M2: Using Visible Middleboxes to Serve Pro-Active Mobile-Hosts

Fahad R. Dogar and Peter Steenkiste
Carnegie Mellon University
Pittsburgh, PA, USA
{fdogar, prs}@cs.cmu.edu

ABSTRACT

While wireless Internet access is getting increasingly more popular, the current Internet architecture is not designed to address issues such as the unpredictable nature of the wireless medium and mobility. Furthermore, in the current architecture it is also difficult to make use of available opportunities like multiple options for a gateway or multiple network interfaces. To address these challenges, we propose M2: an architecture that relies on using *visible Middleboxes* to serve a *pro-active Mobile-Host*. M2 enables a number of techniques such as transport layer connection splitting, caching, pre-fetching, self-addressable ADUs, and cross layer information exchange. The combined use of a pro-active mobile-host and visible middleboxes specifically improves data transfer throughput in the presence of intermittent connectivity and variable network conditions. Our preliminary evaluation highlights the problems that exist in the current architecture and provides an estimate of the benefits that are achievable through M2.

Categories and Subject Descriptors

C.2.1 [**Computer Communications Network**]: Network Architecture and Design—*Wireless Communication*

General Terms

Design, Performance

Keywords

architecture, middlebox, mobile-host, mesh, vehicular

1. INTRODUCTION

The use of wireless devices to access Internet has seen a sharp increase in the last few years. The primary reason is the flexibility that it provides to the users, combined with its cost effectiveness. As a result, there have been widespread wireless deployments in the form of chaotic wireless networks [1] and enterprise/campus wifi networks, with other emerging scenarios such as wireless access using community mesh networks [3] or from moving vehicles [14] also getting more popular. This trend creates several challenges, such as the unpredictable nature of the wireless medium and handling mobility which may result in intermittent network access. The current Internet architecture, however, is not designed to address these challenges: the assumptions of negligible transmission losses and end-hosts always being reachable are unlikely to be true in a mobile wireless scenario. Furthermore, the current Internet architecture also makes it difficult to leverage several opportunities that have been identified in wireless scenarios. For example, it is difficult to switch between proxies/APs because the TCP connection is tied between two IP endpoints. This also makes it difficult to utilize multiple network interfaces, thereby resulting in under-utilization of available resources.

To address the above challenges, we propose M2: an architecture that uses *visible Middleboxes* to serve a *pro-active Mobile-Host*. Visible middleboxes result in flexible data transfer options and allow the mobile-host to make pro-active decisions regarding data transfer. In contrast, a pro-active mobile-host manages the pool of available M2 middleboxes that may be available and makes the decision of which middlebox to use, when to use them, and how to use them. This enhanced interaction between mobile-host and middlebox facilitates the use of existing techniques like transport layer connection splitting, caching, pre-fetching, self-addressable ADUs, and cross layer information exchange. Our contribution lies in combining these known techniques and showing how they can be implemented and better leveraged by incorporating visible middleboxes and pro-active mobile-hosts in the Internet architecture. We also conduct a preliminary evaluation to understand the extent of the problems in the current architecture and how M2 can make an improvement. Our results indicate that the ability to switch between middleboxes can improve the throughput in a mesh setting by approximately 30%. Similarly, the ability to cache and pre-fetch can also significantly improve the download time in a freeway setting where cars get Internet access through road-side access points.

The rest of the paper is organized as follows: In Section 2, we discuss the problems that exist in the current architecture using three motivating scenarios. In Section 3, we present M2 and discuss its design and some usage scenarios. We present our preliminary evaluation results in Section 4 and discuss related work in Section 5.

MobiArch'08, August 22, 2008, Seattle, Washington, USA.

2. MOTIVATING SCENARIOS

In this section, we present three problem scenarios that exemplify the shortcomings in the current architecture in addition to highlighting the available opportunities that remain un-utilized.

Flexible switching between Middleboxes: Recent studies [1, 6] have shown that multiple wireless access points are generally available to the user. Similarly, multiple options are available for use as a gateway in a mesh setting. Moving/Switching between these APs/gateways is hard because it may require change of an IP address or moving connection state that may be present at the middlebox. Consequently, once a user selects a certain middlebox as a gateway to the Internet, and starts a TCP connection, the connection is pinned down with that gateway for its whole duration. Even if the initial decision was judicious, it may not remain an optimal decision due to changes in network conditions. Therefore, this results in under-utilization of available resources because the connection is pinned down with a certain gateway that may no longer be the best one to use.

