Adaptive Interest Lifetime for Content-Centric Requests

Bachelorarbeit
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Lukas von Rotz
2015

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Abstract

Content Centric Networking (CCN) is a new networking approach based on the following ideas:

- Content is accessed by name, not by host identifier
- Each host has a cache to temporarily store received data
- Content is secured and not the connection as in current networking protocols

CCN moves away from the host based communication model towards a content-based approach. Interests get transmitted with the intention of retrieving content quickly from a nearby network node. Because content is identified by name, it can be retrieved from any node that has it and does not have to come from a specific host. Data is composed of multiple segments, where each segment is represented by a unique name. To retrieve content, an Interest has to be transmitted for each of these segments.

Wireless communication is broadcast in nature: every transmitted message can theoretically be received by any node in range. Connections to specific hosts in traditional host-based communication may break in case of mobility and so is the transmitted content. However, with CCN, the requester can retransmit the Interest to quickly find the content at another node, which may have cached the content from the previous transmission, or from an alternative content source.

Every host maintains a Forwarding Information Base (FIB), which contains information where to forward Interests similar to IP tables in host-based communication. Every transmitted Interest is included in the Pending Interest Table (PIT). The PIT table prevents the forwarding of the same Interests (duplicate Interests) and enables data to find its way back to the requester by consuming the information in the PIT. At maximum if no data is received, an Interest stays in the PIT for the duration of the Interest lifetime. If the value times out (Interest timeout), it can be retransmitted. In CCNx, the open-source reference implementation of the CCN concept, the Interest lifetime is set by default to 4 seconds. However in practice, the Interest lifetime cannot be statically set to a fixed value because it depends on the distance to the nearest content source.

Imagine a wireless network where several nodes communicate simultaneously and a lot of data packages are exchanged. In such a scenario, collisions, which can be the reason for an Interest reexpression, are unavoidable. The faster collisions are detected (Interest timeout), the faster a reexpression can be performed resulting in a higher throughput. However, if the Interest lifetime is too low, unnecessary Interest retransmissions would be performed because data could not travel back over multiple hops to the requester. Unnecessary retransmissions increase the traffic on the wireless medium and therefore, also the collision probability.

The goal of this work was the development and evaluation of adaptive algorithms to dynamically set the Interest lifetime. The algorithms have been implemented in CCNx 0.8.2 by extending the existing ccncat application. The ccncat application requests content via a stream, i.e., multiple Interests are transmitted concurrently to subsequently request all segments. For each incoming segment, the round trip time (RTT) is measured, which is used as a basis to adaptively set the Interest lifetime. The round trip time is the time between sending an Interest and receiving the corresponding segment. If the round trip time is larger than the specified Interest lifetime, a timeout is triggered. After a timeout, the Interest lifetime is increased and the Interest is retransmitted. The developed algorithms have been evaluated and compared to existing algorithms in wireless multi-hop scenarios with one or more concurrent requesters.

In our evaluations, we vary the number of hops and concurrent ccncat streams. A stream corresponds to a requester, retrieving a file of 5MB in size. Besides the transfer time, also the number of transmitted Interest messages have been evaluated. The evaluation results are obtained by emulations with NS3 - Direct Code Execution. The network topology, the timing of the simulation and the execution of the CCN commands are written in C++ scripts. All the evaluations were performed on Ubelix, a Linux cluster system of the University of Bern.

Evaluations show that the developed algorithms result in throughput twice as large as existing algorithms while transmitting up to 2.1% fewer Interest messages. Compared to the default Interest lifetime of 4 seconds, the algorithms are up to 4.2 times faster and transmit up to 1.69% fewer Interests, depending on the evaluation scenario.

Adaptive Interest Lifetimes for Information-Centric Wireless Multi-hop Communication

Carlos Anastasiades, Lukas von Rotz, and Torsten Braun
Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland
Email: {anastasiades, vonrotz, braun}@iam.unibe.ch

Abstract— **Information-centric networking (ICN) is a new communication paradigm that is very promising for wireless networks. Users can exploit the broadcast nature of the wireless medium to quickly find desired content at nearby nodes. Wireless multi-hop communication is prone to collisions and it is crucial to quickly detect and react to them to optimize transmission times and avoid spurious retransmissions. Several adaptive retransmission timers have been used in related ICN literature but they have not been compared and evaluated in wireless multi-hop environments. In this work, we evaluate all existing algorithms in wireless multi-hop communication. We find that existing algorithms are not optimized for wireless communication but slight modifications can result in considerably better performance without increasing the number of transmitted Interests.**

Index Terms—**Information-centric networks, wireless, multi-hop, dynamic Interest lifetime**

I. Introduction

Information-centric networking (ICN) is a new communication paradigm to address the shortcomings of today's communication, namely scalability and security. Every content object has a unique name and can, therefore, be routed based on demand and availability. By that, every node can identify what content is exchanged (although the content may be encrypted for security reasons), which facilitates caching and eases content distribution in case of high demands. This is in contrast to current IP based communication, where only the endpoints are known.

Most ICN works have focused on wired communication to replace the current Internet architecture, but ICN is also beneficial for wireless communication, where connectivity to certain nodes may be intermittent due to changing channel conditions and mobility. Information-centric communication can exploit the broadcast capability of the wireless medium to quickly find a suitable content source. Instead of maintaining connectivity to a specific host, content can be retrieved from any neighbor node that holds the desired content.

Wireless communication is error-prone but collision probability in multi-hop communication is even higher because received and forwarded packets can collide as well. If collisions occur near destinations, retransmissions need to be performed over the entire path to the content source. In case of frequent collisions, retransmissions can consume most of the bandwidth, which may even result in a complete communication collapse. ICN can mitigate this problem because content is automatically cached in intermediate nodes, such that retransmissions are only performed over the hop where the collision occurred.

Many ICN architectures have been proposed [1], [2], [3], [4] but in this work, we focus on the Named Data Networking (NDN) architecture [4]. In this architecture, requesters transmit Interest messages that establish a soft state in forwarding nodes such that Data messages can travel the reverse path back to the requester. Every Interest has a lifetime that determines how long the soft state is kept valid at maximum, i.e., if no Data message is received in return. Since Interests are only forwarded upstream if the same Interest is not already forwarded (no soft state), the Interest lifetime has a direct impact when Interests are retransmitted at requesters. In earlier work [5] we have shown that the Interest lifetime has a significant impact on the achievable throughput in wireless communication. Because collisions cannot be detected in contrast to wired communication, short Interest lifetimes enable faster recovery from collisions.

