

# PiSA-SA: Municipal Wi-Fi Based on Wi-Fi Sharing

Tobias Heer, Thomas Jansen\*, René Hummen, Stefan Götz, Hanno Wirtz, Elias Weingärtner, Klaus Wehrle  
RWTH Aachen University, Distributed Systems Group, Germany

Email: {heer, hummen, goetz, wirtz, weingaertner, wehrle}@cs.rwth-aachen.de, \*jansen@mithi.net

**Abstract**—Large-scale municipal wireless networks are currently being established all around the world. These networks provide a rich set of local services, such as tourist guides, environmental information, pedestrian navigation, and local shopping guides. As recent financial failures of prominent municipal wireless networks show, it is economically challenging to achieve the bandwidth and coverage that is necessary for such a network. At the same time, Wi-Fi-sharing communities achieve high bandwidth and good coverage at a very low cost by capitalizing on the dense deployment of private access points in urban areas. However, from a technical, conceptual, and security perspective, Wi-Fi sharing community networks resemble a patchwork of heterogeneous networks instead of one well-planned, uniform and secure network as required for the economic success of a municipal Wi-Fi project. In this paper, we show how to realize municipal wireless services on top of a Wi-Fi-sharing infrastructure in a technically sound and economically attractive fashion while taking into account legacy devices and mobile clients. Our solution cleanly separates the roles of controlling and administering the network from providing bandwidth and wireless access. This allows municipalities to focus their resources on municipal wireless services instead of providing Wi-Fi access.

## I. INTRODUCTION

Municipal Wi-Fi (Muni-Fi) networks are being installed in many cities<sup>1</sup> around the world. Their goals include ubiquitous Internet access, localized services (e.g., city and event guides, traffic information, etc.), and simplified data collection (e.g., traffic monitoring and meter reading). Besides these classical goals, a ubiquitous network can provide a platform for third-party service providers, enriching the service set with new and innovative mobile services, such as parking spot search, digital orders and reservations in restaurants, and mobile gaming. It is hoped that Muni-Fi networks will help bridging the digital divide, stimulate innovation, support economic growth, and increase city operations efficiency [1]. However, the cost of deploying, maintaining, and operating such networks has hampered or even prevented their proliferation in many cases. Recent prominent examples of the financial risks involved in building and operating a Muni-Fi are the discontinuation of the public wireless service in St. Cloud and the continuous economic struggles of Wireless Philadelphia.

Wi-Fi sharing communities are a very cost-effective alternative to providing city-wide Wi-Fi since the financial burden is split among all members. Their concept is that the community members make their privately owned Wi-Fi access points (APs) available to each other. This approach initially

emerged from grass-root movements such as the Freifunk [2] communities and was later adopted by companies such as FON [3].

However, community models generally do not meet all requirements of Muni-Fi networks. On the one hand, open and de-centralized communities lack security and trustworthiness. On the other hand, approaches managed by a single central provider can ensure security but not openness and competition within in the system. A central provider has no incentive to delegate access control or to invite competitive third-party service providers that are desirable in Muni-Fi networks.

In previous work, we presented the Peer-to-Peer Internet Sharing Architecture (PiSA) [4], which can establish security and trust in open, de-centralized Wi-Fi sharing communities. Our contribution in this paper is the PiSA Service Architecture (PiSA-SA), which generalizes secure, community-based Wi-Fi sharing to meet the demands of Muni-Fi networks. In particular it enables secure and mobile service provisioning based on the incorporation of private and municipal wireless AP operators and service providers. PiSA-SA clearly separates the roles of community operators, wireless access providers, and service providers. It shifts the requirements for a community operator from being a network service provider (i.e., providing wireless access and bandwidth) to being a pure control and management instance. At the same time all other parts of the network can be implemented in a peer-to-peer-like fashion between users and service providers. Hence, PiSA-SA enables Muni-Fi networks that can be considered as provider-less because no classical provider-centric networks are required.

The remainder of this paper is structured as follows: In Section II, we give an overview of the roles and stakeholders in a Muni-Fi system. Section III provides an introduction to the PiSA system. Section IV extends the concept of PiSA to meet the requirements for a decentralized municipal Wi-Fi sharing system. Section V shows the performance of our prototype and evaluates the feasibility of our approach. Section VI discusses deployment considerations, Section VII presents related work, and Section VIII concludes the paper.

## II. MUNICIPAL WI-FI MODEL

A Muni-Fi model has to accommodate four different stakeholders that exist at the core of virtually all of today's Muni-Fi networks: a *community operator*, municipal *service providers*, *Wi-Fi providers*, and *users*.

The *community operator* manages the Muni-Fi system as a whole. Its responsibility is to define usage and access

This work is supported by the Ziel2.NRW program and the ERDF fund of the European Union. Europäische Union - Investition in unsere Zukunft

<sup>1</sup>In January 2008, <http://www.muniwireless.com/> listed 395 planned and completed Muni-Fi projects in the US alone.

rules for the system. Typical community operators are city administrations or companies that manage a Muni-Fi.

