Multipath TCP
Internet-Draft
Intended status: Standards Track
Expires: December 17, 2011

G. Hampel
T. Klein
Alcatel-Lucent

June 15, 2011

Enhancements to Improve the Applicability of Multipath TCP to Wireless Access Networks
draft-hampel-mptcp-applicability-wireless-networks-00

Abstract

This document analyses the applicability of Multipath TCP to wireless access networks with overlapping coverage area, and it discusses potential protocol extensions that aim to improve operation in such environments. The analysis attempts to identify use cases, benefits as well as technical and functional obstacles encountered in the current version of the protocol. Based on this analysis, recommendations are made on feature-, signaling- and policy extensions that promise to enhance Multipath-TCP's value, versatility and market acceptance in wireless access networks.

Table of Contents

1. Introduction

Multipath TCP (MPTCP) is a reliable stream-based transport protocol which permits simultaneous utilization of multiple data delivery paths. Each data delivery path appears like an independent TCP connection on the wire and is generally referred to as a subflow pertaining to a superseded MPTCP connection [1].

MPTCP can be run in two distinct operation modes referred to as multipath operation mode and path-selective operation mode. In the former, multiple paths are used simultaneously while in the latter only one path is used at a time for data exchange.

MPTCP's main goal has been to maximize the aggregate throughput of all available subflows subject to a fairness constraint [1], [3]. A fair amount of effort has been invested into finding an appropriate congestion algorithm for such an operation mode [4].

MPTCP has targeted wireless access networks as well as data centers as potential environments for multipath utilization [4]. The present document exclusively focuses on MPTCP's applicability to wireless access networks. MPTCP's principal fit for such environments can be motivated by the following factors:

- Densely populated areas provide a multitude of spatially overlapping data access networks which could be used for multipath operation. These networks may support different access technologies, i.e. such as WCDMA, EVDO, LTE, WiMAX or Wifi.

- The wireless air interface is usually the main throughput bottleneck, hence multiplexing data along multiple paths should be beneficial (at least from the myopic viewpoint).

- The tremendous growth of mobile data traffic demands more efficient use of available spectrum. MPTCP addresses this demand.

- A growing fraction of mobile devices is multi-homed, i.e. simultaneous access is supported for one cellular technology (3G/4G) as well as for WiFi. It is foreseeable that multi-homing capabilities will increase in the future.

- Many users have access permissions to more than one access network due to independent subscriber relationships (e.g. MNO, ISP, company networks, etc.) or because access is free (public hotspots).

While this high-level picture makes a strong case for MPTCP's principal applicability to wireless access networks, a detailed study

presented in this document reveals a variety of issues that are related to the current design and may jeopardize MPTCP's usability or acceptance in the wireless market segment. Specific recommendations are made that should help to overcome these issues and to improve MPTCP's versatility in general as well as its applicability to wireless network environments.

2. Strengths of MPTCP

MPTCP's present architecture incorporates a variety of upfront architecture and design decisions, which make it suit well to wireless access environments. Some of these strengths are:

- Focus on host-based solution: Overlapping access networks are frequently owned by different access providers resulting in a large topological distance between the access points in the network graph. When using a network-based solution as currently supported by the relevant 3G/4G standard bodies for mobility, multipath operation would introduce a triangular routing problem. This is averted by MPTCP due to its end-host-based nature. A host-based solution makes MPTCP further independent of operator policies and inter-operator trust relationships.

- Compliance with existing network infrastructure: MPTCP has been architected to comply with the existing infrastructure (e.g. middleboxes and routers) of access providers. This makes network upgrade or reconfiguration largely unnecessary and lowers the threshold to market acceptance. MPTCP has specifically included mechanisms to overcome firewalls by allowing mechanisms to perform hole punching. This opens opportunities for P2P applications such as VoIP and multi-player gaming.

- Low initial cost of transport: The MPTCP design has tried to keep the initial cost for transport establishment comparable to that of a conventional TCP connection. Cost is measured in signaling effort and state information held on the end nodes. This is an important factor for use cases, where the need for additional paths is not known a priori.

3. MPTCP Multipath Operation Mode

The following observations and recommendations apply to MPTCP's current multipath solution, i.e. where multiple paths are simultaneously used for data exchange.