Enabling Wifi Access from Moving Vehicles in a Campus Network: Recent work has explored the possibility of providing wifi access from moving vehicles in a campus setting [14]. In this setting, mobility can be handled at the link layer because the APs are owned by a single enterprise. However, performance can degrade because of changes in network conditions and frequent switching between Access Points(APs). Some applications may not be tolerant of such disruptions and their performance may be severely impacted in these settings. Moreover, these disruptions may also have an adverse impact on TCP as packets are lost due to handovers. Finally, in the current architecture it is also difficult to use middleboxes(APs) in intelligent ways: to transfer content in case of mobility, to pre-fetch content at the next AP in the path of the vehicle etc.

Reducing Load on the Gateway in Mesh Networks: In a mesh network, nodes access the Internet through a gateway node. This results in the gateway becoming a hotspot and thereby causing congestion in the network. This compounds the existing problem of loss in throughput due to increasing hop length in a mesh setting [8, 3]. As locality may exist in the data that is transferred over the mesh network, there is a need for effective caching at the intermediate middleboxes (mesh proxies) in order to reduce hop length of data transfers. In the current architecture this requires application specific proxies but deploying them for a wide set of applications may be infeasible.

3. PROPOSED ARCHITECTURE

As the scenarios in Section 2 suggest, M2 is targeted towards the scenario where a user is interested in communicating with a fixed host while passing through one or more infrastructure based middleboxes, such as APs, mesh proxies or a road side units. The user can also be mobile, thereby possibly resulting in network disruptions.

The research community has developed a number of techniques that improve data transfers for mobile or wireless users. These include caching [8], transport layer connection splitting [2], cross layer information exchange [21] etc. Unfortunately, deploying these solutions in today's Internet is hard. The primary reason is that middleboxes are supposed to be transparent, thus limiting the functions they can perform. M2 combines the concepts of a visible middlebox and a pro-active client(mobile-host) to support such techniques in a general and elegant way. A visible middlebox means that the two endpoints of the connection know that the middlebox is part of the connection, so they can explicitly coordinate their data transfer related actions with it. We rely on a pro-active client since the client is often in the best position to control the optimization. For example, it knows which middleboxes are reachable, can monitor the wireless network, and knows what information the user still needs. As the middlebox is infrastructure based with only the end-host being mobile, it results in a more constrained setting compared to a Delay Tolerant Network (DTN) [10], where intermediate nodes can also be mobile. This constrained setting allows the mobile-host to be pro-active, enabling it to leverage visible middleboxes for achieving the desired objectives. We provide a detailed comparison of M2 with the DTN architecture in section 5

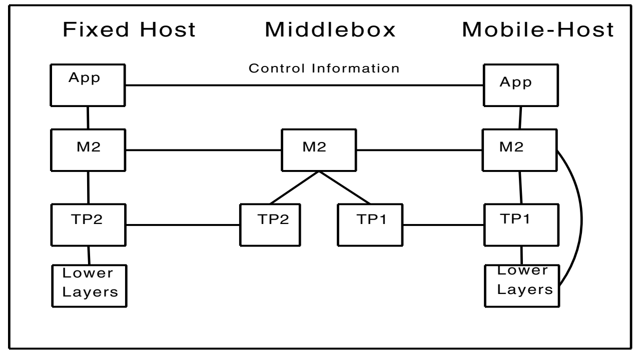

Figure 1: High Level Overview of M2

Fig 1 shows the high level overview of M2 where a mobile-host communicates with a fixed host through a single middlebox. The two main concepts of M2 are: visible middlebox and pro-active mobile-host. The key functionality is added through an M2 layer that sits in between the application and transport layers. The M2 layer also uses the basic building blocks like cross layer information exchange, split connection approach etc., for effective data transfer. Next we briefly outline these building blocks of M2 which is followed by a discussion on the mechanisms that need to be employed in order to fully leverage the key concepts and building blocks of M2.