*Service providers* offer services in the Muni-Fi network. Municipal services can either be offered by the municipality or by third-party service providers. Possible services are manifold and range from simple WWW-like services to interactive location-based services. Service providers can also offer special services to government staff (e.g., access to environmental sensor data, remote meter reading, and emergency information). The community operator controls the set of offered services to prevent misuse of the Muni-Fi network by rogue service providers. In a Muni-Fi, there is typically a clear distinction between municipal services and Internet access: Municipal services can be reached directly and freely from within the municipal network, whereas access to the Internet often requires additional user registration or payment for security and profitability reasons.

*Wi-Fi providers* own and operate wireless access points that give access to the municipal network and its services. Such providers can be formed by companies, for example ISPs, or by governmental and non-profit institutions, for instance municipalities. In addition, even citizens may act as micro operators by sharing their APs and Internet connection.

Finally, mobile and nomadic *users* use the wireless access offered by Wi-Fi providers to access the Internet or special municipal services offered by service providers.

In practice, a single organization can assume several roles. For example, in a provider-centric network, the provider alone holds the roles of the community operator, service provider, and Wi-Fi provider, offering network and services to users. A second example is a municipality that sub-contracts the provisioning of the wireless infrastructure to an ISP. In such a scenario, the municipality embodies the community operator and the service provider, whereas the ISP acts as a Wi-Fi provider. However, when an organization holds multiple different roles or monopolizes one role, it can easily dictate network access conditions or prevent service diversity. Thus, restricting the right to provide wireless access or services to a single organization can hamper competition and innovation.

### III. THE PiSA SYSTEM

The basis of this work is the Peer-to-Peer Internet Sharing Architecture, a Wi-Fi sharing system that puts special emphasis on scalability, openness, security, and user mobility. Figure 1 depicts its four main components and their relations: A *mobile device* or *mobile guest* is the device of a mobile or nomadic user. The *host AP* is the community access point that the mobile device is currently connected to. The *trusted relay* is a router that the user has access to (typically, the user's own AP at home). The *community operator* is a logical entity that certifies community membership of the trusted relay by means of a digital community certificate.

The basic concept of PiSA is to allow mobile users nothing but to open a single encrypted tunnel to their trusted relay over the Internet connection provided by the host AP. This tunnel between the mobile device and its trusted relay is

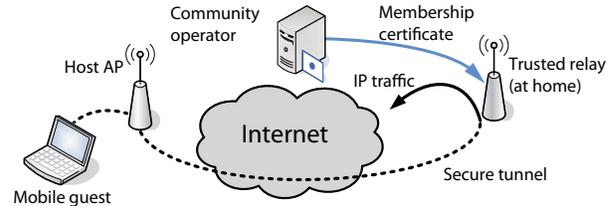


Fig. 1. The PiSA Wi-Fi sharing model.

the only communication permitted by a host AP. Thus, the trusted relay acts as the ingress point to the Internet for the mobile guest. This VPN-tunnel-like setup protects the mobile user from eavesdroppers at any point in the Wi-Fi sharing infrastructure. It also protects the AP provider from being liable for the actions of mobile users since potentially illegal traffic to and from the Internet always relates to a mobile device's trusted relay.

PiSA uses the Host Identity Protocol (HIP) [5] for security and mobility. Located between the network and transport layer, HIP provides a cryptographic name-space such that a host is not identified by an IP address but via a public key, the Host Identity (HI). At layer 3.5, HIP and PiSA provide their services transparently to protocols from the transport layer upwards. For compatibility with IPv6, HIP uses self-certifying hashes of the HI as Host Identity Tag (HIT). When two peers establish a connection via their HITs, HIP translates the HITs to IP addresses, verifies the identity of the peers, and establishes an IPsec tunnel between them. In PiSA, this is the tunnel between a mobile guest and its trusted relay. PiSA extends HIP for end-to-middle authentication and signaling [6], [7]. This enables middleboxes (e.g., the host APs) to authenticate the connection between the mobile guest and the trusted relay. In addition, the SPKI digital certificate support in HIP [8] is employed for representing and validating the community membership of the mobile device and its trusted relay.

Apart from private, personal trusted relays at the user's premises, PiSA also supports central, shared, and potentially commercially operated trusted relays. The latter are of interest for larger organizations, such as commercial providers, universities, or the municipality, with trusted user groups. Typically, such organizations would not only bring a large number of mobile users into the community, but in return make larger amounts of existing or newly installed access points available to the community.

#### A. PiSA for Municipal Wi-Fi

PiSA enables secure large-scale Wi-Fi sharing without a dedicated network provider. Therefore, it is a candidate for supplying mobile users with Internet access in urban areas. However, one of the main distinctions between a Wi-Fi sharing community and a Muni-Fi network is the possibility to provision municipal services. In centralized provider-centric scenarios, the provider is in control of a homogeneous network with a common trust and security level. Hence, services can be implemented and accessed in this single provider-managed network. For the original PiSA Wi-Fi sharing system, as a provider-less and peer-to-peer-like network, the provision of

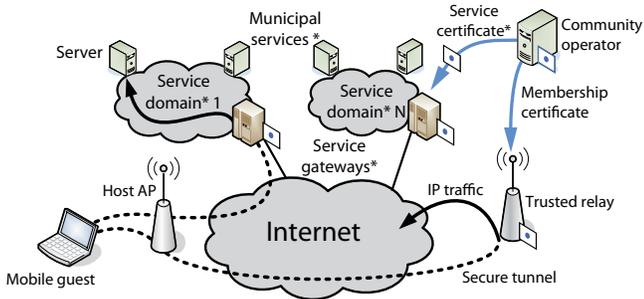


Fig. 2. PiSA-SA overview. Elements marked with \* are special to PiSA-SA.

