network comprises of users, MEC servers, centralized cloud servers (CCS). User and MEC to which it is associated are connected via wireless communication, while MEC is linked with other MECs (its neighbors and non-neighbors) and NEF by single hop or multiple hops fiber optic communication. The components are discussed below:

1) User lyaer: As content requesters, users can fetch the popular contents from its associated MEC. Users set can be indicated by $U = \{u_k \mid u_k =< (x_k, y_k), e_k, t >, 1 \le k \le n\}$. u_k 's tuples include its current location and its directly connected MEC, total number of users is n . User device is capable of sharing its absolute geospatial location through GPS or GIS with MEC network.

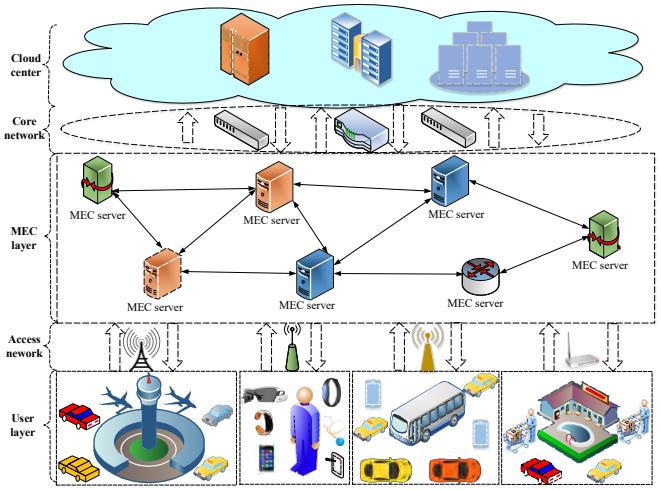

Fig.1 Architecture of MEC Network

2) MEC layer: MEC serves as a service node with computing and caching capabilities, besides routing and forwarding functionalities. MEC can deliver content to user without any disruptions due to mobility-support of MEC. MECs set is $E = \{e_i \mid e_i =< (x_i, y_i), E_i, O_i, U_i, c_i >, 1 \le i \le m\}$, comprising of its location, neighbor MECs set, content set cached locally, users set within its working range, storage capability, and the total number of MECs is m . MEC e_i maintains three major data structures: the Neighbor Information Table (NIT), the Content Object Table (COT), the Original Request Table (ORT). The NIT is used to supervise its neighbor nodes' status information, including ID, cached contents, workload. The CDT holds contents cached locally. The ORT keeps the original request queue receiving from the terminal users, and it is applied to continue to forward the request to corresponding MEC caching the requested content. MEC can aggregate similar content requests and calculate the request arrival rate. MEC servers are deployed together with micro and pico base stations. *Neighbor is the nodes connected with single hop.* At the MEC layer, as the centralized administrator, network exposure function (NEF) keeps track of the requests received by MECs and collects users' current location. The whole content requests reported by all the MECs are aggregated to generate a holistic content popularity ranking. NEF is depicted as $N =< E, U >$, including MECs set and users set. Similarly, NEF maintains the Global Information Table (GIT), which is used to supervise the status information of all the MECs, such as index of contents cached, workload (service intensity, service arrival rate).

3) CCS: CCS stores all the content objects, any content requests originated from users can always be served by OS. Many studies shows that the content (Web and video).

B. Features of Mobile Edge Network

The characteristics of mobile edge computing can be summarized as follows:

- Proximity: Being deployed close to the network end users, MEC is particularly useful to better serve and understand

users' preferences on content. MEC may also have direct access to the devices, which can easily be leveraged by applications;

- Low latency: As edge services run close to end-devices, latency is considerably reduced. This can be utilized to react faster, to improve user experience, or to meet the requirements of delay sensitive applications;
- Location-awareness: A locally-deployed service can leverage low-level signaling information to anonymously determine the location of each connected device. This enables various applications, such as Location-based Services and analytics solutions;
- Network context-awareness: Real-time network data (such as network conditions, radio status and more) can be used by applications/services to offer context-related services that can differentiate the mobile broadband experience and be monetized. New applications can be deployed to connect mobile devices with local points-of-interest, events, among many other possibilities.