3.1 M2 Building Blocks

We now discuss the various building blocks that are used in M2. Although these techniques have been used earlier, M2 combines and uses them with the two key concepts (visible middlebox and pro-active mobile-host) to achieve the desired objectives.

Splitting of Transport Connection: In M2, the end-to-end transport layer connection can be split into two connections: one on the wireless segment and the other on the wired segment. This was first proposed in I-TCP [2] with the vision that standard protocols which are known to work well on the wired segment, like TCP, will operate on the wired segment while customized protocols for the wireless environment and/or for the wireless device in use can be used on the wireless segment. This approach is also very much in line with the requirements of several recent proposals like ExOR [4] and COPE [13], that do not necessarily work well with TCP on the wireless segment and prefer a split connection approach.

Another key advantage of having a split connection approach is the decoupling of connections between the wired and wireless segments: it separates congestion control, flow control, and reliability on each segment. This ensures that TCP running on the wired segment does not start congestion control as a result of losses due to mobility on the wireless segment. Furthermore, frequent changes on the wireless segment are also hidden from TCP on the wired segment. Therefore, from the perspective of TCP running on the fixed host, the other end is very stable and conventional (middlebox in this case) compared to a highly volatile mobile-host.

In the current Internet architecture, however, the splitting approach results in *weaker* end-to-end communication semantics because an acknowledgement from the middlebox does not guarantee that the data has been received by the end party. Furthermore, connection state at the middlebox makes it harder to handle mobility because the state needs to be transferred to the new middlebox. Finally, the split connection results in *slight coupling* between congestion control on the wireless segment and flow control on the wired segment if the middlebox has a limited buffer. For example, if the wireless segment is the bottleneck, the buffer at the middlebox would fill up and then flow control would start taking place on the wired segment. We will later discuss how these issues can be addressed in M2.

Persistent Storage: In M2, middleboxes have persistent storage that can be used for a variety of purposes. During a transfer, it can be used to *hide* the variations on the wireless segment from the wired segment. This may reduce the slight coupling that may exist with a limited buffer, thereby ensuring that the TCP behavior on the wired segment is unaffected by the network conditions on the wireless segment. Once a transmission is over, the data can also be *cached* at the middlebox and later served to other mobile-hosts. As the mobile-host generally has multiple options for connecting to a gateway and is also moving, we can better exploit locality by having middleboxes cache data. The persistent storage can also be used for *pre-fetching* data. This option can be used if the mobile-host is moving and is likely to connect to a certain middlebox in the future. In this case, the persistent storage at the middlebox can be used to store data that would be served in future.

Self Addressable Application Data Units (ADU): In M2 the granularity of data transfer is at the level of ADUs [7]. Applications specify their ADUs to the M2 layer along with the corresponding delivery requirements. For example, a simple file transfer application can just divide the whole file into fixed size chunks, and specify TCP like reliability requirements while a video transfer application can use different frames (I, P etc.) as ADU and specify different requirements for them. In M2, these ADUs are named through their content hash (similar technique at the granularity of chunks has been used in [22]). This self-addressing nature of ADUs mean that we can do away with application level names and potentially need not implement application specific logic at the middleboxes. Because middleboxes can cache ADUs, future requests, even from a different application, can be served if there are common ADUs between applications. Therefore, we can exploit similarity across different applications as well.

Cross Layer Information Exchange: In the context of wireless networks, the use of cross layer information exchange is a well known technique [21]. The goal is to make it easier for the higher layers to adapt to the underlying wireless conditions, although in the current Internet architecture this goal may not be completely realizable. For example, even if the lower layer is reporting a better AP, it may not be possible to switch the TCP connection because of the requirement of changing IP address. In contrast, the pro-active mobile-host in M2 is in a better position to use this cross layer information exchange. The lower layer information that may be communicated to the M2 layer includes: available middleboxes and the corresponding available resources, network interfaces that may be available etc. We can also envision the use of metrics like ETT [9] or WCETT [9]. This is important because while the network conditions are more stable in static Wifi networks, there are significant variations in wireless mesh networks where the best gateway can change frequently or in vehicular networks where mobility makes the network conditions highly variable [14].

