A Dynamic Link Aggregation Scheme for Heterogeneous Wireless Networks

Bala Murali Krishna K, Madhuri S, Vanlin Sathya and Bheemarjuna Reddy Tamma
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: [bala, madhuris, cs11p1003, tbr]@iith.ac.in

***Abstract*—User Equipments (UEs) are integrated with multiple interfaces to facilitate the usage of various wireless technologies. However, due to the limitations of each technology, expected data rates are not achieved even when the best of available interfaces is chosen for data transfer. Link aggregation is used to aggregate data rates over multiple interfaces for achieving higher data rates. Even though there are multiple link aggregation schemes proposed, they are highly complicated and do not consider dynamics of wireless links. In this paper, we propose a simple and application layer Dynamic Link aggregation Scheme (DLAS) where data can be dynamically transferred through multiple links parallelly or sequentially. We tested the performance of the proposed DLAS schemes in NS-3 simulator and found that DLAS schemes achieve considerable improvement in the data rates in heterogeneous networks comprising of LTE and Wi-Fi technologies. In our experiments for 10 UEs, DLAS achieved 65% improvement in maximum achievable throughput when compared to an existing application layer link aggregation technique.**

I. INTRODUCTION

The use of Internet over mobile networks has increased enormously in the recent times and so does the need for higher data rates. To provide such higher data rates, 3GPP proposed 4G technology, which is also known as Long Term Evolution (LTE). The major drawback using any of cellular technologies is their poor coverage in the indoor environments. According to Huawei and Nokia-Siemens [1], [2] 60% of the mobile voice and video traffic comes from indoor environments. So, increasing the data rates in such environments became essential. To solve the above mentioned problem small cells concept was introduced [3]. There are many small cell networks available for outdoor and indoor usage like Femtos, Picos, Wi-Fi and so on. Femtos are installed by the end user and it gets connected to the macro through backhaul connection. Femtos are mainly used in home and enterprise scenarios [4] to provide seamless mobile connection anywhere. Wi-Fi is another variant of small cell network which is being deployed in pair with cellular networks by mobile network providers and operates in unlicensed spectrum. Most enterprise and educational institutes provide Wi-Fi services in their organizations. Wi-Fi *hotspots* are also being installed in population dense areas to offload traffic from cellular networks. So, using such a widely spread Wi-Fi technology helps us further increase in data rates offered to the end users. Heterogeneous Network (HetNet) is a collective deployment of various technologies like LTE Femtos, Wi-Fi AcessPoint (APs) and LTE macros, which results in highly overlapped coverage regions. A typical HetNet is shown in Fig. 1.