services is particularly challenging for two reasons:

a) *No common network*: PiSA reduces the responsibilities of the community operator to pure membership management without furnishing network connectivity. Instead, Internet access is handled in a decentralized fashion between the mobile device and its personal trusted relay at home. Hence, PiSA lacks a common openly accessible network for all users, in which services could be provisioned, accessed, and controlled. Thus, PiSA requires a way of managing services independent of a common shared network and without a single dedicated service provider to preserve its distributed character.

b) *No support for legacy clients and open services*: PiSA requires mobile devices to run customized software and to have access to a PiSA-enabled trusted relay. Clearly, these requirements cannot be met in Muni-Fi networks in general. For example, tourists and travelers without these prerequisites should still be able to use the network to access publicly available information and services. To serve as a general Muni-Fi solution, PiSA must implement basic support for openly accessible services and legacy clients without special software.

#### IV. THE PiSA SERVICE ARCHITECTURE - PiSA-SA

The PiSA Service Architecture is an extension of the basic PiSA Wi-Fi sharing system and enables the decentralized provision of Muni-Fi services as well as legacy client support in PiSA. In contrast to provider-oriented Muni-Fi models, PiSA-SA separates the roles of community operator, service provider, and Wi-Fi provider, and allows distributing these roles to many parties. For example, in PiSA-SA the system may consist of several trustworthy community operators, a heterogeneous set of different Wi-Fi providers (e.g., user-provided micro operators), and a heterogeneous set of service providers (e.g., the municipality, third-party companies, or even users). PiSA-SA reduces the role of the community operator to merely controlling the set of services available in the network. This control is achieved by issuing certificates to trusted services instead of providing connectivity and bandwidth to these.

This section discusses the additional elements introduced by PiSA-SA (see Figure 2). In the following, we illustrate A) how services can be addressed and contacted, B) how community operators can control the set of services to protect the network from misuse, C) how the system can support legacy clients and applications, and D) how user mobility is handled for native PiSA-SA clients as well as for legacy clients without PiSA-SA support.

#### A. Services and Service Gateways

PiSA addresses the legal issues and security threats in Wi-Fi sharing by using a secure tunnel between the mobile device and its trusted relay at home. For a well-defined closed set of benign services in a Muni-Fi network, these legal problems and security concerns are far less problematic. Hence, requiring every user to operate a trust point at home is neither required nor does it provide additional benefits for the Muni-Fi network, but on the contrary limits the performance because of triangular routing. To prevent these issues, PiSA-SA thus grants mobile clients direct access to a defined set of municipal services in addition to PiSA's secure tunnels.

The trusted relays in PiSA can already be seen as a service that provides Internet connectivity to mobile devices. PiSA-SA extends this concept by treating municipal services as additional relays to remove the dependency on the trust point. Services are not accessed through a single tunnel to one trust point, but instead via multiple direct tunnels to the services. Therefore, PiSA-SA must address the question how to set up the different tunnels and how to route packets to the correct services.

Since PiSA and HIP conceptually operate between the network and the transport layer, we approach the addressing and routing problems by introducing a managed virtual IP address space  $S$ . Every service is uniquely addressable via an IP address  $IP_s$  in  $S$ . Addresses in  $S$  are used by the transport layer and above. Thus, upper layers only handle addresses in  $S$  and are unaware of the different tunnels at the PiSA-SA layer. Whenever the PiSA-SA layer receives a packet from an upper layer with an address  $IP_s$  for which it does not have a previously established tunnel readily available, PiSA-SA resolves  $IP_s$  to the cryptographic identity  $HIT_s$  and the routable IP addresses  $IP_r$  of the corresponding service. The mobile client then opens an IPsec tunnel to the service by initiating a HIP and PiSA handshake to it. The address resolution is performed by a dedicated DNS server operated by the community operator.

In the course of the handshake between the mobile client and the service, both the client and the service authenticate against the host AP and provide a certificate that certifies that the service is approved by the community operator. The host AP checks the identity and the certificate of the service and grants or denies forwarding for the tunneled payload packets between the mobile device and the service. The certificate also attests the mapping between  $IP_s$  and  $HIT_s$  to avoid spoofing of service addresses. The PiSA-SA instance on the mobile client forwards all packets with addresses not belonging to  $S$  to the trusted relay of the mobile client, thereby enabling concurrent Internet and service access.

For PiSA-SA we propose to manage the cryptographic identity of a service not on the server itself (e.g., a simple third-party web server) but on a *service gateway*. First, this achieves legacy support for services since all PiSA-SA-related functionality, such as HIP and IPsec communication, identity and certificate management, can be handled transparently by

the gateway on the communication path between the mobile device and the legacy server. Second, it decouples the service provider from the gateway provider. For example, the community operator itself may decide to operate gateways to benign public services (such as a weather forecast web site). Third, a single gateway can provide access to multiple services through a single IPsec tunnel. Hence, a service gateway provides access to a *service domain*. This reduces the resource requirements for establishing and maintaining encrypted tunnels to multiple services within one service domain behind one gateway, which can be of particular interest on mobile devices.