Several adaptive Interest lifetime algorithms have been used in ICN literature [6], [7], [8], [9]. However, these works focus on throughput in wired networks and do not evaluate the number of unnecessary retransmissions due to too short Interest lifetimes. Wireless communication experiences higher variability in RTTs than wired communication due to varying channel conditions and MAC layer buffering. If a timeout is triggered too early, spurious retransmissions may be performed. Related work in TCP literature [10] suggests that spurious timeouts can be avoided by introducing additional random delays. However, it is difficult to decide how large these delays should be. Due to caching in information-centric networks, the calculation of retransmission timers becomes even more difficult.

In this work, we evaluate two slightly modified algorithms for wireless multi-hop communication, i.e., one that considers RTT variability and one that does not, and compare it to available algorithms from the literature. We design three traffic scenarios for wireless multi-hop communication and evaluate all algorithms considering both throughput and transmitted messages. All algorithms have been implemented in CCNx 0.8.2 [11] and evaluated by emulations in wireless multi-hop communication with NS3-DCE [12].

The remainder of this paper is structured as follows. In Section II, we describe the NDN architecture and related work on adaptive retransmission timers. In Section III, we describe the modified algorithms for adaptive Interest lifetime. Evaluation results are shown in Section IV and we conclude our work in Section V.

II. Related Work

A. Basic NDN Concepts

NDN communication [4] is based on two messages: *Interests* to request content and *Data* to deliver content. Typically in NDN, files consist of multiple segments that are included in Data messages. Then, in order to retrieve a file, users need to express Interests in every segment of the file. The CCNx [11] project provides an open source reference implementation of NDN. The core element of the implementation is the CCN daemon (CCND), which performs message processing and forwarding decisions. Links from the CCND to applications or other hosts are called *faces*. A CCND has the following three memory components:

1) The Forwarding Information Base (FIB) contains forwarding entries to direct Interests over specific faces towards content sources. An entry contains a prefix, the face where to forward Interests, e.g., IP address and UDP port if NDN is run as overlay over IP, and a lifetime value. To avoid loops, an Interest is never forwarded on the same face from where it was received.
2) The Pending Interest Table (PIT) stores unsatisfied forwarded Interests together with the face on which they were received. If Data is received in return, it can be forwarded based on face information in the PIT. Existing PIT entries prevent forwarding of duplicate Interests.
3) The Content Store (CS) is used as cache in a NDN router storing received Data packets temporarily.

The *Interest Lifetime* in the Interest header determines how long an Interest stays in the PIT and, therefore, how quickly the Interest can be retransmitted.

B. Adaptive Interest lifetime

The default lifetime for Interests in CCNx is 4 seconds, but there are existing works that apply adaptive Interest lifetimes for congestion control in ICN communication. Early works [6] use an Interest lifetime based on TCP's retransmission timer [13]. TCP's retransmission timer uses an exponential moving average ($sRTT$) of current and past round trip times (RTTs) and the variance of RTT values (rttVAR) to compute the retransmission timeout as follows.

$$\begin{aligned} sRTT &= 0.8 \times sRTT + 0.2 \times RTT \\ \text{rttVAR} &= 0.75 \times \text{rttVAR} + 0.25 \times abs(sRTT - RTT) \\ RTO &= sRTT + 4 \times \text{rttVAR} \end{aligned}$$

The RTO has a lower bound of 1s and in case of a timeout, it is doubled up to a maximum value. TCP's retransmission timer has also been directly applied in simulations to wireless VANET networks [14] but without evaluating timeouts and number of retransmissions.

A slightly modified retransmission timer similar to TCP [15] but with a smaller gain for the RTT variance, i.e., 0.125 instead of 0.25, has been proposed [8], [16]. There, the retransmission timer is additionally multiplied with a factor 2 to ensure that the Interest lifetime is larger than RTT values.

$$\begin{aligned} err &= RTT - sRTT \\ sRTT &= sRTT + 0.125 \times err \\ \text{rttVAR} &= \text{rttVAR} + 0.125 \times (abs(err) - \text{rttVAR}) \\ RTO &= sRTT + 4 \times rttVAR \\ requestTO &= 2 \times RTO \end{aligned}$$

ICP [7] considers the history of RTT measurements. The retransmission timer is based on the minimum value and the midrange of minimum and maximum samples from the last 20 samples as follows.

$$RTO = RTT_{min} + 0.5 \times (RTT_{max} - RTT_{min})$$

Weighted averages without variance considerations have also been applied in CHoPCoP [9]. In this work, the variance of RTT samples is not considered but only $sRTT$ is multiplied with a large constant of value "6" to ensure that Interest lifetimes are not a limiting factor. Routers can then signal congestion back to requesters if they are overloaded via explicit congestion signalling.

Recent works argue that RTT calculations need to consider the IDs of content sources that served the content due to caching [17] and multi-homing [18], [8], which may result in varying path delays. In this work, we assume that wireless requesters first transmit broadcast requests to identify a content source, potentially multiple hops away, and then dynamically create a unicast path to it [19]. By that, all segments are requested from the same content source until the path breaks and a new content source is searched via broadcast. If the path breaks resulting in multiple subsequent Interest timeouts, the Interest lifetime is reset to the default value of 4 seconds. However, multi-homing solutions could of course also be integrated into wireless multi-hop networks by introducing source identities as proposed for wired multi-homing [17], [18], [8].

III. Adaptive Interest Lifetimes

In this section, we describe two algorithms to adaptively set the Interest lifetimes, namely CCNTimer and WMA, which are based on TCP's retransmission timer [13] and CHoPCoP's retransmission timer [9].