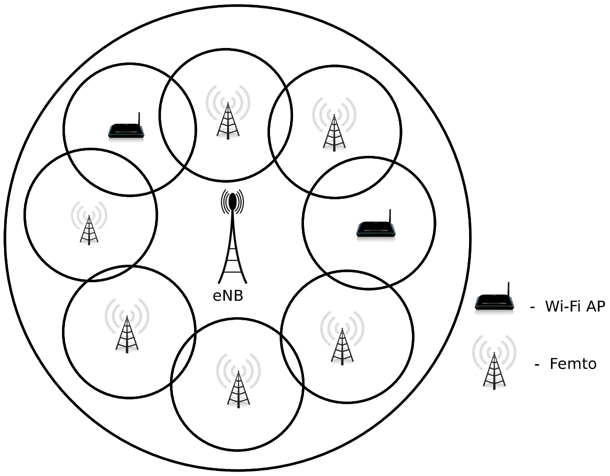

Fig. 1: An example of HetNet comprising of LTE, Wi-Fi, and Femto networks

User Equipments (UEs) like laptops, tablets, smart phones come with multiple radio interfaces. But each interface is given a precedence over other and the applications are not designed to take advantage of multiple interfaces. Only one interface will be kept active at a time, to save battery power, which restricts the use of multiple interfaces simultaneously even though the UE resides in the coverage area two or more macro BS/Femtos/APs. To overcome this problem, Link Aggregation Scheme (LAS) or Bandwidth Aggregation (BAG) was introduced [5]. The purpose of LAS is to aggregate bandwidth over two or more interfaces, which are available at a particular time for the UE. Aggregation can be performed at several layers of the protocol stack to improve the overall throughput of applications. But inefficient link usage still exists if the aggregation scheme fails to fully capture the dynamics of changing network scenario. For example, if data is split equally and sent among multiple interfaces, the end-to-end delay would be equivalent to delay of the interface with lowest bandwidth. Hence, data distribution must be according to the bandwidths (data rates) available over multiple interfaces. So, we propose an application-level link aggregation scheme, "DLAS" (Dynamic Link Aggregation Scheme), which uses multiple interfaces efficiently and provides fairness to other users in the HetNets.

The rest of this paper is organized as follows. Section II describes the related work. Section III describes the proposed work. Section IV describes experimental setup and analysis of algorithms under several scenarios. Finally, the paper is concluded with the future work in Section V.

II. RELATED WORK

In this section, we review existing works on link aggregation due to incorporation of different small cells (Femto, Wi-Fi) into HetNet. In [6], the bandwidth aggregation is done for real-time traffic to reduce the delay and packet loss. Even though the delay is reduced, equal amount of data (packets) is requested on individual interfaces. Where as data requested on each interface must be proportional to the available bandwidth of the interface. In [7], due to high deployment of Wi-Fi APs, message flooding leads to congestion and high amount of packet drop in the network. According to the authors of [8], the bandwidth aggregation is done over Wi-Fi interfaces for both TCP and UDP traffic. Because of un-licensed spectrum in Wi-Fi, there will be a lot of contention for the voice (UDP) traffic and this leads to degradation in voice quality. HetNet scenario is not at all considered where voice traffic would have been routed over Macro BSs.

In [9], the authors proposed an application-level link aggregation (ALA) scheme. But, they demonstrated link aggregation over two Wi-Fi interfaces only. The ratio of data sent on each interface would be proportional to the instantaneous throughputs obtained on corresponding interfaces. Although it gives better aggregate throughput, the effect of key issues such as mobility of nodes, fairness of the algorithm or performance of the algorithm in HetNet scenarios are not studied in their work.

Peer-to-peer Bit-Torrent application [10] uses a method called chunking, where original file is split into smaller parts (in the order of 64KB-2MB) known as chunks as shown in Fig. 2. Different chunks can be downloaded from different peers in the p2p network. Similar to Bit-Torrent protocol, proposed work uses chunking method where original file is requested by UE in the order of chunks over multiple available interfaces.

The goal of this paper is to achieve higher throughputs by devising an application-level link aggregation scheme, DLAS, which allows UEs to request chunks of original file from the source/server through multiple interfaces as per their dynamically changing link bandwidths in HetNets comprising of LTE, Femto, and Wi-Fi networks.

III. PROPOSED WORK

In this work, we consider a HetNet system comprising of macro BSs, Femto BSs and Wi-Fi APs. We assume that a large number of Femtos and Wi-Fi APs are deployed under one macro BS coverage region. The constraint on UE is its inability to utilize multiple interfaces, although UE is in an overlapping region of two or more BSs/APs. Since any overlap region is itself an indication of having more resources to be utilized, proposed work aims at utilizing available multiple interfaces in an optimized way as much as possible by considering the time varying link capacities of individual interfaces. But it also tries to provide fairness to other UEs legacy flows in the network. The DLAS algorithm takes the network congestion into consideration and dynamically changes the number of chunks being transferred on each individual link. Depending on available bandwidth and user density two DLAS approaches have been introduced, parallel and sequential. Sequential approach deals with chunk level sequential transfer of flow's data at any given time instant on each link, where as parallel approach deals with multiple sub-chunk level parallel data transfers at any given time instant on each link.