3.1. Throughput Maximization

MPTCP's primary focus has been on multipath multiplexing to maximize the aggregate connection throughput. In today's wireless environments, the value of this objective may be questionable. While many populated areas offer a plethora of overlapping access networks, limitations to access permissions (due to the necessary subscriber-operator relationship) and radio capabilities considerably restrict the actual number of available access interfaces. Further, multipath multiplexing provides a noteworthy gain only if the paths are approximately equal in throughput, which is rarely the case in natural propagation environments.

It may also be argued that the nature of traffic used by mobile devices allows other and simpler approaches to exploit excess capacity of overlapping air interfaces, e.g. by distributing applications or individual connections (e.g. HTTP object requests) over all available interfaces.

MPTCP may therefore emphasize on other inherent advantages to motivate its value in wireless access networks. The remainder of this document identifies a few of such opportunities and it proposes associated enhancements to the protocol as needed.

3.2. Support of Multiple Radio Links

Simultaneous support of multiple active air interfaces requires that multiple radios are run at the same time. This has impact on the aggregate usage of air interface capacity and mobile battery power. In 3rd- and 4th-generation access technologies, radio bearer support consumes control channel capacity and draws battery power for transmission even if only few data are sent. Such bearer support is not necessary for subflows that stay idle and are used only as backup as it is the case for MPTCP's path-selective operation mode. It is therefore important to also emphasize on the benefits of MPTCP's path-selective operation in case the costs for multi-radio support do not justify multipath operation.

3.3. Multipath Diversity vs. Spatial Multiplexing

Under weak coverage conditions, multipath support could provide additional resilience to connection failure. In such scenarios, a

"multipath diversity scheme" may be more promising than MPTCP's present multipath multiplexing scheme. In the multipath diversity scheme, the same data are simultaneously sent along multiple paths. Such a scheme could substantially reduce head-of-line blocking on connection level when individual paths "choke".

The tradeoff between diversity- and multiplex operation has been well studied in the context of MIMO [6]. The potential benefits of multipath diversity are also known from CDMA IS95, CDMA2000 and W-CDMA air interfaces [7]. Similar principles should apply to MPTCP.

It may be beneficial to investigate a multipath diversity scheme as an alternative solution to MPTCP's present multipath multiplexing scheme. The increase in connection resilience and improvements in head-of-line blocking could justify the associated costs in bandwidth efficiency and battery drainage due to multi-radio operation. More research is required in this area.

3.4. Dynamic Path Adaptation

Since channel conditions and cell loading can rapidly change in wireless settings, the appropriate and timely decision on how load is distributed (and retransmitted) across available paths determines the effective end-to-end throughput. Multipath operation should do well under such conditions since RTT and congestion information is available from all paths and can be used to drive this decision-making process on a per-packet level.

While great effort has been invested into MPTCP's aggregate congestion control and fairness, little guidance is provided on how to optimize its response to fluctuations in path throughput and delay. It may be beneficial to invest further research into this area. The outcome could provide detailed policies on cross-subflow retransmissions and selection of subflow subsets for multipath operation.

3.5. Negotiation of Operation Mode

Currently, MPTCP supports only one connection-level congestion control algorithm, which is applied by the sender. The above recommendations propose additional multipath operation modes among which the data sender could potentially select (max throughput vs. high resilience vs. fast response).

Provided availability of multiple multipath operation modes, additional features would be necessary that allow the data receiver to negotiate the operation mode applied by the data sender.

3.6. Signaling of Path Availability

Mobile devices usually have up-to-date information about interface link quality and interface availability. Such information can be used by the MPTCP sender to make quick decisions on what paths it should use for data transmission.

MPTCP does not provide any method for the receiving host to signal its interface-availability status to the sending peer. If such information were provided, the peer's sender could react within 1/2 RTT and start or stop traffic transmission on the corresponding subflows.

Without such message, the peer's congestion control will indirectly learn about the host's change in interface availability, which will take at least RTO in case of interface loss and multiple RTTs due to slow-start when an interface is brought up again.

The MP_PRIO option is not well suited to provide interface-availability information since it cannot be sent along unavailable paths in order to mark them unavailable.