### B. Service Admission

In PiSA, not the mobile client but the trusted relay authenticates to the host AP – clients are never required to authenticate to the access points but only to the relays. This seemingly reverse authentication is performed because the host AP must determine whether the *destination* of a mobile client’s communication (it’s trusted relay) is part of the community. To prevent malicious and unauthorized services from being offered in the Muni-Fi, PiSA-SA extends this authentication and gives the community operators access control over the services. A community operator certifies the admissibility of a service by issuing a digital service certificate, including the service’s virtual address in  $S$  and its cryptographic identity  $HIT_s$ . This certificate is presented to a host AP by the service whenever a client connects to the service via the AP. Based on the certificate and the end-to-middle authentication in the HIP and PiSA handshake between the client and the service, every host AP enforces that mobile clients can only connect to and communicate with authenticated and certified services.

The employed signature and certificate based authentication can become a performance bottleneck, as public-key-based operations cause high CPU load on commodity routers [4] when performed for every connection setup between a mobile device and a service. A viable performance optimization is to partially eliminate cryptographic operations from the service validation procedure at host APs through caching of once verified certificate information to avoid multiple verifications per gateway. Alternatively, community operators can publish cryptographically signed lists of approved services, including their HITs and proxy IP addresses  $IP_s$ . Host APs can validate such a list via the community operator’s signature. Then, host APs verify the admissibility of a service by looking up the proxy IP ( $IP_s$ ) of the service in the lists. This is computationally trivial compared to the original approach of validating one service certificate per service connection. However, the verification of the HIT and Host Identity of the gateway are still required to verify its authenticity.

### C. Legacy Client and Service Support

PiSA and PiSA-SA make the assumption that all end-hosts – mobile devices as well as services – run PiSA and HIP for authentication, tunnel management, and mobility support. However, especially in Muni-Fi networks, this assumption is not viable because of a large user fluctuation (e.g., travelers

and tourists) and a large variety of hardware, potentially including closed source and embedded systems, such as environmental sensors and surveillance cameras. Therefore, this section discusses how PiSA-SA can be improved to support legacy client devices that expect to use plain IP as if directly connected to a classical Muni-Fi network.

For legacy clients, the host APs in PiSA-SA act as HIP and PiSA proxies, which provide the address space  $S$  directly to clients. Hence, PiSA-SA unaware clients can associate with the open Wi-Fi network of a host AP and directly send packets to addresses in  $S$ . In turn, the host AP establishes the tunnel to the respective service gateways for the client and forwards the legacy client’s packets to the service. Since legacy clients lack the tunnel management and mobility features of PiSA, these devices cannot open a direct secure tunnel to a trusted relay at home. Thus, they can only use the unencrypted Wi-Fi link to access a fixed set of openly available services in  $S$  excluding Internet access. Moreover, in lack of a secured tunnel between the mobile guest and the service, sensitive services should apply additional security measures (e.g., TLS via HTTPS in combination with application-specific authentication mechanisms) to protect against eavesdropping and identity theft. In addition, services must distinguish between PiSA-SA clients and legacy clients to reflect the different security properties.

The service gateway concept already allows for operating legacy services without HIP and PiSA support behind a gateway. Legacy services do not have access to the cryptographically secure identifiers of HIP-enabled clients because the secured HIP connections terminate at the gateway. Yet, the HITs of authenticated clients can be mapped to a locally managed virtual address space  $C$  by the gateway in the service domain to assign trustworthy and consistent IP addresses for authenticated clients. Thus, legacy services can use these addresses in applications, firewalls, and access control lists and enforce policies based on these IP addresses. Again, service gateways must distinguish between PiSA-SA and legacy clients and should map authenticated PiSA-SA clients and legacy clients to a different subspaces of  $C$  to support differentiated treatment by the legacy service.

### D. Client Mobility and Network Heterogeneity

When connecting to a city-wide network, a user expects to perceive the network as *one* single system. However, due to its underlying principles, the network structure of PiSA-SA resembles a patchwork of different networks, owned and operated by individuals. In [9] we showed the practical limitations of using indoor access points for creating a Muni-Fi network. Although the coverage of collaborative networks can be quite high, the range of each access point is short in terms of path coverage on the streets. For the vast majority of the observed access points, the path that a user can expect to walk without losing the connection was below 30 meters in urban areas. Hence, the PiSA-SA provides mechanisms to overcome this patchwork-like character. In particular, PiSA-SA integrates automatic network authentication and end-host mobility via HIP to provide a seamless user experience. In our discussion

we distinguish between mobility support for PiSA-SA-enabled clients and *legacy clients* (i.e., mobile clients without PiSA-SA support, c.f. IV-C). PiSA-SA clients implement host-based mobility management in contrast to the infrastructure-based mobility support for legacy clients. In the remainder of this section we first focus on PiSA-SA-enabled hosts before discussing the case for legacy clients.

PiSA-SA cannot leverage Layer-2 mobility and fail-over because there is no homogeneous Wi-Fi network infrastructure (e.g., a distribution system). In particular, a mobile client not only changes its IP subnet but also enters a network to which no a-priori trust relations exist. Therefore, a re-authentication is required whenever a mobile device accesses a service or trusted relay via a new host AP. After associating to the new host AP, the mobile device updates all tunnels to the gateways and its trusted relay to maintain its connections. For PiSA-SA enabled clients, PiSA uses HIP and our HIP middlebox authentication extension [7], [6] for mobility signaling and authentication towards the new host AP. During this mobility signaling, the new host AP validates the authenticity of the service to assure that it is part of the Muni-Fi network. The verification process is fully integrated in the three-way HIP update process and does not create any additional delay.