TABLE I: Glossory

Notation	**Meaning**
$Nch(I_j)$	Number of Chunks to be sent on interface I_j
Ch_{size}	Chunk Size
Ch_{sm}	Sub-chunk Size
$Th_i(I_j)$	Instantaneous Throughput of the flow on I_j
$Th_{ov}(I_j)$	Overall Throughput of the flow on I_j
$Th_{old}(I_j)$	Past overall throughput of the flow on I_j
n	total number of chunks in the file
m	total number of interfaces available at UE
RCOIS(I_j,$Nch(I_j)$)	A function used by UE which requests $NchI_j$ chunks from Server on I_j in SDLAS
RCOIP(I_j,$Nch(I_j)$,p)	A function used by UE which requests $NchI_j$ chunks from Server on I_j in PDLAS by splitting each chunk flow into p parallel sub-flows
CRCI(I_j,$Th_i(I_j)$,$Th_{old}(I_j)$)	A function used by UE to calculate number of chunks to be sent on I_j in next interval

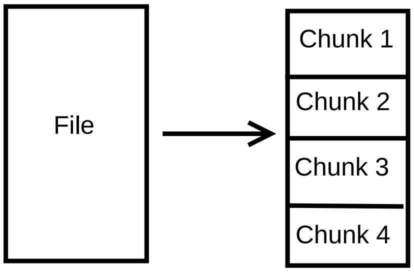

Fig. 2: Fragmentation of original file into multiple chunks

Sequential DLAS: UE requests for one or more chunks through each interface in a transmission round (i.e., burst) and the server responds by sending corresponding chunks. Only after receiving the requested chunks, UE calculates the instantaneous throughput obtained on each interface, which is further used in updating the value of overall throughput of the interface. For example in a two-interface scenario, if the ratio of obtained throughputs on two interfaces is $x : y$, then x chunks would be requested on one interface in a burst and y chunks would be requested on another interface in a burst. A scenario where UE having two interfaces requesting chunks at $1 : 1$ ratio is shown in Fig. 3.

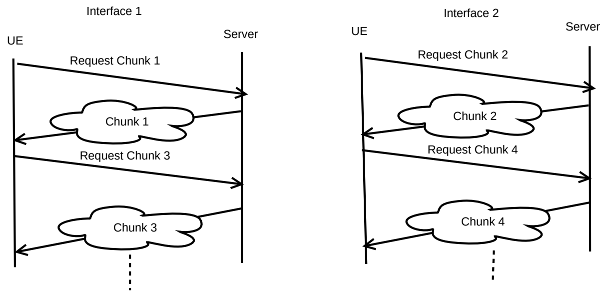

Fig. 3: Example of Request-Response scenario in Sequential DLAS

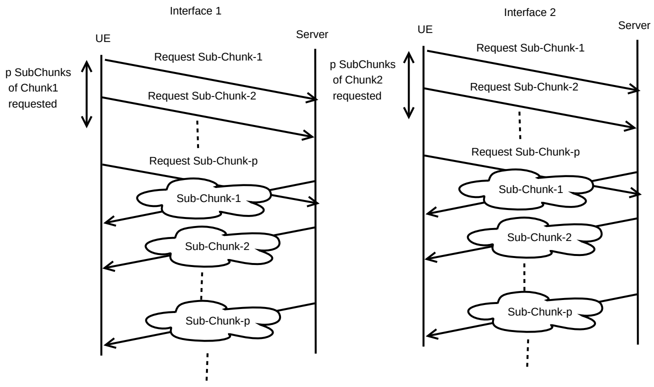

Fig. 4: Example of Request-Response scenario in Parallel DLAS

Parallel DLAS: Like in Sequential DLAS, UE requests for one or more chunks through each interface and the server responds by sending corresponding chunks. But here, there will be p number of parallel TCP flows (i.e., one for each sub-chunk of the chunk under consideration) established over each individual interface, so that the rate of data transfer increases. Since the connections are multiplied, we are dividing the chunk of size Ch_{size} into p sub-chunks of size Ch_{sm}. Hence, $Ch_{sm} = \frac{Ch_{size}}{p}$. This technique leads to more efficient utilization of available link bandwidth. An example showing single chunk request in PDLAS is given in Fig. 4.