It is recommended to introduce an additional signaling mechanism for interface-availability. This mechanism must allow that a message sent from one interface can refer to the availability of other interfaces of the same host. Such messages, referred to as AVAIL_ADDR and UNAVAIL_ADDR, can be designed analogous to the REMOVE_ADDR option and signal availability/unavailability of an enclosed address id. Both messages can be combined into one by adding a binary availability flag.

For this mechanism to function, the peer must hold a mapping between the host's address values and address ids. A host that wishes to use AVAIL/UNAVAIL options can introduce such mapping by sending the ADD_ADDR option before or by enclosing it into the same packet. In this case, the ADD_ADDR option should only provide the mapping between address value and address id, but it should not file a request for subflow initiation. Since the ADD_ADDR option currently combines both of these functions, it is recommended to separate the request for subflow establishment and assign it to a new option referred to as JOIN_ADDRESS option. The JOIN_ADDRESS option solely includes the corresponding address id. This separation is also necessary for other enhancements as discussed in sections Section 4.4 and Section 5.1.

The ADD_ADDR option can be simplified when it refers to the packet's source address. In this case, it only needs to enclose the first 4 octets and it may omit the actual address value itself.

3.7. DSS Insertion

When a bulk of packets is sent in sequence along the same path, only the first packet has to carry a DSS option to provide the peer with the necessary mapping information. The current MPTCP protocol leaves it open to the data sender to enclose further DSS options on subsequent packets of this bulk.

As long as packets are delivered in order and the packet loss rate is small, one DSS option on the first packet should do fine. In wireless access networks, however, these conditions are usually not met. When the first packet carrying the DSS option is lost, the receiver needs to allocate a separate buffer to store the remaining bulk of packets until it receives an adequate mapping from a DSS retransmission. This adds unnecessary complexity to the receiver. Alternatively, the receiver can drop the bulk, which invokes a large number of retransmissions.

To avoid these shortcomings, the data sender should insert DSS options on all packets until the first data ACK is received to packets contained in the bulk. This tells the sender that the receiver has obtained the mapping information, and it can omit the DSS option on all further packets of this bulk.

4. MPTCP Path-Selective Operation Mode

The following observations and recommendations apply to MPTCP's path-selective operation mode, i.e. where only one path is selected for data exchange.

4.1. Principal Benefits

MPTCP's path-selection capabilities facilitate connection migration across access networks pertaining to one or to different access providers. This feature has significant value since (1) there is principal demand as outlined in the introduction, and (2) there is only one alternative solution [5] which bears various drawbacks.

(Note that conventional layer-3 mobility solutions as provided by Mobile IPv4/6, Proxy Mobile IP, 3GPP and 3GPP2, for instance, are not considered in this discussion since they rely on roaming agreements between access operators as well as roaming-compliant infrastructure. These requirements do not apply to MPTCP or [5].)

Path-selective operation may find broader acceptance in the wireless community than multipath operation since its principal procedure is better known and better understood. Further, path-selective operation bears the advantage that it does not require simultaneous operation of multiple radios. It may therefore be possible that path-selective operation becomes a main driver for MPTCP's deployment in wireless environments.

While path-selective operation is a border case of multipath-operation, MPTCP's signaling and design may not have been optimized for this border case. The next sections makes specific recommendations on how design and signaling could be tailored to better support path-selective operation.

4.2. Reduction of Design Complexity

MPTCP has been designed with multipath operation in mind. This goal makes the solution very complex, and it adds a lot of processing, state- and signaling overhead to the end nodes. While such complexity is the price for multipath operation, a simpler solution would be adequate when path-selective operation is satisfactory. To avoid suporting two different protocols, full inter-operability is required between full-fledged solution and simplified alternative.

In the following, a simplified design is proposed for both MPTCP sender and MPTCP receiver. The associated complexity reduction is substantial and permits implementations on lower-layer packet filters (often referred to as "bump in the stack" implementation), i.e.

outside the kernel.

The design simplifications do not affect MPTCP's support of multiple parallel subflows. Also, MPTCP's middle-box compliance remains unaffected.