For legacy clients we cannot assume host-based mobility support. However, mobile legacy clients without any mobility support would suffer from frequent disconnects due to frequent changes in their network attachment. Moreover, the roaming functions of their operating systems may cause legacy clients to roam between access points with identical SSID at any time, possibly leading to disconnects even for nomadic users. To avoid these inconveniences, PiSA-SA host APs and services cooperate to provide infrastructure-based mobility management. Whenever a client roams from AP to AP, its packets can still be routed towards the right service because of the unified address architecture. However, packets from a service gateway to a mobile client are wrongly sent to the previous access point because the gateway's mapping between the client and its return path is invalid. In order to support roaming of legacy clients, the mapping for the gateway's return packets must be adjusted so that the proper tunnel, reflecting the new position of the mobile client, is used.

In our prototype we implemented a reactive approach that updates the mapping at the gateway whenever it receives a packet from a client via a different tunnel (i.e., from a different host AP). To this end, the host APs include the MAC address of the client in the tunnel packets which is then used to identify packets from different clients. On receiving a packet of an already connected client through a new tunnel, the gateway adjusts its mapping and continues to use the new tunnel. We acknowledge that MAC addresses are not secure identifiers and that spoofing the MAC address may allow an attacker to perform DoS or rerouting attacks on a client. However, we argue that confidentiality and data integrity are not at stake because services that require secure connections are expected to use additional security measures (c.f. Section IV-C).

This traffic-driven mobility management can only work if

the clients send packets to the service in short succession or immediately before expecting a response. For request/response applications (e.g., classical WWW), such behavior suffices. However, connections for which the client only reacts to packets of the server (e.g., pure TCP downloads in which the client merely transmits acknowledgments) will break because the data packets from the server that trigger the transmission at the client side do not arrive. In such cases, supplementary applications that exhibit a more chatty character (e.g., a tailor-made web 2.0 application in a browser frame) may generate a steady packet flow from the client to the gateway.

A legacy client may communicate with a diverse set of services, located in different service domains. Thus, a host AP must open several tunnels to different services for the legacy client. Establishing these tunnels can either be performed in a proactive or reactive manner. In a proactive scenario, the host AP opens connections to all possible services at startup. In a reactive scenario, the host AP only opens connections that are explicitly addressed by an IP packet from the legacy client. Proactive connection establishment leads to lower latencies for clients but is not suited for networks in which the set of services is large or not clearly defined. The reactive approach is more flexible regarding the set of services, however, it introduces additional latency during the initial connection establishment and the connection maintenance during legacy client handovers. We implemented both approaches for our prototype and provide an evaluation of the delay caused by reactive on-demand connections in the next section.

## V. EVALUATION

To demonstrate the feasibility of the PiSA Service Architecture, we evaluate the performance of our PiSA-SA prototype. We extended our existing PiSA implementation with the features of the service architecture. Specifically, we added the tunneling from the mobile device to the service gateways and the firewall functionality at the host AP to filter the tunnel packets. Furthermore, we implemented the legacy application support at the wireless routers consisting of the address-space mangling and mobility support. The prototype was implemented for Linux and is also available for embedded Linux platforms (OpenWRT for wireless routers and Maemo 5 for mobile phones). Our prototype is a proof of concept and is not optimized for performance. In particular, all components are implemented in userspace (vs. more efficient yet more complex kernel space implementations) and do not utilize multi threading and multi-processor systems. However, this evaluation still shows the feasibility of our approach.

In our evaluation, we used four different devices to represent the hardware expected in a collaborative Muni-Fi network: A Linksys WRT160NL commodity wireless router with a 400 MHz Atheros 9130-BC1E CPU, running OpenWRT and the PiSA-SA software as host AP. The WRT160NL is an inexpensive consumer-class wireless router without special features like cryptographic acceleration. Its price, openness, and availability make it a valid choice for the use in a

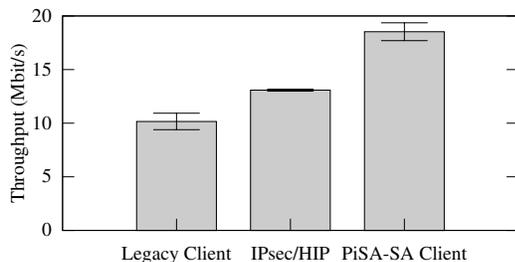


Fig. 3. TCP throughput and Std. Dev. for PiSA-SA on a WRT160NL router.

collaborative Wi-Fi network. We used two different resource-constrained mobile devices for evaluating the performance of the PiSA-SA client implementation. A netbook-class device Asus Eee PC T91 with a 1,333 MHz Z520 Intel Atom CPU and a Nokia N900 mobile phone with an ARM Cortex A8 CPU running at 600 MHz. Finally we used three PCs with AMD Athlon 64 X2 Dual Core 4800+ processors and 4 GB RAM as service gateways and as load generators for evaluating the host AP and gateway performance. In the following we evaluate each component of PiSA-SA separately to minimize the limiting side-effects of other PiSA-SA components.