In proposed DLAS schemes, aggregation is performed at the application layer of the protocol stack to make it simple and without complexities of modifying the network or transport layers. The additional advantages of having aggregation at the application layer are:

1) No additional devices such as proxy servers are needed as per [6].
2) It can support link aggregation between any two technologies even though they have no protocol stack layers in common except the application layer.

IEEE standards like Wi-Fi and WiMax have upper four layers of protocol stack in common and hence transport layer aggregation can be realized for them. But to have aggregation between technologies like 3GPP LTE and IEEE Wi-Fi which have no layer in common other than the application layer, it becomes difficult to have such proxy servers at the transport layer. So, we are using aggregation at the application layer. The only overhead of using aggregation at the application layer is re-ordering of packets/chunks which is minimal and can be taken care easily to reap in the benefits of huge gain in the bandwidth aggregation.

A. Algorithm Description

We propose DLAS algorithm which is explained as follows. Table I shows the notations used in this work. Original file is divided into several chunks as shown in Fig. 2. Instantaneous Chunk Throughput $Th_i(I_j)$ is estimated as follows:

$$Th_i(I_j) = \frac{Nch(I_j) * Ch_{size}}{RTD} \tag{1}$$

where RTD is the round-trip-delay for $Nch(I_j)$ chunks. RTD is given as time interval between first request made by client to last chunk received at the client.

In the initialization phase, only one chunk is requested on each interface to obtain the corresponding instantaneous throughputs for both DLAS schemes. For each individual interface, overall throughput $Th_{ov}(I_j)$ is calculated as follows:

$$Th_{ov}(I_j) = \gamma * Th_{old}(I_j) + (1 - \gamma) * Th_i(I_j) \tag{2}$$

for some γ, $0 < \gamma < 1$.

We assume that cumulative throughput reflects the current link capacity, the past obtained $Th_{old}(I_j)$ is given weightage in order to negate any fluctuations in the link, i.e., instantaneous throughput $Th_i(I_j)$ does not necessarily indicate the channel condition. We experimented with different values of γ (refer Fig. 5), and found that γ=0.3 gives optimal result. The ratio of chunks to be transferred on each interface is directly proportional to the ratio of the overall throughputs obtained on each interface and normalized.

$$\begin{aligned} Nch(I_1) : Nch(I_2) : \ldots : Nch(I_m) = \\ Th_{ov}(I_1) : Th_{ov}(I_2) : \ldots : Th_{ov}(I_m) \end{aligned} \tag{3}$$

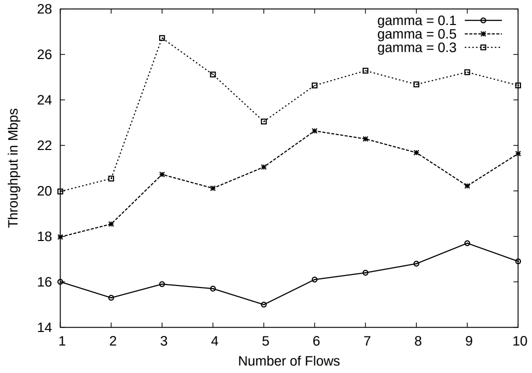

Fig. 5: Throughputs for different values of γ for Wi-Fi and LTE with parallel connections

1) Algorithm for Sequential Dynamic Link Aggregation

```
Input: $n$ , $m$
For all the available interfaces, j:1 to m
do
    Call func RCOIS($I_j$,1);// Initially one chunk is requested on $I_j$
    Calculate $Th_i(I_j)$ // Refer Equation (1)
    $Th_{old}(I_j) \leftarrow Th_i(I_j)$;
    while $n > 0$ do
        $Nch(I_j) \leftarrow$ CRCI($I_j$, $Th_i(I_j)$, $Th_{old}(I_j)$) //calculates $Nch(I_j)$ to be sent on $I_j$ in next interval, by calculating $Th_{ov}(I_j)$ according to Equation (2) and calculating $Nch(I_j)$ according to Equation (3)
        Call func RCOIS($I_j$, $Nch(I_j)$)// Request $Nch(I_j)$ chunks on $I_j$
        Calculate $Th_i(I_j)$;
        $Th_{old}(I_j) \leftarrow Th_{ov}(I_j)$;
        $n \leftarrow n$ - $Nch(I_j)$;
    end while
done
End of SDLAS
```