3.2 M2 Mechanisms

We now discuss the key mechanisms that we need to employ in order to effectively make use of the concepts and building blocks of M2. Towards this end, we discuss the *flexible transfer options* that are available in M2 and how they can address the semantics problem that is present in the current architecture. We also discuss how a mobile-host needs to *manage a pool of visible middleboxes* and employ *pro-active decision making* in order to fully leverage the M2 architecture.

Flexible Transfer Options: Having visible middleboxes allow applications to implement data transfers in diverse ways and to support them with appropriate communication semantics. For example, the server can delegate its responsibility to the middlebox: as a result it can leave/exit after transferring all the contents to the middlebox without worrying about the subsequent transfer to the mobile-host. In this case, the middlebox can also cache the data and act as an authoritative server for other clients. Alternatively, the mobile-host can give complete authority to the middlebox. The middlebox can act on behalf of the mobile-host while communicating with other peers in a peer-to-peer setting or with a server. Finally, we can also envision a middle ground where role of the middlebox is limited and its nature is defined by the two end parties.

These elaborate transfer roles can be suitably supported by richer communication semantics. Existing TCP-like semantics or semantics that are more/less strict compared to TCP can be implemented at the M2 layer. For example, we can address the semantics issue arising out of the use of split connection approach by adding global acknowledgements at the M2 layer. These richer semantics not only allow diverse application requirements to be implemented but also makes the communication more secure and accountable as the role of every party is clear and known to everyone.

Managing pool of visible middleboxes: As the middleboxes are visible, the mobile-host can explicitly use them. Towards this end, it has to manage information about the available middleboxes and how they can help in data transfers. This management requires knowledge about the ADU availability at each middlebox as ADUs may be present in the persistent storage of the middlebox. Furthermore, the mobile-host can maintain and exploit information regarding state of the connection with the middlebox. This information can be obtained through cross layer exchange of lower layer information such as signal strength, available bandwidth etc. Therefore, managing the pool of visible middleboxes allows the mobile-host to know: i) which middleboxes are available, ii) ADU availability at these middleboxes, and iii) the path characteristics between the mobile-host and the middleboxes.

Pro-Active Decision Making: The management of information regarding visible middleboxes allows the mobile-host to make decisions regarding data transfers. It makes the decision of *which* middlebox to use, in case multiple options are available, *when* to use them if the user is mobile,and *how* to use them(caching, pre-fetching etc.). Towards this end, it employs a pull-driven approach and makes the decision at the granularity of ADUs.

One of the benefits of the use of ADUs is that they can be used as a unit of recovery. This ensures that no transfer state needs to be maintained at the middleboxes to handle mobility. For example, if the mobile-host is disconnected during the transfer of a certain ADU, once it is re-connected, it can again make a pull request for the same ADU. This makes mobility much easier to handle because there are no handoffs or large scale disruptions — the mobile-host simply requests again for the lost ADU. Finally, by pre-fetching ADUs, it is easier to avoid losses or disruption: If the mobile-host is likely to experience dis-connectivity because of mobility, it can pre-fetch ADU/ADUs at the next middlebox.

A pull based approach towards transferring ADUs is based on the rationale that the mobile-host is in a better position to know its requirements, as well as its available options with respect to transferring data. Furthermore, because mobile-host can move around, causing short or long disruptions, the onus of getting data is on the mobile-host rather than on the network infrastructure. This makes routing easier in the presence of frequent disruptions, because the mobile-host can pull data on its own, whenever it is connected to any infrastructure.

3.3 M2: Usage Scenarios

In order to illustrate how M2 can improve performance through the interplay of visible middleboxes and pro-active mobile-host, we now describe the working of M2 in the three motivating scenarios that we described in Section 2.

Flexible switching between Middleboxes: As the mobile-host manages information about the visible middleboxes, it knows the availability of ADUs at a certain middlebox as well as whether the path to a middlebox is productive or not. Based on this information, it will know that the old middlebox is no longer the best one to use and therefore it will switch to another(better) middlebox. Note that this switching would be quite simple as the granularity of decisions is an ADU — the mobile-host can just request the new middlebox to resume transfer of the remaining ADUs.