#### A. Throughput on Host Access Points

We measured the throughput of the host AP in both described scenarios: a) for native PiSA-SA clients, for which the host AP only acts as a modified HIP firewall and b) for legacy clients, for which the host AP performs packet tunneling, address space mangling, and encryption. To ensure that the mobile devices are not the limiting factor for the host AP throughput measurements, we used the Athlon PCs as service gateway and as mobile nodes. The PCs were connected to the WRT160NL via a wired Ethernet connection. We decided to avoid wireless transmissions in our performance measurements to eliminate the artifacts of the wireless channel because the PiSA-SA leaves the Wi-Fi interface untouched. The throughput was measured with *iperf* over TCP running on the service gateway and the client. For each result, we performed 10 individual throughput measurements, each lasting 60 seconds with 30 intermediate samples.

Figure 3 summarizes the results of the throughput of the host AP and the respective standard deviations from the perspective of the host AP. For native PiSA-SA mobile devices, which perform address space management, authentication, and encryption themselves, the mean throughput of the host AP was 18.5 Mbit/s and for legacy clients it was 10.2 Mbit/s. Figure 3 also depicts the throughput that can be achieved with HIP and IPsec alone for a connection between the host AP and the service gateway. This throughput of 13.1 Mbit/s can be seen as a benchmark, as it effectively shows the IPsec throughput of the router without any modifications. In contrast to all other PiSA-SA components, IPsec encryption is performed by a Linux kernel module rather than by a userspace process. Hence, the IPsec throughput shows the cost for the cryptographic processing while the difference to the legacy case shows the impact of the additional PiSA-SA processing and the address space mangling. Compared to

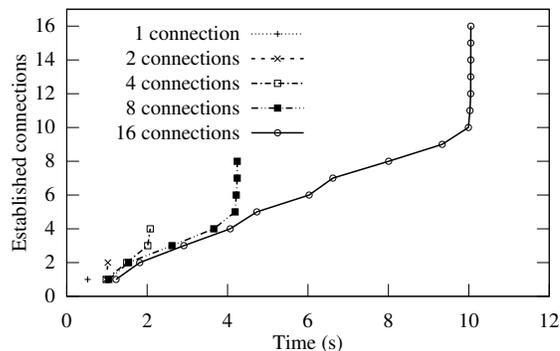


Fig. 4. Registration delay on a WRT160NL for legacy clients.

IPsec, PiSA-SA in legacy client mode achieves 77% of plain IPsec throughput whereas the throughput of PiSA-SA is 141% of the IPsec throughput. The higher value of the native PiSA-SA clients shows the reduced computational burden at the host AP because in this scenario the mobile client performs address space mangling and encryption, reducing the task of the router to mere stateful IPsec filtering.

When compared to an unmodified WRT160NL with a throughput of 93.6 Mbit/s on the wired and 71.5 Mbit/s over the wireless interface, the obtained PiSA-SA results appear relatively small. Evidently, the commodity router has a limited throughput capacity if per-packet filtering or encryption is applied without hardware support. Moreover, implementing per-packet filtering or address space mangling in userspace leads to performance penalties because the packet contents are copied from kernelspace to userspace and back before they are forwarded. However, all throughput measurements have to be seen in the context of the expected use of the device. Since such commodity hardware will most likely be used by private persons, the ADSL Internet uplink and downlink will probably become the limiting factor in many cases.

#### B. Delay for On-Demand Connections

As described in Section IV-D, host APs can reactively open the required tunnels to service gateways for legacy clients. Since establishing a new HIP connection involves the use of CPU-intensive public-key cryptography, we evaluated the resulting connection delay. The delay for establishing parallel on-demand connections not only shows the initial delay for new connections but also reflects the service disruption for legacy clients in case of mobility as well as the capabilities of a host AP when serving multiple clients.

We used the same setting as in Section V-A, consisting of two PCs and one wireless router. One PC acted as legacy client and initiated a measurement run by sending packets to different destination IP addresses in  $S$ . For each packet, the WRT160NL AP performs a handshake and opens a new tunnel to the respective service. We did not use optimizations like caching and proactive tunnel establishment, as discussed in Section IV-D, to model a worst-case scenario.

Figure 4 shows the results for 1, 2, 4, 8, and 16 parallel connection requests. A single PiSA-SA tunnel can be established within 520 ms. Establishing multiple connections in

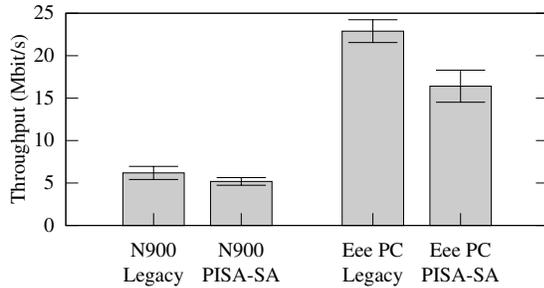


Fig. 5. PiSA-SA client performance in Mbit/s and Std.Dev.

parallel increases the delay linearly. Each graph in the Figure shows two different slopes caused by the sequential packet processing in the HIP implementation. During the first phase, the parallel connection establishments are slowed down by the processing of packets that cause CPU-intensive cryptographic operations (RSA signatures and Diffie-Hellman shared key generation). After completing this CPU-intensive phase, the remaining operations are completed in rapid succession.

