Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks

Giovanna Carofiglio, Luca Muscariello, Michele Papalini, Natalya Rozhnova, Xuan Zeng

Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks

Giovanna Carofiglio
Cisco Systems
gcarofig@cisco.com

Luca Muscariello
Cisco Systems
lumuscar@cisco.com

Michele Papalini
Cisco Systems
micpapal@cisco.com

Natalya Rozhnova
Cisco Systems
nrozhnov@cisco.com

Xuan Zeng
IRT SystemX, UPMC
xuan.zeng@lip6.fr

ABSTRACT

One of the most appealing features of Information-Centric Networking (ICN) is its agile connectionless transport model based on consumer requests and hop-by-hop forwarding.
By relaxing end-to-end constraints, ICN empowers a distributed in-network control with the potential to improve congestion management over heterogeneous wired/wireless media and in presence of mobility. However, little effort has been devoted so far to the exploration of ICN capabilities in this space. In this paper, we contribute an understanding of the opportunities for ICN in-network control over wireless mobile networks and a proposal for simple, yet very effective mechanisms for in-network loss detection and recovery to complement receiver-driven control. More precisely, we introduce *(i) WLDR*, a mechanism for in-network Wireless Loss Detection and Recovery that promptly identifies and recovers channel losses at wireless access point and *(ii) MLDR*, a mechanism for preventing losses due to consumer/producer mobility via explicit network notification and dynamic on-the-fly request re-routing.
We setup a realistic wireless simulation environment in ndnSIM using IEEE 802.11n connectivity and evaluate WLDR-MLDR performance. The results show significant benefits over consumer-based solutions with or without explicit loss notification, while also removing any dependency from network and application timers.

CCS Concepts

•Networks → Network protocol design; Network simulations; Network performance analysis; Mobile networks;

Keywords

in-network control, loss recovery, wireless networks, mobility

1. INTRODUCTION

Next generation 5G networks are expected to support communications over an heterogeneous wireless access and in the presence of mobility. The high level of diversity of wireless media characteristics, combined with frequent end-point mobility, clearly present a challenge.

Despite the numerous ad-hoc TCP flavors and layer-2 to layer-4 workarounds proposed in the last decade, TCP-like congestion control still fails to effectively incorporate identification and management of wireless channel or mobility losses. To cite a few of the traditional concerns: misinterpretation of wireless losses as congestion signals and consequent throughput degradation, delays due to loss detection and recovery at sender-side, losses generated by connection state migration and re-establishment in case of mobility.
Most of the issues have the same root cause: the end-to-end nature of traditional TCP-based control delegating connection management at the sender node only.

Information-Centric Networking (ICN) has the potential to overcome all such limitations in virtue of a radically different transport model that centers the communication around named data packets retrieved by the consumer in a request/reply fashion. In this paper we consider NDN as reference ICN architecture, even though the basic ideas of our proposal can be applied with minor modifications to other ICN architectures where the forwarding decisions are taken hop-by-hop. In this architecture the end-to-end principle is totally revisited: the unique end-point is the consumer and its requests can be satisfied by any traversed router storing the corresponding data in its local cache. Otherwise, requests are dynamically forwarded in a hop-by-hop fashion based on content names with no knowledge of the sender a priori. Soft-state associated to pending requests enables fully distributed in-network decisions that may help rate and congestion control management, otherwise performed at the consumer side only.

More precisely, ICN in-network control can allow (i) to shield consumer-side rate and congestion control from wireless events of different nature, such as wireless channel or mobility losses, (ii) to disentangle and separately manage such events at the optimal place in the network, (iii) to achieve fastest detection and recovery. Several papers have studied rate and congestion control in ICN, but only a few ([10], [1]) have specifically focused on wireless mobile networks and none has explored in-network control operations.

The goal of this paper is to address the impact of wireless and mobility losses on overall rate and congestion control performance and to propose simple, yet effective, mechanisms for in-network loss detection and recovery to complement receiver-driven rate and congestion control. More precisely, we introduce *(i) WLDR*, a mechanism for in-network

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Wireless Loss Detection and Recovery to promptly identify and recover channel losses at wireless access point and *(ii) MLDR*, a mechanism for preventing losses due to consumer/producer mobility via explicit network notification and dynamic on-the-fly request re-routing. Further, we setup a realistic wireless simulation environment in ndnSIM using Wi-Fi 802.11n connectivity and assess MLDR/WLDR performance in comparison with consumer-driven alternatives based on timers or on Explicit Loss Notification (ELN).

The results show a significant reduction in terms of flow completion time or request satisfaction time, i.e. the time between the first request emission and the corresponding data packet reception at the consumer, which is particularly important in case of latency-sensitive applications. In addition, our proposal provenly removes any dependence from network/application timers that existing ICN solutions rely upon and, hence, issues on their setting.

The remainder of the paper is organized as follows: in Sec.2 we give an overview of previous work in ICN and TCP/IP literature. Sec.3 introduces our proposal and describes its implementation (Sec.4). The evaluation is gathered in Sec.5, while conclusions are reported in Sec.6.

2. RELATED WORK

Reliable communications in wireless mobile networks are realized either at the transport layer or at the link layer. Link layer solutions can cope with channel losses, whereas they cannot manage mobility efficiently. Therefore the literature has focused on L4 techniques. A large body of work, beyond ICN, has highlighted the issues of traditional TCP-based congestion control over wireless channels and due to mobility. To cite the main known drawbacks: *(i)* poor performance due to higher error rates on wireless links misinterpreted by TCP sender as congestion notifications; *(ii)* performance degradation due to connection migration and slow start at every mobility event; *(iii)* packet losses due to mobility in absence of soft handover mechanisms. Classical countermeasures can be classified into three categories:

Transport-layer solutions isolating wired and wireless segments: e.g. I-TCP [2], Freeze TCP, Mobile TCP [7] and SplitTCP [11]. Besides the differences, the advantages of such category of approaches relate to the capability of shielding the wired segment from the lossy wireless part without requiring end-host modification. As a drawback, the buffering at the proxy between wired and wireless segment can be considerably high and may introduce additional latency. Also, some of these approaches break the end-to-end semantics, complicating rate control at server side. TCP Westwood [12] is an end-to-end transport-layer scheme based on bandwidth estimation: it continously monitors the rate of returning ACKs and uses it in computation of window and ssthresh. However, this implies a huge dependance from ACKs.

Link-layer solutions: Snoop TCP [4], TULIP [15], MAC MIB [18] to cite a few. They all leverage ACKs or MAC layer observations to identify wireless losses and to react by recovering them or avoiding unnecessary congestion window reduction. For instance, Snoop TCP involves link interface sniffing at the base station for any segment to and any ACK coming from the mobile host and performs buffering intended to support local retransmissions. All such approches are MAC-Layer specific and consider the case of mobile consumer only. Also they are limited to single-hop wireless communications.