4.3. Complexity-Reduced Path-Selective Sender

The complexity-reduced MPTCP sender presides over only one flow- and congestion engine, which operates in the data sequence number space. This engine can be provided by a conventional TCP control block, for instance. When a packet departs the flow engine, the decision is made on what subflow it has to be transmitted, and the mapping from data sequence- and data acknowledgement numbers (DSN and DAN) to subflow sequence- and acknowledgement numbers (SNs and ANs) is performed accordingly. This process is straightforward in between path re-selection events.

When path re-selection occurs, the sender determines a cutoff DSN and transmits all data with DSN above or equal to the cutoff value along the new path. Retransmissions are sent along the old path if their DSN is below the cutoff value. This procedure is simple since only one cutoff DSN has to be cached.

In case the old subflow becomes unavailable, retransmissions can occur across subflows in the same manner as supported by the full-fledged version of MPTCP.

The subflow sequence numbers are derived from data sequence numbers via a subflow-specific offset, which only changes at the moment of path re-selection or when cross-subflow retransmissions occur. In these cases, the sender inserts DSS options into all packets with subsequent DSNs until data ACKs are received that indicate successful arrival of the latest mapping update. At this point, both hosts are synchronized and the sender can omit further DSS options. This procedure guarantees that the peer has complete mapping information even if packets get dropped or delivered out of order. Note that this procedure is in full compliance with current MPTCP. It follows along the same lines as the recommendation made for multiflow operation in Section 3.7.

The host should not engage into another path re-selection until complete re-synchronization between both hosts has been achieved.

When operating with only one flow/congestion engine, each subflow still has to support its own TCP signaling handshakes to make it appear like an independent TCP connection on the wire. This is important for interoperability with hosts running the full-fledged

version of MPTCP and to ensure compliance with middle boxes.

In addition, care has to be taken that subflow ANs match the actual subflow SNs sent on the same path. When path re-selection occurs, new data move out on the new path while acknowledgements may still refer to packets that arrived on the old path. In this case, a separate ACK has to be generated which holds the corresponding subflow AN and is sent on the old path. The data packet obtains a subflow AN, which is equal to the last AN sent on the new path. This procedure can be accomplished via a lookup table. It is recommended to generate a few examples as guidance to implementors.

Using only one flow/congestion engine significantly simplifies the sender-side implementation. This simplification may have slight performance impact during the path re-selection phase since congestion control has to adapt to the conditions of the new path. This performance impact, however, should not be worse than experienced by standard mobility protocols such as Mobile IP.

Note that the complexity reduction on the sender does not require any change to MPTCP's present signaling. It is further possible to furnish a host with a simplified sender (using path-selective mode) and a full-fledged multipath-capable receiver. This may reintroduce operational complexity to the sender since it has to frequently split acknowledgements from data and send them on different paths.

4.4. Complexity-Reduced Path-Selective Receiver

By confining the sender to path-selective operation mode, the receiving host can substantially reduce buffer space needed for data assembly. Further, the assembly process becomes easier since data arrive on one subflow for most of the time.

The complexity-reduced receiver must therefore have the means to enforce path-selective operation on the remote sender. It must further be able to give directions regarding the specific path to be (re-)selected.

The present MPTCP protocol provides the MP_PRIO option, which could serve for this purpose. For path re-selection, the receiver has to send one MP_PRIO option with B=1 on the old path and one option with B=0 on the new path.

This solution has the following drawbacks:

- The MP_PRIO option is not binding. Hence there is no guarantee that the remote host follows the directives and reduces data delivery to only one subflow (or the desired subflow).

- Delivery of MP_PRIO options is unreliable. Therefore, the remote sender may engage into multipath operation in case the MP_PRIO option gets lost on the old path.

- There is no policy that requires confirmation of MP_PRIO messages. Therefore, the host must derive the successful delivery of all MP_PRIO messages by analyzing packet arrival on the various paths.

- At every path re-selection, two messages have to be sent while in principle, one message would be sufficient.

- Between arrival of the first and the second MP_PRIO option, the peer may assume an undefined state.

These drawbacks show that the MP_PRIO option is not well suited for the present purpose. This is understandable since the MP_PRIO option was designed for multipath operation rather than complexity reduction of path-selective operation.