The delay measurements were performed in a local network with a notably low RTT of 0.5 ms. Therefore, the results mainly show the delay caused by the packet processing on the host AP and the service gateway. The results include three RTTs, which would be higher in a real Muni-Fi network.

The results show that opening a high number of on-demand connections in parallel can lead to prolonged delays. However, we assume that many clients will use a similar set of popular services in the Muni-Fi network. Hence, host APs are likely to already have established connections to these services when a new client associates with the host AP. Secondly, the high delay shows the benefit of the service gateway concept because many different services can be reached through a single gateway. Finally, the results indicate that careful consideration is necessary for connection timeouts (i.e., the time before idle connections between the host AP and the service gateway are closed). Due to the negligible maintenance cost of idle connections, connections can stay open for long times.

### C. Service Gateway Throughput

While the previous measurements focused on the throughput of the host APs, this section evaluates the TCP throughput of a service gateway. We used a setup of three Athlon PCs, directly connected by a gigabit Ethernet switch. Two PCs acted as PiSA-SA clients and the third PC acted as gateway. We did not use the host AP in this setting because it would have limited the throughput of the clients.

The mean TCP throughput per client was 77.6 Mbit/s with a standard deviation of 6.3 Mbit/s, resulting in an mean throughput of 155.1 Mbit/s for the service gateway. The numbers show that a considerable number of mobile clients can be served with moderate speeds. For services requiring a larger bandwidth, support for multi threading and cryptographic accelerators can be implemented to further increase performance.

### D. Client Throughput

Client devices can either be PiSA-SA capable or legacy devices without modifications. We evaluated the throughput

for both cases with the Asus Eee PC and the N900 mobile phone. Each device was connected to the WRT160NL via an 802.11g wireless link. Since we aim at determining the client performance only, the WRT160NL wireless router did not run any PiSA-SA related software during the measurements. Hence, the legacy case represents the maximum throughput of the client devices without PiSA-SA involved, whereas the PiSA-SA client performed encryption and address space mangling in addition. Figure 5 shows the throughput of both devices with and without PiSA-SA.

The N900 used only a fraction of the available bandwidth of 802.11g in native and legacy mode. Using the N900 as a PiSA-SA-capable device slightly reduced the overall throughput from 6.2 Mbit/s to 5.2 Mbit/s in comparison to its plain IP performance. During the measurements, the CPU utilization of the N900 was higher when used as PiSA-SA capable device (100% vs. 25%). With constant high-volume downloads, this higher utilization may lead to a shorter battery runtime for the PiSA-SA client. With a throughput of 22.9 Mbit/s, the Eee PC almost fully used the practically available 802.11g bandwidth when acting as legacy client. Running PiSA-SA on the Eee PC considerably reduced the available bandwidth to 16.1 Mbit/s. However, the Eee PC can still saturate state-of-the-art ADSL lines. Similar to the N900, the CPU utilization of the Eee PC was higher when running PiSA-SA (70% vs. 13%).

## VI. DEPLOYMENT CONSIDERATIONS

The main strength of PiSA-SA is its openness that turns the community operator from a provider to a distributed control instance that defines the set of available services and manages their address space. Since the set of services is not controlled or dominated by a single ISP or commercial organization, competition among municipal service providers becomes possible. However, the successful distribution of the community operator function requires a trustworthy and independent management of the root certificate and the community root DNS server for name-space control. Since the community operator is not required to operate provider infrastructure (e.g., APs and service connectivity), it becomes feasible for non-profit organizations to assume this role.

Although PiSA-SA-based systems grow with the contribution of citizens to the access network, bootstrapping the system is critical. Especially in the early phases, the benefit to early adopters is limited because of insufficient coverage. Additional incentives like the provision of public access points in well frequented places can resolve the initial deadlock situation.

An often stated goal for Muni-Fi systems is to bridge the digital divide. To achieve this goal, the municipality can operate a trusted relay with Internet access for citizens with poor economic background. However, coverage of community Wi-Fi access points may be low in poor areas of the city. Hence, stimulus programs like free or cheap Internet connections bundled with host APs for some citizens are an option to increase Wi-Fi coverage in such areas. We provide further considerations regarding the use of collaborative Wi-Fi in [9].

## VII. RELATED WORK

In the past, a number of successful federated or collaborative Wi-Fi access networks have been established. A number of commercial providers offer Wi-Fi sharing communities (e.g., FON [3], and Wippies [10]). These networks use provider overlays above existing broadband connections to offer services. However, the selection of services is fully controlled and typically restricted to the needs of a provider (e.g., registration and payment services). Hence, new services must earn the merit of the provider, which reduces the opportunity to implement innovative services without provider benefit.

An example for a federated non-profit network is the eduroam network, a European Wi-Fi access network for education facilities, such as Universities and high-schools [11]. eduroam consists of many organizations, of which each operates its own RADIUS [12] authentication server. The system is tightly controlled and professionally managed, making it difficult to include user-provided networking or services.