Explicit notification solutions leveraging NACKs and Loss Differentiation Algorithms (LDA) to acknowledge the sender and to hence trigger retransmission. To this aim, [5] leverages inter-arrival times, [19] uses a relative one-way trip time (ROTT) for congestion signaling. More precisely, the spikes in ROTT measurements are used to differentiate different degrees of congestion. When spikes are observed, losses are considered as due to congestion, otherwise due to wireless environment. In [9] authors compare different LDAs and propose their own ZigZag scheme. The decision about the nature of losses (congestion ot wireless) is based on the number of losses and the difference between ROTT and its mean value. The comparison shows that each of them may perform well in some particular conditions and be less effective otherwise.

ICN literature only marginally considered congestion control implications of mobile and wireless communications. [1] deals with wireless ad hoc networks and proposes Interest rate regulation and retransmission at the receiver based on a timer that adapts to RTT delay variations. The advantage is clear over other ICN approaches relying on a fixed retransmission timer. However, the solution does not distinguish the nature of losses nor copes with faster in network recovery, which is the primary goal of this paper. The work in [10] also tries to dynamically tune the Interest lifetimes based on RTT measurements in order to improve the effectiveness of retransmissions for video streaming applications at the receiver. The nature of losses is differentiated with simple ECN scheme in the network that marks the incoming Data packets in case of congestion and considers all losses not due to congestion as due to wireless channel errors. Mobility is not taken into account.

3. PROPOSAL

In this section we present our proposal for in-network wireless and mobility loss detection and recovery, respectively *WLDR* (Sec.3.1) and *MLDR* (Sec.3.2). While addressing two separate problems, their design shares the same principles. We aim for a solution that

- is *layer-2 agnostic*, applicable to any wireless medium irrespectively of specific access characteristics;
- *differentiates the nature of losses*, to distinguish and separately handle wireless channel losses from buffer overflows due to congestion or from losses due to mobility timeouts.
- decouples the point in the network where to perform detection and recovery of losses by leveraging *Explicit Loss Notification* (ELN);
- leverages *fast in-network loss detection and recovery* in sub-round trip time scale or before the expiration of an Interest timer at the consumer.

Besides reactive detection and recovery, the rationale behind in-network loss management is to make NDN/CCN data plane robust to losses and insensitive to pending interest table (PIT) timer management, which is a non trivial operation in very lossy environment. Also in-network loss management reduces misguided congestion and flow control decision at the consumer which ultimately is responsible for reliable transport. Finally, in WLDR/MLDR definition we seek for lightweight mechanisms in terms of signaling overhead, additional state at routers and complexity. Implementation considerations are reported in Sec.4.

3.1 Wireless Loss Detection and Recovery

WLDR is implemented at face level and introduces an additional sequencing on packets to detect losses. Two neighbors are interconnected using adjacencies, called faces in NDN/CCN. An adjacency is a unicast bidirectional channel between two nodes. It may also be established among nodes connected to the same broadcast medium, as in IEEE 802.11 for instance. Sequentiality is then guaranteed on a per-face basis. In the same way, using multiple wireless faces in parallel, the stream of packets generated by each face is associated to a different sequencing. WLDR is not able to detect losses end-to-end, such task being the responsibility of the transport protocol. However, by applying WLDR at each hop, WLDR can be simply extended to the multi-hop wireless case.

WLDR introduces new fields in the headers of Interest and Data packets (see Sec. 4) to store the sequence numbers used by the algorithm. These values have limited scope on a single wireless link and they may be modified every time a packet traverses a new wireless link implementing WLDR. It is important not to confuse the sequence number used by WLDR with the one that may be present in the Interest/Data names.

To illustrate WLDR basic functioning let us consider the simple example in Fig. 1. In such scenario, a consumer is connected to a wireless Access Point (AP hereinafter, with no reference to any specific wireless technology) through a wireless link and sends Interest packets to the AP to request a given content item. To keep track of Interests sent through a given face, the consumer maintains a counter per-face indicating the sequence number for the *next* Interest to be sent (hence, indicated with *next* in Fig. 1, Alg.1). Such sequence number is also added in the header of every Interest to be sent, then the counter is increased by one. In Fig. 1, *next* is equal to 3, i.e. the consumer will associate the label 3 to the next Interest to be sent and update it to 4.

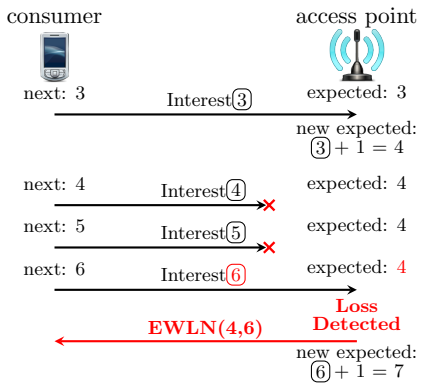


Figure 1: WLDR wireless loss detection

Loss Detection. The sequence number in the Interest is used by the AP to reconstruct the sequence of packets and so detect potential losses. To verify whether the incoming Interest is the expected one, the AP keeps an *expected* sequence number value. Upon Interest reception, it compares the sequence number in the Interest packet with such expected value. If they coincide, the AP simply updates its expected value (increasing by 1). E.g. in the example, the expected value at AP is 3 and at reception of an Interest with the same sequence number, the AP updates it to 4.

If the expected value and the Interest sequence number are not the same, the AP detects a loss. This is the case in Fig. 1 at reception of the Interest packet with sequence number 6.

Fig. 1 describes WLDR between a wireless consumer and an AP. However, WLDR applies to any couple of nodes connected through a wireless link without requiring a distinction between consumer/producer or wireless node/AP. In contrast, Previous mechanisms depend on the role of the wireless node. E.g. two close algorithms are presented by Balakrishnan et al. [3] and Biaz et al [6], the first one working in case of wireless producer only, while the second one working for wireless consumer node only. The only distinction required by WLDR is the one between sender and a receiver node, since WLDR is a directional protocol. The sender is responsible to enumerate the packets and recover losses, while the receiver checks the sequence number in the packet to detect losses and to notify the sender. A node can be a sender and a receiver at the same time and this is the case over a bidirectional link.

In addition, since WLDR is implemented at the face level, it does not keep any per-flow information, since it does not make any distinction between packets from different applications, nor between Interest and Data packets. As consequence, mechanism failure at the base station does not lead to connection disruption, as it is the case for alternative proposals in the TCP/IP world (I-TCP [2] or WTCP [17]).

Loss Distinction. The mechanism above described distinguishes wireless channel losses from losses due to mobility in virtue of the sequence number labeling performed at the output face at the sender. Indeed, losses due to mobility are caused by the absence of available output faces to reach the mobile consumer/producer (respectively for Data/Interest packets). Thus, they occur before WLDR labeling. Packets queued in the output buffer may still suffer from drops due to congestion: while managing congestion losses is out of the scope of this paper, we observe that WLDR intervenes only at service time before packet transmission. In this way, packet losses due to congestion do not interfere with WLDR mechanism.