CCNTimer is described in Algorithm 1. It uses the same exponential moving average for $sRTT$ and rttVAR as TCP [13]. We measure the RTT as the time between the transmission of the first Interest in a segment and the actual reception of the corresponding segment. In case of timeouts, Interests may be retransmitted but the RTT start time of the corresponding segment is not changed because the segment has possibly been successfully transmitted over some hops and been cached in

Algorithm 1 CCNTimer

```
initial values: IL = 4s, recv = 0, initial = true

function RECEIVE_CONTENT(RTT, Tout)
    recv++
    if Tout == 0 then
        if recv ≥ Γ then
            if initial == true then
                sRTT = RTT
                rttVAR = RTT / 2
                initial = false
            else
                sRTT = 0.8 × sRTT + 0.2 × RTT
                rttVAR = 0.75 × rttVAR + 0.25 × abs(sRTT − RTT)
            RTO_TCP = sRTT + 4 × rttVAR
            IL = max(4s, γ × RTO_TCP)

function TIMEOUT_OCCURRED(Tout)
    if Tout > Φ then
        IL = 4s
        initial = true
    else
        IL = max(4s, (1 + ϵ) * IL)
```

Algorithm 2 Weighted Moving Average (WMA)

```
initial values: IL = 4s, recv = 0, initial = true

function RECEIVE_CONTENT(RTT, Tout)
    recv++
    if Tout == 0 then
        if recv ≥ Γ then
            if initial == true then
                sRTT = RTT
                rttVAR = RTT / 2
                initial = false
            else
                sRTT = 0.8 × sRTT + 0.2 × RTT
            IL = max(4s, γ × sRTT)

function TIMEOUT_OCCURRED(Tout)
    if Tout > Φ then
        IL = 4s
        initial = true
    else
        IL = max(4s, (1 + ϵ) * IL)
```

intermediate nodes. This is slightly different to TCP, where the RTT is reset after retransmissions since there is no caching and all segments need to be transmitted over the entire path again. $Tout$ defines the number of timeouts for each segment. In this algorithm, we only process RTT measurements if no timeouts occurred, i.e., no retransmissions of the same Interest. If a received segment experienced a timeout, the $sRTT$ value remains unchanged because the RTT sample is not considered. Therefore, the old Interest lifetime IL is used for the next transmission.

In addition to [13], the moving average $sRTT$ is weighted by the factor γ to obtain the Interest lifetime IL. This ensures that the Interest lifetime is slightly larger than the usual RTT, which avoids spurious retransmissions. While too large Interest lifetimes have only an influence in case of collisions (retransmissions), too short Interest lifetimes may already trigger retransmissions if Data messages are returned with a delay. Therefore, too short lifetimes may cause unnecessary traffic and in the worst case, result in a collision with the delayed Data packet.

In contrast to [13] we do not drastically increase the Interest lifetime in case of collisions (timeouts) by doubling the Interest lifetime because collisions may happen regularly in wireless networks. Higher Interest lifetimes, e.g., after burst errors, cannot avoid future collisions but only delay the detection and reaction to it. Therefore, if a retransmission is required for a segment, the Interest lifetime is only slightly increased by the factor ϵ to account for delay variability. If Φ retransmissions are required for the same segment, we reset the Interest lifetime to 4s, which we also set as maximum Interest lifetime in the algorithm (default CCNx Interest lifetime). For stability reasons and to avoid high variability in the beginning of a file transfer, we ignore the first Γ messages before processing the samples. In our evaluations, we use the following default numerical values, Γ = 4, γ = 1.2, ϵ = 0.2 and Φ = 3.

WMA (Algorithm 2) is simpler because it does not consider the variance of the samples. Similar to [9], the Interest lifetime is only based the exponential moving average of RTT values, which is weighted by a factor γ. Since the specified γ is 6 [9], which would result in very large Interest lifetimes close or equal to 4 seconds (maximum value), we use a smaller value of 1.5 to ensure that the Interest lifetime stays close to the RTT values, i.e. only 50% higher. In case of timeouts, we only slightly increase the Interest lifetime similar to CCNTimer in Algorithm 1.

IV. Evaluation

We implemented CCNTimer and WMA as well as other well-known retransmission timers from literature (specified in the next subsection) in CCNx 0.8.2 [11] by extending the *ccncat* application. We also compared it to a constant Interest lifetime of default 4 seconds. In this section, we show evaluation results in wireless multi-hop communication, which are obtained by emulations with NS3-DCE [12] using the 802.11g MAC layer and the free-space path loss model. In all evaluations, we use UDP as underlying transport protocol to avoid any conflicts with TCP's congestion avoidance mechanisms. All evaluations have been performed 100 times.

A. Implementation Details

We compare CCNTimer and WMA with retransmission timers, which we listed in Section II, namely, TCP's retransmission timeout (RTO) [13], an adapted RTO variant [8], and ICP [7].

TCP's RTO uses an exponential moving average with a lower bound of 1 second. In case of timeouts, the Interest lifetime is doubled. In our evaluations, we set TCP's RTO as Interest lifetime.

We denote the adapted RTO [8] as Timeest. As shown in Section II, it is similar to TCP but uses a lower weight for the variance. The RTO is multiplied by 2 to ensure that Interest lifetimes are larger than RTT values. Since the behavior in case of timeouts is not specified [8], we assume the same behavior as for TCP, i.e. doubling the RTO.

CCNTimer is similar to Timeest, but we use the same weight for the variance as TCP and multiply the RTO only with 1.2 to keep the Interest lifetimes closer to the RTT values. In case of a timeout, we increase the Interest lifetime by only 20% to account for higher RTT variability. In case of 3 consecutive timeouts of the same segment, the Interest lifetime is reset to the default value of 4 seconds.

WMA is simpler because it only considers the exponential moving average and not the variance of RTT values similar to CHoPCoP's retransmission timer [9]. However, we only multiply the exponential moving average with a factor of 1.5 instead of 6 [9], because otherwise, the Interest lifetime would be close or equal to 4 seconds most of the time. The weighting factor is larger than with CCNTimer because WMA does not consider the variance of the samples. By this, we can keep the Interest lifetime close to the RTT values and evaluate the influence of variance values.

The RTT is the time between the transmission of the first Interest and the reception of the corresponding segment. Thus, in case of timeouts, the RTT may include times for retransmissions of Interests. However, for TCP's RTO, Timeest, CCNTimer and WMA, we only process RTT values of segments that have not experienced timeouts to avoid large RTT values that distort the exponential moving average $sRTT$ excessively.

ICP's retransmission timer [7] is completely different than the other algorithms. As described in Section II, the Interest lifetime is based on the history of the last 20 RTT values. For ICP we consider all measured RTT values, even for segments that experienced timeouts and required Interest retransmissions. We have also implemented an ICP variant that ignores timeouts but the performance was significantly worse. If timeouts are ignored, the algorithm can not adapt the Interest lifetime during timeouts until the next segment is received in time. However, by considering RTTs with timeouts, higher RTT values are possible, which automatically increase the next Interest lifetime.

For all algorithms, we set the maximum Interest lifetime to 4 seconds because higher values resulted in worse performance. In addition, for all algorithms (except TCP's RTO), we set the Interest lifetime's lower bound to 100ms, which ensures that transmissions to other hosts are possible.

B. Scenarios

We evaluate all algorithms in wireless multi-hop communication with 1 to 5 hops. The network nodes are placed in a row so that only immediate neighbors see each other and every node has a unicast face configured to its neighbors.