Sastry et al. [13] use a tunnel mechanism similar to PiSA to enable Wi-Fi sharing between untrusted parties. The authors show how to architect a citywide cooperative network based on it. The solution allows for distributing the function of the Wi-Fi provider but does not take into account the provisioning of municipal services or legacy client support. Johansson et al. propose to integrate user-operated APs into existing networks [14] to extend the efficiency of cellular networks with 802.11 access points without considering heterogeneous collaborative networks. Heikkinen proposed to integrate user-provided services into IMS [15]. The system allows users to offer services in a provider context but does not focus on a provider-less case.

Kuptsov et al. use HIP to implement a Wi-Fi authentication system that allows HIP clients to connect to a single HIP relay in a city-wide Wi-Fi system [16]. Similar to PiSA and PiSA-SA, the authors use our HIP middlebox authentication extension [6], developed for PiSA. The system is centralized and does not lend itself to the community-based approach. Kuptsov et al. also consider legacy client support by using a port switching technique. However, they do not take mobility support for legacy clients and access to multiple service gateways into account. Hence, they do not reach the same level of decentralization as PiSA-SA.

## VIII. CONCLUSION

In this paper, we present the PiSA Service Architecture. It generalizes the PiSA Wi-Fi sharing system to be applicable to distributed municipal Wi-Fi networks. It reduces the cost of municipal Wi-Fi projects by involving users and companies as Wi-Fi service providers (e.g., micro operators) and municipal service providers. The proposed architecture provides a large degree of openness through decentralization to foster competition and diversity for services in the network. It incentivizes users to make their wireless access points available to the municipal Wi-Fi system and obtain mobile Internet access through other access points in return. This incentive can be the

driver for constant growth while the costs are shared among many parties.

We have implemented a fully functional Linux-based prototype for mobile clients, embedded routers, and servers and evaluated its performance in the relevant settings. Our evaluation shows that current router and client hardware is sufficient to operate PiSA-SA at DSL line speed. Hence, even the prototype can adequately serve collaborative Muni-Fis.

We conclude that PiSA-SA eases the deployment of municipal Wi-Fi services, as it relieves the community operator from its dependence on a specific network operator or the burden of providing access to the municipal network by itself. PiSA-SA allows municipal Wi-Fi networks to grow with their user-base without making compromises regarding usability and flexibility.

## IX. ACKNOWLEDGMENTS

We thank Miika Komu, Diego Biurrun, Henrik Ziegeldorf, Jahn Bertsch, Tim Just, Mircea Gherzan, Nicolai Viol, and all other contributors of PiSA for all their support and dedication.

## REFERENCES

- [1] L. van Audenhove, P. Ballon, M. Poel, and T. Staelens, "Government policy and wireless city networks: a comparative analysis of motivations, goals, services and their relation to network structure," *The Southern African Journal of Information and Communication*, vol. 8, 2009.
- [2] Freifunk Community, "Freifunk Website," [Online] Available <http://start.freifunk.net/>, January 8, 2008.
- [3] FON WIRELESS, Ltd, "FON Website," [Online] Available <http://www.fon.com/>, August 10, 2009.
- [4] T. Heer, S. Götz, E. Weingärtner, and K. Wehrle, "Secure Wi-Fi Sharing on Global Scales," in *Proc. of 15th International Conference on Telecommunication (ICT '08)*, 2008.
- [5] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," RFC 5201 (Experimental), 2008.
- [6] T. Heer, R. Hummen, M. Komu, S. Götz, and K. Wehrle, "End-host Authentication and Authorization for Middleboxes based on a Cryptographic Namespace," in *Proceedings of the IEEE International Conference on Communications 2009 (ICC 2009)*, 2009.
- [7] T. Heer, "End-Host Authentication for HIP Middleboxes," IETF, Internet-Draft draft-heer-hip-midauth-02, Feb. 2008, work in progress.
- [8] S. Varjonen and T. Heer, "HIP Certificates," IETF, Internet-Draft draft-ietf-hip-cert-01, 2009, work in progress.
- [9] T. Heer, R. Hummen, N. Viol, H. Wirtz, S. Götz, and K. Wehrle, "Collaborative Municipal Wi-Fi Networks - Challenges and Opportunities," in *Proc. of 6th IEEE PerCom Workshop on Pervasive Wireless Networking*, 2010.
- [10] Saunalahti Group Oyj, "Wippies Website," [Online] Available <http://www.wippies.com/>, August 10, 2009.
- [11] K. Wierenga and L. Florio, "Eduroam: past, present and future," *Computational Methods in Science and Technology*, vol. 11, no. 2, 2005.
- [12] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, Jun. 2000. [Online]. Available: <http://www.ietf.org/rfc/rfc2865.txt>
- [13] N. Sastry, J. Crowcroft, and K. Sollins, "Architecting Citywide Ubiquitous Wi-Fi Access," in *Proceedings of ACM SIGCOMM HotNets (HOT Topics in Networks)*, 2007.
- [14] K. Johansson, J. Lind, M. Berg, J. Hultell, N. Kviselius, J. Markendahl, and M. Prytz, "Integrating user deployed local access points in a mobile operator's network," in *Proc. of Wireless World Research Forum WWRF*, 2004.
- [15] S. Heikkinen, "Providing Identity Assured User Generated Services Using IMS," in *2nd International Workshop on Mobile and Wireless Networks Security*, 2009.
- [16] D. Kuptsov, A. Khurri, and A. Gurtov, "Distributed User Authentication in Wireless LANs," in *Proc. of the 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'09)*, 2009.