Loss Notification. In cases of losses, the AP notifies the consumer with an *Explicit Wireless Loss Notification* (EWLN) message. The EWLN contains the current expected sequence number (4 in Fig. 1), and the sequence number of the last received Interest (6 in the figure) to notify the loss of the expected packet, namely 4, and packets in between, namely 5. Once EWLN is sent, the AP updates its expected sequence number to the last received Interest plus 1 (7 in the example).

The usage of EWLN packets is in contrast with most of the MAC layer retransmission algorithms, where ACKs are used to track losses. ACKs are used because they are not subject to the communication patterns, since a node can detect a loss using timeouts. The drawback of ACKs is that a node may keep retransmitting packets even if the wireless channel is down, and this is what happen today in the IEEE 802.11 standard. Instead, with EWLN packets a node receives a loss report only if the channel is good enough to transport some packets. In this way packets are retransmitted only when the channel conditions are good enough, with a resulting better performance. Notice that in case an EWLN message gets lost, the baseline case of retransmission by timer at the receiver happens.

In-network recovery. WLDR is designed to recover the

losses in-network, without sending any signal to the application/transport layer running at consumer side, but a different recovery strategy can be implemented (e.g. explicit loss notification to consumer, see Sec. 5.2). Loss recovery is enabled by maintaining a buffer of Interest/Data packets or, depending on the forwarder implementation, by reference to the content store (CS), for data, or the PIT, for interests with no need to copying data. In Fig. 1, when the consumer receives the EWLN packet it retransmits all Interests indicated by the EWLN. Interests packets retransmitted by the consumer are sent with a new sequence number to keep the two nodes synchronized (in the example, the AP is expecting packet 7) and to enable future retransmission in case of channel losses. In Fig. 1 the new sequence values would be 7 and 8, respectively, for the two lost Interests 4 and 5. The detailed WLDR algorithm is reported in Alg.1,2.

Algorithm 1: WLDR algorithm (sender side)

```
buffer [] ;                    // Local buffer with sent packets
bufferSize ;                              // Local buffer size
next ;                                  // Next sequence number
Function OnSendPacket (packet)
    packet.setSeqLabel(next);
    buffer[next % bufferSize] = packet;
    next++;
    send(packet);
Function OnEWLN (ewlnPkt)
    expectedPkt = ewlnPkt.getExpectedPkt();
    lastReceivedPkt = ewlnPkt.getLastReceivedPkt();
    if ((next - expectedPkt) <= bufferSize) then
        // lost packets are in the buffer
        while (expectedPkt < lastReceivedPkt) do
            lostPkt = buffer[expectedPkt % bufferSize];
            if (lostPkt is not expired) then
                lostPkt.setSequenceLabel(next);
                buffer[next % bufferSize] = lostPkt;
                send(lostPkt);
                expectedPkt++;
                next++;
```

3.1.1 WLDR enhancements

Adjusting Interest/Data lifetime: To avoid retransmissions for packet with expired PIT entry, we use the lifetime field in the Interests and we add an equivalent one in the Data that contains the copy the Interest lifetime. When the sender node labels a packet (Interest or Data) for the first time, it stores a timestamp that works in a similar way to the PIT timer. In case of retransmission, the sender computes the time elapsed from the first packet transmission and the packet is retransmitted only if such time is less then $\alpha \times lifetime$, where $\alpha \in [0, 1]$. With α close to 1 we have more chances to retransmit packets for which the PIT timer is already expired, while with α close to 0 we may not retransmit valuable packets.

Algorithm 2: WLDR algorithm (receiver side)

```
expected ;                          // Expected sequence number
Function OnReceivePacket (packet)
    pktLabel = packet.getSeqLabel();
    if (pktLabel != expected) then
        ewlnPkt.setExpectedPkt(expected);
        ewlnPkt.setLastReceivedPkt(pktLabel);
        send(ewlnPkt);
    expectedLabel = pktLabel + 1;
```

Reinitialization of sequence number: A critical component of WLDR is to keep nodes in sync in case of handovers. There are two possible scenarios: *(i)* the mobile node temporally disconnects from an AP and reconnects to the same one, or *(ii)* the mobile node migrates to a new AP. In the first case there is just a temporary disconnection, so we keep using the same counting sequence. This has the advantage that, if some packets got lost during the disconnection, the two nodes may recover them. Instead, in the second case, we reset all the WLDR state on both the nodes. In this case, the losses due to mobility are handled by MLDR.

3.2 Mobility Loss Detection and Recovery

NDN/CCN name-based connectionless transport significantly simplifies mobility management. However, mobility events may still lead to losses, as a result of Interest expiration in PIT due to temporary unavailability of forwarding output face on the path between consumer and producer. The goal of MLDR is to handle losses due to either consumer and producer mobility, during the time period required by mobility management protocols to update network forwarding state. In the following, we consider separately consumer and producer mobility and introduce MLDR countermeasures accordingly.

3.2.1 Consumer mobility

Consumer mobility is natively supported in NDN/CCN, since after moving and attaching to a new base station, a consumer can reissue lost Interests to retrieve data available from the closest cache. The main problem when a consumer node changes its point of attachment is that the Interests that are already pending in its PIT will never be satisfied due to the symmetric routing property. Such losses due to mobility might be mistakenly assimilated to congestion, which may further affect flow control through, for instance, a window or rate decrease.

Loss detection. Interest losses are typically detected by means of PIT timer expiration, with the timer being set equal to the Interest lifetime. Besides known timer setting difficulties, waiting for timer expiration implies retransmission delays and negatively affects congestion window evolution, regardless of the specific rate controller used at the consumer. In MLDR, we base consumer mobility loss detection on local "face up/down" signaling, also distinguishing them from losses of different nature. In practice, this is a task of link-layer technologies. For MLDR to benefit from such signals they should be as reactive as possible. Indeed, upon change of AP by the consumer, the corresponding face goes down until the connection with another AP is successfully established. A signal of "face down" triggers in MLDR a PIT lookup in order to filter all entries associated to pending Interests forwarded through such face.