Alternatively, a reduced-complexity receiver could enforce single-path operation as well as path reselection through dynamic subflow setup/teardown procedures: When a new path is to be selected, the host creates the corresponding subflow via MP_JOIN and kills the old subflow via TCP RST. Obviously, this procedure impairs robustness and adds delay since the new subflow cannot be established unless path reselection is imminent, and since the 3-way SYN/ACK handshake takes a considerable amount of time. Also, TCP RST cannot be considered a clean TCP termination procedure in the present scenario. Using TCP FIN instead may not have the desired effect in case the peer has still data to send and insists on the present path.

4.4.1. The MP_SELECT Option

Given these drawbacks, it would be beneficial to introduce a separate TCP option that enforces path-selective operation on the remote sender indicating the preferred path. This option is referred to as MP_SELECT.

When the host wishes to (re)-select a certain path, it sends the MP_SELECT option on the selected path only. Upon reception of the MP_SELECT option, the peer responds with an MP_SELECT option on the proposed path to confirm delivery of the MP_SELECT option it received. These steps apply to a complexity-reduced MPTCP sender in the same way as to a full-fledged MPTCP sender. In case the remote sender is complexity-reduced, it initiates path re-selection according to Section 4.3 as soon as it receives the MP_SELECT option.

While path selection via MP_SELECT option is principally binding,

situations may occur where both hosts have conflicting interests. Also, conflicting MP_SELECT options may cross on different paths. Hence a conflict resolution policy has to be introduced that regulates such situations.

An appropriate policy can be derived from the premise that each host is satisfied when permitted to select its own local interface. When sending an MP_SELECT option on a desired path, the sending host indicates the local interface it wishes to use, which is the source address of the MP_SELECT packet. A universally satisfactory path is defined by this interface and the peer's preferred local interface.

The peer can send the MP_SELECT response along this universally satisfactory path. If this path is not supported by a subflow, the peer can establish this subflow via MP_JOIN. To avoid unnecessary delays, the peer may temporarily accept the selection of a sub-optimal path until the universally satisfactory subflow has been established.

While the conflict-resolution policy restricts each host to determine its interface rather than the entire path, it only applies to situations of competing interests. If the peer has no specific preferences for a certain interface, it should follow the path selection provided by the MP_SELECT option it receives.

Note that conflicts do usually not occur for mobile clients supporting multiple subflows to one server interface. This applies for the majority of mobile internet traffic.

To avoid a time-consuming retransmission schedule for path re-selection, the host should attach MP_SELECT options on all packets it sends on the new path until it receives the first MP_SELECT delivery confirmation on the new path. The peer keeps sending delivery confirmations until it stops receiving MP_SELECT options on the new path. Such procedure guarantees synchronization between both hosts within 1RTT. The procedure is the equivalent to that recommended for DSS synchronization in Section 3.7 and Section 4.3.

4.4.2. Break Before Make

Under some circumstances, the host may want to use the old path to inform the peer about an imminent re-selection decision. This applies to break-before-make scenarios, where only one radio is available to support both the old and the new interface.

Since the lower-layer handover, i.e. tear-down of the old and setup of the new air interface, consumes significant time, all data transmitted by the peer during that time frame get lost and have to

be retransmitted.

If the host requested path re-selection on the last packet of the old path, the peer could start sending data on the new path while the host switches lower-layer interfaces. This would significantly reduce the performance impact due to this type of hard handover.

To provide such means, MP_SELECT could be furnished with an explicit reference to a particular subflow. Such a solution requires availability of mutually agreed subflow identifiers, which are currently not supported by MPTCP. It would be possible to use the random number R_A and R_B exchanged during MP_JOIN for this purpose.

Alternatively, the host can insert the address id of its new interface address into the MP_SELECT option. This allows the peer to select a path compliant with the host's new address. Obviously, the host must have announced the mapping between address id and address value prior to the handover using the ADD_ADDRESS option. For this purpose, it is necessary to strip the ADD_ADDRESS option from the additional request for subflow-generation as discussed in Section 3.6 and Section 5.1

Providing solely an address-id instead of a subflow-id is sufficient as it circumvents the performance degradation due to hard handoff. In case multiple subflows are available for this new address, the peer can select a universally satisfying candidate among them.

4.5. Dynamic Overhead Shedding

Path-selective operation requires substantially less overhead in processing and buffer space than multipath operation. This applies to both the complexity-reduced- as well as the full-fledged design.

