From the Institute of Telematics
of the University of Lübeck

Director: Prof. Dr. Stefan Fischer

BEAR: Bandwidth-Estimation-based Flow Admission Control and Routing in IEEE 802.15.4-based Ad-hoc Sensor Networks

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Science

Submitted by
Muhamad Omer Farooq
from Faisalabad, Pakistan

Lübeck, 2014

First Referee: Prof. Dr. rer. nat. Stefan Fischer
Second Referee: Prof. Dr.-Ing. Thomas Kunz
Date of oral examination: 11.06.2014
Approved for printing. Lübeck, 12.06.2014

Zusammenfassung

Die vorliegende Dissertation behandelt die Unterstützung von Echtzeit-Multimedia-Übertragungen in ad-hoc drahtlosen Sensornetzwerken (WSNs). Echtzeit-Multi-media-Übertragungen stellen Anforderungen an die Qualität des Services (Quality of Service, QoS) in Form von unteren Schranken für Verzögerungen und Paketverlusten sowie Bandbreitenausnutzung. Typischerweise verwenden WSNs den IEEE 802.15.4-Standard in der Medium Access Control- (MAC) sowie der Bitübertragungs-Schicht (PHY). Dieser Standard unterstützt Echtzeit-Multim-edia-Übertragung jedoch nicht zufriedenstellend, sodass der Fokus Unserer Arbeit auf der Unterstützung von Echtzeit-Multimedia-Übertragungen in ad-hoc basierten WSNs mittels dem Standard IEEE 802.15.4 liegt.

Naturgemäß weist die drahtlose Kommunikation Störungen durch Interferenzen auf. Jene Störungen in Verbindung mit dem Overhead des MAC-Protokolls und der Implementierung des Netzwerk-Protokoll Stacks limitieren die verfügbare Bandbreite in Drahtlosnetzwerken erheblich, sodass es zu einem Datenstau kommen kann, obwohl die Übertragungsrate der Knoten deutlich unter der maximalen Bandbreite der darunter liegenden Kommunikationstechnik liegt. Um die QoS-Anforderungen innerhalb des IEEE 802.15.4-Standards sicherzustellen, sollte jeder Knoten im Netzwerk nur soviele Daten übertragen, dass sich diese zusätzlichen Übertragungen nicht negativ auf die bereits existierenden Echtzeit-Multimedia-Übertragungen auswirken. Ein Routing-Protokoll sollte daher einen Pfad wählen, der den QoS-Anforderungen besser gerecht wird.

Die MAC-Schicht bestimmt die Aufteilung des Kommunikationsmediums. Diese Dissertation analysiert die Ergebnisse der Aktivierung oder Deaktivierung von Carrier Sense Multiple Access Collision Avoidance (CSMA-CA) in der MAC-Schicht, indem untersucht wird, welche Auswirkungen auf den Kanaldurchsatz und der Verzögerung/Latenz der Paketlieferung zu erwarten sind. Die Parameter, die die Wahl in Bezug auf Aktivierung oder Deaktivierung der ACKs in der MAC-Schicht beeinflussen, sind: (I) Ende-zu-Ende-Verzögerung und Anforderungen an den Paketverlust in Echtzeit-Multimedia-Übertragungen, (II) Datenlast innerhalb des Interferenzbereiches der Sender entlang dem Datenleitungspfad und (III) Länge des Datenleitungspfads.

In dieser Dissertation werden die Einschränkungen von aktuellen Regelalgorithmen für ad-hoc Netzwerke herausgestellt. Die Ergebnisse zeigen, dass jene Regelalgorithmen ihrer Aufgabe nicht gerecht werden. Es wurden mehrere Faktoren identifiziert, die ein effektiver, bandbreitenbasierter Regelalgorithmus berücksichtigen sollte. Erstens: Ein erhöhter Datenverkehr in einem Netzwerk erhöht den CSMA-CA MAC-Schicht Overhead. Zweitens: Der Zugangskonflikt

zu einem Knoten, der nicht Teil eines Pfades ist, ist abhängig von der Anzahl der Sender (entlang des Pfades) im Störungsbereich des Knotens. Drittens: Die Selbstinterferenz eines Flusses (entlang des Pfades) ist abhängig vom Abstand einzelner Knotens von der Quelle bzw. zum Zielknoten. Unter Berücksichtigung dieser Faktoren wurde BandEst entworfen und implementiert. BandEst ist eine Kombination aus messungsbasierter Technik zur Schätzung der verfügbaren Bandbreite und eines Regelalgorithmus für auf IEEE 802.15.4-basierte ad-hoc WSNs. Die Ergebnisse zeigen, dass BandEst für ad-hoc-Funknetze deutlich besser als aktuell verfügbare Regelalgorithmen für ad-hoc-Funknetze ist.

Abschließend wurde ein bandbreitenbasiertes proaktives Routingprotokoll für ad-hoc WSNs mit einzelnen oder mehreren Zielknoten für IEEE 802.15.4 entworfen und implementiert. Der Routingprotokoll ermittelt den besten Pfad in Bezug auf die verfügbare Bandbreite eines jeden Zielknoten in einem Netzwerk. Darüber hinaus kann ein Knoten mehr als einen Datenweiterleitungspfad in Richtung des gleichen Senkungsknoten speichern. Umfangreiche Experimente zeigen die Unterschiede zwischen einem aktuellen opportunistischen Routing-Protokoll und dem proaktiven Routing-Protokoll auf. Es zeigt sich, dass opportunistische Routing-Protokolle Datenlasten ungleichmäßig (bei mehreren Zielknoten) verteilen, was in einer hohen Ende-zu-Ende-Verzögerung und einer niedrigen Packet Delivery Ratio (PDR) resultiert. Im Falle des proaktiven Routing-Protokolls führt die Auswahl der Weiterleitungspfade durch Berücksichtigung lediglich der verfügbaren Ende-zu-Ende-Bandbreite häufig zu längeren Datenweiterleitungspfaden. Dies hat eine höhere Selbstinterferenz der Datenflüsse zur Folge, mit negativen Folgen sowohl für den PDRs als auch für die Ende-zu-Ende-Verzögerung. In einem der experimentellen Szenarien mit mehreren Zielknoten konnte gezeigt werden, dass das proaktive Routingprotokoll, bei einer sorgfältigen Auswahl der Datenweiterleitungspfad(e), die nicht zu lang sind, jedoch eine bessere Ende-zu-Ende Bandbreite aufweisen, erheblich verbesserte Leistung aufzeigen kann.

Darüber hinaus wurde BandEst in das proaktive-Routingprotokoll integriert. Die Ergebnisse zeigen, dass im Allgemeinen das Zusammenspiel aus Ende-zu-Ende-Bandbreite und der Länge eines Datenweiterleitungspfades die Verzögerung minimieren sowie Ende-zu-Ende PDR verbessern kann.

Abstract

This dissertation is in the context of supporting real-time multimedia flows in ad-hoc Wireless Sensor Networks (WSNs). Real-time multimedia flows require Quality of Service (QoS) provisioning in terms of bounds on delay and packet loss along with a soft bandwidth guarantee. Typically, WSNs use the IEEE 802.15.4 standard at the Medium Access Control (MAC) and Physical (PHY) layers. The IEEE 802.15.4 standard does not support real-time multimedia flows well. Therefore, our work focuses on supporting real-time multimedia flows in IEEE 802.15.4-based ad-hoc WSNs.

The shared nature of the wireless communication medium results in interference. Interference combined with the overheads associated with a MAC protocol, and the implementation of a networking protocol stack limit the available bandwidth in wireless networks, and can result in congestion, even if the transmission rates of nodes are well below the maximum bandwidth supported by an underlying communication technology. Therefore, to satisfy real-time multimedia flows' QoS requirements inside IEEE 802.15.4-based ad-hoc WSNs, each node inside the network should determine the amount of data that the node can transfer without negatively impacting the performance of real-time multimedia flows. Moreover, a routing protocol should select a forwarding path that can better satisfy the real-time multimedia flows' end-to-end QoS requirements.

The MAC layer decides the sharing of the communication medium, and in this dissertation our results demonstrate that enabling or disabling the IEEE 802.15.4's unslotted Carrier Sense Multiple Access Collision Avoidance (CSMA-CA) MAC layer ACKs impacts channel throughput and packet delivery delay. The parameters that affect the choice regarding enabling or disabling the MAC layer ACKs for real-time multimedia flows are: (i) end-to-end delay and packet loss requirements of real-time multimedia flows, (ii) data load within the interference range of transmitters along the data forwarding path, and (iii) length of the data forwarding path.

In this dissertation, we highlight limitations of the state-of-the-art flow admission control algorithms for ad-hoc wireless networks. Our results demonstrate that the state-of-the-art flow admission control algorithms for wireless ad-hoc networks fail in their task. We identified multiple factors that an effective available-bandwidth-based flow admission control algorithm should consider. First, increased data traffic in a network increases the CSMA-CA MAC layer overhead. Second, the contention count on a node that is not on a flow's data forwarding path is a function of the number of transmitters (along the flow's forwarding path) within the interference range of the node. Third, a flow's intra-

flow contention count on a node (along the flow's forwarding path) depends on the hop-count distance of the node from the source and the destination nodes. Taking these factors into account, we designed and implemented BandEst; combination of a measurement-based available bandwidth estimation technique and a flow admission control algorithm for IEEE 802.15.4-based ad-hoc WSNs. Our results demonstrate that BandEst significantly outperforms the state-of-the-art flow admission control algorithms for ad-hoc wireless networks.

Finally, we designed and implemented an available-bandwidth-based proactive routing protocol for IEEE 802.15.4-based single-sink and multi-sink ad-hoc WSNs. The available-bandwidth-based proactive routing protocol maintains the best forwarding path in terms of the end-to-end available bandwidth towards each sink node present in a network. Moreover, a node can maintain more than one data forwarding path towards the same sink node. We performed extensive experiments, and compared our proactive routing protocol with a state-of-the-art opportunistic routing protocol. Our results demonstrate that the opportunistic routing protocol can distribute data load unevenly (in case of multiple sink nodes), hence results in high end-to-end delay and low Packet Delivery Ratio (PDR). In case of our proactive routing protocol, selecting forwarding paths by only considering the end-to-end available bandwidth frequently results in lengthy data forwarding paths. Lengthy data forwarding paths result in higher intra-flow contention, hence PDR and end-to-end delay are impacted. One of the experimental scenarios, using multiple sink nodes, demonstrates that in case of our proactive routing protocol, carefully selecting the data forwarding path(s) that are not too long compared to the shortest available data forwarding path(s), but have better end-to-end available bandwidth significantly improves the performance of the proactive routing protocol. Moreover, we integrated BandEst with the available-bandwidth-based proactive routing protocol. Our results indicate that, in general, trading off end-to-end available bandwidth and the length of a data forwarding path may improve end-to-end PDR and delay.

Acknowledgments

Thanks to Almighty Allah for blessing me with the strength to complete this PhD thesis. I would like to express my deepest gratitude to my PhD thesis supervisors Dr.-Ing. Thomas Kunz (Carleton University, Canada) and Dr. rer. nat. Stefan Fischer (University of Luebeck, Germany). This thesis would not have been possible without their guidance and support. The comments/suggestions provided by Dr.-Ing. Thomas Kunz during the thesis were really valuable, and I believe that the training I got from both supervisors will play an immense role in my future career.

I would like to thank my parents for being the best parents. I would like to thank my sisters for their love and care, especially Humaira for hosting me in Germany. I would like to thank my nephew Sahil and niece Sara for making my stay in Germany a good experience. Moreover, I would like to thank my wife for her care and support in general, and particularly during the thesis. I would like to thank my sons Hamza and Subhan for their innocent pranks, those typically, served as a refresher during the thesis.

I would like to thank Muhammad Ashraf (late); my uncle for his support and motivation in pursing a PhD.

To my parents; for unconditional and unmatched love, support, and motivation

Contents

1. Introduction

According to the ITU, Quality of Service (QoS) is defined as the "totality of characteristics of a telecommunication service that bear on its ability to satisfy stated and implied needs of the users of the service". In computer networks, QoS refers to a network's ability to deliver predictable results in terms of certain metrics, especially network availability, bandwidth, delay, error rate, and packet loss. QoS is categorized as either hard QoS or a soft QoS. If an application requires guaranteed QoS in terms of certain metrics, and momentary violation of an application's QoS requirement is not tolerable, the application QoS requirement is called a hard QoS requirement. If an application can tolerate momentary violation in its QoS requirement, and such violation does not result in a system malfunction, the application QoS requirement is called a soft QoS requirement. QoS requirement of a real-time multimedia application can be categorized as a soft QoS requirement as such an application requires bounds on delay, jitter, and bandwidth, and at the same time such an application can tolerate momentary QoS violation, but momentary violations must be limited. Generally, QoS models/solutions can be categorized as per-flow or class-based QoS models. In a per-flow-based (a flow is an end-to-end data transfer connection between the source and the destination nodes) QoS model, QoS guarantees are enforced on per-flow basis, i.e., a per-flow-based QoS model ensures that end-to-end QoS requirements of each flow must be satisfied with no or minimum violation of a flow's QoS-agreement. There is a scalability issue with a per-flow-based QoS model, i.e., if the number of QoS flows is very large (in this case, the definition of "very large" depends on network, computing, and memory resources), available resources may become insufficient. In a class-based QoS model, QoS guarantees are enforced on aggregates, i.e., a class-based QoS model defines different traffic classes, and group of flows are mapped to a certain traffic class based on some criteria, e.g., flows' priorities, application type, pricing policy, etc. Class-based QoS provisioning does not necessarily translate into per-flow QoS guarantees.

IEEE 802.15.4-based ad-hoc WSNs have their application in many smart environments such as visual surveillance [14], assisted living [66], intelligent transportation [53], and habitant monitoring [41] to name but a few. Therefore, applications running on such ad-hoc WSNs can generate real-time multimedia data [3], and QoS provisioning is essential to support these applications in WSNs.

The IEEE 802.15.4 unslotted Carrier Sense Multiple Access Collision Avoidance (CSMA-CA) MAC layer protocol can work in reliable and unreliable mode. Working in the reliable mode, a node waits for a constant time period to receive

an ACK for the transmitted data frame. Therefore, the reliable CSMA-CA protocol may result in increased end-to-end delay, and decreased end-to-end throughput. In the unreliable mode, the CSMA-CA MAC layer ACKs are not transmitted, hence a node does not wait for ACKs after transmitting data frames (this discussion is only relevant to unicast data frames). Using the IEEE 802.15.4 unslotted CSMA-CA MAC layer protocol in the unreliable mode may increase a node's transmission rate, packet loss rate, and at the same time it may also decrease a data frame's end-to-end delay, as nodes do not wait to receive an ACK frame. As the unreliable CSMA-CA protocol does not retransmit data frames, a flow's end-to-end throughput may not be predictable, primarily due to the characteristics of the wireless communication channel, i.e., multi-path fading, scattering, etc. End-to-end delay and predictable end-to-end throughput are important QoS metrics. Therefore, considering the QoS requirement of real-time multimedia applications, and the possible pros and cons of reliable and unreliable CSMA-CA MAC layer protocols, a through study of reliable and unreliable CSMA-CA MAC layer protocol is required to select the appropriate MAC layer protocol for effectively supporting real-time multimedia applications in IEEE 802.15.4-based ad-hoc WSNs.