Loss notification and recovery. For each of these entries, a *Notify and Retransmit* procedure is initiated. Its goal is twofold: *(i)* to inform the application of the mobility event in order to prevent congestion window reduction upon PIT timer expiration and *(ii)* to trigger Interest retransmission, on another available face, if any, or later in time on the same face when available. To this purpose, MLDR first checks whether alternative output faces are available. If so, it retransmits the Interests. Otherwise, if no alternative face is available, MLDR adds a special mobility flag, denoted as

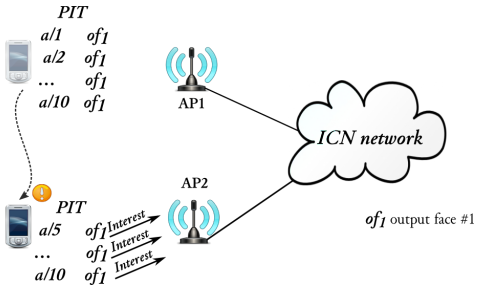


Figure 2: On detection of consumer mobility

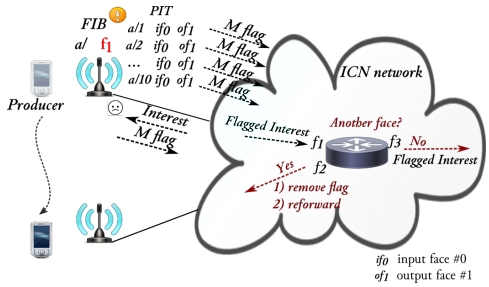


Figure 3: On detection of producer mobility

M to the Interest and sends it back to the application to inform it about the mobility event. On a "face up" signal, MLDR retransmits all pending Interests that have no output face. It is important to remark that PIT entries are not removed by MLDR, rather updated in the implementation as far as concerns the pointers to the output faces. In presence of multiple output faces corresponding to a PIT entry, only the face that is down is removed.

An example of consumer mobility detection is illustrated in Fig. 2. The consumer sends Interests $a/1$ to $a/10$, while connected to $AP1$. The first four Interests are satisfied before its disconnection from $AP1$. When the consumer connects to $AP2$, $a/5$ to $a/10$ Interests are still pending in its PIT, so a retransmission to $AP2$ can be triggered.

3.2.2 Producer mobility

We now address producer mobility, namely the case where a change of AP for the producer breaks the path between consumer and producer with consequent loss of Interest and Data packets in both directions. MLDR objective is to recover such losses during the time period required by the mobility management protocol to update network forwarding state according to new producer location. To do so, MLDR notifies in-path routers about the mobility event, to enable a retransmission of the Interests over alternative paths within PIT entry/application lifetime. Indeed, the proposed solution is a generalization of MLDR detection and notify plus retransmit procedure described above.

Loss detection. A mobility event detection occurs in two cases: first, for an Interest arriving at the old AP where it cannot be forwarded anymore and, second, like in consumer mobility case, upon "face down" signaling at the same AP.

Loss notification and recovery. In the case where the AP cannot forward an incoming Interest, a flag M is set in the Interest packet and the latter is sent back on the incoming face. A "face down" signaling at AP triggers MLDR detection procedure as previously described and the same procedure as for consumer mobility applies with possible retransmission over alternative available faces. Otherwise, the $M-$flagged Interests will be generated and sent through the corresponding incoming faces (whose pointers are stored in PIT entries) to propagate back to the consumer an explicit mobility notification. Corresponding entries are then removed from the PIT.

An example is reported in Figure 3. When a face of_1 goes down, the pending Interests $a/1$ to $a/10$ will be flagged and sent back on their incoming face(s). If the AP is still receiving the Interests for this particular content, it will flag them and forward back.

Propagation of $M-$flagged Interests may give rise to retransmission on other available faces, e.g. those dynamically created by mobility management protocols to reconnect producer at new location. Upon reception of an $M-$flagged Interest at a given network node, a decision about Interest re-forwarding should be taken. The Interest is re-forwarded in case where the face from which the flagged Interest arrived is a unique output face for the corresponding PIT entry. If this face is not unique, the flagged Interest is rejected and the corresponding output face is just removed from the PIT entry. If the flagged Interest can be re-forwarded,a FIB lookup is performed to find another face to use according to the node forwarding strategy, which can support multipath or not, independently from our scheme. Note that multipath does not require any change to MLDR. The M flag is then removed from the packet and the Interest is re-forwarded through the selected face.

If no other face is available, the $M-$flagged Interest is forwarded to the list of corresponding incoming faces. The PIT entry is then removed. It is easy to see that if no nodes of the path have alternative faces to re-forward the $M-$flagged Interest, it will finally arrive to the consumer to notify it about the mobility loss. As for wireless channel losses, unnecessary congestion window reduction may be prevented by the explicit notification of mobility loss. It is important to observe that FIB entries are not altered, in fact, MLDR never removes a face from the corresponding FIB. The algorithm in case of producer mobility is detailed in Alg. 3.

Algorithm 3: MLDR algorithm (Producer mobility)

```
Function OnProducerWirelessFaceDown (face)
    while (PIT.hasNextEntry()) do
        reforwardInterests(face,PIT.getNextEntry());
Function OnMobilityFlag (interest,face)
    interest.unsetMobilityFlag();
    reforwardInterests(face,PIT.match(interest));
Function reforwardInterests (face,entry)
    if (face ∈ entry.outFaces() &
                    |entry.outFaces()| = 1) then
        interest = entry.getInterest();
        nextHopFaces = FIB.match(interest);
        forall (outFace ∈ nextHopFaces) do
            if (outFace ∉ entry.InFaces() &
            outFace ∉ entry.usedForRetransmission()) then
                residualTime = entry.expiration() - now;
                interest.setLifetime(residualTime);
                outFace.send(interest);
                entry.addUsedForRetransmission(outFace);
                return;
        interest.setMobilityFlag();
        forall (inFace ∈ entry.inFaces()) do
            inFace.send(interest);
        PIT.remove(entry);
```

3.2.3 MLDR Enhancements

Adjusting Interest lifetime: One of the operations

that each node (consumer included) should perform before re-forwarding a $M-$flagged Interest, is to adjust the lifetime value carried by the re-forwarded Interest to the residual time left until the PIT entry expiration. In addition, as for WLDR in case of retransmissions, the decision about re-forwarding may be based on this value. For example, only if residual time is large enough the Interest is re-forwarded, otherwise, it will be sent back to downstream nodes as in case of no alternative faces. Note that we do not modify the PIT timers.

Preventing retransmission loops: To avoid loops due to re-forwarding a $M-$flagged Interest over the same output face, a list of faces used for retransmission is added to the subset of PIT entries affected by a mobility loss notification. Every time a $M-$flagged Interest is forwarded, the corresponding output face is added to the list. Only if such face has not yet been used for retransmission of an Interest with the same name during PIT entry lifetime, the Interest is effectively re-forwarded. Otherwise it is further propagated with the M flag in the direction of the consumer.