A MPTCP-aware application server supporting many simultaneous multipath connections can apply an overhead-shedding mechanism by switching to path-selective operation when the aggregate traffic load runs too high.

For this purpose, the server needs to enforce path-selective operation in the same manner as discussed in Section 4.4. for the complexity-reduced receiver. Hence the same signaling features, i.e. MP_SELECT option and ADDR_AVAIL/ADDR_UNAVAIL options need to be supported to enable this feature.

5. Incremental Deployment of MPTCP

MPTCP is based on the premise that both end hosts support the MPTCP protocol. In wireless access networks, such a requirement may create a burden to deployment since both end points are represented by different parties and only one of them may see a benefit in using MPTCP. (This for instance is different in data centers where the end points are controlled by the same party). This burden may jeopardize MPTCP's market acceptance.

In some deployment scenarios, MPTCP may provide sufficient benefit to both sides to overcome this burden. This may apply to P2P services, such as VoIP and VidIP, where both end points are mobile, and they both have a vested interest to upgrade to MPTCP. This scenario, however, fails in case back-to-back agents are inserted between the mobile end points as it is often the case for SIP- and IMS-based traffic. Further, many P2P services are of conversational nature and rendered via UDP.

There may be an incentive for some network-based services to upgrade to MPTCP, especially if their service offerings are tailored toward mobile devices. It is not clear, however, how strong this incentive is and if it supports MPTCP deployment on a large scale.

One way to lower the deployment threshold for MPTCP is through the introduction of proxies as proposed by [8]. Since such proxies require only one end point to be MPTCP-compliant, they facilitate an incremental deployment process.

In the most general scenario [8], no restrictions are made to the location where the proxy resides. As a result, the MPTCP-aware host has to undergo a signaling procedure to authenticate itself to the proxy and to provide it with information about the remote peer with whom it wishes to establish a connection. Such a procedure substantially extends the present MPTCP signaling protocol.

In a more restrictive scenario, the proxy resides on a central router in the MPTCP-host's network. Being integral to the host's network is important since it eliminates the need for a separate authentication procedure. The central location further allows the proxy to derive all information through interception of passing traffic. Hence no additional signaling between host and proxy is needed for connection establishment and the proxy becomes transparent to the end hosts.

While a transparent proxy can also be introduced for MPTCP, some minor issues arise due to MPTCP's support of multiple simultaneous subflows, which make the "on-path" condition ambiguous. These issues are discussed in the next section. The next following section

discusses the relevance of transparent proxies in the context of 3rd- and 4th-generation mobile-network deployments.

5.1. Transparent Proxy

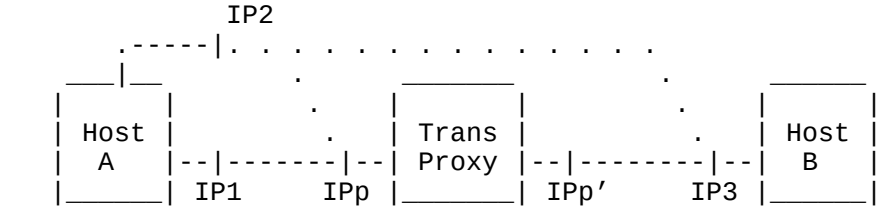

Fig.1: Path ambiguity in presence of transparent proxy

The MPTCP transparent proxy must reside on the initial path used for the first subflow between both connection end points. When one end point (host A) starts the SYN/ACK handshake with its peer (host B), the proxy intercepts the initial packet, derives all connection-relevant information and lets the packet pass.

In case host A and host B are MPTCP-capable, they mutually engage into a MPTCP connection and the proxy stays out of the picture. In case host A is MPTCP-capable but host B is not, the proxy finds out since host B's SYN/ACK packet does not contain the MP_CAPABLE option. At this point, the proxy steps in and provides all MPTCP signaling on behalf of host B throughout the duration of the connection.

Figure 1 illustrates such a situation: Host A initiates a subflow from IP1 to host B's IP3. The transparent proxy sits on this path. While host A believes it sustains an MPTCP connection with host B, host B believes it sustains a conventional TCP connection with host A.