Bandwidth is a shared and scarce resource in WSNs. Interference along with the overheads associated with the IEEE 802.15.4's unslotted CSMA-CA MAC protocol, and a networking protocol stack's implementation further limits the available bandwidth. This can result in congestion even if nodes' transmission rates are well below the bandwidth supported by the IEEE 802.15.4 standard. Congestion increases delay and packet loss, hence it may result in a performance degradation of real-time multimedia flows inside a network. Therefore, each node inside a WSN should estimate the available bandwidth. Based on the available bandwidth estimate, a flow admission control algorithm can be used to limit the amount of data inside a network so that the QoS requirements of real-time multimedia flows can be satisfied. Designing a flow admission control algorithm for ad-hoc wireless networks is a challenging task in general, primarily because of the shared nature of the wireless communication medium [71].

A routing protocol helps to relay data from the source node to the destination node, therefore the state of the relaying nodes (along a flow's data forwarding path) in terms of congestion, available bandwidth, and node traversal delay can affect the performance of real-time multimedia applications. Hence, in an attempt to satisfy a real-time multimedia flow's QoS requirements, a routing protocol must select the data forwarding path that best suits the real-time multimedia flow's QoS requirements.

1.1. Research Motivation

Typically in ad-hoc wireless networks, QoS provisioning is done through estimating the available resources such as the available bandwidth, and then restricting the amount of data traffic inside a network w.r.t. the available

bandwidth and flows' QoS requirements through a flow admission control algorithm [58]. In IEEE 802.15.4-based ad-hoc WSNs the sharing of the communication medium is decided by the MAC layer, therefore the amount of bandwidth available to applications running on WSNs is dictated by the MAC layer [58]. Moreover, a routing protocol selects a data forwarding path and the state of links on the selected data forwarding path in terms of the available bandwidth may impact the performance of a real-time multimedia application [71]. In this dissertation, our work focuses on the following for QoS provisioning in IEEE 802.15.4-based ad-hoc WSNs.

- Suitability of IEEE 802.15.4's CSMA-CA MAC layer protocol for ad-hoc WSNs.
- Available-bandwidth-based flow admission control algorithm.
- Available-bandwidth-based routing protocol and its integration with the flow admission control algorithm.

1.1.1. Suitability of the IEEE 802.15.4 CSMA-CA MAC layer for Real-Time Applications

The CSMA-CA MAC layer protocol standardized in the IEEE 802.15.4 specification can work in reliable and unreliable mode. Working in the reliable mode, the IEEE 802.15.4 CSMA-CA protocol waits for a constant period of time after transmitting a data frame to receive an ACK frame. If an ACK frame is not received, the MAC layer backs-off, and a retransmission attempt is made after a random exponential back-off delay. This process increases delay, and it seems that a flow's end-to-end throughput may decrease, primarily due to ACK overhead. Using the IEEE 802.15.4 CSMA-CA MAC layer protocol in the unreliable mode may increase a node's transmission rate, and packet loss rate, but at the same time it can also decrease a data frame's end-to-end delay, as nodes do not wait to receive an ACK frame. Unreliable CSMA-CA does not retransmit data frames, therefore a flow's end-to-end throughput may not be predictable, primarily due to the characteristics of the wireless communication channel. QoS in general, is related with the predictability of the service being offered. In case of real-time multimedia applications, one QoS metric of interest is predictable end-to-end throughput along with bounded delay and packet loss. Therefore, a thorough simulation-based study of reliable and unreliable CSMA-CA protocols is carried out to explore their suitability for real-time multimedia applications in IEEE 802.15.4-based ad-hoc WSNs. Based on our work, we present different factors that affect the IEEE 802.15.4 channel throughput and per-packet delivery ratio, moreover we highlight different factors that affect the suitability of the IEEE 802.15.4 CSMA-CA (reliable or unreliable) MAC layer for real-time multimedia applications.

1.1.2. Available-Bandwidth-based Flow Admission Control Algorithm

To satisfy a real-time multimedia application's delay and bandwidth requirements, an estimate of the available bandwidth is essential [12]. Exceeding what is available in terms of the available bandwidth can result in congestion, hence increased delay and decreased throughput. Bandwidth is a shared resource in a wireless network, therefore the amount of available bandwidth is affected by interference, intra-flow contention, channel access overhead, and transmission errors. Hence, estimating the available bandwidth in wireless networks is fundamentally different as compared to wired networks, as estimating the available bandwidth in a wired communication link does not require considering intra-flow contention and a flow's contention on non-relaying nodes. In this work, we highlight different factors that result in an effective available-bandwidth-based flow admission control algorithm in ad-hoc wireless networks. Based on the identified factors, we designed BandEst; a measurement-based available bandwidth estimation and flow admission control algorithm. Our results demonstrate that BandEst significantly outperforms the state-of-the-art available-bandwidth-based flow admission control algorithms for ad-hoc wireless networks.

1.1.3. Available-Bandwidth-based Routing Protocol and its Integration with the Flow Admission Control Algorithm

Based on some metric or a combination of metrics, e.g., hop-count, delay, congestion status, available bandwidth, etc, a routing protocol selects a single or multiple data forwarding paths from the source node to the destination node. The state of communication links on a particular data forwarding path impacts the performance of an application in terms of delay and packet delivery ratio. In our work, we designed a proactive available-bandwidth-based routing protocol. The routing protocol uses end-to-end available bandwidth as a routing metric. Afterwards, we integrated BandEst with the available-bandwidth-based proactive routing protocol. Our results indicate that trading off end-to-end available bandwidth and the length of a data forwarding path may improve end-to-end Packet Delivery Ratio (PDR) and delay.

1.2. Thesis Contributions

The contributions of this thesis are as follows.

- Our survey of WSNs and Wireless Multimedia Sensor Networks (WMSNs) testbeds considering their capabilities and scale, along with the capabilities of state-of-the-art wireless multimedia sensor nodes are documented in the following papers.

  - M. O. Farooq and T. Kunz,"Wireless Multimedia Sensor Networks Testbeds, and State-of-the-Art Multimedia Sensor Nodes", Applied Mathematics and Information Sciences Journal vol. 8 no. 3, 2014.
  - M.O. Farooq and T. Kunz, "Wireless Multimedia Sensor Networks Testbeds and State-of-the-Art Hardware: A Survey", In Future Generation in Communication and Networking (FGCN), Jeju Island, South Korea, December 8-10, 2011.

- The papers listed below document the following: (i) our analysis of an operating system's and a networking protocol stack implementation impact on the IEEE 802.15.4 channel utilization and a node's throughput, (ii) the relationship of the IEEE 802.15.4 channel bandwidth with the real-time multimedia applications end-to-end throughput and delay requirements, and (iii) a study of the suitability of the IEEE 802.15.4 CSMA-CA MAC layer for real-time multimedia applications.

  - M.O. Farooq and T. Kunz, "Contiki-based IEEE 802.15.4 Node's Throughput and Wireless Channel Utilization Analysis", In 5^{th} IFIP Wireless Days, Dublin, Ireland, November 21-23, 2012.
  - M.O. Farooq and T. Kunz, "On Determining Bandwidth Usage Threshold to Support Real-Time Multimedia Applications in Wireless Multimedia Sensor Networks", In 27^{th} International Conference on Advanced Information Networking and Applications Workshops (WAINA), Barcelona, Spain, March 25-28, 2013.
  - M. O. Farooq and T. Kunz, "Contiki-based IEEE 802.15.4 Channel Capacity Estimation and Suitability of its CSMA-CA MAC Layer Protocol for Real-Time Multimedia Applications", to appear in Mobile Information Systems Journal.

- The papers listed below document the following: (i) Key factors for an effective available-bandwidth-based flow admission control algorithm in ad-hoc wireless networks, (ii) our general framework for the available-bandwidth-based flow admission control algorithm and the routing protocol for IEEE 802.15.4 ad-hoc networks, and (iii) our proactive available bandwidth estimation technique that considers the complete impact of the IEEE 802.15.4 CSMA-CA MAC layer protocol on the available bandwidth.

  - M.O. Farooq and T. Kunz, "Key Factors for a Proper Available-Bandwidth-based Flow Admission Control in Ad-hoc Wireless Sensor Networks", In 8^{th} International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN), Benidrom, Spain, June 22-27, 2014.
  - M.O. Farooq and T. Kunz, "BEAR: Bandwidth Estimation-based Admission Control and Routing for IEEE 802.15.4-based Networks", In 6^{th} Joint IFIP Wireless and Mobile Networking Conference (WMNC),

Dubai, UAE, April 23-25, 2013.

. M.O. Farooq and T. Kunz, "Proactive Bandwidth Estimation for IEEE 802.15.4-based Networks", In IEEE 77^{th} Vehicular Technology Conference (VTC2013-Spring), Dresden, Germany, June 2-5, 2013.

- The following paper document our end-to-end available-bandwidth-based proactive routing protocol for IEEE 802.15.4 ad-hoc networks and its comparison with an opportunistic available-bandwidth-based routing protocol.

  . M.O. Farooq and T. Kunz, "Available-Bandwidth-based Routing in IEEE 802.15.4-based Ad-hoc Networks: Proactive vs. Opportunistic Technique, In 28^{th} IEEE International Conference on Advanced Information Networking and Applications (AINA), Victoria, Canada, May 13-16, 2014..

1.3. Thesis Organization

The rest of this dissertation is organized as follows: Chapter 2 presents key factors for a proper flow admission control in ad-hoc wireless networks. Related work is presented in Chapter 3. Chapter 4 presents our suitability analysis of the IEEE 802.15.4 CSMA-CA MAC protocol for real-time multimedia applications, and limitations of the Contiki operating systems and its implementation of the IEEE 802.15.4 CSMA-CA MAC protocol. Chapter 5 presents our proposed proactive available bandwidth estimation technique. Chapter 6 presents BandEst; a measurement-based available bandwidth estimation technique and a distributed flow admission control algorithm for IEEE 802.15.4-based ad-hoc networks. In this chapter, a through performance comparison of BandEst with the state-of-the-art available-bandwidth-based flow admission control algorithms for ad-hoc wireless networks is also presented. Chapter 7 presents our available-bandwidth-based proactive routing protocol for IEEE 802.15.4-based ad-hoc networks and a performance comparison of the routing protocol with a state-of-the-art routing technique for IEEE 802.15.4-based ad-hoc WSNs. This chapter also discusses integration of the routing protocol with BandEst, moreover the results obtained by running BandEst over the proactive routing protocol are compared with the results obtained by running BandEst over a simple hop-count-based routing protocol. Finally, this dissertation concludes in Chapter 8.

2. Key Lessons for a Proper Flow Admission Control

In ad-hoc wireless networks, typically, QoS provisioning is done through estimating the available resources such as the available bandwidth, and then restricting the amount of data inside a network w.r.t. the available bandwidth and flows' QoS requirements through a flow admission control algorithm. In this chapter, we identify various factors that must be considered for a proper flow admission control in ad-hoc wireless networks.

2.1. Empirically Estimating the MAC Layer Overhead

To measure the IEEE 802.15.4's unslotted CSMA-CA MAC layer overhead (back-off and retransmission) with an increased data load inside a network, we conducted multiple simulations using a simple line topology of 4 nodes in which each intermediate node is within the transmission range of its immediate upstream and downstream nodes. General simulation parameters are shown in Table 2.1. We selected a line topology to estimate the MAC layer overhead (back-off and retransmission) because it can capture the effects of intraflow contention, and at the same time it facilitates reporting of the MAC layer overhead at each node present inside the network. The simulations are performed on the Cooja WSN simulator. To estimate the MAC layer overhead (back-off and retransmission) we consider the aggregate data rate, but there are other parameters that may affect the MAC layer overhead such as packet size, nature of traffic (burst, constant bit rate), and number of flows inside a network. We expect that, beyond the aggregate data rate, other parameters will only have a modest impact. We created 10 different simulation scenarios, and in each scenario we vary the offered data load inside a network. In the first simulation scenario node 1 transfers 2 kbps to node 4, as nodes 2 and 3 are acting as the relaying nodes, therefore in this case, the total data load within the interference range of nodes 1, 2, and 3 is 6 kbps. In subsequent simulation scenarios, node 1 increases its data generation rate in such a manner that it increments data load within the interference range of nodes 1, 2, and 3 to 12, 18, 24, 30, 36, 42, 48, 54, and 60 kbps. Each simulation runs for 110 seconds, and to determine the mean value each simulation is repeated 10 times. The backoff overhead is measured in time, but Figure 2.1 reports the MAC layer overhead in bps. We converted the mentioned overhead to bps by multiplying the accumulated time duration (during each second) a node spends in the back-off mode with the channel rate.

Table 2.1.: General Simulation Parameters

Parameter	Value
MAC layer	Unslotted CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 frames
Channel rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Simulated node type	Tmote sky

Figure 2.1 shows that with a substantial increase in the data load the average back-off and retransmission overheads increase, therefore in such cases it is essential to proactively consider the back-off and retransmissions overhead by taking into account the additional data load inside a network. If the anticipated data load within interference range of a node inside a network is in excess of 60 kbps, extrapolation techniques can be used to determine the additional back-off and retransmission overheads. Cooja emulates the Contiki operating system's [1] CSMA-CA MAC layer, and Contiki CSMA-CA uses a constant contention window size, therefore we can derive contention window overhead by knowing the number of additional packets a node intends to transmit.

2.2. Proactively Considering Additional MAC Layer Overhead on Non-Relaying Nodes

To demonstrate the affects of an increased data load on the MAC layer overhead at nodes within the interference range of transmitters along the data forwarding path, we create two simulation scenarios (general simulation parameters are shown in Table 2.1, and total duration of each simulation is 100 seconds), using the topology shown in Figure 2.2. In Scenario 1, node C transmits 10 kbps to node D (10 data packets per second). In Scenario 2, in addition to the flow from node C to ode D, node A transmits 10 kbps to node B (10 data packets per second) and node E transmits 10 kbps to node F (10 data packets per second). In both scenarios, we keep track of the mean MAC layer overhead at node C. We repeated each simulation scenario 10 times. One thing to notice is that node C is neither on the data forwarding path of node A's flow nor it is on the data path of node E's flow, but it is within the interference range of nodes A and E. Moreover, node C is also transmitting data.