III. DISTRIBUTED AND LOCAL-FIRSR COMPUTING MODEL

User sends content request to MEC network, and it contains content name as the special content identity. If the requested content is cached in the MEC network, it will be sent to the requester. Otherwise, OS satisfies the content request and delivery it to requester. However, if original server directly serve all the requests, its load will go up, thus seriously affecting experience of user. The detailed working mechanism of M-DCDN is further elaborated as follows:

In the MEC network, user's corresponding MEC (local MEC) receives and process the content request, store it to the ORT. Local MEC search the OCT to check whether requested content name is matched, if so, it will directly be sent to the user through the reverse path. If not, local MEC then retrieve NIT to inspect whether its neighbor MECs cache the requested content. If requested content is found in its neighbor MECs, local MEC then redirect the content request to the neighbor MEC. If both local MEC and its neighbor MECs don't cache requested content, it sends a query request to NEF to find the non-neighbor MEC through checking the GIT, if found, then NEF send this MEC's ID back to local MEC, and local MEC redirects the request to that non-neighbor MEC for fetching the requested content. Otherwise, NEF redirects this content request to the original server, then original server serves the request and send requested contents back to user using the same communication path traversed by the request message.

A. Computing Latency

d_r denotes the size of request packet (or computation task), c_e and f_e are MEC server's CPU number required for processing 1-bit data and MEC 's CPU frequency. Hence, MEC's computing latency is given by

$$L_e = \frac{d_r c_e}{f_e} \tag{1}$$

Similarly, the computing latency of NEF and cloud center for processing one user request are shown as follows:

$$L_n = \frac{d_r c_n}{f_n} \tag{2}$$

$$L_c = \frac{d_r c_c}{f_c} \tag{3}$$

where, c_n and f_n, c_c and f_c are similar to MEC server's c_e and f_e.

B. Transmission Latency

Let H denote the channel power gain for MEC that is constant during sending content, and p_{eu} its transmission power via wireless communication. The achievable rate (in bits/s) is:

$$r_{ue} = B\log_2(1+\frac{p_{eu}H}{N_0}) \tag{4}$$

where B and N_0 are the bandwidth and the variance of the complex additive white Gaussian noise, respectively.

Thus, the transmission latency between user and MEC servers is given by

$$L_{eu} = \frac{d_r}{r_{ue}} \tag{5}$$

MEC server communicates with NEF by optical The transmission latency between MEC server and NEF

MEC server interacts with the adjacent MEC servers, NEF and cloud center servers through fiber optic communication. The transmission rate of link between MEC server and NEF is r_{en}, the transmission latency between them is :

$$L_{en} = \frac{d_r}{r_{en}} + \frac{NL_{en}}{C} \tag{6}$$

where L_{en}, N, C are fiber length, refractive index of fiber (for the 1310nm fiber, it is about 1.5) and speed of light in the vacuum (3×10^8m/s) respectively.

Similarly, the transmission latency between MEC server and cloud center server is given by

$$L_{ec} = \frac{d_r}{r_{ec}} + \frac{NL_{ec}}{C} \tag{7}$$

C. Total Service Latency

Total service latency is the time experienced by user from the time the terminal initiates the service request to the time the terminal receives the service response, including the transmission delay of the service request, the calculation processing delay of the service request, and the backhaul delay of the service response.

The non-cooperative working mode: only MEC process user requests, the corresponding total service latency is :

$$\begin{aligned} L_{MEC} &= L_{ue}^r + L_e^r + L_{eu}^o \\ &= \frac{d_r + d_o}{B\log_2(1+Hp_{eu}/N_0)} + \frac{d_r c_e}{f_e} \end{aligned} \tag{8}$$

The cooperative working mode: MEC collaborates with NEF server for processing user requests, the corresponding total service latency is :

$$\begin{aligned} L'_{MEC} &= L^r_{ue} + L^r_e + L^r_{en} + L^r_n + L^o_{en} + L^o_{eu} \\ &= \frac{d_r + d_o}{B\log_2(1+Hp_{eu}/N_0)} + d_r\left(\frac{c_e}{f_e} + \frac{c_n}{f_n}\right) \\ &+ \left(\frac{d_r + d_o}{r_{en}} + \frac{2NL_{en}}{C}\right) \end{aligned} \tag{9}$$

The centralize cloud computing mode: Only centralized cloud server process user request without MEC server. The corresponding total service latency is :

$$\begin{aligned} L_{Cloud} &= L^r_{ue} + L^r_{ec} + L^r_c + L^o_{ce} + L^o_{eu} \\ &= \frac{d_r + d_o}{B\log_2(1+Hp_{eu}/N_0)} + \frac{d_r c_c}{f_c} \\ &+ \left(\frac{d_r + d_o}{r_{ec}} + \frac{2NL_{ec}}{C}\right) \end{aligned} \tag{10}$$

Latency difference between cloud computing and mobile edge computing models is shown below:

$$\begin{aligned} \Delta &= L_{Cloud} - L_{MEC} \\ &= d_r\left(\frac{c_c}{f_c} - \frac{c_e}{f_e}\right) + \left(\frac{d_r + d_o}{r_{ec}} + \frac{2NL_{ec}}{C}\right) \end{aligned} \tag{11}$$

Under the hybrid working model of mobile edge computing and cloud computing, the mathematical expectation of the service delay for processing the service request is given:

$$L_{a\,var\,age} = PL_{MEC} - (1-P)L_{Cloud} \tag{12}$$

Where P is the hit rate for requests in the MEC layer.