2) Algorithm for Parallel Dynamic Link Aggregation

```
Input: $n$, $m$
Input: $p$ //number of parallel connections per chunk
For all the available interfaces, j:1 to m
do
    Call func RCOIP($I_j$,1,$p$);
    Calculate $Th_i(I_j)$; // Refer Equation (1)
    $Th_{old}(I_j) \leftarrow Th_i(I_j)$ ;
    while $n > 0$ do
        $Nch(I_j) \leftarrow$ CRCI($I_j$, $Th_i(I_j)$, $Th_{old}(I_j)$) //calculates $Nch(I_j)$ to be sent on $I_j$ in next interval, by calculating $Th_{ov}(I_j)$ according to Equation (2) and calculating $Nch(I_j)$ according to Equation (3)
        Call func RCOIP($I_j$,$Nch_{I_j}$, $p$);
        calculate $Th_i(I_j)$; // Refer Equation 1
        $Th_{old}(I_j) \leftarrow Th_{ov}(I_j)$;
        $n \leftarrow n$ - $Nch(I_j)$;
    end while
done
End of PDLAS
```

IV. SIMULATION SETUP AND PERFORMANCE RESULTS

To measure the performance of proposed DLAS algorithms, we have simulated the DLAS algorithms in NS-3 [11]. An experimental setup where a grid consisting several UEs deployed within a HetNet is created as shown in Fig. 6. The UEs are distributed uniformly all over the grid. UEs are initially connected to the nearest Femto cell/Wi-Fi AP through one interface and to the Macro BS through another interface. Four experimental scenarios are considered. In first scenario, only single LTE interface is connected to Macro BS for data transfer. In second scenario, only single interface is connected to Wi-Fi/Femto for data transfer. In third scenario, one interface is connected to Macro and other interface to Wi-Fi/Femto. Reference link aggregation scheme (ALA) given in [9] is implemented and final scenario is where one interface is connected to macro and other interface connected to Wi-Fi/Femto. A scenario where one interface is connected to Femto and other interface to Wi-Fi is out of scope of this work.

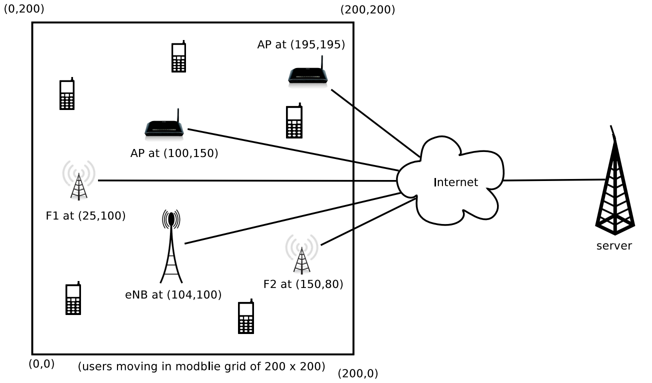

Fig. 6: Experimental setup in NS-3

User mobility is another important aspect to be considered. Even though the user is static, to mimic real-time scenario the user is assumed to have some mobility. Users mobility might trigger hand-offs. To replicate the current LTE deployment scenario, hand-offs between Wi-Fi APs (non trusted network) have not been implemented where as X2-interface which facilitates handoff between any two eNB (Macro) or HeNB (Femto (trusted network)) is implemented [12]. The Random walk mobility model was used for movement of UEs inside the grid with a velocity of 1 to 3 m/s.