The results shown in Table 2.2 demonstrates that with an increased data load within the interference range of node C, the MAC layer overhead has increased at the node. Therefore, a good flow admission control algorithm must proac-

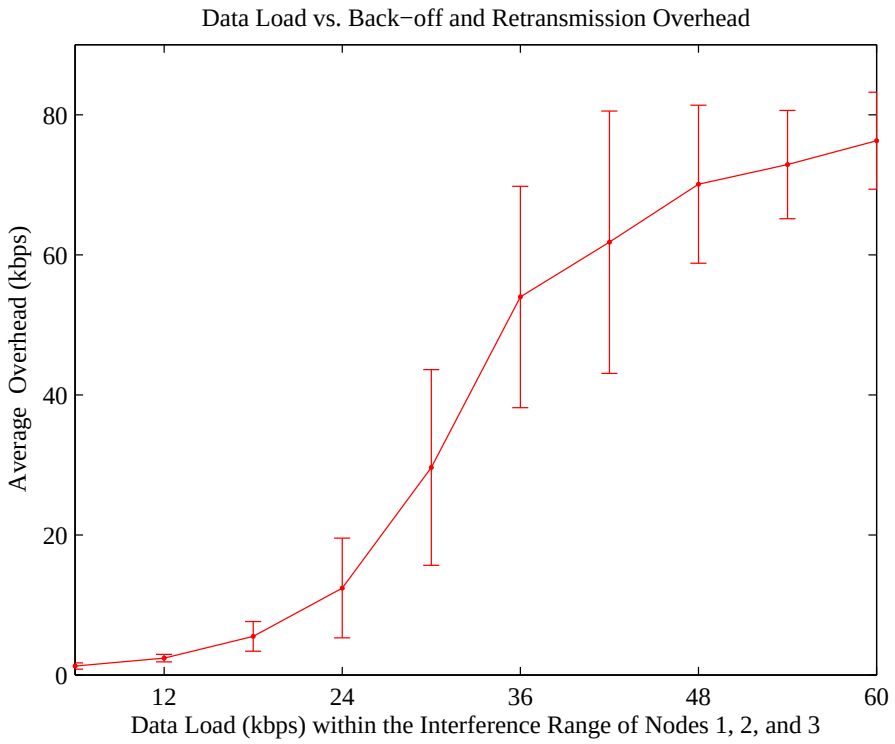


Figure 2.1.: Data Load vs. Average Back-off and Retransmission Overhead

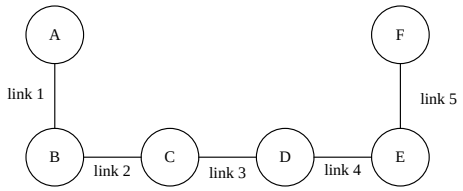


Figure 2.2.: Simulated Network Topology

tively consider the additional MAC layer overhead at nodes which are not on a new flow's data path, but are within the interference range of transmitters along the data forwarding path. This factor must only be taken into account for nodes which are transmitting/relying data.

Table 2.2.: MAC Layer Overhead at Node C

Scenario	**Mean Overhead**	**95% Confidence Interval**
1	91.67 kbps	90.20 - 93.14 kbps
2	103.55 kbps	100 - 107.16 kbps

Table 2.3.: Data Activity as Measured by Nodes

Node ID	Mean Data Activity	95% Confidence Interval
A	31.41 kbps	31.21 - 31.61 kbps
B	41.35 kbps	41.16 - 41.53 kbps
C	51.85 kbps	51.53 - 52.17 kbps
D	43.48 kbps	43.33 - 43.63 kbps
E	33.18 kbps	32.87 - 33.49 kbps
F	22.85 kbps	22.64 - 23.06 kbps

2.3. Determining Maximum Intra-Flow Contention Count

It has been claimed in [45], [58], and [73] that the maximum intra-flow contention count on a node along the data forwarding path is 4. To verify the claim, we carried out a simulation-based experiment. In our experiment, using the topology shown in Figure 2.2 node A is the source node and node F is the sink node. Node A transmits at the rate of 10 kbps (10 data packets per second) to the sink node. Node A starts the transmission at simulation time of 5 seconds and terminates the data packets transmission at the simulation time of 105 seconds. In our experiment, we measured the data activity at nodes inside the network via wireless channel-sensing throughout the duration of node A's flow. We repeated the experiment 10 times, and the mean data load observed by the nodes while node A is transmitting data packets is shown in Table 2.3 along with the 95% confidence interval. The end-to-end flow's throughput was perfect, i.e., 10 kbps.

Table 2.3 demonstrates that the mean data load observed by node C is approximately 50 kbps, which is 5 times the transmission rate of node A. The data loads observed by nodes A, B, D, E, and F are approximately 30 kbps, 40 kbps, 40 kbps, 30 kbps, and 20 kbps respectively. Therefore, the contention counts at nodes A, B, D, E, and F are 3, 4, 4, 3, and 2 respectively. The maximum contention count is 5, the reason is that nodes A, B, D, and E are within the interference range of node C, hence node C cannot transmit while nodes A, B, D, and E are transmitting. Moreover, node C also relays node A's data, therefore the correct maximum contention count is 5.

2.4. Determining Correct Contention Count on Nodes not on a Flow's Data Forwarding Path

There may be nodes inside a network that are not on a new flow's (flow requesting admission) data forwarding path, but other flows' data is being relayed by those nodes. Therefore, it is necessary to determine the new flow's correct contention count on those nodes, otherwise end-to-end QoS requirements of admitted flows may be compromised.

Table 2.4.: Data Activity as Measured by Node G

Scenario	Mean Data Activity	95% Confidence Interval
1	10.20 kbps	10.05 - 10.35 kbps
2	20.36 kbps	20.06 - 20.66 kbps
3	31.90 kbps	31.25 - 32.55 kbps
4	42.95 kbps	42.25 - 43.65 kbps

The state-of-the-art flow admission control algorithms for ad-hoc wireless networks, e.g., [58], [45], and [73] do not take into account the correct contention count on nodes which are not on the new flow's data path. When a flow's admission request arrives at a node, typically, the contention on such nodes are only considered by determining the minimum available bandwidth within the interference range of a node (hereafter, in this chapter, we refer to this technique as locally estimating the contention count). Hence, the algorithm is assuming a contention count of 1 for all such nodes. This technique suffers from the following two problems.

- If a common node (node that is not on a new flow's data forwarding path) is within the interference range of more than one transmitter (nodes on the data forwarding path of a new flow), the contention count on the node is equal to the number of transmitters within the interference range of the node. Hence, the algorithm may wrongly admit a flow.
- Locally determining the contention count on nodes within the interference range of a transmitter can result in wrong admission decisions. Let us assume that a common node (i.e., a node that is not on a new flow's data forwarding path) is within the interference range of two transmitters and both transmitters relay data of two different flows. Let us further suppose that the admission request for both flows is initiated at the same time, moreover we assume that the available bandwidth at the common node is enough to accommodate a single contention count of anyone of the two flows. In this case, locally estimating the contention count will allow admission to both flows (considering the flow passes the intra-flow contention test), hence one of the two flows is wrongly admitted to the network.

We performed a number of simulations to show that the contention count on a node that is not on a flow's data forwarding path, but is within the interference range of transmitter(s) along the data forwarding path is a function of the number of transmitters within the interference range of the node. We added one more node in the network shown in Figure 2.2, and created 4 different simulation scenarios by changing the location of the additional node inside a network. In these simulations, node G measures the data activity using the wireless channel-sensing technique. Each simulation scenario is repeated 10 times. Figure 2.3 shows the modified network topologies for different simulation scenarios. Table 2.4 shows the mean data activity measured by node G during the duration of node A's flow.

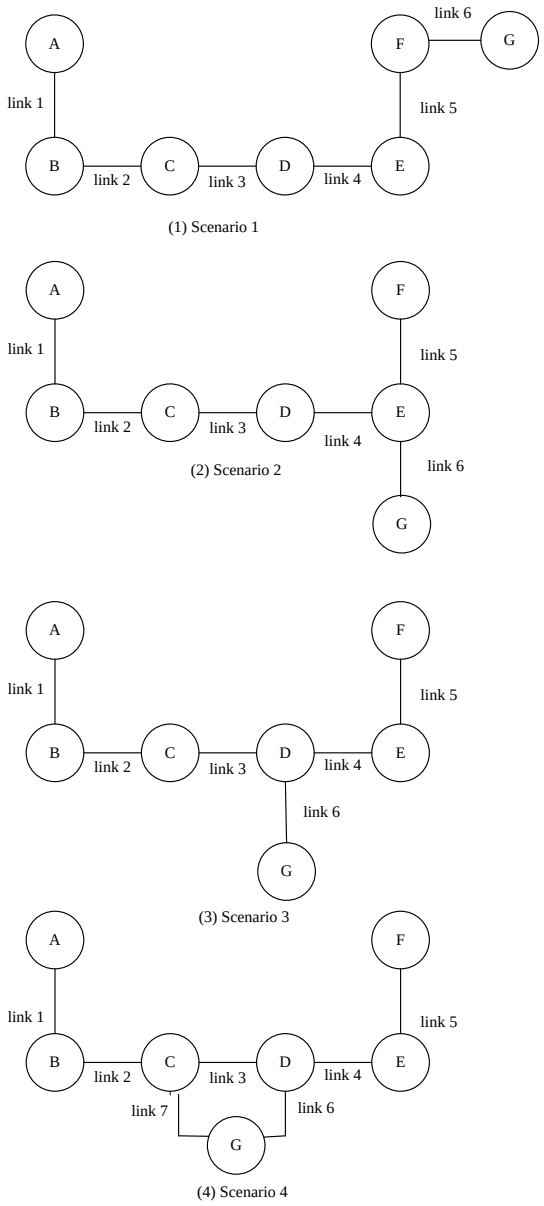


Figure 2.3.: Simulated Network Topologies

In Scenario 1 node G is within the interference range of one transmitter, i.e., node E, and Table 2.4 shows that the contention count at node G in this case is approximately 1. Similarly, in Scenarios 2, 3, and 4, node G is within the interference range of 2, 3, and 4 transmitters, hence the contention count at node G in these cases is 2, 3, and 4 respectively. Node A was transmitting at the rate of 10 kbps, and the end-to-end throughput was 10 kbps. The data activity measured by node G in different simulation scenarios is greater than the contention count at node G multiplied by the node A's flow data rate, because of retransmitted data packets.

2.5. Key Lessons

The following factors must be considered for a flow admission control algorithm for ad-hoc wireless networks.

- Determine the correct intra-flow contention factor.
- Determine the correct contention factor on nodes that are not on a data flow's forwarding path, but are within the interference range of transmitters along the data forwarding path.
- When a new admission request is received at a node, the node must proactively take into account the complete additional CSMA-CA MAC layer overhead with an increase in the data traffic load and the number of transmitters (due to the flow's admission) not only at the node, but also on nodes which are within the interference range of the node (if those nodes are transmitting data).

3. Related Work

Realizing the importance of QoS in the Internet, the Internet Engineering Task Force (IEFT) has standardized two QoS architectures, namely: Integrated Services (IntServ) [10], and Differentiated Services (DiffServ) [9]. IntServ is a complex architecture that aims to satisfy QoS requirements on a per-flow basis. IntServ reserves resources and maintains flows' state on the flows' data relaying nodes. The resource reservations are carried out using a two-pass signaling protocol called Resource Reservation Protocol (RSVP) [10]. DiffServ is designed to provide QoS provisioning in aggregate form, hence it defines various traffic classes. DiffServ defines forwarding behaviors corresponding to each defined traffic class, and ingress routers are responsible for enforcing the forwarding behaviors. Application data is mapped to a particular traffic class possibly depending on one or a combination of the following metrics: priority, application type, pricing policy, etc. Wireless sensor nodes are severely resource constrained devices in-terms of computational power, memory, and available power, secondly bandwidth is also a scarce resource in ad-hoc WSNs. Therefore, QoS provisioning in ad-hoc WSNs requires different solutions than the solutions designed for the Internet.

In most of the cases, the QoS requirements of the real-time multimedia application are satisfied using a flow admission control algorithm. The main purpose of the flow admission control algorithm is to restrict the amount of data inside a network in such a way that the QoS requirements of the admitted real-time multimedia applications are satisfied. Typically, the amount of available bandwidth is used as an input to a flow admission control algorithm [58] [45]. In a IEEE 802.15.4 ad-hoc network, the MAC layer decides the sharing of a communication medium, hence the amount of available bandwidth is impacted by the MAC layer protocol. Moreover, the state of nodes along a data forwarding path in terms of the available resources, e.g., the available bandwidth may impact the performance of a real-time multimedia application. As a routing protocol selects a data forwarding path, therefore the routing protocol has a pivotal role in satisfying the QoS requirements of a real-time multimedia application. In the remainder of this chapter, we review the following: (i) IEEE 802.15.4 unslotted CSMA-CA MAC layer protocol, (ii) state-of-the-art work on IEEE 802.15.4 CSMA-CA MAC layer protocol's throughput analysis, (iii) state-of-the-art available bandwidth estimation methods and available-bandwidth-based flow admission control algorithms for ad-hoc wireless networks, and (iv) QoS-based routing protocols for ad-hoc wireless networks in general, and for ad-hoc WMSNs in particular.

3.1. IEEE 802.15.4 Medium Access Control Protocol

The MAC layer decides the amount of bandwidth available to the applications running on a node [72]. Therefore, in this section, our main focus is to discuss the operations of MAC layer protocols standardized in the IEEE 802.15.4 specification.

The IEEE 802.15.4 standard defines two channel access methods, namely beacon enabled channel access and non-beacon enabled channel access. The beacon enabled channel access method requires a central coordinator called Personal Area Network (PAN) coordinator. The PAN coordinator periodically broadcasts a beacon frame to identify its PAN and to synchronize nodes associated with it. The time between two beacon intervals is partitioned into an active period, and optionally an inactive period. The length of an active period constitutes the superframe. In an inactive period nodes can switch to sleep mode to preserve energy. The active period is partitioned into 16 equally sized time slots. The active period consists of a contention access period and a contention-free period. In the contention access period, nodes contend for access to the channel using a slotted CSMA-CA protocol, and in the contention-free period nodes reserve time slots with the PAN coordinator, and transmit data during their assigned time slots. Non-beacon enabled mode is based on the contention access period and there is no contention-free period. No beacon frames are broadcasted, hence there is no need for a PAN coordinator. Non-beacon enabled mode uses an unslotted CSMA-CA MAC protocol for channel access.

In an unslotted CSMA-CA protocol, two variables affect the wireless channel access: BE; which is the current back-off exponent, and NB; a count of the number of back-offs. Before transmitting a data frame, a node waits for a random number of backoff slots ranging from 0 to $2^{BE} - 1$. Initially, BE is initialized to BE_{min} and its maximum value is BE_{max}. By default the value of BE_{min} is 3 and the value of BE_{max} is 5. Initially, the value of NB is set to 0. After waiting for some time, a node performs a Clear Channel Assessment (CCA) , and if the wireless channel is sensed idle, the node transmits. Otherwise a node backs-off after incrementing BE and NB by one unit. This process continues till a node successfully transmits a packet or NB reaches its maximum limit. Unslotted CSMA-CA can work in reliable and unreliable mode. In reliable mode, transmission of ACKs is supported to indicate successful delivery of data frames. In unreliable mode, transmission of ACKs is not supported.

Channel access methods supported by the IEEE 802.15.4 standard provide limited support for real-time applications in the form of a contention-free period. But the contention-free period only works in a star topology. Moreover, there is a serious scalability issue, i.e., there are up to seven Guaranteed Time Slots (GTSs) during each superframe duration, therefore only a limited number of nodes can consume these GTSs, and if there are additional nodes requiring GTSs, GTSs are denied to them. Moreover, if the application data rate is low compared to the length of a GTS, the additional time period goes unused.