IV. PERFORMANCE COMPARISON ANALYSIS

In this section, we evaluate the performance of mobile edge computing, compared with the traditional cloud computing in term of service latency. Parameter setting is given in Table Ⅰ.

TABLE I. PARAMETER SETTING

Parameter	Value	Parameter	Value
d_r	1Mb	N_0	10^{-9}W
d_o	10Mb	P_{eu}	1W
d_s	50Mb	H	10^{-6}
f_e	1×10^9cycles/s	D_{en}	5.0×10^3m
c_e	1×10^3cycles/bit	D_{ec}	50×10^3m
f_c	5×10^9cycles/s	r_{en}	100Mb/s
c_c	10×10^3cycles/bit	r_{ec}	100Mb/s
f_n	1×10^9cycles/s	N	1.5
c_e	1×10^3cycles/bit	C	3×10^8m/s
B	10MHz	P	90%

Simulation experiment is achieved by Matlab in the Windows7 platform, its basic parameter includes 3.60GHz Inter Core (TM)i7-4790 processor, 12GB memory. In this section, we first perform numerical experiment on the effect of CPU frequency, distance between base station and cloud center, transmission packet size, and hit rate on service latency. Then, we analyze the effect of computing capability on Latency difference between cloud computing and mobile edge computing models.

A. *Effect of CPU frequency, distance between base station and cloud center, transmission packet size, and hit rate on service latency*

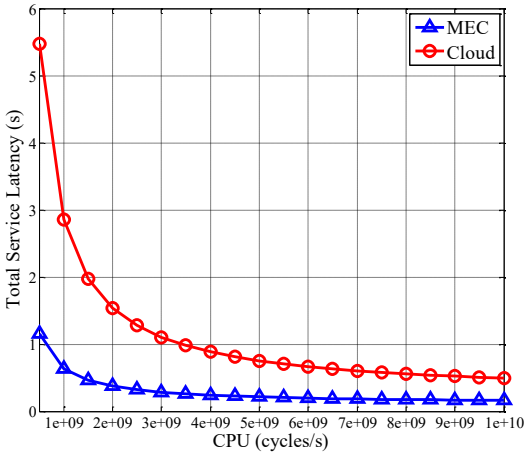

Fig.2 CPU vs total service latency

Fig.2 shows the effect of CPU frequency on total service latency (d_o =10Mb, D_{ec} =1000km): The larger the CPU frequency, the smaller the service delay of the MEC or cloud center, and the lower the CPU frequency, the MEC service delay is lower than that of cloud computing.

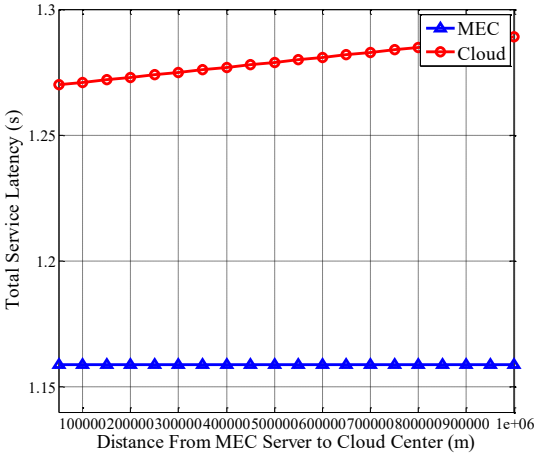

Fig.3 Distance from MEC server to cloud center vs total service latency

Fig.3 illustrate the relationship with distance from MEC server to cloud center and total service latency (f_e =1×10^9cycles/s, f_c =5×10^9cycles/s, d_o =10Mb): As the communication distance between the base station and the cloud center increases, the total service delay of the cloud computing increases linearly. Since the MEC does not need to be transmitted through the core network, it has no effect on the total service delay of the MEC.

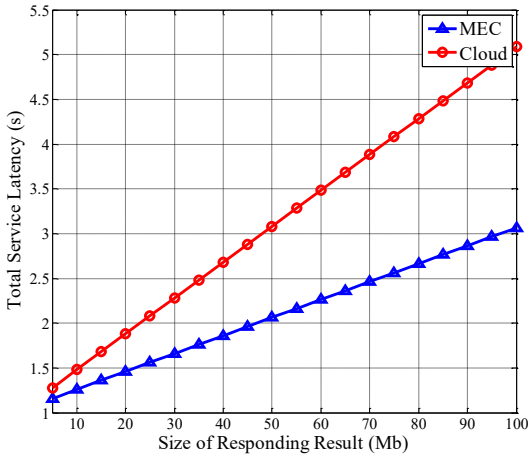

Fig.4 Size of responding result vs total service latency

From Fig.4, we observe the effect of Size of responding result on total service delay (f_e =1×10^{9}cycles/s, f_c =5×10^{9}cycles/s, D_{ec} =1000km): As the length of response packets increases, the total service delay of cloud computing and MEC also increases. However, the service delay of the MEC is lower than the service delay of the cloud computing, because the MEC does not transmit through the core network of the base station and the cloud center.