The various simulation parameters used in the simulation are given in the Table II. The performance of proposed algorithm is measured with the following metrics:

Aggregate Throughput: Throughput is aggregated over two interfaces. Since the maximum throughput is closer to link capacity, we use it to essentially identify the link usage efficiency.

End-to-End Delay: The overall delay refers to the time taken for whole file to be transmitted across a network from server to client.

Fairness: It determines whether users or applications are receiving a fair share of system resources (bandwidth in this case). TCP fairness requires that a newly implemented TCP scheme should not have larger share of the network resource than a regular TCP scheme. This is important as TCP is the dominant transport protocol on the Internet, and if a newly implemented network protocol acquires unfair capacity of a link it may lead to large data transfers resulting in unfair distribution of bandwidth amongst other users in the network. We are using Jain's Fairness Index [13] to calculate the fairness of our algorithms. Formula to calculate Jain's fairness index is:

$$J(x_1, x_2, ..., x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2} \quad (4)$$

where, n is the no of users and x_i is throughput for the i^{th} connection/user. The result ranges from 1/n (worst case) to 1 (best case), and it is maximum when all users receive the same allocation. This metric identifies underutilized channels and is not unduly sensitive to a typical network flow patterns.

TABLE II: Simulation Parameters

Parameters	Values
Network Dimensions	200 m X 200 m
Number of Wi-FI APs	5
Number of Femto, Macro BSs	5, 1
Number of UEs	10
Wi-Fi Technology	802.11g
Wi-FI Modulation	ERP-OFDM
Wi-Fi PHY rate	24 Mbps
Macro, Femto PHY rates	25 Mbps, 24 Mbps
Macro, Femto BS Tx powers	46 dbm, 20 dbm
TCP File Size	1 MB
TCP MSS	512 B
Mobility Model	Building
UE Velocity	1m/s to 3 m/s
Path Loss Model	Indoor
Femto Coverage range	40 m
Number of seeds	10

A. Performance Results

Performance Analysis of DLAS vs other ALA schemes in LTE-Wi-Fi HetNet

Fig. 7 shows aggregate throughput versus number of flows for various schemes under study. In Fig. 8 end-to-end delay versus number of flows is plotted for various schemes. Aggregate throughput values and end-to-end delay values correspond to all the four experimental scenarios mentioned above. In the four experimental scenarios, UEs are having dual interfaces, one interface is connected to Wi-Fi network and other is connected to Macro/femto BS of LTE network as point of access.

There is a 9 - 10% increase in aggregate throughput with the SDLAS algorithm when compared to ALA algorithm and 65% increase in aggregate throughput with the PDLAS algorithm over ALA algorithm. The reason for improvement in throughput and decrement in delay for proposed algorithm over ALA algorithm is that ALA had considered only instantaneous throughput for splitting the chunks of data packets over different interfaces. Since instantaneous throughput does not indicate the actual channel capacity: a high value of instantaneous throughput does not necessarily indicate good overall channel capacity where as lower value does not necessarily indicate bad channel capacity, hence weightage must be given to previous throughput measures. In ALA algorithm, file is divided in terms of packets (MSS size of TCP) which does not utilize the physical channel capacity to the fullest and results in higher end-to-end delays and lower aggregate throughputs.

There is a 60% increase in aggregate throughput of PDLAS when compared to SDLAS. PDLAS performs better than SDLAS, as in PDLAS data is transferred through multiple parallel flows, which decreases time required to finish the file transfer. In SDLAS the data transfer takes place only after receiving the requested chunks, hence results in more end-to-end delay and lowering of aggregate throughput.

Performance Analysis of DLAS vs other ALA schemes in LTE-Femto HetNet

In this section the effect of mobility over DLAS is discussed. Figs. 9 and 10 show aggregate throughput versus number of flows for non-mobile UEs and mobile UEs, respectively. Figs. 11 and 12 show end-to-end delay versus number of flows for non-mobile UEs and mobile UEs, respectively. Aggregate throughput values and end-to-end delay values correspond to all the four experimental scenarios mentioned above are plotted. In the four experimental scenarios UEs are having two interfaces, one interface is connected to Femto and other connected to Macro as the point of access.