3.2. IEEE 802.15.4's MAC Throughput Analysis

In [38] an analytical throughput analysis of the slotted CSMA-CA MAC layer protocol of IEEE 802.15.4 is presented. An analytical model typically requires simplifying assumptions to produce results. In real WSNs, these assumptions may not be true [26], hence such analysis may not accurately predict achievable throughput. Analytical models only consider the working of the IEEE 802.15.4 slotted CSMA-CA protocol, therefore such analytical models do not predict throughput from an application's perspective. I.e., an application normally runs over an operating system, and data transmitted by an application program traverse the networking protocol stack. Therefore, the operating system architecture and networking protocol stack overhead have an impact on a node's ability to transmit data. An operating system's design is affected by the resources available on the target hardware. Sensor nodes are severally resource-constraint devices, therefore a WSN operating system designer has to consider these limitations. For example memory available on a contemporary Tmote sky node is 10 KB, hence from the networking protocol perspective the operating system may allocate small buffers for the networking protocols [22]. Networking protocols are invariably implemented using timers and events, therefore the way events are handled in an operating system has an impact on a node's throughput. Therefore, if an analytical approach states that IEEE 802.15.4 slotted CSMA-CA can achieve a certain throughput, we cannot conclude that the stated throughput can be achieved by user applications. Hence, a good analysis must consider all the factors that limit the channel capacity.

Testbed-based throughput measurements of the IEEE 802.15.4 MAC layer are presented in [37] and [20]. Furthermore, [20] also presents simulation-based throughput measurements of the IEEE 802.15.4 MAC layer. These papers focus on overall channel capacity, and do not consider the node level throughput. The testbed results reported in [37] and [20] show that the upper limit on channel throughput is in the range of 35 to 40 kbps. The simulation results reported in [20] show that the maximum channel capacity is approximately 65 kbps. It is important to note that in [20] simulations were performed using NS 2.34, which does not capture the impact of an operating system on the channel capacity.

In [31] and [21], a throughput analysis of IEEE 802.15.4 with the Guaranteed Time Slot (GTS) algorithm is presented. The model for the IEEE 802.15.4 GTS algorithm is build with the help of OPNET Modeler [43] and simulations were performed using the same simulator. Both papers focus on maximum channel throughput. Again limitations imposed by the WSN operating systems are ignored, hence the reported maximum channel capacity may not be an accurate estimate from an application's perspective.

Most of work on the IEEE 802.15.4's MAC throughput analysis is focused on estimating the overall channel capacity. As far as we are aware, no work focuses on analyzing the impact of an operating system's networking protocol stack's implementation and offered load on a node's throughput and per-packet delay.

3.3. Available Bandwidth Estimation Methods and Flow Admission Control Algorithms

As of today, the available bandwidth estimation methods for computer networks can be categorized as active or passive methods. A discussion on improving QoS through bandwidth estimation can be found in [44]. A detailed discussion on bandwidth estimation methods, metrics, and tools is presented in [51]. A detailed survey of state-of-the-art flow admission control algorithms for ad-hoc wireless networks is presented in [33], and a detailed survey on a radio's link quality estimation is given in [8].

3.3.1. Active Bandwidth Estimation Methods

Packet pair probing [54] is an active bandwidth estimation method. A number of packet pairs are transmitted to the receiver and an estimate of the available bandwidth is derived through statistical methods such as the dispersion experienced by the packet pairs. To mitigate the effects of cross-traffic, multiple packet pairs are injected on the route between the source-destination pair.

SLoPS [29] uses a train of equal-sized packets to estimate the available bandwidth on the data forwarding path. The estimate of the available bandwidth is made using the one-way delay experienced by a stream of packets. The main idea is that a sender varies the transmission rate, and approximates the available bandwidth. When the transmission rate exceeds the available bandwidth at the bottleneck node, the queue at the bottleneck node quickly starts to fill up, thus delay increases and SLoPS uses the feedback to estimate the available bandwidth.

BNeck [75] is an active bandwidth estimation technique that combines the functionalities of measuring the link capacity and locating the bottleneck nodes. The source node estimates the bandwidth by measuring the time gap between two Internet Control Message Protocol (ICMP) packets. For estimating the available bandwidth, BNeck uses probe packets of different sizes.

For ad-hoc wireless networks, active bandwidth estimation is not an ideal technique due to the following reasons:

- Active bandwidth estimation techniques use probe packets to measure the available bandwidth between a given source-destination pair. When the number of source-destination pairs is large, the number of probe packets is large as well. This may require a substantial amount of bandwidth.
- Due to the time-varying nature of the wireless links, the wireless network topology is not as stable as the wired network topology. This requires active bandwidth estimation techniques to conduct bandwidth estimation at a higher frequency. This can result in extra bandwidth requirements.

3.3.2. Passive Bandwidth Estimation Methods and Flow Admission Control Algorithms

In [71], a contention-aware flow admission control for ad-hoc wireless networks (CACP) is presented. Flow admission control is performed based on the available bandwidth estimate. An estimate of the available bandwidth is provided through the wireless channel-sensing mechanism, and it considers back-off periods as idle periods. Here the assumption is that the back-off periods are negligible even if the channel is saturated. The algorithm considers both intra-flow and inter-flow contention counts in a distributed manner. The drawbacks associated with this scheme are: the impact of the MAC layer on the available bandwidth is not considered and the impact of the MAC layer overhead on the available bandwidth with an increased data traffic load inside a network is not considered.

In [73], a distributed flow admission control for assuring QoS in ad-hoc wireless networks is presented. It is claimed in [73] that before deciding about a flow's admission request both intra-flow and inter-flow contention have been taken into account. The authors of this work claim that the maximum intra-flow contention count on an intermediate node along the data forwarding path is 4. Inter-flow contention is considered by matching a flow's required bandwidth with the minimum available bandwidth within the interference range of a node deciding about the flow's admission request. An estimate of the available bandwidth is provided through a wireless channel-sensing mechanism that uses both the virtual and physical carrier sensing mechanisms of the IEEE 802.11 MAC layer protocol. The proposed flow admission control algorithm uses a two-pass signaling mechanism to reserve resources along the data forwarding path. The drawbacks associated with this technique are: incomplete inter-flow contention, i.e., each node only checks the minimum available bandwidth within its interference range. In some scenarios, the calculated intra-flow contention count will be wrong. Also, the impact of an increased MAC layer overhead with an increased data traffic load inside a network is not considered.

In [68], an analytical capacity estimation based flow admission control scheme for multi-hop wireless networks is presented. Each node uses an analytical model to decide about a flow's admission request. A node inside a network accepts a flow if λ_{new} is smaller than the available capacity. The incoming data packets arrival rate is calculated using the equation ($\lambda_{new} = \lambda + K\lambda_{flow}$). In the given equation λ represents the data packet arrival rates of all nodes within the transmission range of the node, λ_{flow} is a new flow's data arrival rate, and K is the contention count. This technique uses $K = 2$ for a source node, $K = 3$ for an intermediate node, and $K = 1$ for a destination node. All nodes processing a flow's admission request evaluate the given equation. The downsides of this scheme are: there are cases in which both the intra-flow and inter-flow contention count estimation will be wrong (given that the interference range of a node is greater than its transmission range), the mathematical model assumes a constant packet size, assumes that at the MAC layer the main factor

that affects the delay is retransmissions, and does not consider the impact of the increased MAC layer overhead with an increased data traffic load inside a network on the available bandwidth.

CapEst [30] is a measurement-based link capacity estimator for wireless networks. It monitors the service time of data packets at each link, and based on this measurement an estimate of a link's capacity is made, hence it is not a MAC layer specific method. CapEst does not consider the increased MAC layer overhead with an increased data load inside a network.

In [58], an available bandwidth-based flow admission control algorithm (ABE) for ad-hoc wireless networks is presented. An estimate of the available bandwidth is provided through the wireless channel-sensing mechanism considering both virtual and physical carrier sensing, and different types of IEEE 802.11 CSMA-CA MAC layer inter-frame spacings. It is argued in [58] that measuring the channel activity considering the time spent in virtual and physical carrier sensing, and different inter-frame spacing results in an overestimate of the available bandwidth. This happens due to the non-synchronization of sender and receiver nodes in an ad-hoc wireless network. Therefore, a mathematical model is presented that takes into account the collision probability to estimate the actual available bandwidth. Hence, it probabilistically takes into account the future back-off overhead through a mathematical model. In [58], nodes periodically broadcast control packets, called HELLO packets. The collision probability is derived from the number of HELLO messages a node has received over the number of HELLO packets the node expected to receive during the last measurement interval. The flow admission control algorithm uses one hop and two hop neighbor information to calculate the intra-flow contention, and the authors claim that the maximum intra-flow contention count on a node is 4. Inter-flow contention is taken into account by determining the minimum available bandwidth within the interference range of a node when deciding about a flow's admission request. The downsides of this technique are: with an increased data traffic load inside a network only additional back-off overhead is considered, additional retransmission and contention window overheads are ignored. The intra-flow contention count estimator does not always provide the right contention count, and the inter-flow contention count estimator is too simple as it only considers minimum available bandwidth within the interference range of a node. Finally, the collision probability is derived without considering the future data traffic load and the number of transmitters.

Retransmission-based Available Bandwidth Estimation (RABE) [45] is a probabilistic mathematical model used to consider the complete impact of the IEEE 802.11 CSMA-CA MAC layer on the available bandwidth. The drawback of this scheme are: it assumes a fixed packet size, the impact of the number of transmitters on the additional MAC layer overhead is not considered, and it does not include a flow admission control mechanism.

Table 3.1.: Evaluation of State-of-the-Art Available-Bandwidth-Based Flow Admission Control Algorithms for Ad-Hoc Wireless Networks

Algorithm	**MAC Layer Effects on the Available bandwidth**	**Intra-Flow Contention**	**Contention Non-Relaying Nodes**	**Add. MAC Layer Overhead**	**Add. MAC layer Overhead on Non-Relaying Nodes**
CACP [71]	No	Yes	Yes	No	No
Distributed Admission Control [73]	Yes	Partially correct	Partially correct	No	No
CapEst [30]	Yes	No	Partially correct	No	No
Analytical-Capacity-based [68]	Yes	Partially correct	Partially correct	No	No
ABE [58]	Yes	Partially correct	Partially correct	Partially correct	No
RABE [45]	Yes	No	No	Yes	No

Table 3.1 summarizes our evaluation of state-of-the-art available bandwidth-based flow admission control algorithms for ad-hoc wireless networks. Table 3.1 demonstrates that none of these algorithms take into account all the identified factors in Chapter 2. Hence, there is a need for an available bandwidth-based flow admission control algorithm for ad-hoc wireless networks that considers all these factors.

3.4. QoS-based Routing Protocols

There exist QoS-based routing protocols that select path(s) to the destination node based on metrics such as a flow's required bandwidth and/or delay. In general, QoS-based routing protocols for WSNs/WMSNs can be categorized into cost, reactive multi-path, delay and reliability, and opportunistic routing based protocols.

3.4.1. Cost-based Routing Protocols

Researchers have proposed cost-function-based QoS routing for WSNs. It involves collecting all or a subset of the following information about downstream nodes: buffer size, signal to noise ratio, residual energy, congestion status, power required to reach the downstream node, and delay. Using the gathered information, a node evaluates a cost function corresponding to each downstream node. The downstream node corresponding to the lowest value of the cost function is selected as the relaying node. To cope with the network topology changes, the cost function is re-evaluated after T time units. Such routing protocols can be grouped into the following categories.

- Classless multi-path routing
- Traffic class-based single path routing
- Traffic class-based multipath routing
- Hierarchical routing protocols
- Agent-based QoS routing

[50] falls under the category of a classless multi-path routing protocol. Multiple downstream nodes are selected as relaying nodes, depending on the value of a cost function (sorted in ascending order). Data packets are relayed on multiple paths in anticipation that this will result in less delay and balanced energy consumption.

QoS-based routing protocols [65] [17] [25] fall under the category of traffic class-based single path routing protocols. Such protocols define different traffic classes and a flow's data is mapped to a particular traffic class depending on the QoS requirements of the flow. A cost function is evaluated corresponding to each downstream node. The decision to select a downstream node as

a relaying node for a particular traffic class depends on the value of the cost function corresponding to the downstream node and the characteristics of the traffic class.

The routing protocol presented in [69] falls under the category of traffic class-based multipath routing. To meet the delay and the bandwidth requirements of real-time applications, the source node discovers multiple node-disjoint paths to the destination node. The protocol provides service differentiation by maintaining separate queues for real-time and non-real-time data. During the route construction phase, each node uses a cost function to determine the next hop for the route request message. Using this approach, a source node discovers N node-disjoint paths to the destination node. The source node reserves the l best node disjoint paths for real-time data and another m paths to forward non real-time data. Each real-time data packet is segmented into l equal-sized segments and an error correcting code is calculated for each segment. Afterwards, data is transferred to the sink node using the l node disjoint paths. Forward Error Correction (FEC) is used to reconstruct any missing data segment.

Hierarchical routing protocols partition the sensing field into various clusters. Each cluster acts as a multi-hop WSN. Each sensor node in the network associates itself with only one cluster head. It is possible that a sensor node reaches its cluster head in multi-hop fashion, hence each cluster is a multi-hop WSN. Within a cluster, downstream relaying nodes are selected depending on the type of data being transmitted and the value of a cost function. The QoS-based routing protocol for WMSNs presented in [34] is an example of a hierarchical routing protocol.

[40] falls under the category of agent-based QoS routing. Software agents are used to monitor changes in a network topology, network communication flow, and node's routing state. For maintaining good routes, a cost function is used to select a downstream forwarding node. Two types of agents are being implemented: forward agent and reverse agent. The forward agent is responsible to find suitable paths and the reserve agent enforces the path taken by the forward agent. The cost function values associated with multiple paths is fed to a Swarm-based optimization algorithm that helps to direct flows on paths that optimizes the utilization of the whole system.

The shortcoming of the cost-based routing protocols is that such protocols take locally optimal decisions. It is possible that a node locally selects the best next hop from the list of possible next hops, but none of the downstream nodes of the selected next hop have enough resources to satisfy a flow's QoS requirement.

3.4.2. Reactive Multi-path Routing Algorithms

QoS-based routing protocols for WSNs presented in [7] [6] can be categorized as reactive multipath algorithms. Reliability and delay are the QoS metrics used in these routing protocols. Packets are routed on multiple paths considering the reliability and delay requirements of a flow. Routes are discovered on demand

by flooding route request messages in a network. When the sink node receives multiple copies of the route request message through different paths, it selects a path depending on the application requirements i.e., if the application is delay sensitive, the path that offers less delay is selected and if the application demands reliability, the path that offers maximum reliability is selected (path selection is done from the set of available paths).

Reactive routing protocols discover a data forwarding path on demand. If delay is a metric of interest, such protocols try to discover a path that can offer minimum delay. Delay for a particular data path is measured by determining the difference of the following two time instances, the time when route request reached the destination node, and the time when the route request was transmitted by the source node. But, measuring path delay in such a manner can be misleading as data traffic on a particular path varies with time, and this can vary delay as well.