RTT reduction: Such enhancement to baseline MLDR only applies to the class of congestion controllers at consumer side leveraging RTT monitoring. Indeed, in-network retransmissions may introduce an additional delay affecting RTT monitoring at the consumer which may be misinterpreted as due to congestion. To avoid this, we aim at correcting RTT estimation at the consumer by removing the RTT component due to retransmission. This scheme updates the PIT entry corresponding to an Interest to be reforwarded with a retransmission timestamp. Similar techniques have been proposed in [17]. Thus, we need two PIT timestamps: the original sending time and the retransmission time. Note that in case of multiple retransmissions of an M-flagged Interest, we record only the latest retransmission timestamp. In this way, the difference between the two timestamps indicates the overall retransmission time for the Interest. Upon reception of Data matching the retransmitted Interest, the difference between these two timestamps is computed and stored in the Data packet. Thus, it can be removed in RTT computation at the consumer side. Such a scheme may also be used by WLDR mechanism to explicitly notify the application about the additional delay introduced by the retransmission process.

General comments: Before the performance evaluation in Sec. 5, we observe that immediate local retransmission at consumer side may already improve Interest satisfaction time w.r.t. timer-based retransmissions, but the latency reduction gains are even more important in presence of in-network retransmissions. In both cases, MLDR enables retransmission at sub-RTT scale unlike any approach based on timer expiration at consumer side or explicit notification and retransmission at the consumer. It is important to observe that even if other nodes than consumer/producer are mobile, MLDR can still deal with that, under a more complex mobility management (out-of-scope here). Finally, from the security point of view, we remark that MLDR does not modify any ICN data plane information that could introduce opportunities for attacks.

4. IMPLEMENTATION

In this section we describe the additional state required in each node by WLDR and MLDR as well as the additional header's fields that we use in Interest/Data packets.

WLDR. The ICN forwarder of a node running WLDR (a WLDR node hereinafter) needs to locally store two values per face: the next sequence number, `next_seqno` and the expected sequence number, `expected_seqno`. Such values are required for packet labeling at the sender and loss detection at the receiver. In addition, a WLDR node keeps track of the sequence of sent packets and temporarily store them (or store a pointer to their copy in PIT/CS) in a circular buffer, one per output face: each packet is hence stored at position `seqno % buffer_size`. The amount of additional state required by WLDR depends on the size of this buffer. A too small buffer may reduce chances to recover losses, because of packets overwritten too frequently. In order to dimension such buffer, we consider a 'bandwidth delay product' rule where delay stands for PIT timer (logically content lifetime which however may vary according to the considered application). In the current implementation this results in a buffer of 8192 packets considering 1sec PIT timer and a Wi-Fi link at 100Mbps, which is close to the number of Data packet (of 1500 bytes) that can be sent in a second on this link. This guarantees that we almost never overwrite any useful packet, for a cost in terms of memory of around 11MB. However, if we record only the pointers to the PIT/CS entries where these packets are already stored, the considered buffer size can be reduced to 64kB.

MLDR. MLDR does not require substantial modifications of the existing data structures or packet format. However, additional information has to be stored by each router to prevent retransmission loops of forwarded Interests and perform the RTT reduction (cfr Sec. 3.2.3).

More precisely, a list of faces used for retransmission is added to PIT entries affected by a mobility loss notification.

To perform the RTT reduction signaling, two additional timestamps need to be stored in the PIT entries: the time of the original sending of an Interest and the moment of its retransmission (if any).

In terms on computational complexity, the basic implementation of the *OnProducerWirelessFaceDown* function (see Alg. 3) requires a linear scan of the PIT on the AP node in case of producer mobility. In order to reduce such complexity, we store the list of pointers to associated PIT entries at each output face. With this data structure we can obtain the required PIT entries in constant time, with no iteration over the entire PIT. The complexity involved by lookup of $M-$flagged Interest is not different than that of a standard Interest lookup.

Packet format. WLDR and MLDR introduce four new fields in the packets: the *sequence number*, the *Data lifetime*, the *mobility flag*, and the *RTT reduction*. The sequence number is an integer value that is introduced both in Interest and Data. The Data lifetime is the equivalent of the Interest lifetime, and is specific to Data (Interests have this field by default). The mobility flag, that requires a single bit, and the RTT reduction, which is an other integer value, are field required only on the Interests.

EWLN packet: WLDR also introduces a new signaling message, denoted as EWLN (Explicit Wireless Loss Notification) which carries: (i) a flag specifying that the packet is an EWLN, (ii) the expected sequence number at the receiver when the loss is detected and (iii) the sequence number of the last received packet. It is worth observing that such signaling message has a one hop validity and it is discarded by the receiver (e.g. the AP or the station) after having

triggered either a retransmission of the missing packet(s) or the creation of explicit notification message(s) carrying the name of the missing packet(s) for further propagation.

5. EVALUATION

To assess WLDR/MLDR performance, we setup a realistic wireless simulation environment in ns3 2.24/ndnSIM 2.1 using IEEE 802.11n access, as further detailed. We suppose that the congestion control is built-in in the transport protocol. Thus, with no loss of generality, we implemented a receiver-driven window based congestion control scheme based on RAAQM[8] at the consumers.

5.1 ICN over IEEE 802.11

We assume all nodes are connected to the same broadcast medium shared in infrastructure mode, namely IEEE 802.11n on 5GHz frequencies, with a single base channel of 40MHz with short guard intervals (SGI), using a single antenna at either the AP and the wireless nodes (denoted as STAtions).

Channel characteristics and contention: The PHY rate adaptation is minstrel [13] for High Throughput (HT) rates, i.e. MCS from 0 to 7 (corresponding to Data rates from 15Mbps to 150Mbps). 802.11 frame aggregation is also enabled with a maximum A-MSDU size of 7935 Bytes and A-MPDU maximum size of 64kB with block ack which enable high application throughputs. When multiple STAs have a face established with the same AP, multiple access is managed by 802.11 EDCF which implies transmission latency and bandwidth sharing among active stations. A face between a STA and the AP is characterized by a time varying capacity that depends on a number of factors like radio conditions, PHY rate selection, medium sharing. In this work, we assume a small cellular Wi-Fi deployment managed by a single entity that can engineer and manage radio planning and 802.11 tuning.

Coverage and mobility: In our simulations each AP operates with a maximum power of 40mW (16dBm), that enables a maximum radio range of 120 meters in outdoor. STAs are assumed to move in a fully covered geographic area performing handover from one cell to another using the Hysteresis handoff algorithm described in [14] to perform handovers among the Wi-Fi cells. The hysteresis handoff has been shown to give the best performance in terms of handover latency. See [16] for more details about 802.11n parametrization.

5.2 WLDR Evaluation

We first evaluate WLDR in the scenario illustrated in Fig. 4, where one consumer and one producer are connected by means of a 802.11n to a wired network represented by $AP1$, $AP2$ and one intermediate router. During the simulations, these two nodes move back and forth from the two APs as indicated by the arrows. We use mobile nodes in order to test our algorithm with different signal conditions that are variable according to the distance between the STA and the related AP (from 0 to 80 m). We let the speed vary between 3km/h to 50km/h. The STAs remain connected to the same AP (no handovers). The propagation delay of the wired links is set to 1ms, while the link capacity changes according to the simulation. The propagation delay of the wireless links depends on the distance between STA and AP. The workload consists in 10 parallel flows of 50,000 packets ('flow' here stands for retrieval of a content item). Intermediate caching is disabled to allow the observation of wireless losses at both ends.