We evaluate the algorithms in three scenarios. Scenario 1 consists of two active nodes: one requester that requests a file and a content source, which is 1 to 5 hops away. Figure 1a shows the topology for scenario 1 with 3 hops. The requester (R) is the leftmost node and the content source (CS) the rightmost node. In scenario 2, there is one requester at every node requesting one distinct file. Figure 1b shows the topology for 3 hops. Scenario 3 is similar to scenario 2, but every requester additionally receives requests from two requesters above and below. Figure 1c shows the topology in scenario 3 with 3 hops on the horizontal.

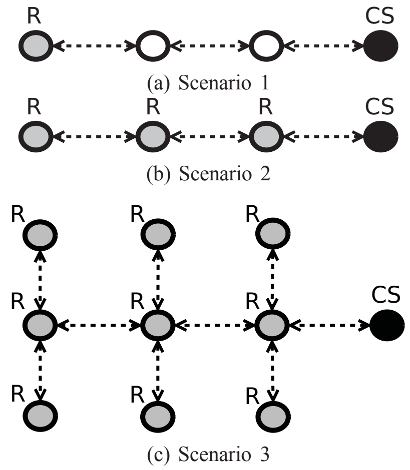


Fig. 1. Evaluated Scenarios between 1 to 5 hops on the horizontal, here shown for 3 hops. The content source is on the rightmost node.

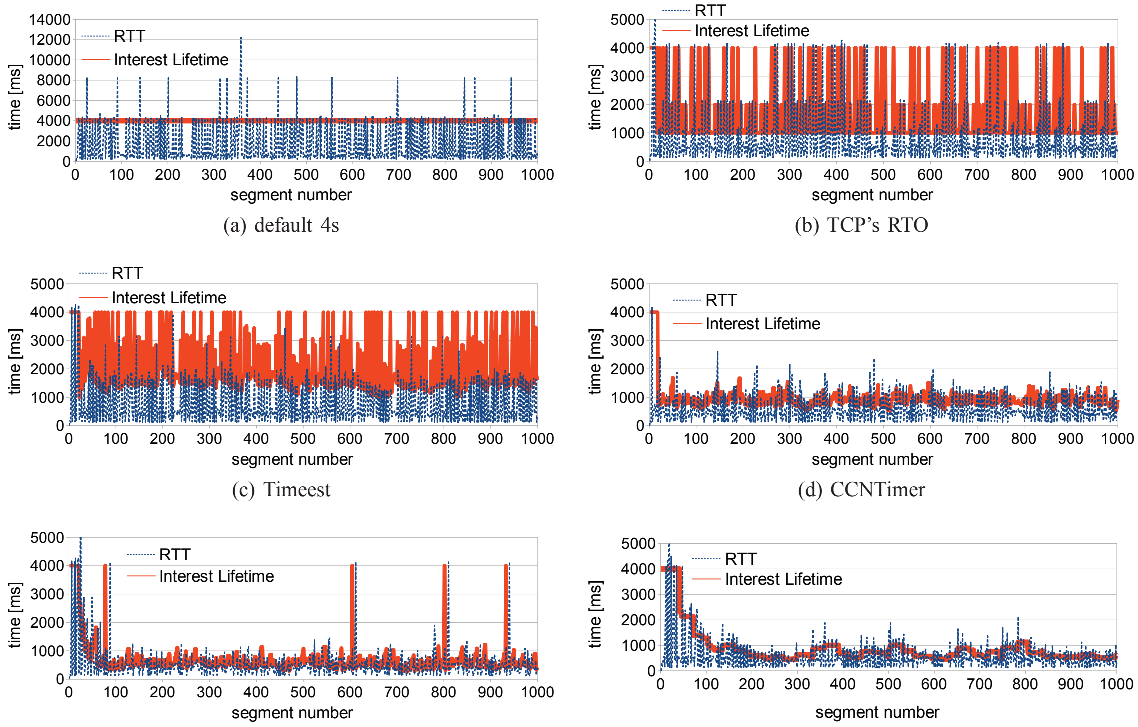
C. One Requester - Varying Hop Distance

In this section, we evalute the algorithms in scenario 1, between one requester and one content source.

To better understand the algorithms, Figure 2 shows the measured RTT (blue dashed lines) and the Interest lifetime values (red solid line) of a sample run for all evaluated algorithms over 3 hops. Since the RTT is measured between the first Interest transmission and the reception of the corresponding segment, i.e., incorporating Interest retransmissions, the RTT values are slightly different for all algorithms. If collisions are detected quicker (due to a shorter Interest lifetime), retransmissions can be performed faster resulting in a lower RTT.

Figure 2a shows the results if the default CCNx strategy, i.e., a constant Interest lifetime of 4 seconds, is used. If the RTT equals 4s a collision occured, because it is only detected after the Interest times out. If RTT values are above 4 seconds, multiple timeouts of the same segment have occured. This shows that even with large Interest lifetimes, collisions can not be avoided, but due to a longer detection time, the RTT values become larger.

Figure 2b shows the values if TCP's RTO is used as Interest lifetime. During these measurements TCP's RTO is only slightly below 1 second, which means that it is set to the lower bound. However, due to high variability, some RTT values may be slightly higher than 1 second and, therefore, cause a timeout. These timeouts are visible in Figure 2b as RTT values slightly above 1 second, which indicates that the retransmission was successful from the cache. However, since the Interest lifetime is doubled in case of timeouts, these unnecessary timeouts cause the Interest lifetime to be increased to 2 seconds. As a result, collision detection takes more time and RTT values increase.



(a) default 4s

(b) TCP's RTO

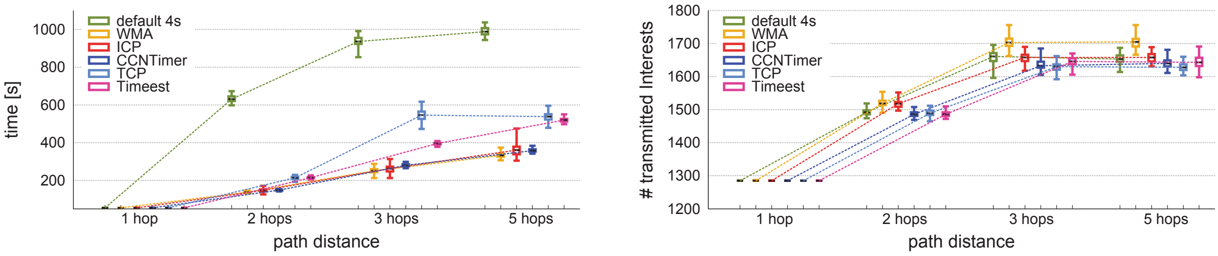
(c) Timeest

(d) CCNTimer

(e) WMA

(f) ICP

Fig. 2. RTT and Interest lifetime samples from 1/100 run over 3 hops for the first 1000 segments.