3.4.3. Delay and Reliability based Routing Protocols

Another class of routing protocols for WSNs tries to provide soft real-time guarantees in terms of delay [24] [27] [74]. At the source node data packets provide their end-to-end delay requirements and the source node calculates the required per-hop delay as $\frac{Delay_{total}}{Hop_{max}}$. $Delay_{total}$ represents the required end-to-end delay and Hop_{max} represents the number of hops between the source and the destination nodes. At each intermediate node, required per-hop delay is calculated as $\frac{\left(Dealy_{total} - Delay_{elasped}\right)}{Hop_{rem}}$. $Delay_{elasped}$ represents the delay that has already elapsed and Hop_{rem} represents the remaining number of hops. Such protocols assumes that the MAC layer provides service differentiation support, therefore data packets with an earlier deadline are mapped to a higher priority class. If an intermediate node concludes that it is impossible to meet the delay deadline of a data packet, this packet is dropped. The next-hop downstream node offering the minimum delay is selected as a relaying node.

The major shortcoming associated with this routing scheme is inefficient use of the available bandwidth and energy. Each node tries to make the optimal forwarding decision locally, i.e., selecting a best downstream node depending on the delay requirement of a flow. Typically, nodes close to the sink node(s) have to relay more data. Hence there is a possibility that nodes closer to the sink node(s) conclude that non of their downstream nodes can meet a data packet's delay requirements, therefore the data packet is dropped. The resources used to relay the data packet up to this point, where the data packets is dropped, are hence wasted.

3.4.4. Opportunistic Routing Protocols

Routing protocols presented in [62], [13], [60], and [61] fall in the category of opportunistic routing protocol. Before initiating a data packet's transmission,

Table 3.2.: QoS-based Routing Protocols Evaluation

Type	**Optimality Level**	**Route Setup Delay**	**Efficient Use of Resources**
Cost-based	Local	No	Not always
Reactive	End-to-End	Yes	In most cases
Delay and Reliability	Local	No	Not always
Opportunistic	Local	No	Not always

opportunistic routing protocols, in most cases, broadcast a Request to Send (RTS) packet. Ideally the RTS packet is successfully received by the direct neighbours of the broadcasting node. Depending on the metric(s) of interest, i.e., delay, energy, or distance, each node that has received the RTS packet calculates its priority. The node having the highest priority replies with the Clear to Send (CTS) packet. On the reception of the CTS packet, the node transmits the packet to the node replied with the CTS packet. CTS packets received from other neighboring nodes are ignored. If the data packet is not received successfully, the node retransmits packet to the same node. The shortcoming associated with the opportunistic routing protocols is exactly similar to the one we have mentioned for cost-based routing protocols.

Table 3.2 shows our evaluation of the different types of QoS-based routing protocols. As far as we are aware, there is not a single QoS-based routing protocol for ad-hoc wireless networks that selects the forwarding paths based on the following: global optimality level, no route setup delay, and efficient use of resources. As part of this research, we focus on designing a routing protocol that considers all of the identified factors.

4. IEEE 802.15.4's Unslotted MAC Layer Analysis

In this chapter, firstly we analyze the impact of the Contiki Opertaing Systems (OS) [1], and its CSMA-CA implementation on an IEEE 802.15.4's node's throughput, node's reception capability, and wireless channel utilization. The analysis is based on Contiki's Rime networking protocol stack [18], and the goal of study is to determine an upper bound for the stated metrics. Moreover, we present Contiki's implementation details that affect a node's throughput, and our modifications to Contiki's CSMA-CA implementation that result in enhanced throughput. Secondly, we experimentally derive IEEE 802.15.4's channel capacity for two cases, i.e., when the CSMA-CA protocol is working without ACKs, and when it is working with ACKs. Furthermore, for both cases, we plot the relationship of offered data load with delay and packet loss rate. Based on the relationship, we present the parameters that affect the choice of a CSMA-CA protocol for real-time multimedia applications.

4.1. Contiki Operating System

In this section, we discuss the design of the Contiki operating system for WSNs. The discussion in this section helps us to understand the Contiki features that limit the transmission capability of a node. Afterwards, we discuss Contiki's CSMA-CA implementation in detail, and we examine how it limits the transmission rate of a node. We conclude this section with a discussion of our modifications to the Contiki's CSMA-CA implementation to increase a node's transmission capability.

4.1.1. Contiki Overview

Contiki is a lightweight open source OS written in the C programming language for WSNs. It follows a modular architecture, and it is build around an event driven kernel. Contiki provides preemptive multitasking at the process level, but to yield the processor to another thread, the running thread has to invoke the yield function explicitly. In other words, if a running thread continues to run, waiting threads are not scheduled.

The Contiki kernel comprises of an event scheduler that dispatches events to the running processes. Process execution is triggered by events dispatched by the

kernel to the process or by a polling mechanism. When an event is dispatched to a process, it runs to completion, however, event handlers can use internal mechanisms for preemption. Contiki maintains a queue of pending events, and events are dispatched to target processes in a First In First Out (FIFO) manner. Interrupts can preempt an event handler, but to avoid synchronization issues, interrupts cannot post an event. To transmit a data packet, Contiki uses a callback timer. The callback timer takes an expiry time and a pointer to a function that acts as an event handler as arguments. When the timer expires, an event is stored in the event queue and the event handler is called eventually. If there are multiple events pending in an event queue, and events are fired in a FIFO manner, it is possible that the event handler for a callback timer does not execute right away. This phenomenon limits the transmission capability of a node. Furthermore, the communication stack overhead, e.g., copying a message and adding headers to a message adds further delay.

4.1.2. The Contiki 2.5 CSMA-CA MAC Layer

When a CSMA-CA MAC layer receives a packet for the upper layer, it enqueues the packet in a MAC layer buffer. If the packet received at the MAC layer is a broadcast packet, it is not enqueued, rather the MAC layer broadcasts it straight-away, without performing carrier sensing. In case of a unicast packet, if no space is available in the MAC layer queue, the unicast packet is treated as a broadcast packet, hence the packet is transmitted without performing carrier sensing. If a unicast packet is treated as a broadcast packet, it is not re-transmitted, in case of data packet collision or corruption.

Whenever the MAC layer has a packet to transmit, it delays carrier sensing by $1/8^{th}$ of a second, using the null radio duty cycling algorithm. Afterwards, it performs carrier sensing and if no carrier is detected, the packet is transmitted. If reliability mode is enabled, the MAC layer waits for a predefined interval of time to detect an ACK, in Contiki 2.5 this interval is 6 real-time ticks for Tmote sky motes. The real-time timer on a Tmote sky mote ticks 16,384 times per second. When an ACK is detected, the system waits for another 10 real-time ticks. If no ACK is detected in the stated time interval, the system backs-off for a random amount of time. The random back-off interval depends on the Channel Check Interval (CCI) used by the radio duty cycling algorithm, which is $1/8^{th}$ of a second for null radio duty cycling. If the MAC layer is about to transmit a packet and it senses that the channel is busy, it backs-off for a random amount of time, as stated above. The CSMA-CA makes three re-transmission attempts and if unsuccessful, the packet is dropped.

After every successful packet transmission, the CSMA-CA MAC layer with null duty cycling waits for $1/8^{th}$ of a second to transmit the next packet in the MAC layer queue. Therefore, CSMA-CA with null radio duty cycling can only transfer MAC_{tx-max} bps, where $MAC_{tx-max} = \sum_{i=1}^{8} FS_i \times 8$, and FS_i is the total size of the i^{th} frame in bytes. A transmission rate of MAX_{tx-max} bps is only possible if MAC layer ACKs are disabled, otherwise node throughput will

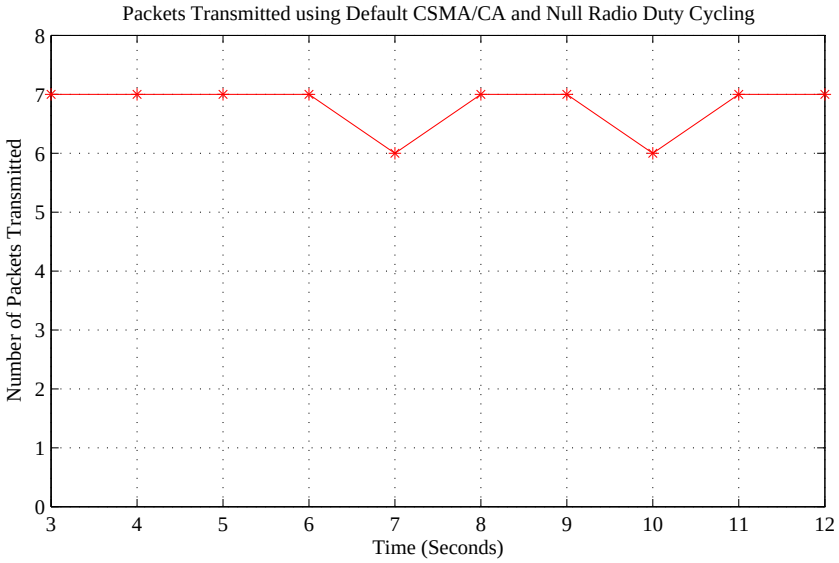


Figure 4.1.: Packets Transmitted Using CSMA-CA and Null Radio Duty Cycling Algorithm

further degrade.

We have performed simulations to validate that, using the default CSMA-CA implementation with null duty cycling, demonstrating that a node running Contiki can only transmit 8 kbps at maximum. Simulations were performed using the Cooja WSN simulator [49]. Two Tmote sky nodes are used; one node sends ten packets per second to the other node. The size of each data frame is 127 bytes, including headers, and the MAC layer queue can store 6 full size frames. Nodes are using Contiki's Rime communication stack with Cooja's Unit Disk Graph Model (UDGM). The two node WSN is simulated for fourteen seconds, and MAC layer ACKs are disabled. If the MAC layer queue is full, the MAC layer broadcasts packets without performing channel sensing. We drop such packets at the MAC layer because broadcasting a packet without performing channel sensing can cause interference, hence might lead to decreased network throughput.

Figure 4.1 shows the number of packets transmitted per unit time between three to twelve seconds of simulation time. The figure shows that a node can only transfer 7 packets per second. The size of each packet was 127 bytes, therefore the node's throughput in this case was 6.94 kbps. This confirms the fact that after every successful transmission a node waits for $1/8^{th}$ of a second. An OS has to add headers to the application data, schedule a packet transfer, and in case of CSMA, there is a need to perform clear channel assessment, therefore a node cannot exactly transfer eight packets per second. The important lesson we take away from this experiment is that a node's transmission ability is limited in terms of number of packets, using the default set up. We conclude that the delay between two successive transmissions along with the FIFO event dispatching mechanism, and events run to completion semantics, reduce node

level throughput.

4.1.3. Modifications to the Contiki Operating System

To enhance network throughput, we modified the time interval for which a CSMA-CA MAC layer waits to perform the clear channel assessment. For the null radio duty cycling algorithm, this time is calculated as $(1/CCI)$ seconds. The value of CCI, in case of null duty cycling is 8. The CSMA-CA MAC layer uses Contiki's callback timer (ctimer) mechanism to invoke the function responsible for performing the CSMA-CA MAC layer activities corresponding to a packet transmission. Our modification was to set the value of the callback timer to 0, so that, in case there is a packet in the MAC layer queue, Contiki's CSMA-CA immediately performs clear channel assessment, and transmits a packet if no carrier is detected, otherwise a node switches to the back-off mode. It is shown through simulations that our modification increases a node's throughput, hence nodes in a network can transmit more data compared to what nodes can transmit with Contiki's original CSMA-CA implementation. The rational for using a CCI value of 8 is to ensure fairness, i.e., reducing the rate at which nodes transmit leaves more bandwidth for other nodes. If our modifications are coupled with a flow admission control algorithm, we not only increase nodes' throughput, but also the flow admission control will impose its notion of fairness. Moreover, Contiki 2.5's feature of sending unicast packet straight away as broadcast packet, if the MAC layer queue is full, bypassing all other packets, seemed strange, therefore as mentioned before, we have disabled that.

Secondly, we are interested in measuring average per-packet delay at the MAC layer. Our system keeps a record of the time that each packet spends in the MAC layer queue. To determine the average per-packet delay per second, the total queuing delay for transmitted packets is divided by the number of packets transmitted per second.

We are interested in studying Contiki's CSMA-CA implementation with and without ACKs. In the latter case, it is possible that two or more nodes sense the wireless channel idle, hence they may start transmission at the same time. This results in a collision of the transmitted packets. If our simulator considers such packets as delivered, we would end up overestimating the channel capacity. Therefore, we added logic to Cooja's UDGM (our simulations use UDGM), to keep track of the total number of collided and corrupted data packets per second.

4.2. Experimental Results

In this section, we present theoretical as well as simulation-based analysis of the IEEE 802.15.4's unslotted MAC layer.

Table 4.1.: General Parameters for Simulation

Parameter Value	Value
MAC layer	CSMA-CA
MAC layer reliability	Disabled
Radio duty cycling algorithm	No duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 frames
Bit rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Simulated node type	Tmote sky

4.2.1. Theoretical Performance Limits of IEEE 802.15.4

The IEEE 802.15.4 WPAN standard working in the 2.4 GHz band supports a bit rate of 250 kbps, therefore channel capacity cannot go beyond 250 kbps. The IEEE 802.15.4 physical frame consists of 4 bytes of preamble, a one byte Start of Frame Delimiter (SFD), and a one byte frame length field. The maximum physical layer payload size is 127 bytes. Furthermore, it is reported in [48] that the throughput is further limited by a 192 μs turnaround time, and it is equivalent to six overhead bytes. A theoretical upper-bound on the single hop throughput is T_{ub}.

$$T_{ub} = \left(\frac{127}{(4+1+1+127+2+6)} \right) \times 250 \approx 225 kbps \tag{4.1}$$

4.2.2. Simulation-based Results

The simulations are performed on Cooja, the WSN simulator provided with the Contiki operating system. The general simulation parameters are shown in Table 4.1.