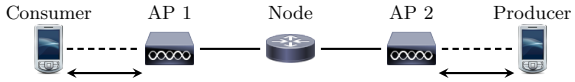


Figure 4: WLDR test topology.

Fig 5(a) shows the average flow duration time for different values of Interest retransmission timer (set equal to PIT timer), when the mobile nodes move at 10km/h. In the figures we compare three alternatives: *NO WLDR* indicating the simulations without WLDR, rather with consumer timer-based retransmissions, *WLDR* and finally, *ELN TO C* indicating the solution leveraging Explicit Loss Notification (ELN) messages to the consumer every time a wireless channel loss is detected. In the latter case, we use WLDR detection, but instead to recover the losses in the network, we notify the consumer who immediately retransmits the Interest, without waiting for the timeout nor decreasing the congestion window.

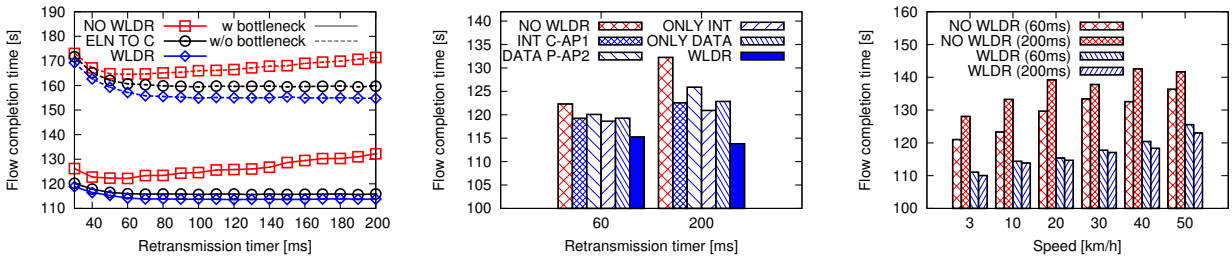
In absence of WLDR, one can observe a significant dependency of flow completion time on retransmission/PIT timer. Indeed, if the timer value is too large, waiting for a timeout to detect a loss is too costly. On the contrary, its the timer is too small (w.r.t. the experience average round trip time, which is of 50ms) unnecessary timer expirations cause Data discard even in absence of losses.

If WLDR in-network detection considerably improves flow completion time, its in-network recovery also enhances overall performance when compared against *ELN TO C* solution. In fact, the recovery time for *ELN TO C* depends on end-to-end network latency, while WLDR recovery time only on wireless hop latency. Hence, the presence of a bottleneck in the wired part of the network (we modify wired link capacities from 300Mbps to 60Mbps) increases the inefficiency gap of *ELN TO C* over WLDR.

We now break down WLDR gains into its components, namely detection and recovery of Interest rather then Data packets and WLDR at consumer side (between consumer and $AP1$) rather than at producer side (between producer and $AP2$). To this purpose, we enable WLDR only partially in the simulation in order to quantify gains due to each component. Results are reported in Fig. 5(b). The speed of the moving nodes is set to 10km/h and we use two values for Interest retransmission/PIT timer: 60 ms (best observed value without WLDR in the previous simulation) and 200ms. The bottleneck is in the wireless part of the network. We show the flow duration associated to 6 cases: (i) WLDR is not active (*NO WLDR*), (ii) Interest recovery between consumer and AP1 only (*INT C-AP1*), (iii) Data recovery between the producer and AP2 only (*DATA P-AP2*), (iv) Interest recovery everywhere (*ONLY INT*), (v) Data recovery everywhere (*ONLY DATA*), (vi) WLDR is fully activated (*WLDR*).

As expected, the timer value has no impact of WLDR, while it visibly effect the performance for *NO WLDR*. Comparing *ONLY INT* and *ONLY DATA*, one can observe that recovering Interest is more advantageous than recovering Data, especially when they are recovered at consumer side. Intuitively, this allows detection and recovery of losses on the first hop, thus significantly reducing recovery time w.r.t. timer-based or ELN-based consumer retransmission.

Data packet recovery is also important, as they are bigger



(a) (b) (c)

Figure 5: (a) Flow completion time with and without bottleneck; (b) Flow completion time with WLDR partially activated; (c) Flow completion time for different speeds.

than Interests in size, hence more prone to losses. The benefits for in-network Data recovery are clearly more important when performing it at producer side on the first hop for Data packet. It is also worth noticing that the difference between the *DATA P-AP2* and *ONLY DATA* is slightly higher in case of timer set to 200ms which allow for more retransmissions of Data before timer expiration. We can conclude that WLDR is insensitive to retransmission/PIT timer value provided that the timer is higher than the average round trip time, so accomodating in-network (possibly more than one tentative) retransmissions.

Finally, Figure 5(c) shows the flow completion time versus mobile nodes speed for 60ms and 200ms timer values, with the bottleneck still in the wireless network. WLDR always outperforms the case with consumer retransmissions (from 7.22% up to 12% for 60ms timer, between 13% and 18% for 200ms timer).

5.3 MLDR Evaluation

In this section we quantify the effectiveness of MLDR scheme by analyzing consumer and producer mobility separately (we consider them jointly in Sec.5.4) The topology under study is reported in Fig. 6 and consists of the root node acting as consumer or producer in case of producer or consumer mobility respectively, of 6 network routers and 6 IEEE 802.11n Access Points(AP_1 to AP_6) at 50m of distance each other. All wired links have a constant propagation delay of 1ms. In the simulations the STA moves linearly across the APs, as described by the dashed arrow in Fig. 6. For the producer mobility, the examples of the directions for in-network reforwarding are illustrated by the dashed blue and green arrows.

To characterize MLDR behavior under a generic mobility management protocol, we implemented an ideal global routing scheme that immediately updates the FIBs of each node as soon as producer mobility is detected (i.e. producer is associated to a new AP). Under real mobility management protocols, the time for updating network forwarding state would be longer, hence higher the gains due to MLDR w.r.t. the case under study.

We start from producer mobility: the producer moves between the APs at different speeds (from 3km/h to 50km/h). We set an Interest lifetime of 500ms, which is an order of magnitude bigger than the average round trip time, here essentially determined by the wireless hop. In all scenarios, the consumer requests 300k packets.

We compare three approaches: (i) the baseline with loss detection and recovery performed at consumer side based on timer expiration (ii) the case with in-network loss detection, Explicit Loss Notification (ELN) to the consumer (iii) MLDR, where either detection and recovery are performed in-network. The ELN scheme exploits $M-$flagged Interests of MLDR, in this case sent directly to the consumer with no interception by in-network routers.