(a) Retrieval Time

(b) Transmitted Interests

Fig. 3. Retrieval Time and Transmitted Interests for requester in scenario 1 for varying hop distance.

Figure 2c shows the Timeest algorithm, which is similar to TCP, but we set the lower bound to a much lower value, i.e., 100ms, which was actually never reached in any measurements. Since the RTO is multiplied by 2, the resulting Interest lifetime is still larger than most RTT values, avoiding too early and unnecessary transmisions. The Interest lifetime can adopt more diverse values because it is not always set to the lower bound as TCP's RTO. More diverse Interest values enable Timeest to adjust the Interest lifetime better to current round-trip times, resulting in slightly lower RTT values. For example, RTT values of 4 seconds are never reached compared to TCP's RTO in Figure 2b. We can therefore conclude, that large values for lower bounds, such as 1 second for TCP are not appropriate for wireless multi-hop communication. However, because the RTO is doubled in case of collisions, the Interest lifetime can quickly reach high values, which causes higher RTT values in case of collisions.

In Figure 2d, we show the Interest lifetimes of CCNTimer. In contrast to Timeest, the RTO is multiplied only with a small factor of 1.2 to account for higher RTT variability and ensure that the Interest lifetime is only slightly larger than most RTT values. In case of timeouts, the Interest lifetime is only slightly increased by 20% for every timeout and not doubled as for TCP and Timeest. Timeouts are mostly caused by higher RTT variability or collisions. In both cases, the content might be found in caches at intermediate nodes. Therefore, the algorithm can react quicker to timeouts and does not overcompensate by increasing the Interest lifetime extensively in case of collisions. This enables CCNTimer to reduce timeout periods, which leads to lower RTT values as Figure 2d shows. There are a few segments that time out and cause RTTs of slightly more than 2 seconds, e.g., segment 149, 232 and 302, but the Interest lifetime does not drastically react to it and only increases slightly. Figure 2d confirms that the next few segments can be received without timeout.

Figure 2e shows the Interest lifetimes and RTT values for WMA. We can see that the RTT values are on a slightly lower level than with CCNTimer because Interest lifetimes are shorter and, therefore, timeout periods can be reduced. However, there are also four peaks of Interest lifetimes and RTT values. Because the Interest lifetime is slightly too short, timeouts happen more frequently and for segments 80, 606, 802 and 942 in Figure 2e, the Interest is reset to 4 seconds because of 3 consecutive timeouts of the same segment. If segments, which are transmitted with an Interest lifetime of 4 seconds after a reset, time out, the RTT values may become considerably larger since it takes more time to detect the timeout. We can see such RTT peaks at segments 90, 614, 812 and 942, i.e., after every lifetime reset, which indicates that WMA results in many timeouts.

Finally, Figure 2f shows the Interest lifetime when using the ICP algorithm. The Interest lifetimes does not change as quickly as for the other algorithms because the Interest lifetime is only based on minimum and maximum RTTs, which do not change that quickly for 20 samples. The Interest lifetime changes in cycles increasing from minimum to maximum values and decreasing again. For small minimum RTT values, the Interest lifetime becomes slightly too small resulting in timeouts. Timeouts require retransmissions, which means that RTT values increase again resulting in larger Interest lifetimes. Larger lifetimes result again in fewer timeouts and lower RTT values (fewer retransmissions) completing the cycle.

To measure the performance of these algorithms, we evaluate *retrieval times*, i.e., the time to request and receive content from a content source multiple hops away, and transmitted messages. Figure 3a shows the retrieval time (y-axis) of a 5MB file over multiple hops (indicated on the x-axis). For two hops or more, Interests need to be forwarded by intermediate nodes and may collide with other Interests or Data messages. As a result, adaptive Interest lifetimes result in shorter retrieval times. Compared to the default Interest lifetime of 4 seconds, CCNTimer results in 4.2 times faster transmissions over 2 hops, 3.3 times faster transmissions over 3 hops and 2.76 times faster transmissions over 5 hops. With increasing hop distance, the performance of adaptive algorithms decreases slightly compared to the default 4 seconds due to longer retrieval times and more collisions. In case of 3 hops, CCNTimer is 1.95 times faster than TCP's RTO and 1.45 times faster than Timeest, but requires 6% more time than ICP and 10.3% more time than WMA. This is because ICP and WMA use shorter Interest lifetimes than CCNTimer. To understand the implications of shorter Interest lifetimes, we evaluate transmitted messages and timeouts.

Figure 3b shows the number of transmitted Interests in this scenario. For 3 hops, the requester sends approximately the same number of Interests with CCNTimer, Timeest, TCP and default 4 seconds. With ICP, requesters send 1% more Interests than with CCNTimer and with WMA 4% more Interests. Although these values seem to be low, it is worth mentioning that ICP results in 3.59 times more timeouts and WMA in 2.2 times more timeouts than CCNTimer, while the other algorithms are approximately on the same level in terms of timeouts. This means that ICP's Interest lifetimes are only slightly too short but retransmissions do not need to be sent on the wireless medium in most of the cases but can be satisfied from the local cache, i.e., delayed Data arrival. However, in case of more traffic, Data messages may not return in time, such that Interests need to be retransmitted, resulting in more traffic and an increased collision probability.

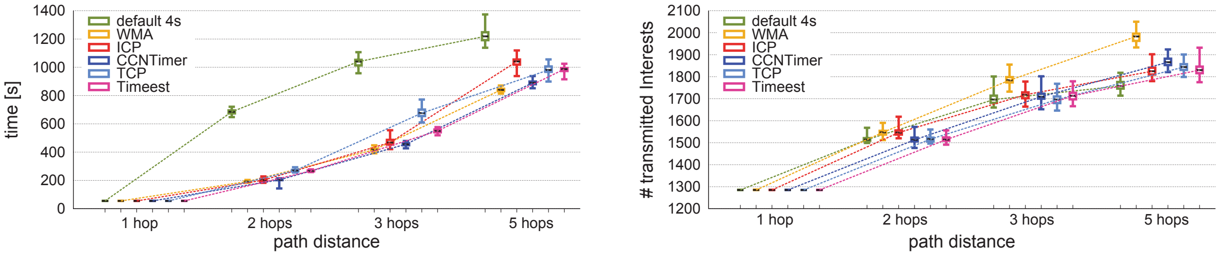
D. Multiple Requesters - Varying Hop Distance

In this section, we evalute the algorithms in scenario 2, where every node requests content from a content source. Every requester uses the same algorithm and we measure the performance at the leftmost requester, however, the relative performance of all algorithms at other requesters is similar.