Enabling Wifi Access from Moving Vehicles: In this scenario, the path is fixed i.e., the sequence of APs that the van is likely to associate with is known a-priori. This information can either be entered by the user or the lower layers can provide this information to the mobile-host. Based on this information, the mobile-host can pre-fetch ADUs that are likely to be affected by the handover. Towards this end, it will contact the next AP[1] and then request for the relevant ADUs to be fetched and stored at the middlebox. The mobile-host will know once it moves into the vicinity of the next AP through lower layer information and it can then decide to fetch the already stored ADUs from the new AP.

Reducing load on the gateway in mesh networks: In this situation, we can envision the use of M2 at all the nodes in the mesh network. Nodes along the path of a gateway will also cache ADUs. As the mobile-host manages the pool of middleboxes available and their associated resources, it would know the ADU availability at each middlebox. This information can then be used to fetch ADUs from middleboxes other than the gateway, thereby reducing the chances of the gateway becoming the bottleneck. Towards exploiting caching in mesh networks, M2 provides two key features: i) caching is application independent because it is done by the M2 layer using content based addressing and ii) constant use of lower layer information in decision making also helps in avoiding hot-spots as those paths/middleboxes are avoided which are congested.

4. PRELIMINARY EVALUATION

In this section, we present some preliminary evaluation results that depict some problems in the current architecture and provide an estimate of the benefits possible through M2. We consider two scenarios: i) data transfer in a mesh setting through a gateway and ii) vehicular wifi access in a freeway setting.

Mesh Setting: As discussed earlier, in a typical mesh setting the connection is pinned down with a gateway for the whole duration of a session. We investigate the impact of *connection pinning* on performance by conducting a trace driven analysis of data from a campus wireless testbed [18]. This data consists of the ETT[2] value for the "link" between every pair of nodes in the network. We randomly select gateway nodes in the topology and based on this selection, cost (in terms of ETT) is calculated between each node and the gateway.

We define four scenarios to capture the various possibilities in these settings. The `First-Best` scenario refers to the current practice of selecting the best gateway at the start and then sticking with this gateway for the whole duration of the session. We compare this with the `Flexible` scenario which shows the cost incurred if the gateway selection is flexible, as is the case in M2. Finally we also provide the upper and lower bounds for the performance in the current architecture: `Worst-Avg` scenario shows the performance if the gateway with the worse average throughput for the whole duration of the session is selected at the start. Likewise, the `Best-Avg` case represents the best case performance achievable through this approach of sticking to a single gateway.

Figure 2(a) shows the performance for a 40 min session with 5 gateways in all four scenarios. The worse case is clearly very bad in terms of cost whereas the `Flexible` approach provides around 30% benefit over the other two scenarios(`Best-Avg` and `First-Best`). Figure 2(b) shows the impact of increasing the number of gateways on the various schemes. In the `Flexible` approach, the cost decreases as

[1]Note that this communication may go through the first AP.
[2]ETT is the Expected Transmission Time of a packet, considering the delivery ratio and available bandwidth on a link [9].

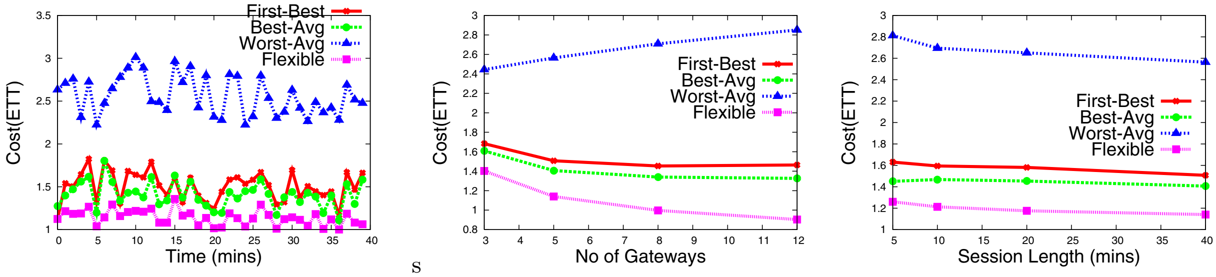

Figure 2: Connection Pinning in a Mesh Network

the number of options increase whereas in the fixed gateway scenarios, the cost remains relatively fixed. This indicates that the flexible approach is essential in order to benefit from the emerging scenario where the mobile-host is likely to have several options for AP/mesh proxies. Finally, figure 2(c) shows that the benefits of `Flexible` approach is there even with shorter session duration. Therefore, it is likely to benefit a wide variety of applications that rely on downloading data.