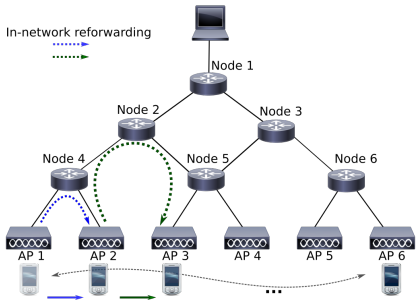


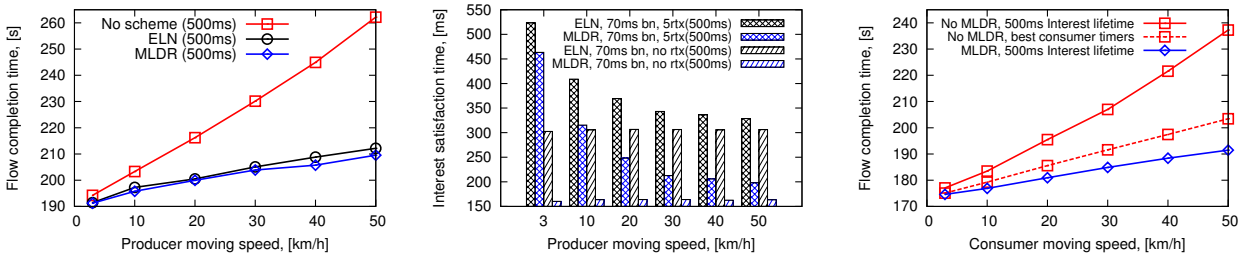
Figure 6: MLDR test topology.

Fig. 7(a) reports flow completion time (or download time) as a function of producer moving speed. MLDR gains in terms of loss detection are striking and higher the speed, higher this gain. Indeed, higher the speed, higher the number of performed handovers corresponding to the number of times MLDR is in action. Overall, in this scenario, the number of re-forwarded Interests grows with the speed value to a maximum value of 0.25% of total traffic at 50km/h.

If the benefits of in-network loss detection are significant, the additional gain of MLDR over ELN due to in-network retransmission is much smaller. This can be easily explained as the additional latency introduced to notify the consumer and let him retransmit the Interest packets is negligible w.r.t. the overall round trip time, mainly affected by wireless hop delay.

To better understand where the difference between these two schemes plays a role, we increase the propagation delay of the links between *Node* 1, *Node* 2, *Node* 3 to 70ms and compute for ELN and MLDR solutions the average "*Interest satisfaction time (IST)*", i.e. the interval of time between the first transmission of an Interest packet and the reception of the corresponding Data packet.

IST is measured at the consumer and takes into account all performed retransmissions. The average values of IST for the retransmitted packets are presented in Figure 7(b). We show the results for the cases with (indicated with *5rtx*) and without (labeled *no rtx*) additional retransmissions by timer performed by the consumer. In case of retransmissions the consumer can issue the same Interest up to 5 times in case of timer expiration, as in all the other simulations. In the no retransmission setting, the consumer retransmits an Interest only on reception of a loss notification.



(a) (b) (c)

Figure 7: (a) Flow completion time as a function of producer moving speed; (b) Interest satisfaction time as a function of producer moving speed; (c) Flow completion time as a function of consumer moving speed

In absence of further retransmissions, we observe that MLDR and ELN show a constant retransmission time as a function of moving speed, with MLDR considerably outperforming ELN by means of much smaller IST (almost half of ELN's IST). The performance gap remains in the case where additional consumer retransmissions are allowed, also increasing as moving speed grows. MLDR's better performance here is a consequence of the increasing amount of re-forwarded Interests that have the effect of reducing the number of timeouts/required retransmission for MLDR.

From the presented results we can conclude that, in terms of IST, MLDR always outperforms ELN in virtue of the additional delay that the $M-$flagged Interests experience to reach the consumer and to be retnasmitted there, rather then being intercepted and immediately re-forwarded by the routers of the path, as it is the case for MLDR.

We now move to the consumer mobility case. We consider the same network topology in Fig. 6, with the producer placed at the root. The consumer moves between the APs at different speeds (from 3kmh to 50km/h).

The results of this scenario are presented in Fig. 7(c). Here, retransmissions can only be performed at the consumer, hence we compare the performance of MLDR against consumer-driven retransmission using the best timer values, i.e. the values associated to each moving speed that give us the best performance as obtained by a set of simulations with different timer values, as we did for WLDR in Fig. 5(a). Here we tested timers from 50ms to 500ms.

In Fig. 7(c), we observe again the significant improvement in terms of flow completion time when MLDR is activated. This is due to both earlier detection and recovery. Indeed, MLDR performs better than consumer retransmissions under either 500ms timer value, either the best consumer timers, in virtue of its quick reaction to mobility events.

5.4 Joint WLDR-MLDR Evaluation

In this section we analyze the performance of the two proposed algorithms combined, in a more realistic scenario. We use the topology in Fig. 8, where APs are positioned in a grid of 6 by 6 nodes at a distance of 80m each other. Each edge router in the lower layer of the fat tree is connected to 3 APs. We run simulations with 10 mobile nodes (5 acting as consumers, 5 as producers). The mobility is simulated using a random waypoint model. Micro-mobility is assumed like in current radio mobile networks as LTE. The mobile nodes move in the area indicated by gray square in Fig. 8. We put extra APs outside the moving area to guarantee homogeneous radio coverage: each STA can sens 9 APs from each point of the simulation area. A consumer retrieves a file composed by 100k chunks from a single producer. We run each simulation 200 times, with the nodes starting at different positions.

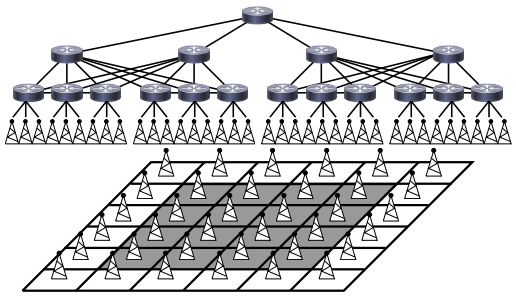
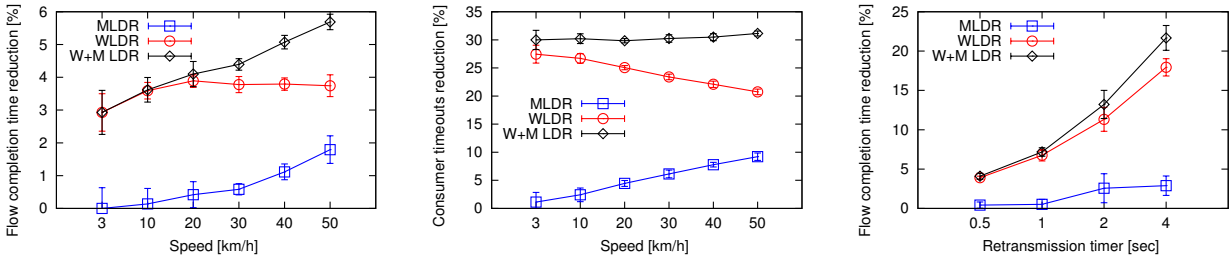


Figure 8: Edge network topology.