Upper-Bound on Node Throughput

To analyze the impact of the Contiki operating system and the networking protocol stack on the node throughput, we conducted simulations with different scenarios. We used two nodes; one node acts as a transmitter and the other acts as a receiver. The purpose of using two nodes with maximum sized packet is to get an upper bound on a node's throughput. The focus of these simulations is to determine how much data a single node can transfer with our modifications to the CSMA-CA protocol, in an ideal situation. In different simulation scenario, the application transmits 20, 30, 40, 50, and 60 packets per second and transmission continues till the transmitters transmits 200, 300, 400, 500,

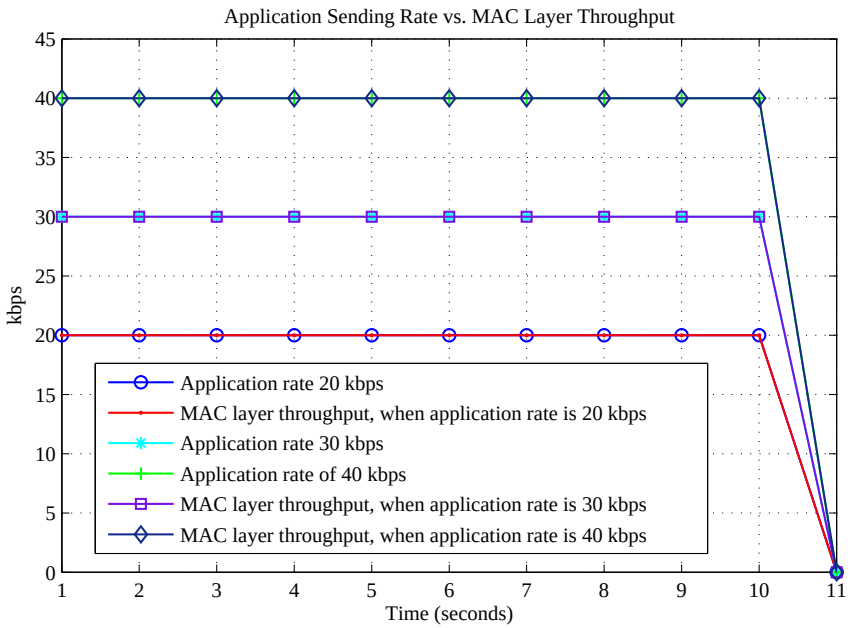


Figure 4.2.: Data Transmission Rate and MAC Layer Throughput

and 600 packets respectively. To enable uniform spacing between packet transmissions, we used Contiki's callback timer. Figure 4.2 shows a comparison of the application transmission rate and the MAC layer throughput for the three lower per-node packet rates.

In this case, MAC layer throughput matches application transmission rate. Furthermore from Figure 4.2, we can conclude that an application can send 40 packets per second using the callback timer, without lose of precision.

Figure 4.3 compares application transmission rate and the MAC layer throughput when the application tries to send 50 and 60 kbps data respectively. It can be seen, in both cases, that the application transmission rate does not increase beyond 41 kbps. We are using Contiki's callback timer mechanism to keep uniform time spacing between two successive data packet transmissions, therefore from Figure 4.3, we can conclude that at maximum, with no changes to the Contiki callback timer module, and with our system, a callback timer can fire 40 times in a second. The application transmission rate and the MAC layer throughput approximately remain the same, therefore in this scenario, it seems that sending data packets using the callback timer limits a node's transmission capability.

To determine whether the MAC layer throughput is limited by the callback timer's overhead, or Contiki's implementation and event handling mechanism limits a node's MAC layer throughput, we create another simulation scenario. In this simulation scenario, our aim is to transfer application data to the Contiki kernel in a burst mode. Therefore, to avoid packet drops at the MAC layer, we increased the MAC layer queue size so that it can store 45 frames. An appli-

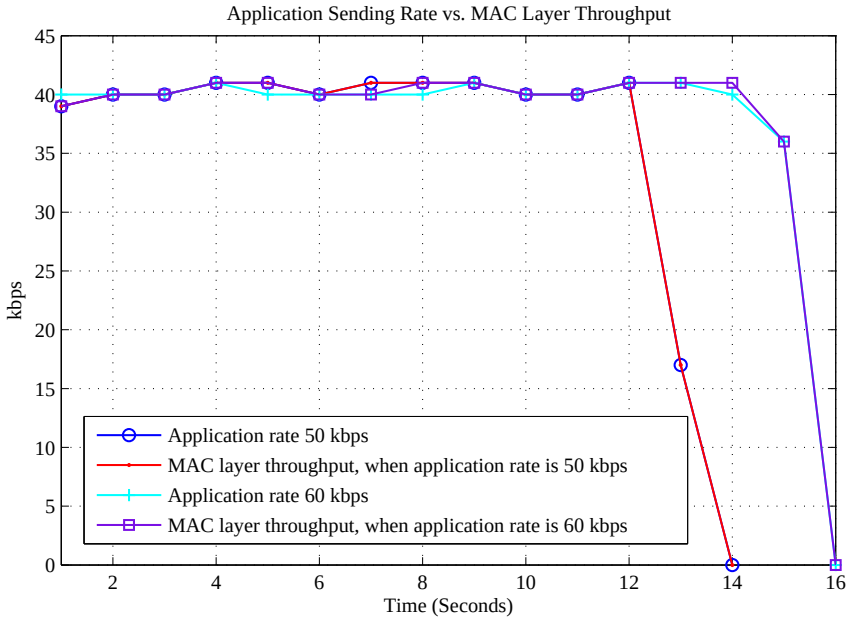


Figure 4.3.: Increased Data Transmission Rate and MAC Layer Throughput

cation transfers packets to the MAC layer in burst mode, i.e., if an application transmits 20 packets per second, it transfers 20 packets to the MAC layer in a loop, and then waits for a second to transmit the next 20 packets. When an application transmits 50 and 60 packets per second it transfers 25 and 30 packets instantly, and then waits for half a second to transfer the remaining 25 or 30 packets (to avoid packet loss at the MAC layer). Figure 4.4 shows a comparison of the application transmission rate, and the MAC layer throughput when the application is transferring data in burst mode.

Figure 4.4 depicts that the MAC layer throughput approximately remains the same as the application transmission rate, when the application transmission rate is 20 and 30 kbps. When the application transmission rate is increased to 40 kbps, the MAC layer throughput decreased and the delay increased as the MAC layer took 11 seconds to transfer packets that should have been transmitted in 10 seconds, had MAC layer throughput corresponded to the application rate. This result demonstrates that the bulk transfer at the rate of 40 kbps is worse than the uniformly spaced packet generation by the application at the same rate. We further increased application sending rate to 50 and 60 kbps and the results are shown in Figure 4.5.

Figure 4.5 depicts that an application can transfer 60 kbps of data to the MAC layer in burst mode without any problem. But the throughput at the MAC layer is 45 kbps, at maximum. Therefore we conclude that with our modification a node's throughout can reach 45 kbps at most in an ideal situation, which is approximately 600 percent better than what a node can achieve without our modifications, in an ideal situation. Furthermore, from these results we conclude that Contiki's callback timer mechanism does not have a substantial

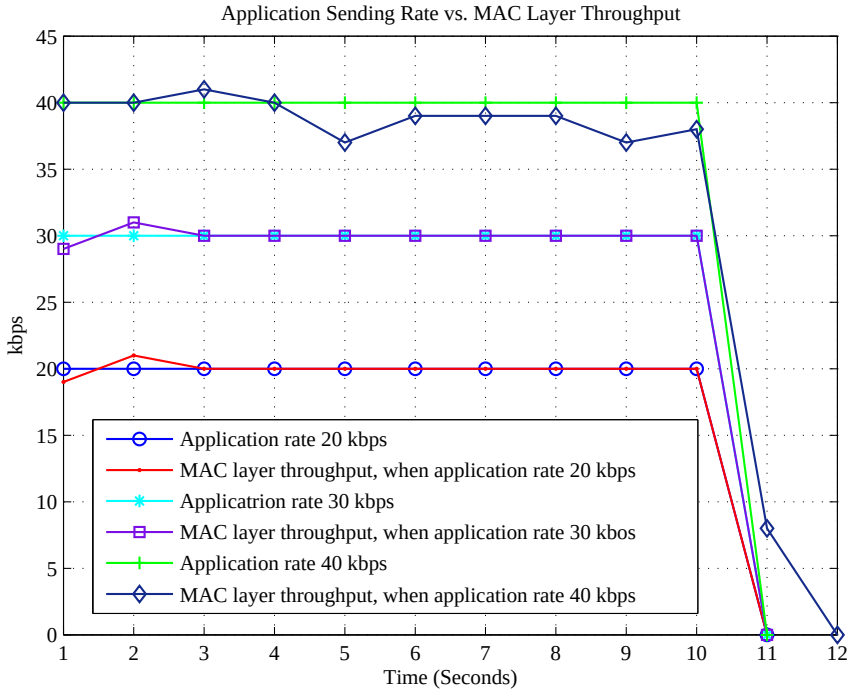


Figure 4.4.: Data Transmission Rate (Burst Mode) and MAC Layer Throughput

impact on node's throughput, rather it is Contiki's event handling mechanisms and its network stack implementation that limits a node's throughput, as discussed in Section 4.1.1.

IEEE 802.15.4 Channel Capacity Estimation

To estimate the IEEE 802.15.4-based WSN channel capacity, and relationship of delay and packet loss rate with offered traffic load, we simulated a WSN with eleven nodes. All nodes are within the transmission range of each other. Ten nodes act as transmitters and one node acts as a receiver. We increase the total offered data load in the network from 20 to 220 kbps (total data generation rate is uniformly distributed among 10 transmitters). Each simulation scenario is repeated three times, and averaged results are reported to account for the random nature of the CSMA-CA protocol. Table 4.1 lists general simulation parameters.

From Table 4.1 it can be seen that we are using full size IEEE 802.15.4 frames to estimate the IEEE 802.15.4 channel capacity. If we use short IEEE 802.15.4 addressing mode, 102 bytes of application data is carried in a MAC layer frame using the Rime communication stack of Contiki operating system. Typically, multimedia applications generate lots of data, therefore it is not uncommon that such applications utilize the maximum possible packet size of 102 bytes in

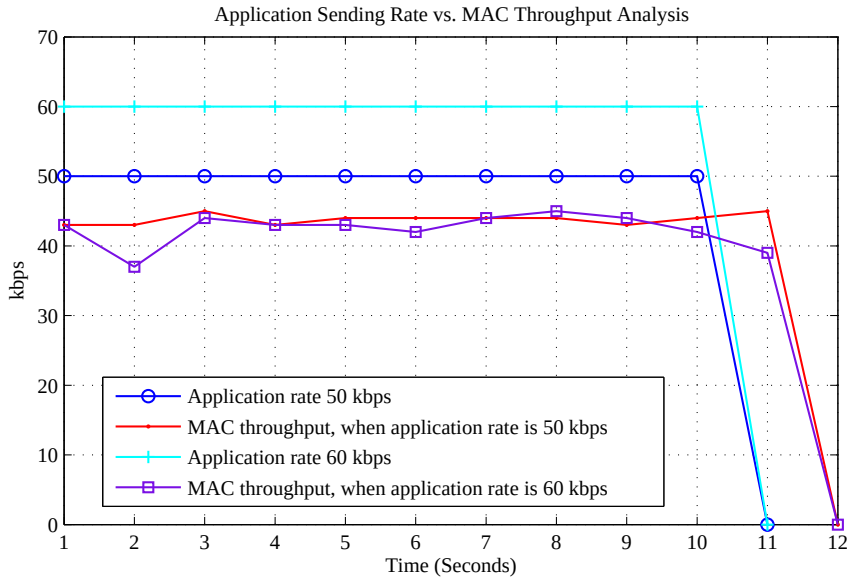


Figure 4.5.: Increased Data Transmission Rate (Burst Mode) and MAC Layer Throughput

each transmitted packet.

Figure 4.6 shows the average channel throughput w.r.t. the offered data load. In case of CSMA-CA without ACKs, the rate at which the channel throughput increases till the offered data load reaches 100 kbps is almost linear with offered data load. When the offered data load ranges from 100 to 180 kbps, the slope of the line showing average channel throughput increases more slowly. This is primarily due to the distributed nature of the CSMA-CA protocol. As data load in the WSN increases, each node has more data to send, hence nodes frequently contend for channel access and in this process nodes go to back-off mode more frequently. It results in an increased delay in getting access to the channel, hence channel throughput increases slowly. In fact, in simulation results we have observed that an offered data load of 180 kbps acts as a threshold point in terms of offered data load, after which channel throughput starts to decrease. This is primarily due to the CSMA-CA protocol and its back-off mechanism.

In case of CSMA-CA with ACKs, an interesting observation is that up to an offered data load of 60 kbps, packet drop rate is almost 0%. The primary reason for 0% packet drop rate till the offered data load is 60 kbps is that MAC layer ACKs are enabled, and if an ACK for a transmitted frame is not received, the frame is retransmitted. Moreover in this case, the channel throughput almost remains constant at approximately 93 kbps from an offered data load of 120 kbps to an offered data load of 200 kbps. The reason for this phenomenon is that the MAC layer retransmits lost/corrupted data frames. The MAC layer was not able to retransmit all lost frames due to the back-off and ACKs overhead. The channel is saturated at an offered load load of 220 kbps as the channel throughput drops to 86 kbps.

For an offered data load between 20 to 100 kbps the average channel throughput of CSMA-CA with ACKs is higher compared to the average channel throughput of CSMA-CA without ACKs. For an offered data load between 120 to 180 kbps, CSMA-CA without ACKs offers higher channel throughput, and lower packet loss rate. The higher packet loss rate, and decreased throughput in case of CSMA-CA with ACKs between an offered data load of 120 to 180 kbps is due to the following reasons: (a) Figure 4.6 shows that an increase in the offered load results in increased packet loss rate, therefore causing a higher number of retransmissions, and (b) an increased offered data load normally means more packets transmitted per second, hence more time a node has to wait for ACKs. Channel throughput results shown in Figure 4.6 show a step decrease in the channel's throughput for an offered data load in excess of 180 kbps, in case of CSMA-CA without ACKs. On the other hand, there is a gradual decrease in the channel's throughput for an offered data load of in excess of 200 kbps, in case of CSMA-CA with ACKs.

CSMA-CA with ACKs offers a packet loss rate of 0% as long as the offered data load is between 0 to 60 kbps. In all other cases, an increase in offered data load implies higher packet loss rate. Therefore, if the only parameter of interest for real-time multimedia application is strict reliability (low packet loss rate), CSMA-CA with ACKs is the only choice, and the system must limit the amount of data within the interference range of transmitters along the forwarding path to 60 kbps. A real-time multimedia application can tolerate end-to-end packet loss rate of 5% [58], in this case, CSMA-CA without ACK is only a feasible choice if total data load within the interference range of nodes along the forwarding paths is less than 30 kbps, and preferably there must be no more than one intermediate node between the source-destination pair. Assuming that end-to-end packet loss rate of 5% is the only requirement, and there are multiple intermediate nodes between source-destination pairs CSMA-CA with ACKs is the only choice, and total data load within the interference range of nodes along the forwarding paths must not exceed 60 kbps.

Figure 4.7 shows the relationship of delay with offered data load. In case of CSMA-CA without ACKs, it can be observed that average per-packet delay does not increase a lot as long as the offered data load ranges between 0 to 100 kbps. Beyond that point, average per-packet delay increases sharply. From Figure 4.6 and Figure 4.7, it can be concluded that the channel capacity in case of CSMA-CA without ACKs is 118 kbps, and it is achieved at an offered data load of 180 kbps. This confirms that operating below the bandwidth supported by the underlying communication technology results in congestion, hence increased delay and packet loss rate.

Figure 4.7 also shows the relationship of delay with offered data load when the MAC layer ACKs are enabled. It can be observed that the average per-packet delay in this case is much higher as compared to the average per-packet delay when the MAC layer ACKs were not enabled. From an offered data load of 60 to 120 kbps there is a sharp increase in the average per-packet delay, and the maximum channel capacity in this case is 94 kbps.