Figure 4a shows the retrieval times for a 5MB file at the leftmost requester over multiple hops and Figure 4b, the number of transmitted Interests at the leftmost requester. CCNTimer shows the best overall performance (retrieval time and transmitted messages). When considering the retrieval time over 3 hops, ICP and CCNTimer perform approximately the same, while Timeest requires 21% more time and TCP's RTO even 48% more time than CCNTimer. For 5 hops, CCNTimer performs better than ICP (17% less time), Timeest (11% less time) and TCP's RTO (10.2% less time). Although WMA results in slightly faster retrieval times, i.e., 6% faster compared to CCNTimer for 5 hops, it is at the expense of 6.27% more Interest transmissions. In fact, Figure 4b shows that WMA results in considerably more Interest transmissions in all scenarios. On the other hand, CCNTimer results in shorter transmissions than Timeest (21% for 3 hops and 11% for 5 hops) but leads to only to 1.9% more transmitted Interests over 5 hops and even to 0.3% less transmitted Interests over 3 hops. We can also see that the overhead of adapting the Interest lifetime is marginal and the default strategy of constant 4 seconds results only in slightly fewer Interest transmissions, i.e., 6% fewer Interest transmissions but 36.9% longer transfer times for 5 hops compared to CCNTimer.

Figure 5 shows the measured RTT (blue dashed line) and Interest lifetime values (red solid line) for the 5 hops scenario. Figure 5a indicates that the number of collisions (and retransmissions) has increased compared to scenario 1, i.e., more RTT values at 4 or more seconds. Figure 5f reveals that the Interest lifetime for ICP stays almost constantly at 4 seconds, which is the reason why ICP degrades so drastically between 3 and 5 hops compared to the other algorithms in Figure 4a. Since ICP incorporates all RTT measurements including retransmissions, the RTT values can be considerably larger than with other algorithms. In case of frequent timeouts, there is always at least one large RTT, which causes the Interest lifetime to remain at 4 seconds (maximum value).

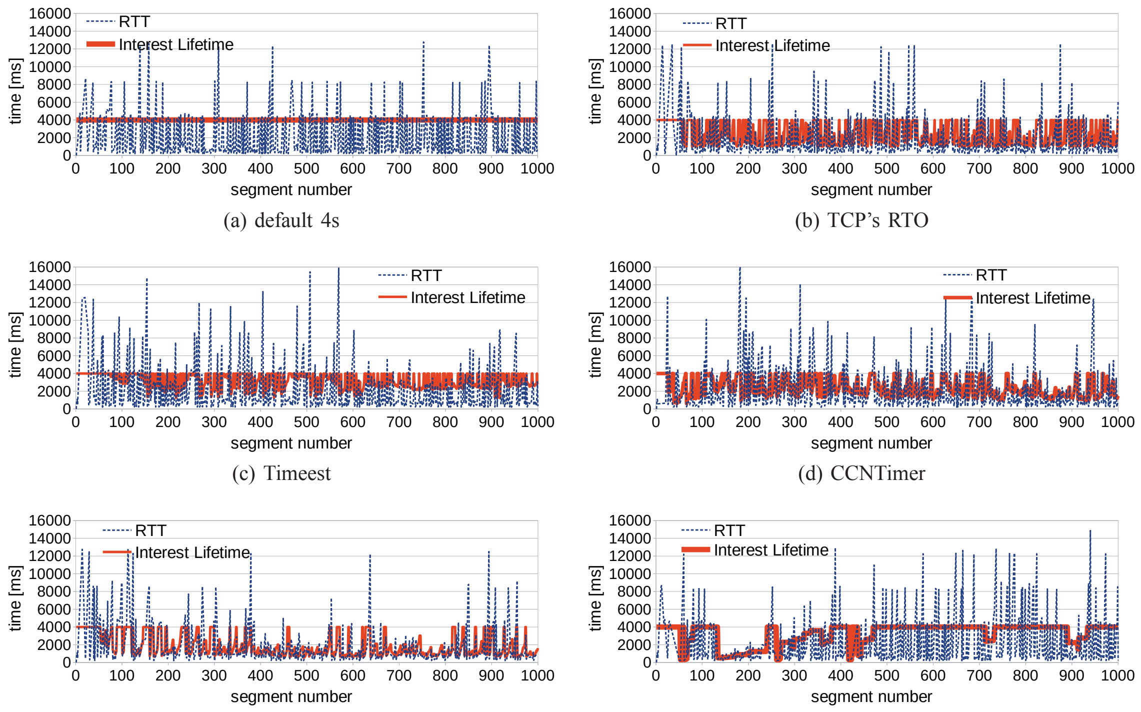
Figures 5c and 5d show that the Interest lifetime using Timeest and CCNTimer is often below 4 seconds. However,



(a) Retrieval Time

(b) Transmitted Interests

Fig. 4. Retrieval Time and Transmitted Interests for requester in scenario 2 for varying hop distance.



(a) default 4s

(b) TCP's RTO

(c) Timeest

(d) CCNTimer

(e) WMA

(f) ICP

Fig. 5. RTT and Interest lifetime from 1/100 run over 5 hops for the first 1000 segments.

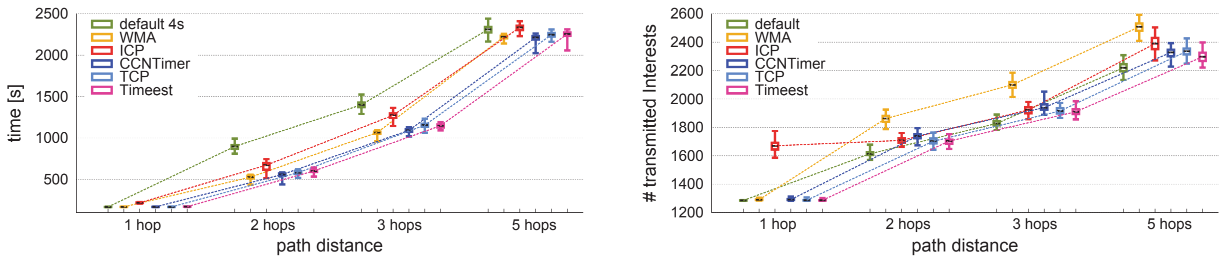
with Timeest the Interest lifetime quickly reaches 4 seconds, because the Interest lifetime is doubled for every collision. Figure 4b illustrates that slightly increasing the Interest lifetime (CCNTimer) after a timeout is often enough to retrieve the content and does not result in significantly more Interest transmissions and timeouts. If RTT variance would not be considered, such as for WMA, Interest lifetimes and RTTs are generally lower as Figure 5e shows. While this seems favorable at first, it also leads to a higher traffic load on the wireless medium as illustrated in Figure 4b.