Fig. 9(a) shows the average and the standard deviation of the flow completion time reduction (in percentage) that we obtain running our proposals with respect to the results obtained recovering losses at the consumer on timer expiration. In this simulations we set the retransmission timer (or Interest lifetime), and so the PIT timer, equal to 500ms. In the figure *MLDR* and *WLDR* indicate the gain that we achieve using only one of the two proposed algorithms, while *W+M LDR* denotes the gain when both WLDR and MLDR are used.

It is easy to see that MLDR benefits increase when the speed of the mobile nodes is higher. This is due to the larger number of handovers at high speed during the simulations, hence of opportunities for MLDR to fast detect and recover packets that would have been lost otherwise. E.g., when the mobiles nodes moves at 3km/h on average 17 handovers per flow occur (considering the sum of the handovers performed by the consumer and the producer), while at 50km/h the average number of handovers is more than 200. WLDR, instead, is more effective at low speeds, when the majority of the losses are due to the wireless channel. Finally, we can see that combining the two algorithms we addition the gains brought by each one separately. In this setting we achieve a maximum gain of almost 6% when the mobile nodes move at 50km/h.

Fig. 9(b) reports the average and the standard deviation of the reduction of the timeouts registered at the consumer when we enable our mechanisms. The simulation setting is the same as the one in Fig. 9(a). This figure confirms our conclusions: WLDR is more effective a lower speeds, when most of the losses are due to the wireless channel, while MLDR becomes more and more effective when increase the number of mobility events. Enabling both WLDR and MLDR we are able to reduce the number of timeouts at the



(a) (b) (c)

Figure 9: (a) Flow duration for different speeds, (b) Timeouts reduction at the consumer, (c) Flow duration for different retransmission timeouts.

receiver by more than 30%.

In Fig. 9(c) we show again the flow completion time reduction, but as a function of the retransmission timer. In this set of simulations we set the node speed to 20km/h. Increasing the retransmission timer from 500ms to 4sec the flow completion time can be reduced by more than 20% combining WLDR and MLDR. This is due to the fact that using in-network retransmissions we remove the dependency from network/application timers that affects the network performances and are really difficult to tune correctly.

6. CONCLUSIONS

ICN with hop-by-hop forwarding transport model offers an opportunity to rethink congestion control over wireless mobile networks beyond the limitations of traditional connection-based approaches, by leveraging in-network control capabilities. Quite some attention has been devoted to congestion control design in the ICN community, but very little to the case of wireless mobile environments, where the distinction of the nature of loss events and the capability to achieve prompt recovery are key factors for an effective rate and congestion control.

In this paper, we analyze the potential for improvement of in-network loss detection and recovery and propose two solutions, WLDR and MLDR, respectively tackling wireless channel losses and losses due to mobility events. WLDR-MLDR follow by the same design principles: they consist in a link-layer agnostic purely distributed approach decoupling in space and in time loss detection and recovery operations by exploiting Explicit Loss Notification messages. Fast recovery at sub-round trip time scale is achieved in the network once the information about loss detection has reached the first potential retransmission point.

The performance evaluation carried out by means of simulations shows significant benefits in terms of reduction of flow completion time or per-packet request satisfaction time over consumer-based solutions. The other advantage over state of the art solutions is that WLDR/MLDR remove the dependency from network/application timers with all setting-related issues.

We plan as future work to carry out a larger scale experimentation in realistic indoor/outdoor Wi-Fi enviroments as well as to extend the analysis to other wireless access technologies in 5G context.

7. REFERENCES

[1] M. Amadeo, A. Molinaro, C. Campolo, M. Sifalakis, and C. F. Tschudin. Transport layer design for named data wireless networking. In *Prof. of IEEE INFOCOM NOM*, 2014.

[2] A. Bakre and B. Badrinath. I-TCP: indirect TCP for mobile hosts. In *Proc. of IEEE ICDCS*, 1995.

[3] H. Balakrishnan and R. H. Katz. Explicit loss notification and wireless web performance. In *Proc. of IEEE GLOBECOM Internet Mini-Conference*, 1998.

[4] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proc. of ACM MOBICOM*, 1995.

[5] S. Biaz and N. Vaidya. Distinguishing congestion losses from wireless transmission losses: a negative result. In *Computer Communications and Networks, 1998.*, Oct 1998.

[6] S. Biaz and N. Vaidya. Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. In *Proc. of IEEE ASSET'99*, 1999.

[7] K. Brown and S. Singh. A network architecture for mobile computing. In *Proc. of IEEE INFOCOM*, 1996.

[8] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and S. Wang. Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks. In *Proc. of IEEE ICNP*, 2013.

[9] S. Cen, P. Cosman, and G. Voelker. End-to-end differentiation of congestion and wireless losses. *Networking, IEEE/ACM Transactions on*, 11(5), Oct 2003.

[10] L. Han, S.-S. Kang, H. Kim, and H. In. Adaptive Retransmission Scheme for Video Streaming over Content-Centric Wireless Networks. *Communications Letters, IEEE*, 17(6), June 2013.

[11] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split TCP for mobile ad hoc networks. In *Proc. of IEEE GLOBECOM*, 2002.

[12] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In *Proc. of ACM MOBICOM*, 2001.

[13] A. Mcgregor and D. Smithies. Rate adaptation for 802.11 wireless networks: Minstrel. In *http://blog.cerowrt.org/papers/minstrel-sigcomm-final.pdf*.

[14] V. Mhatre and K. Papagiannaki. Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks. In *Proc. of ACM MobiSys*, 2006.

[15] C. Parsa and J. J. Garcia-Luna-Aceves. Improving TCP Performance over Wireless Networks at the Link Layer. *Mob. Netw. Appl.*, 5(1), Mar. 2000.

[16] E. Perahia and R. Stacey. *Next Generation Wireless LANs 802.11n and 802.11ac*. Cambridge University Press, 2 edition, 2013.

[17] K. Ratnam and I. Matta. WTCP: an efficient mechanism for improving TCP performance over wireless links. In *Proc of IEEE ISCC*, 1998.

[18] K. Shin, J. Kim, and S. B. Choi. Loss Recovery Scheme for TCP Using MAC MIB over Wireless Access Networks. *Communications Letters, IEEE*, 15(10), 2011.

[19] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda. Achieving moderate fairness for UDP flows by path-status classification. In *Proc. of IEEE LCN*, 2000.