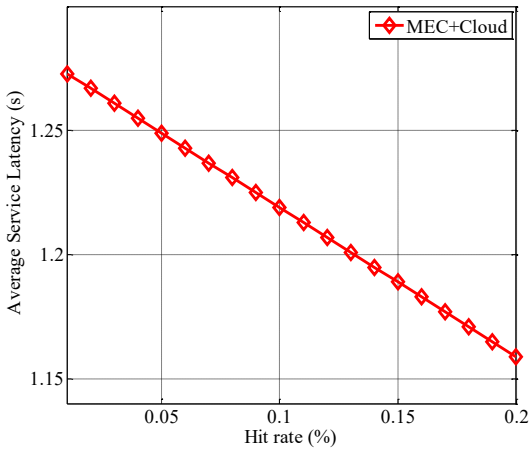

Fig.5 Hit rate vs total service latency

Fig.5 indicates the effect of hit rate on total service latency (f_e =1×10^{9}cycles/s, f_c =5×10^{9}cycles/s, D_{ec} =1000km, d_o =10Mb): As the hit rate in the MEC increases, the mathematical expectation of the average service delay decreases.

B. *Effect of MEC CPU frequency and Cloud CPU frequency on* Δ

From Fig. 6, when the CPU frequency of the cloud is fixed, the service delay difference increases as the CPU frequency of the MEC increases, indicating that as long as the CPU frequency of the MEC is sufficiently large, the feature of the service latency in MEC obtaining the response is more obvious.

On the contrary, when the CPU frequency of the MEC is fixed, the service delay difference decreases as the CPU frequency of the Cloud increases. This shows that when the CPU frequency of the Cloud exceeds a certain value, the advantage of the MEC is no longer obvious.

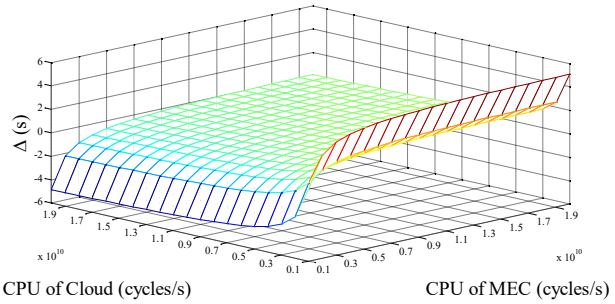

Fig.6 CPU vs Δ

V. CONCLUSION

With the development of 5G network, netowrk edge provides the computation and caching functionality. Mobile Edge Computing has emerged as a promising paradigm to handle the exponentially increasing data traffic and to alleviate the communication burden. In this paper, we first introduce the framework of mobile edge computing, including its components and features. Then, from the perspective of computation and communication, we conduct the analysis of computing and transmission models in term of service latency. Finally, we evaluate the comparison between mobile edge computing and clouding computing, and analyze the effect of systerm parameters on the performance.

REFERENCES

[1] CISCO, "Transformation through innovation." [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/transformation-through-innovation.html.

[2] Wang S , Zhang X , Zhang Y , et al. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications[J]. IEEE Access, 2017:1-1.

[3] Mao Y, You C, Zhang J, et al. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.

[4] Mach P, Becvar Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading[J]. IEEE Communications Surveys & Tutorials, 2017, PP(99):1-1.

[5] Tran T X, Hajisami A, Pandey P, et al. Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges[J]. IEEE Communications Magazine, 2017, 55(4):54-61.

[6] Ren D, Gui X, Dai H, et al. Hierarchical Resource Distribution Network Based on Mobile Edge Computing[C]// 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). 2017.

[7] Tran T X, Pandey P, Hajisami A, et al. Collaborative multi-bitrate video caching and processing in Mobile-Edge Computing networks[C]// Wireless On-demand Network Systems & Services. 2017.

[8] Wang C , Liang C , Yu F R , et al. Computation Offloading and Resource Allocation in Wireless Cellular Networks With Mobile Edge Computing[J]. IEEE Transactions on Wireless Communications, 2017, 16(8):4924-4938.

[9] Barbera M V, Kosta S, Mei A, et al. To offload or not to offload? The bandwidth and energy costs of mobile cloud computing[C]// Infocom, IEEE. 2013.