E. Multiple Streams - Varying Hop Distance

In this section, we evaluate the algorithms in scenario 3, where each requester additionally receives and transmits Interests from two requesters (above and below) as Figure 1c shows. This results in considerably more traffic in the network.

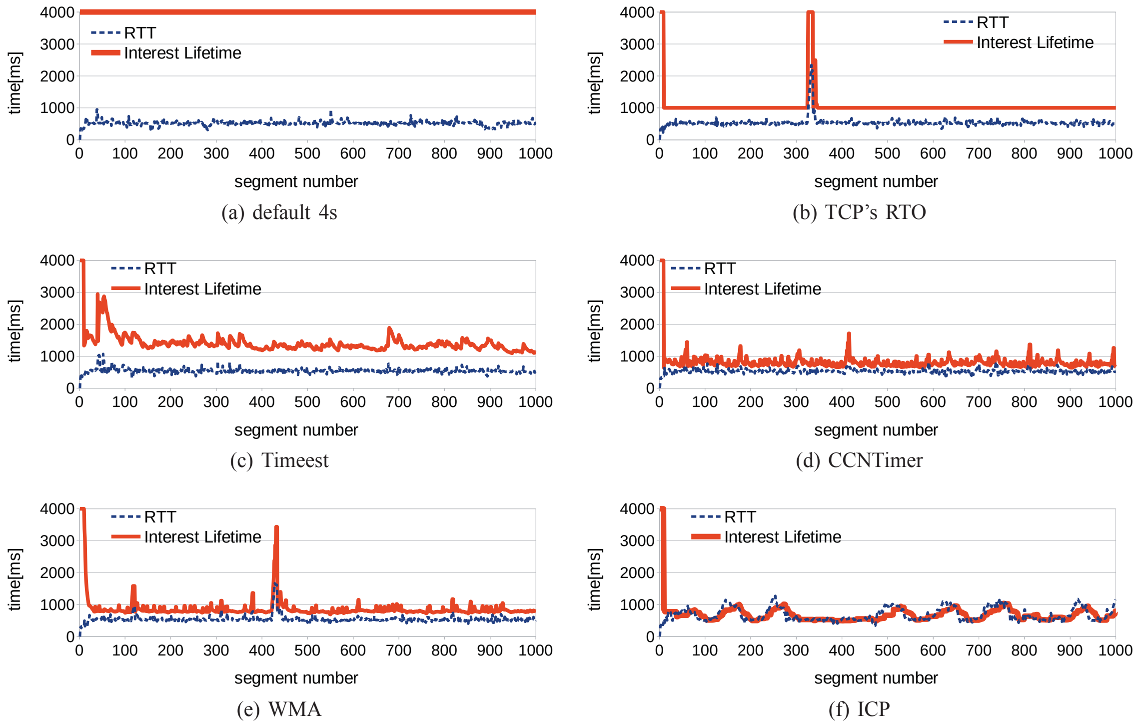
Figures 6a and 6b show the retrieval times and transmitted Interests for a 5MB file at the leftmost requester in the middle. We show the results for the same requester as in previous scenarios. Figure 6a shows that TCP's RTO, Timeest, CCNTimer and WMA perform similar with respect to retrieval times because there are many transitions to Interest lifetimes of 4 seconds. Only in case of 2 hops, WMA results in 6% faster transmission time but with increasing hop distance, WMA requires even slightly more time than CCNTimer. ICP needs between 29.36% (1 hop) and 5.76% (5 hops) more time than CCNTimer because the Interest lifetime remains mostly at 4 seconds similar to scenario 2.

The number of transmitted Interests is also similar for all algorithms in Figure 6b except for ICP and WMA. Compared to CCNTimer, WMA sends 7% more Interests over 2 hops, 8.3% more over 3 hops and 7.7% more over 5 hops. ICP sends significantly more Interests than the others over 1 hop, i.e., 29.36% more than CCNTimer and approximately the same



(a) Retrieval Time (b) Transmitted Interests

Fig. 6. Retrieval Time and Transmitted Interests for requester in scenario 3 for varying hop distance.



(a) default 4s (b) TCP's RTO (c) Timeest (d) CCNTimer (e) WMA (f) ICP

Fig. 7. RTT and Interest lifetime from 1/100 run over 5 hops for the first 1000 segments.

number of Interests (1.8% and 0.9% fewer for 2 and 3 hops, and 2.7% more for 5 hops) for more hops. This means that in high traffic scenarios, slightly too short Interest lifetimes (WMA, ICP) do not result in any benefits.

For 5 hops, the Interest lifetimes stay amost constantly at the maximum of 4 seconds due to many collisions, which is the reason why the algorithms perform approximately the same in terms of retrieval time (Figure 6a). To better see differences and understand why ICP sends more Interests over 1 hop, we show the Interest lifetimes and RTT values in Figure 7 for 1 hop. Recall that there are two additional requesters above and below the requester, which cause significantly more variability and processing delays compared to scenario 1.

For TCP's RTO, the Interest lifetime stays almost constantly at the lower bound of 1 second (Figure 7b). The RTT samples have a high variance and if the RTT values become slightly larger than 1s, an Interest retransmission needs to be performed. In high traffic scenarios, unnecessary transmissions should be avoided because they increase the collision probability, e.g., collision with the delayed Data message. For example between segment 324-230, there are 1-2 timeouts for the same segments, which causes the Interest lifetime to be quickly increased to 4 seconds. Consequently, also the RTT increases because of a higher Interest litetime, i.e., longer collision detection time.

With Timeest (Figure 7c), the Interest lifetime adapts better to varying RTT values. For example, at segment 43, the Interest lifetime increases quickly due to large RTT variations. We can see that the Interest lifetimes are significantly larger than most RTT values, which of course eliminate all unnecessary timeouts, but can result in long timeout periods in case of collisions.