Vehicular Wifi: We consider a setting where a car is moving on a freeway at typical driving speeds of around 100 km/h. It connects to an AP and tries to download data from a server. The connection between the AP and the server is the bottleneck with available bandwidth of 1 Mbps. Based on recent measurement studies [11, 15] we select values for parameters, including the session duration between the car and the gateway (15 seconds), and expected throughput (4 Mbps). Furthermore, we also assume that the distance between two APs is 2 kms which is the typical distance between two emergency phones on a freeway and thus can be used as a minimum distance between two APs.

Based on these settings, we calculate the time taken to download typical mp3 and movie files. In the caching/pre-fetching case, we assume that the required data is available at the AP, either because it was cached due to an earlier transfer or it was pre-fetched (as we know the route and thus possibly the APs). The estimate shows that caching/pre-fetching can significantly improve the data transfer throughput and enable faster downloads of likely file sizes, such as an mp3 file (5MB) or a movie file (700MB). For the mp3 file, with caching/pre-fetching we can complete the download from a single AP in `10 seconds` whereas without caching it requires intermittent download from multiple APs requiring `150 seconds`. Similarly, for a movie file without any caching/pre-fetching, it takes `7.5 hours` while with caching/pre-fetching the time reduces to `1.8 hours`.

5. RELATED WORK

Proposals on delay tolerant networking [10, 19, 17] aim to take a holistic view of the challenges involved in the emerging wireless scenario. They consider mobility, intermittent access, and opportunistic communication in one way or the other. The design of M2 is inspired by several key components of DTNs: Like the bundle layer in DTNs, the M2 layer also sits in between the transport and application layers. Similar to DTN intermediaries, M2 middleboxes also have persistent storage that can be used to cache the content. Finally, notion of ADUs that can be specified by the application is also common in both M2 and DTNs.

Unlike DTNs, M2 considers a more limited scenario where only the end-host is mobile — middleboxes are infrastructure based. This limited scenario entails that routing in M2 is less complicated compared to DTNs where the routing algorithm has to account for network partitions that can happen anywhere in the network. In M2, disruptions can only occur at the edges where the end host is mobile. Therefore, whenever the mobile-host is connected to a middlebox, end-to-end reachability is similar to what we observe in the current Internet. This allows us to employ a pull-driven approach where the mobile-host makes active decisions of how to fetch data. The differences in the rights of data transfer result from different goals of the two architectures, with DTNs providing resilience and connectivity in the presence of extremely adverse network conditions while M2 tries to improve/optimize data transfer in the presence of mobility at the edges only. Furthermore, because M2 relies on existing routing techniques for connectivity between nodes, transition to the traditional scenario (no mobility, single interface/middlebox) is easier compared to DTNs. Finally, unlike DTNs, M2 also introduces the notion of using content based addressing to name ADUs, thereby providing an application independent way to cache data.

There have been several proposals to address the challenge of reachability as a result of user mobility [16, 20]. Compared to these proposals, the focus of M2 is more on improving data transfer performance rather than addressing the issue of reachability. In M2, reachability is ensured through the use of pull based decision making at the granularity of ADUs. As a result, M2 focuses on the situation where the mobile-host initiates the data transfer. Other proposals can be used in case where another node initiates communication with a mobile-host [16, 20]. Similarly, there is a whole body of work on performance enhancing proxies and making middleboxes *work* in the current Internet architecture [5, 12]. We believe that by making middleboxes visible we reduce the problems that arise out of using hidden middleboxes and enable more effective use of middlebox functionality.

The M2 building blocks have been used in other contexts. Ideas of caching, pre-fetching, content based addressing, splitting transport layer connections, and ADUs are well known. Our contribution lies in combining these techniques with a *visible middlebox* and a *pro-active mobile-host*. For example, separating data transfer from application control logic through content based naming has been used in DOT [22]. It also has a notion of dividing the file into chunks and employing a pull-driven approach towards data transfer. Similarly, the idea of splitting connections has been the focus of several studies [2]. M2 extracts more benefits out of these techniques compared to the current Internet architecture by having visible middleboxes and pro-active mobile-hosts.