A problem arises when host A wishes to establish a new subflow to host B from another interface, which connects to a different network. Since the new path to host B does not cross the transparent proxy, subflow establishment via MP_JOIN will fail. Instead, host A should establish the new subflow to the proxy's IP address. Host A, however, does not know about the proxy due to its transparence.

In the illustration of figure 1, host A would try to establish the new subflow from the new interface IP2 to host B's interface IP3. Instead, it should establish a subflow to the proxy's IP address marked with IPp.

In order to support establishment of additional subflows, the proxy

has to tell host A to use the proxy's address rather than host B's address as the destination for new subflows.

Currently, MPTCP could accomplish this through a rather awkward procedure:

- The proxy sends the ADD_ADDR option to host A advertising its own IP address (IPp).
- Host A interprets this message as a request for immediate subflow establishment and acts upon it using the same interface it used for the first subflow (i.e. IP1). As a result, both subflows (IP1< =>IP3 and IP1<=>IPp) run along the same path between host A and the proxy.
- The proxy terminates the first subflow (IP1<=>IP3) with host A via a FIN exchange and relays all packets it exchanges with host B (IPp'<=>IP3) to the second subflow (IP1<=>IPp). Then it sends the REMOVE_ADDR option to host A pointing to host B's IP address (IP3).
- Upon reception of the REMOVE_ADDR option, host A removes host B's address (IP3) and talks directly to the proxy (IPp). All future subflows will now be established with the proxy rather than with host B.

While this procedure works, it requires a lot of effort at the beginning of each connection even though it is not known if host A ever wishes to establish other subflows. This is against MPTCP's spirit to keep the initial cost of connection establishment low.

It is recommended to provide an alternative approach, which does not require such effort. This can be done through the following signaling enhancements:

- The ADD_ADDR option only represents a request to cache the enclosed address value together with an address id. This request is independent of further actions or intentions associated with this address. Such a modification of the ADD_ADDR option also supports the enhancements proposed in Section 3.6 and Section 4.4. As mentioned before, the address value does not have to be included in the option in case the ADD_ADDR option refers to the source address of the packet itself.
- The new JOIN_ADDR option is introduced. It requests that the receiving host establishes a new subflow to the address id specified in the option.

- The new DEFER_ADDR option is introduced. It requests that the receiving host uses the designated address id as the destination of all future subflows.

After establishment of the first subflow, the transparent proxy can announce its own address via the ADD_ADDR option and subsequently send the DEFER_ADDR option. No further action has to be taken until host A wishes to start a new subflow to host B. In this case, host A uses the proxy's address as the destination of the new subflow.

5.2. Applicability to 3G/4G Mobile Network Deployments

The transparent proxy is in line with present 3G/4G mobile network deployments, which rely on macro-cellular standards using centralized architecture. Given such infrastructure, the MPTCP transparent proxy can reside on the central router of the 3G/4G network (e.g. packet data gateway node). MPTCP-compliant terminals can initiate connections via the macro-cellular network, which offers wide-area coverage at the price of throughput. Based on availability, the terminal can start additional subflows with other access networks (e.g. WLANs), which are local in nature but usually offer higher data rates.

The MPTCP transparent proxy allows the cellular operator to dynamically offload traffic from licensed to unlicensed spectrum and eventually away from the cellular core in case both end hosts support MPTCP. MPTCP can further leverage off from 3GPP's security since the proxy's initial key is forwarded through the secured cellular network. This thwarts hijacking attacks by outside hosts. A more detailed analysis on security requirements would be desirable in this context.

Note that path-selective MPTCP with transparent proxy provides the same functionality as 3GPP's WLAN internetworking solution [5]. At the same time MPTCP is a simpler and more versatile solution since it does not need tunnel support while providing better middlebox compliance. In addition, it supports multi-flow capabilities and it permits operation as true end-host based protocol. Since operating on layer 4, MPTCP should further be compliant with existing 3GPP standards.

6. Summary of New Messages

This section summarizes the new MPTCP options introduced in the prior sections, and it riefly states their purpose:

MP_SELECT:

This option enforces path-selective operation on the receiving host. It is generally sent on the designated subflow. The option may enclose an address id in case it is sent preemptively, i.e. in break-before-make scenarios before the designated path becomes available. By enforcing path-selective operation, the MP_SELECT option permits low-complexity MPTCP receiver solutions (Section 4.4) as well as dynamic overhead shedding for heavily loaded servers (Section 4.5).