Figure 7d shows that CCNTimer can also adapt quickly to changing RTT values. For example, at segment 418 and 811 the Interest lifetime increases due to higher RTT variance. Similar to Timeest, Interest lifetimes are also larger than RTT values, but Interest lifetimes are at a much lower level, which enables almost immediate reaction to collisions.

With WMA (Figure 7e), Interest lifetimes can also stay close to RTT values but since RTT variance is not considered, Interest lifetimes may be slightly too short in case of high RTT variability. For example, segment 435 is returned slightly too late, which causes an Interest retransmission that collides with a delayed Data arrival similar to TCP's RTO in Figure 7b.

Finally, Figure 7f shows the Interest lifetime and RTT values for ICP. ICP's retransmission timer may calculate a good average estimation, i.e., $RTT_{min}+0.5\times(RTT_{max}-RTT_{min})$, but in case of high variation, the algorithm underestimates Interest lifetime values. In case of very low RTT values, e.g., for segment 475, the Interest lifetime becomes very small causing many timeouts. Low RTT values have a larger influence than for other algorithms that use an exponential moving average because it is remembered and kept as the base for the next 20 Interest lifetimes. Figure 7f shows that Interest lifetimes and RTT values are continuously increasing due to many timeouts up to a maximum value, where no timeouts occur anymore. Then, RTT and Interest lifetimes decrease again until the next timeouts occur. While CCNTimer has approximately 2-3 timeouts over one hop, ICP has 1670 timeouts. This illustrates the importance of considering RTT variance and ensuring that Interest lifetimes are larger than current RTTs.

V. Conclusions

Adaptive algorithms for Interest lifetimes are beneficial in wireless networks to control how quickly retransmissions can be performed in case of collisions. Fast retransmissions have a large impact on transmission times in wireless information-centric networks. Since the number of transmitted packets is limited by the pipeline size, new Interests can only be transmitted if Data is received in return or the Interests time out. All evaluated algorithms resulted in considerably faster transmission times than the default 4 seconds in CCNx. A large lower bound for the retransmission timer such as with TCP's RTO is disadvantageous because RTT values may be much lower. Interest lifetimes in wireless networks need to account for high RTT variability. Thus, RTT variance and moving average must be multiplied by a small factor to avoid spurious transmissions.

Collisions in wireless communication are experienced frequently and are independent of the Interest lifetime. Therefore, a quick reaction to timeouts by doubling the Interest lifetime such as in most existing algorithms would only increase the detection time. Thus, it is better to only slightly increase it. Our evaluations have shown that doubling the lifetime does not result in fewer transmitted Interests but transmission times may increase due to longer timeout periods by a factor of 1.45 up to 4.2 depending on the scenario.

Strategies that are based on the history of RTT samples are much more complex than exponential moving averages because they need to maintain recent values in a buffer as well as compare and update minimum and maximum values for every new sample. Our evaluations did not show any benefits of such strategies compared to exponential moving averages. If the midrange of the samples is considered, the Interest lifetime may often be slightly too small resulting in more frequent timeouts. In addition, in case of frequent timeouts, the Interest lifetime may remain at the maximum value because of large maximum RTT samples.

In this work, we did not consider multi-homing because we assumed that a requester builds a unicast route to a content source and finds a new source to connect to if the path breaks. However, all algorithms could be extended to differentiate different content sources similarly as described in the corresponding related work.

References

[1] M. Caesar, T. Condie, J. Kannan, and K. Lakshminarayanan. ROFL: Routing on Flat Labels. In *ACM SIGCOMM*, pages p. 363–374, Pisa, Italy, September 2006.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A Data-Oriented (and Beyond) Network Architecture. In *ACM SIGCOMM*, pages p. 181–192, Kyoto, Japan, August 2007.

[3] M. Srel, T. Rinta-aho, and S. Tarkoma. RTFM: Publish/Subscribe Internetworking Architecture. In *ICT-MobileSummit*, pages p. 1–8, Stockholm, Sweden, June 2008.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Network Named Content. In *5th ACM CoNEXT*, pages p. 1–12, Rome, Italy, December 2009.

[5] C. Anastasiades, T. Schmid, J. Weber, and T. Braun. Opportunistic Content-Centric Data Transmission During Short Network Contacts. In *IEEE WCNC*, Istanbul, Turkey, April 2014.

[6] S. Oueslati, J. Roberts, and N. Sbihi. Flow-aware traffic control for a content-centric network. In *IEEE Infocom*, pages 2417–2425, Orlando, FL, USA, March 2012.

[7] G. Carofiglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networks. In *IEEE Nomen*, pages 304–309, Orlando, FL, USA, March 2012.

[8] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin. Ccn & tcp co-existence in the future internet: Should ccn be compatible to tcp? In *IFIP/IEEE IM*, pages 1109–1115, Ghent, Belgium, May 2013.

[9] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu. A transport protocol for content-centric networking with explicit congestion control. In *23rd ICCCN*, Shanghai, China, August 2014.

[10] T. E. Klein, K. K. Leung, R. Parkinson, and L. G. Samuel. Avoiding spurious tcp timeouts in wireless networks by delay injection. In *IEEE Globecom*, pages 2754 – 2759, Dallas, Texas, USA, November - December 2004.

[11] CCNx. http://www.ccnx.org/, November 2014.

[12] NS-3: Direct Code Execution, August 2014. http://www.nsnam.org/overview/projects/direct-code-execution/.

[13] M. Allman. Rfc 2988, November 2000.

[14] M. Amadeo, C. Campolo, and A. Molinaro. Design and analysis of a transport-level solution for content-centric vanets. In *IEEE ICC*, Budapest, Hungary, June 2013.

[15] V. Jacobson. Congestion avoidance and control. In *SIGCOMM Symposium on Communication Architectures and Protocols*, pages 314–329, New York, NY, USA, 1988.

[16] S. Braun, M. Monti, M. Sifalakis, and C. Tschudin. An empirical study of receiver-based aimd flow-control strategies for ccn. In *22nd ICCCN*, pages 1–8, Nassau, Bahamas, July - August 2013.

[17] L. Saino, C. Cocora, and G. Pavlou. Cctcp: A scalable receiver-driven congestion control protocol for content-centric networking. In *IEEE ICC*, pages 3775–3780, Budapest, Hungary, June 2013.

[18] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papalini. Multipath congestion control in content-centric networks. In *IEEE Nomen*, pages 1–6, Turin, Italy, April 2013.

[19] C. Anastasiades and T. Braun. Dynamic transmission modes to support opportunistic information-centric networks. In *IEEE NetSys*, Cottbus, Germany, March 2015.