It is observed that aggregate throughput is decreased incase of mobile users using DLAS (PDLAS and SDLAS) compared to non-mobile users, this is due to high mobility of users which causes frequent handover among Femtos. Handovers result in redirection of chunks and decrease overall throughput and increase overall delay. Even though mobility affects the performance of DLAS, still it performs superior over individual links.

The percentage of increase in throughput of PDLAS when compared to SDLAS is 65% without mobility and with mobility it is 60%. This reduction in percentage is due to the buffering data at the source Femto. When the nodes are mobile, source Femto buffers data and routes the data to the destination Femto by serving gateway (S-GW) using X2 interface. This routing has to happen less number of times in SDLAS, since it does not have parallel TCP connections. But in case of PDLAS, this routing has to happen multiple times as there are multiple TCP connections per chunk. Hence, there is a slight increase in the delay which leads to the decrease in expected throughput.

Fairness of DLAS Algorithm

In Figs 13 and 14 Jain's fairness index is calculated for DLAS algorithms. Here UEs are having two interfaces: one interface is connected to Wi-Fi/Femto and other is connected to Macro as point of access. So fairness index is calculated for both the interfaces individually, since DLAS algorithm must be fair on both the connected links/interfaces. Its observed that though the number of flows increase, fairness index does not dip much, which indicates that the DLAS algorithms are fairer to other TCP users in the network.

V. CONCLUSIONS

We proposed DLAS algorithm which considerably increases achievable throughput over multiple interfaces. It also dynamically alters the packet request depending on the available