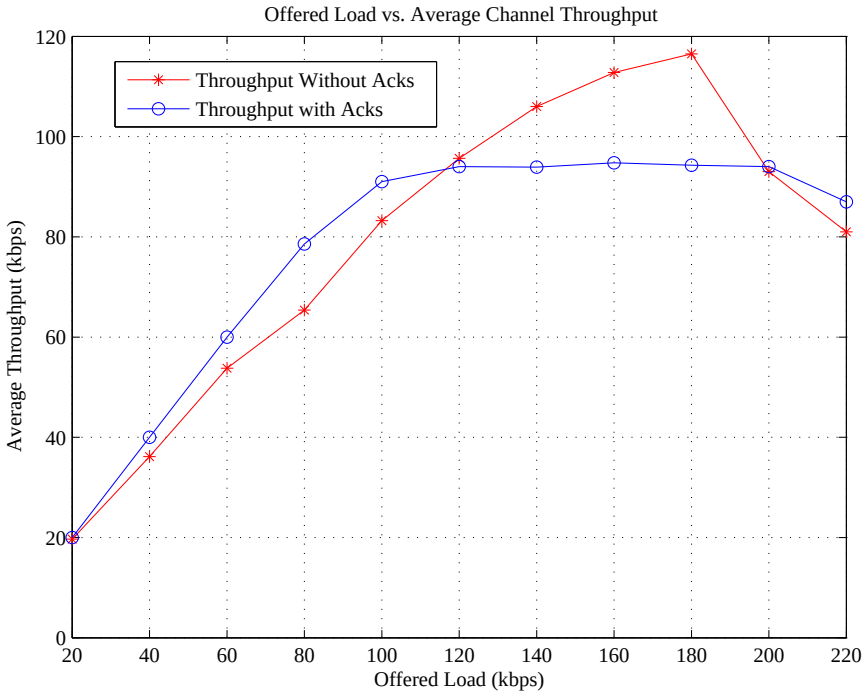


Figure 4.6.: Offered Data Load vs. Throughput

A real-time multimedia flow can tolerate end-to-end packet delay of 250-300 ms [46]. Figure 4.7 shows per-hop average per-packet delay, therefore end-to-end delay depends on the number of hops between the source and the destination node. If a real-time multimedia flow requires 5% end-to-end packet loss rate and end-to-end delay should remain within 250-300 ms, using CSMA-CA without ACKs is only possible if there is at most one relay node between the source and the destination node, and data load is limited to at most 30 kbps. In the same scenario, CSMA-CA with ACKs can fulfill a flow's requirement even if the source and the destination nodes are three hops away, and data load must be limited to 50 kbps. In general, the choice of a CSMA-CA protocol depends on an application's requirements, forwarding path's length, and the data load within the interference range of transmitters along the forwarding path.

4.3. Result Verification

In this section, we simulate a multi-hop WSN and we assume that nodes inside the network generate real-time multimedia flows. The main purpose of these simulations is to validate the conclusions we drew from the results presented in Section 4.2.2. We run separate simulations to validate the results for CSMA-CA without ACKs, and the results for CSMA-CA with ACKs. In each simulation scenario, average per-packet delay (total time spent by a packet in the MAC layer queue), and node's average throughput between the simulation

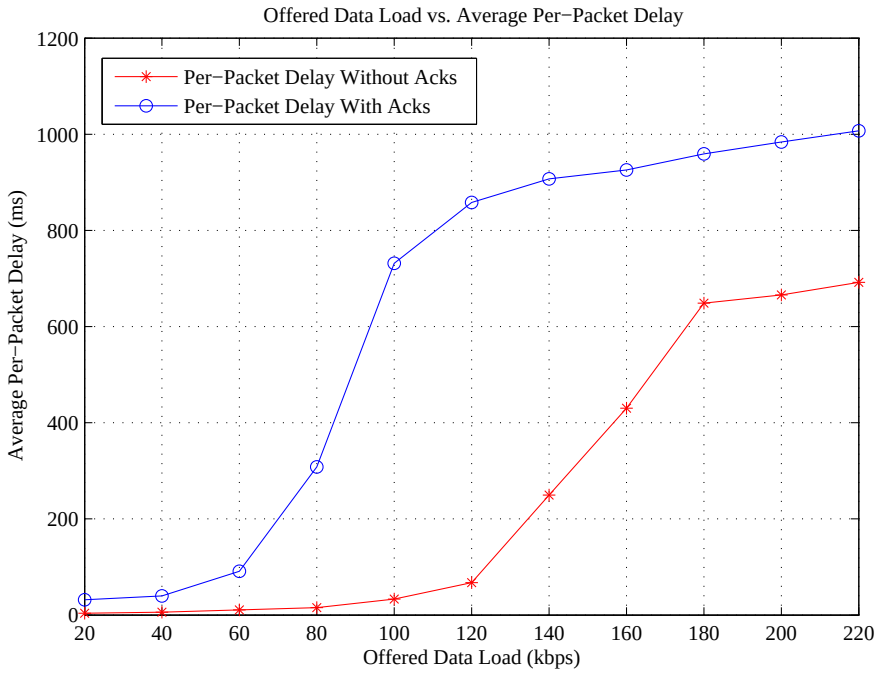


Figure 4.7.: Offered Data Load vs. Average Per-Packet Delay

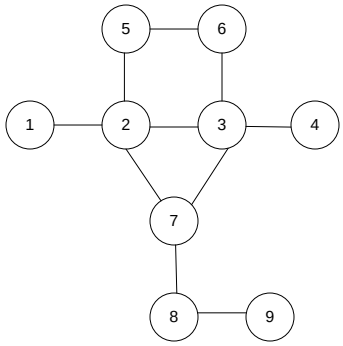


Figure 4.8.: Simulated Network Topology

time interval of 20 to 100 seconds are measured at each node. Figure 4.8 shows the simulated network topology. Table 4.1 lists general simulation parameter. Simulations are performed on the Cooja WSN simulator. Moreover, Table 4.2 shows nodes within the interference range of each node present in the simulated network.

4.3.1. Results Verification of the CSMA-CA Protocol without ACKs

The test-case to validate the results for CSMA-CA without ACKs assumes that a flow can tolerate the per-hop packet loss of up to 8% (which is significantly higher than the 5% end-to-end packet loss we stipulated above), but requires bounded delay (per-hop packet delay less than or equal to 30 ms). As per the

Table 4.2.: Nodes' Interference Set

Node ID	Interfering Nodes
1	1, 2, 3, 5, 7
2	1, 2, 3, 4, 5, 6, 7, 8
3	1, 2, 3, 4, 5, 6, 7, 8
4	2, 3, 4, 5, 6, 7
5	1, 2, 3, 4, 5, 6, 7
6	2, 3, 4, 5, 6, 7
7	1, 2, 3, 4, 5, 6, 7, 8, 9
8	2, 3, 7, 8, 9
9	7, 8, 9

Table 4.3.: Simulation Scenario 1

Flow ID	Source Node	Destination Node	Start Time (Sec)	Pkts/Sec	Total Packets to Transmit
A	1	4	4	10	1000
B	7	9	10	10	1000

experimental result shown in Figure 4.6 and Figure 4.7, offered data load inside the network should not exceed 60 kbps to provide an acceptable level of service. To validate the results presented in Figure 4.6 and Figure 4.7, we created three simulation scenarios. Each simulation scenario is repeated three times, and we report average results in this section.

Scenario 1

Table 4.3 summarizes the flows in Scenario 1. It is evident from Figure 4.9 that the average per-packet delay at each node is less than 30 ms corresponding to Scenario 1. Data loads within the interference range of nodes 1, 2, 3, 4, 5, 6, 7, 8 and 9 are 40, 50, 50, 30, 40, 30, 50, 40, and 20 kbps respectively. Figure 4.6 does not plot average throughput corresponding to the offered data load of 30 kbps. Considering Figure 4.6 it can be observed that the average throughput for an offered data load between 20 and 40 kbps increases almost linear. Therefore, we can use Equation 4.2 to estimate packet loss rate at an offered data load of 30 kbps via linear interpolation.

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \tag{4.2}$$

In this case, $x_1 = 20, x_2 = 40, y_1 = 19.7$, and $y_2 = 38$. Solving Equation 4.2 for the given values yields: $y = 0.915x + 1.4$. The average throughput for an offered data load of 30 kbps is 28.85 kbps, therefore per-hop packet drop rate is 3.3 percent. Similarly, Figure 4.6 does not plot average throughput for an offered data load of 50 kbps, but we can solve Equation 4.2 using the closest two data points: $x_1 = 40, x_2 = 60, y_1 = 38$, and $y_2 = 55$. Solving Equation 4.2 for these values yields: $y = 0.85x + 4$. Therefore, the estimated per-hop packet drop rate

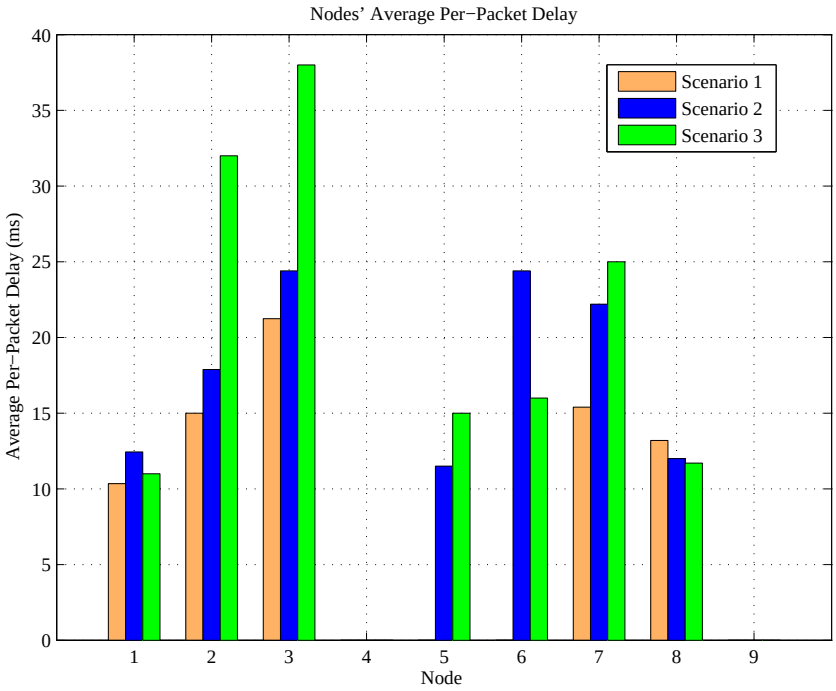


Figure 4.9.: Nodes' Average Per-Hop Packet Delay

is 7% for an offered data load of 50 kbps. The average number of bits that node 4 anticipates to receive as per the result present in Figure 4.6 can be calculated as: $(10 \times 0.95 \times 0.93 \times 0.93) = 8.21$ kbps. Simulation results show that node 4 received 9.2 kbps. Hence, the per-hop packet loss rate for flow A is certainly below 8 percent. Similarly node 9 must receive $(10 \times 0.95 \times 0.93) = 8.83$ kbps. Simulation results show that the average number of bits received by node 9 is 9.80 kbps. Therefore, the requirements in-term of delay and per-hop packet loss rate are met for both flows.

Scenario 2

Table 4.4 summarizes the flows in simulation Scenario 2. This scenario has the same two flows as scenario 1, with an additional flow from node 5 to node 4 (flow C), sending data at a rate of 5 kbps. In this scenario, the maximum data loads that can be ideally observed (i.e., if no packets were lost) at nodes 1, 2, 3, 4, 5, 6, 7, 8, and 9 are 45, 60, 60, 40, 50, 40, 60, 40, and 20 kbps respectively. Data load within the interference range of nodes 2, 3, and 7 is maximum, i.e., 60 kbps w.r.t. the chosen threshold level. In our experiments the average number of bits received at node 4 is 12.96 kbps. Node 4 receives 8.24 kbps for the flow A, and 4.72 kbps for flow C. From Figure 4.6 we can see that at the offered data load of 40 kbps, the packet drop rate is approximately 5 percent. Figure 4.6 does not plot average throughput for the offered data load of 45 and 50 kbps, but again we can use Equation 4.2 to estimate average packet loss at an offered

Table 4.4.: Simulation Scenario 2

Flow ID	Source Node	Destination Node	Start Time (Sec)	Pkts/Sec	Total Packets to Transmit
A	1	4	4	10	1000
B	7	9	10	10	1000
C	5	4	15	5	500

data load of 45 and 50 kbps, as done before.

The estimate of the average throughput for an offered data load of 45 kbps is 42.25 kbps, therefore in this case, we estimate that the per-hop packet drop rate is 6 percent. The per-hop packet drop rate for the offered data load of 50 kbps is 7 percent. If we use these derived values, the average number of bits that node 4 expects to receive for flow A is: $(10 \times 0.94 \times 0.92 \times 0.92) = 7.66$ kbps, which is lower than 8.24 kbps, and the average per-packet delay at nodes 1, 2, and 3 is less than 30 ms as shown in Figure 4.9, hence the QoS requirements of flow A are met. Similarly, node 4 should at least receive $(5 \times 0.93 \times 0.95) = 4.42$ kbps for flow C. But our results show that node 4 has received 4.72 kbps, which is more than 4.42 kbps. Moreover, Figure 4.9 shows that the average per-packet delay at nodes 5 and 6 is less than 30 ms. Hence, the requirements of flow C are also fulfilled. Node 9 is the destination of flow B, and as per our simulation results, it has on average received 9.80 kbps. Considering the results presented in Figure 4.6 and the data load within the interference range of nodes 7 and 8, node 9 should have received 8.74 kbps on an average. In this case, node 9 has received 12 percent more data compared to the results presented in Figure 4.6. The average per-packet delay as per Figure 4.9 is less than 30 ms at nodes 7 and 8, hence the requirements of flow B are also met.

In this scenario, the average number of bits received at destination nodes is more than what is anticipated. In both scenarios we have observed that the packet drop rate is lower than what is shown in Figure 4.6. A plausible explanation is that we derived the expected number of received bits based on the assumption that the offered load is equal to the sum of all flows without any packet loss. In reality, as soon as a packet is dropped, the offered load is reduced, resulting in a lower offered load and hence a lower packet loss rate, increasing the actual observed packet delivery rate. One can consider this as a positive development, because requirements of real-time multimedia flows are fulfilled. But at the same time, one can argue that our threshold of 60 kbps is overly conservative, hence we are missing out on opportunities to admit more flows in a network. Therefore, to analyze the impact of operating marginally above the threshold, let us consider Scenario 3.