ADD_ADDR:

This option should be re-interpreted. In the new interpretation, it only provides a mapping between address id and address value but abstains from further advice or request for action. When the ADD_ADDRESS option refers to the source address of the packet it is enclosed in, it can omit the address value.

AVAIL_ADDR/UNAVAIL_ADDR:

These options inform the receiving host about the availability/ unavailability status of an interface referred to via an address id. The enclosed address-id permits sending the option from an available interface to refer to an unavailable interface. The options can be combined into one option by including a binary availability flag. They permit the remote host to switftly adjust data transmission to interface tear-down and setup of the local host as outlined in Section 3.6, Section 4.4 and Section 4.5.

JOIN_ADDR:

This option requests that the receiving host initiate a subflow to an address referred to via the enclosed address id. Currently, the functionality of this option is melted into the ADD_ADDR option.

DEFER_ADDR:

This option instructs the receiving host to use a specific address referred to via an address id as the destination for

all future subflows. This option is required for transparent-proxy operation (section 6).

DSS insertion policy for bulk transfer:

To reduce receiver complexity, DSS options should be inserted into all packets of a bulk until the first data ACK is received for a packet contained in the bulk (Section 3.7).

7. Security Considerations

The security considerations established in RFC6181 [2] apply. Additional considerations can be found in [3]. No additional security risks have been introduced through the enhancements proposed in this document.

8. Conclusion

MPTCP has great potential in its applicability to wireless access networks. Especially MPTCP's path-selective operation mode can be considered an attractive solution that facilitates connection migration across access providers and/or access technologies meeting an existing demand. It is strongly recommended to add the proposed enhancements that permit a substantial reduction in design complexity.

MPTCP's multipath capabilities may provide additional benefit in wireless environments. For that to happen, further exploration of the multipath operation space is recommended. In this context, multipath diversity and dynamic path adaptation have been named as principle objectives that may add substantial value beyond that of throughput aggregation. Features to support signaling for path-availability may add further performance benefit. The outcome of such efforts should provide specific guidance to implementors on how design and configuration parameters have to be set.

In wireless environments, MPTCP's core problem is incremental deployment. This problem can be overcome through transparent proxies. While this falls in line with existing mobile network deployments it requires small modifications and enhancements to MPTCP signaling.

9. References

[1] Ford, A., Raiciu, C., Greenhalgh, A., and M. Handley, "Architectural Guidelines for Multipath TCP Development", RFC 6182, March 2011.

[2] Bangulo, M., "Threat Analysis for TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6181, March 2011.

[3] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", I-D ietf-mptcp-multiaddressed-03, March 2011.

[4] Wishik, D., Raiciu, C., Greenhalgh, A., and M. Handley, "Design, Implementation and Evaluation of Congestion Control for Multipath TCP", 8th USENIX Symposium on Networked Systems Design and Implementation , March 2011.

[5] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects, 3GPP system to Wireless Local Area Network (WLAN) interworking; System description (Release 9)", 3GPP TS 23.234 , December 2009.

[6] Wikipedia, "MIMO", http://en.wikipedia.org/wiki/Multiple-input_multiple-output_communications , July 2010.

[7] Karim, M. and M. Sarraf, "W-CDMA and cdma2000 for 3G Mobile Networks", McGraw-Hill Telecom , 2002.

[8] Raiciu, C., Niculescu, D., Bagnulo, B., and M. Handley, "Opportunistic Mobility with Multipath TCP", http://nrg.cs.ucl.ac.uk/mptcp/mobility.pdf , 2011.

Authors' Addresses

Georg Hampel
Alcatel-Lucent
600 Mountain Ave
Murray Hill, NJ 07974
US

Phone: +1 908 582 2377
Fax: +1 908 582 8222
Email: georg.hampel@alcatel-lucent.com

Thierry Klein
Alcatel-Lucent
600 Mountain Ave
Murray Hill, NJ 07974
US

Phone: +1 908 582 3585
Fax: +1 908 582 8222
Email: thierry.klein@alcatel-lucent.com