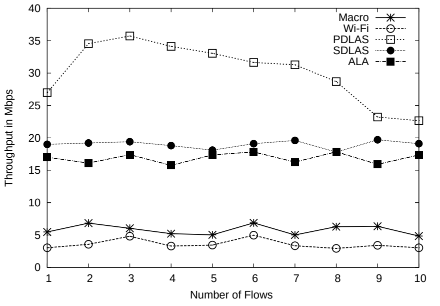

Fig. 7: Throughput for non-mobile users using Macro and Wi-Fi

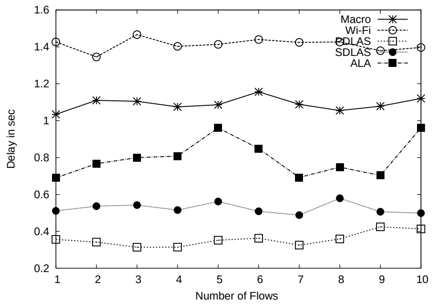

Fig. 8: Delay for non-mobile users using Macro and Wi-Fi

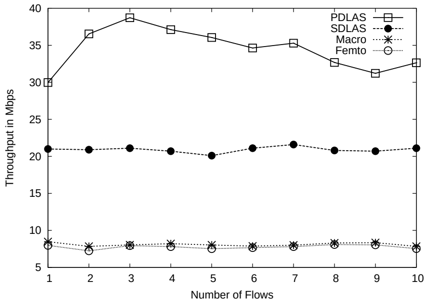

Fig. 9: Throughput for non-mobile users using Macro and Femto

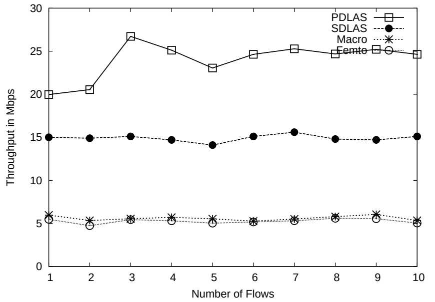

Fig. 10: Throughput for mobile users using Macro and Femto

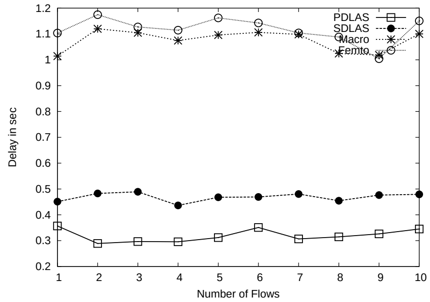

Fig. 11: Delay for non-mobile users using Macro and Femto

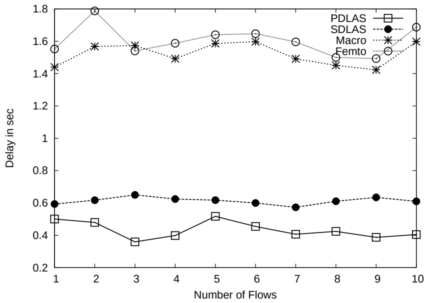

Fig. 12: Delay for mobile users using Macro and Femto

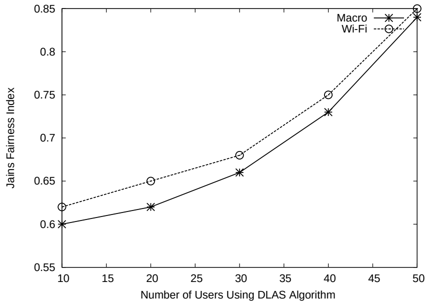

Fig. 13: Jains Fairness Index between Macro and Wi-Fi

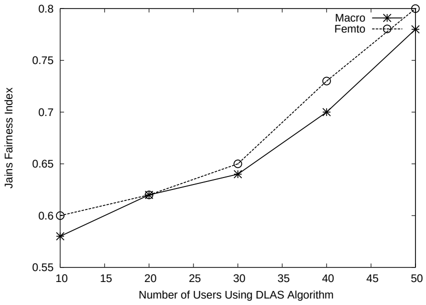

Fig. 14: Jains Fairness Index between Macro and Femto

bandwidth over each interface. In accordance to link capacity of the individual links, data is transferred over singular flows or Multiple parallel flows. It also incurs lower end-to-end delay and it is fair to other users in the network. We intend to extend this work with vertical hand-off scenarios.

REFERENCES

[1] S. Nielsen, "Lte evolving towards local area in release 12 and beyond." Future Radio in 3GPP, Nokia Corporation, 2012.

[2] "Views on rel-12 and onwards for lte and umts." Future Radio in 3GPP, Huawei Technologies, 2012.

[3] "Small Cells Forum." http://smallcellforum.org/smallcellforum.

[4] "Enterprise multi-femtocell deployment guidelines." Qualcomm Inc., 2011.

[5] M. Guijarro and R. Gaspar, "Experience and lessons learnt from running high availability databases on network attached storage," *Journal of Physics: Conference Series*, vol. 119, no. 4, p. 042015, 2008.

[6] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 4, pp. 388–403, 2006.

[7] D. Zhu, M. W. Mutka, and Z. Cen, "Qos aware wireless bandwidth aggregation (qawba) by integrating cellular and ad-hoc networks.," in *QSHINE*, pp. 156–163, Citeseer, 2004.

[8] P. Sharma, S.-J. Lee, J. Brassil, and K. G. Shin, "Handheld routers: Intelligent bandwidth aggregation for mobile collaborative communities," in *Broadband Networks, 2004. BroadNets 2004. Proceedings. First International Conference on*, pp. 537–547, IEEE, 2004.

[9] T. Takiguchi, A. Hidaka, H. Masui, Y. Sugizaki, O. Mizuno, and K. Asatani, "A new application-level link aggregation and its implementation on android terminals," *Wireless Communications and Mobile Computing*, vol. 12, no. 18, pp. 1664–1671, 2012.

[10] R. Rodrigues and P. Druschel, "Peer-to-peer systems," *Commun. ACM*, vol. 53, pp. 72–82, Oct. 2010.

[11] "Ns-3." http://www.nsnam.org/.

[12] "3GPP X2 application protocol (X2AP),TS 36.423 V10.4.0,Dec. 2011." http://www.3gpp.org/releases.

[13] "Jains Fairness Index." http://en.wikipedia.org/wiki/Fairness_measure.