6. CONCLUSION

In this paper, we proposed M2: an architecture that relies on using *visible Middleboxes* to serve a *pro-active Mobile-Host*. Compared to other architectural proposals, M2 considers a more specific scenario where a mobile-host accesses the Internet through infrastructure based middleboxes. M2 targets better throughput in this scenario by addressing key challenges such as connection pinning, mobility, and changing network conditions. Similarly, it makes use of available opportunities that may be present in the form of multiple options of gateways(middleboxes) or network interfaces. We discussed how a visible middlebox and pro-active mobile-host can address these challenges and make use of the available opportunities. Our preliminary evaluation also highlighted the need for M2 by showing its potential benefit in specific scenarios such as mesh networks and vehicular wifi environments. Our future work entails building a prototype of M2 and evaluating it under several existing and emerging wireless scenarios.

7. REFERENCES

[1] A. Akella, G. Judd, S. Seshan, and P. Steenkiste. Self-management in chaotic wireless deployments. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 185–199, New York, NY, USA, 2005. ACM Press.

[2] A. V. Bakre and B. Badrinath. Implementation and performance evaluation of indirect tcp. *IEEE Transactions on Computers*, 46(3):260–278, 1997.

[3] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 31–42, New York, NY, USA, 2005. ACM Press.

[4] S. Biswas and R. Morris. Exor: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.*, 35(4):133–144, 2005.

[5] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance enhancing proxies intended to mitigate link-related degradations, 2001.

[6] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 50–61, New York, NY, USA, 2006. ACM.

[7] D. D. Clark and D. L. Tennenhouse. Architectural considerations for a new generation of protocols. In *SIGCOMM '90: Proceedings of the ACM symposium on Communications architectures & protocols*, pages 200–208, New York, NY, USA, 1990. ACM.

[8] S. M. Das, H. Pucha, and C. Y. Hu. Mitigating the gateway bottleneck via transparent cooperative caching in wireless mesh networks. *Ad Hoc Networks (Elsevier) Journal*, Special Issue on Wireless Mesh Networks, 2007.

[9] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128, New York, NY, USA, 2004. ACM.

[10] K. Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03*, pages 27–34, New York, NY, USA, 2003. ACM.

[11] R. Gass, J. Scott, and C. Diot. Measurements of in-motion 802.11 networking. In *WMCSA '06: Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications*, pages 69–74, Washington, DC, USA, 2006. IEEE Computer Society.

[12] S. Guha and P. Francis. An end-middle-end approach to connection establishment. In *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 193–204, New York, NY, USA, 2007. ACM.

[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: practical wireless network coding. In *SIGCOMM '06*, pages 243–254, New York, NY, USA, 2006. ACM Press.

[14] R. Mahajan, J. Zahorjan, and B. Zill. Understanding wifi-based connectivity from moving vehicles. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 321–326, New York, NY, USA, 2007. ACM.

[15] J. Ott and D. Kutscher. Drive-thru internet: Ieee 802.11b for automobile users. In *Infocom '04*, pages 373 –. IEEE, 2004.

[16] C. Perkins. Mobile ip. In *International Journal Of Communication Systems*, pages 3–20, 1998.

[17] M. J. Pitkänen and J. Ott. Redundancy and distributed caching in mobile dtns. In *MobiArch '07*, pages 1–7, New York, NY, USA, 2007. ACM.

[18] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth. Routing stability in static wireless mesh networks. In *Passive and Active Network Measurement*, pages 73–82, Berlin/Heidelberg, Germany, 2007. Springer.

[19] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A networking architecture designed around mobile users. In *The Third Annual IFIP Conference on Wireless On-demand Network Systems and Services (WONS 2006)*, 2006.

[20] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *SIGCOMM '02*, pages 73–86, New York, NY, USA, 2002. ACM.

[21] K. Sundaresan and K. Papagiannaki. The need for cross-layer information in access point selection algorithms. In *IMC '06*, pages 257–262, New York, NY, USA, 2006. ACM.

[22] N. Tolia, M. Kaminsky, D. G. Andersen, and S. Patil. An architecture for Internet data transfer. In *Proc. 3rd Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, CA, May 2006.