Scenario 3

Table 4.5 summarizes the flows in simulation scenario 3. This scenario further extends scenario 2 by increasing the data rate of flow C so that some nodes experience an offered load (as the sum of the transmission rates of all nodes in the interference range) above the 60 kbps threshold. In this scenario, the

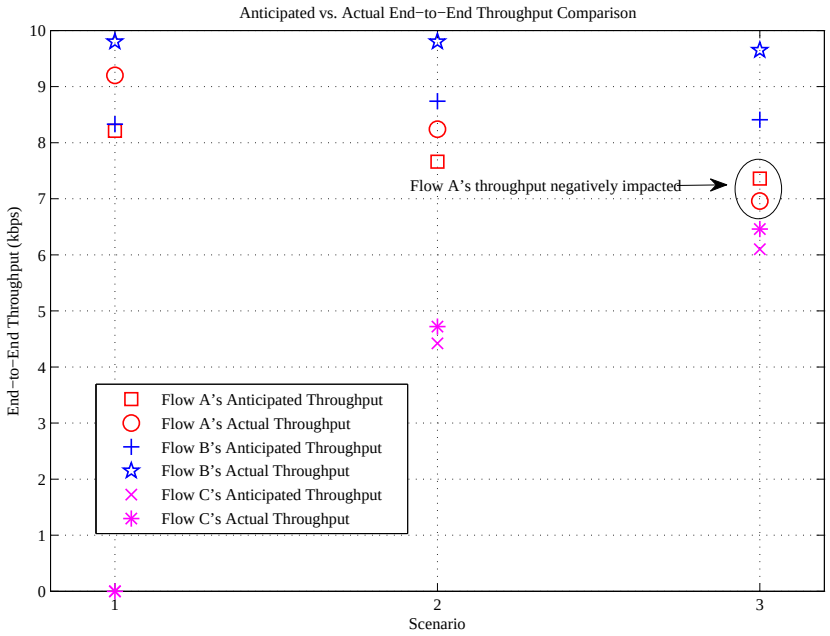


Figure 4.10.: Anticipated vs. Received End-to-End Throughput

Table 4.5.: Simulation Scenario 3

Flow ID	Source Node	Destination Node	Start Time (Sec)	Pkts/Sec	Total Packets to Transmit
A	1	4	4	10	1000
B	7	9	10	10	1000
C	5	4	15	7	700

maximum data loads that can be observed at nodes 1, 2, 3, 4, 5, 6, 7, 8, and 9 are 47, 64, 64, 44, 54, 44, 64, 40, and 20 kbps respectively. For simplicity we assume that the packet loss rate corresponding to the offered data load of 44, 47, 54, and 64 kbps is the same as for the offered data load of 45, 45, 55, and 65 kbps respectively. The per-hop packet drop rate corresponding to the offered data load of 45 kbps is 6 percent, as derived in Scenario 2. We need to solve $y = 0.85x + 4$ to estimate the per-hop packet loss rate corresponding to the offered data load of 55 kbps, which is 7.72 percent. Similarly, for the offered data load of 65 kbps we need to solve Equation 4.2 with values: $x_1 = 60, x_2 = 80, y_1 = 55$, and $y_2 = 65$ resulting in $y = 0.5x + 25$. Hence, the per-hop packet loss rate when the offered data load is 65 kbps is 11.5 percent. Node 4 has received 13.41 kbps, 6.96 kbps for flow A and 6.46 kbps for flow C.

The average number of bits that node 4 should expect to receive for flow A is $(10 \times 0.94 \times 0.885 \times 0.885) = 7.36$ kbps. Similarly, as per Figure 4.6, the average number of bits that node 4 expects to receive for flow C is $(7 \times 0.923 \times 0.94) = 6.10$ kbps. Our results show flow A has suffered more packet loss, moreover Figure 4.9 shows that the average per-packet delay at nodes 2 and 3 has significantly increased and now exceeds the target of 30 ms per hop. Therefore, in this case the flow A experiences degradation in its performance. Node 9 can

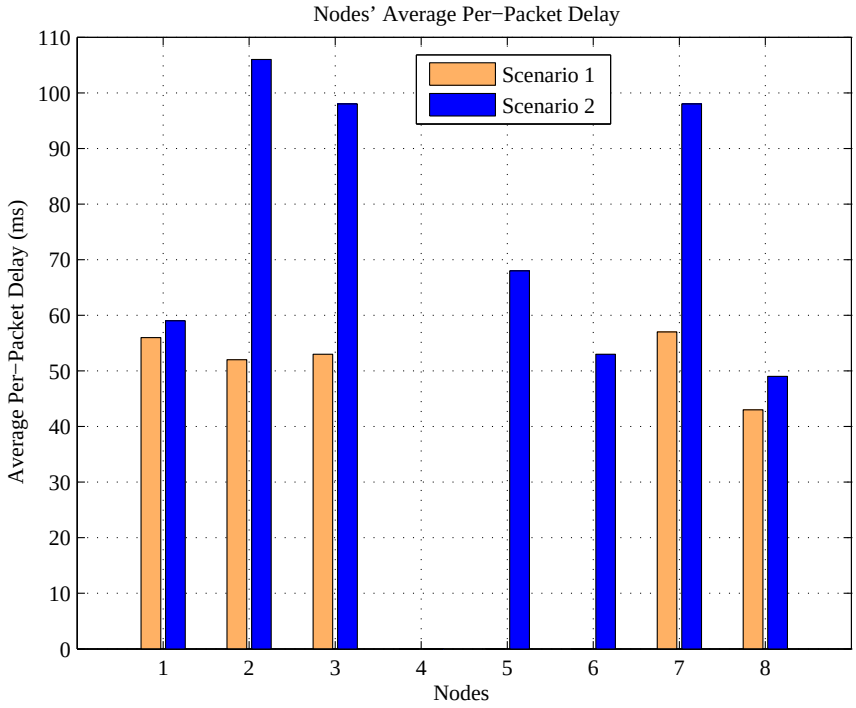


Figure 4.11.: Nodes' Average Per-Hop Delay (ACKs Mode)

expect to receive $(10 \times 0.885 \times 0.95) = 8.41$ kbps, but in simulation node 9 receives 9.65 kbps, which is 15 percent more than what is expected. Nevertheless, we have shown that exceeding 60 kbps deteriorates the performance of at least one real-time multimedia flow. Furthermore, Figure 4.9 shows increased delay especially at nodes 2 and 3 which further supports the tightness of the chosen threshold. Figure 4.10 shows the comparison of flows' anticipated and actual throughput.

4.3.2. Results Verification of the CSMA-CA Protocol with ACKs

The test-case to validate the results for CSMA-CA with ACKs assumes that a flow cannot tolerate packet loss, and it requires per-hop packet delay of less than 70 ms. As per the experimental results shown in Figure 4.6 and Figure 4.7, offered data load inside the network should not exceed 50 kbps to provide an acceptable level of service. We created two simulation scenarios. Each simulation scenario is repeated three times, and we report average results in this section.

Scenario 1

Table 4.3 summarizes the flows in Scenario 1. It is evident from Figure 4.11 that the average per-packet delay at each node is less than 70 ms corresponding

to Scenario 1. Data loads within the interference range of nodes 1, 2, 3, 4, 5, 6, 7, 8 and 9 are 40, 50, 50, 30, 40, 30, 50, 40, and 20 kbps respectively. As per Figure 4.6 the per-hop packet drop rate is 0% till the data load exceeds 60 kbps. In our simulation results, the end-to-end throughput of both flows is 100%. Therefore, the results corresponding to the first simulation scenario validates our determined statistics for the wireless channel under study.

Scenario 2

Table 4.4 summarizes the flows in simulation Scenario 2. In this scenario, the maximum data loads that can be ideally observed at nodes 1, 2, 3, 4, 5, 6, 7, 8, and 9 are 45, 60, 60, 40, 50, 40, 60, 40, and 20 kbps respectively. Data load within the interference range of nodes 2, 3, and 7 is 60 kbps, and it is above the chosen threshold. In our experiments the end-to-end throughput of all three flows is perfect, and it is in accordance with the results presented in Figure 4.6. The data load within the interference range of nodes 2, 3, and 7 is 60 kbps which is above the chosen threshold, and if we consider per-hop packet delay results presented in Figure 4.11, it can be observed that per-packet delay at these nodes is in excess of 70 ms. Hence, simulation results for this scenario demonstrates that our determined thresholds are tight.

4.4. Conclusions

The Contiki OS limits a node's transmission capability, primarily due to its event handling mechanism and implementation of the networking protocol stack. We experimentally derived an upper bound on a node's transmission capability and a wireless channel utilization using Contiki's implementation of the IEEE 802.15.4's unslotted CSMA-CA MAC layer protocol. Furthermore, we showed the relationship of offered data load with the per-hop packet delay, and the per-hop packet loss rate. Based on the experimatal results presented in this chapter, we demonstrated that the CSMA-CA protocol without ACKs offers lower end-to-end delay compared to the CSMA-CA protocol with ACKs. Furthermore, we showed that the CSMA-CA protocol with ACKs offers 0% packet loss rate if the data load within the interference range of a node is within 60 kbps. If the data load within the interference range of a node is below 120 kbps, the CSMA-CA protocol with ACKs offers better throughput as compared to the CSMA-CA protocol without ACKs. The CSMA-CA protocol without ACKs achieves better throughput if the data load is between 120 to 160 kbps. We conclude that, in most cases, the CSMA-CA protocol with ACKs demonstrates better end-to-end throughput, but the choice of a suitable IEEE 802.15.4 CSMA-CA MAC layer protocol depends on the requirements of a real-time multimedia flow, data load within the interference range of transmitters along the forwarding path, and the length of the data forwarding path. The experimental results presented in this chapter, can help to derive tight bounds on the offered data load, if the path length and QoS requirements of a flow are know. Hence, the relationship

of offered data load with the packet loss rate and the delay can be a useful information for a flow admission control.

5. Proactive Available Bandwidth Estimation

Estimating the available bandwidth in IEEE 802.15.4-based ad-hoc networks is a difficult and challenging task due to the shared nature of the wireless communication medium. Moreover, the MAC layer decides the sharing of a communication medium, therefore the MAC layer dictates the amount of bandwidth available to a node. Some recent solutions consider the partial impact of the MAC layer due to collision and transmitter and receiver non-synchronization on the available bandwidth, Available Bandwidth Estimation (ABE) [58] is one such example. ABE does not pro-actively consider the complete impact of the unslotted CSMA-CA MAC layer protocol on the available bandwidth, with an increased data load inside a network. Therefore the amount of reported available bandwidth is not fully available to a node, hence this can result in poor admission decisions.

For estimating the available bandwidth, the proactive bandwidth estimation mechanism considers the impact of back-off on the available bandwidth due to collision and sender-receiver non-synchronization, similar to ABE. Moreover, our proactive bandwidth estimation mechanism considers the impact of the contention window size on the available bandwidth, which is again similar to ABE. The major difference is that instead of predicting the impact of back-off and contention window size on the available bandwidth using a mathematical model and existing data traffic load inside a network, the proactive bandwidth estimation uses an empirical approach to estimate the impact of back-off and contention window (hereafter in this chapter, we refer to back-off and contention window size as the MAC layer overhead) on the available bandwidth using anticipated future data load inside the network. Therefore, the proactive bandwidth estimation minimizes the estimation error in ABE as reported in [45].

5.1. Bandwidth Estimation Module

The bandwidth estimation module obtains the MAC layer overhead information directly from the MAC layer, and for this purpose we have modified the Contiki operating system's CSMA-CA implementation to keep track of the real MAC layer overhead per unit time. Periodically the bandwidth estimation module retrieves the MAC layer overhead information from the MAC layer and stores this information in its internal data structure. In our current implementation,

we choose a reporting period of 1 second. To estimate data activity within the interference range of a node, the node monitors the wireless channel. The MAC layer overhead is measured in time. The estimate of the available bandwidth is made in bits per second (bps). Therefore, the MAC layer overhead is converted to bps by multiplying the overhead with the channel rate. The random nature of the wireless communication medium can result in significant deviations in the available bandwidth per unit time. To address this issue, we used a window-based averaging mechanism to estimate the available bandwidth, using a window size of α. At each bandwidth measurement instance, the available bandwidth is calculated by obtaining average overhead and the result is subtracted from the channel rate. Nodes inside a network inform their neighbors about their and their neighbors' available bandwidth using control messages.

5.2. Estimating Additional MAC Layer Overhead

An increased data load inside a network may increase the CSMA-CA MAC layer overhead. Therefore, an effective bandwidth estimation/flow admission control algorithm must take into account the extra MAC layer overhead associated with an increased data load inside a network. To consider the impact of the MAC layer overhead two methodologies can be used: analytical modeling and estimating the MAC layer overhead through empirical methods.

Analytical modeling makes simplified assumption about traffic generation pattern, error rates, and node synchronization. Therefore, analytical methods based on such assumptions can only give correct estimates if these assumptions hold true in real scenarios, which is not always the case.

An empirical approach involves setting up a network and measuring the MAC layer overhead with different values of the offered data loads. One can run such experiments before the network becomes operational, and such experiments can be repeated multiple times to obtain an average estimate of the MAC layer overheads for a certain data load inside a network. There are other parameters such as packet size, nature of traffic (burst, constant bit rate), and number of flows that can have an impact on the MAC layer overhead. We also vary the packet size parameter in the simulation section, and we expect that, after considering the aggregate data rate, other factors will have little impact. The advantage associated with the empirical method is that one does not need to make assumptions, as the estimate is made considering the real network conditions. The overhead associated with the empirical method is that a lookup table is stored on nodes that provide the estimated MAC layer overhead corresponding to a certain offered data load inside a network. It is not possible to store the MAC layer overhead in terms of bps corresponding to each possible offered data load, but an algorithm can estimate the MAC layer overheads for an offered data load not present in the lookup table by linear interpolation, using the two closest available data points.

To estimate average back-off overhead and standard error, we conducted mul-

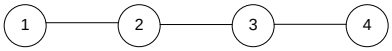


Figure 5.1.: Network Topology

Table 5.1.: General Simulation Parameters

Parameter	**Value**
MAC layer	CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 packets
Channel rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Simulated node type	Tmote sky

tiple simulations with different offered data loads. Simulations are performed using the Cooja WSN simulator. For simulations we choose the network topology shown in Figure 5.1, because it captures the effects of intra-flow contention and at the same time it is simple enough to measure back-off overhead at each node present in the network. Ten different simulations scenarios were created, and each scenario corresponds to a different offered data load inside a network. In the first simulation scenario, node 1 transfers 2 kbps to node 4, hence total data rate within the interference range of nodes 1, 2, and 3 is 6 kbps. In subsequent simulation scenarios, node 1 increases its data rate in such a manner that it increments offered data load to 12, 18, 24, 30, 36, 42, 48, 54, and 60 kbps within the interference range of nodes 1, 2, and 3. Each simulation runs for 110 seconds, and to collect average values, each simulation is repeated 10 times. Table 5.1 lists our general simulation parameters.

Figure 5.2 shows average back-off overhead and the measured standard error. It shows that with a substantial increase in the data traffic load inside a network, average back-off overhead increases, therefore it is essential to consider this overhead pro-actively, i.e., before admitting a new flow. In Contiki's IEEE 802.15.4 implementation, the contention window size remains constant, i.e., whenever the MAC layer has to transfer a new MAC layer frame, it defers the data frame's transmission for a constant amount of time. Once the contention window timer expires, a node performs clear channel assessment. The frame is transmitted if no carrier is detected, otherwise the node enters into exponential back-off mode. Therefore, the overhead associated with the contention window is a function of packets transferred per unit time. If CW_{size} is the size of the contention window, and on average a node transmits AN_{tx} packets per second, the contention window overhead is calculated using Equation 5.1. $ACW_{overhead}$ represents the average contention window overhead. In our experiments, we simulated Tmote sky motes, and the size of the contention window is equivalent

to 7812 bps with a channel rate (ρ) of 250 kbps.

$$ACW_{overhead} = \rho \times (CW_{size} \times AN_{tx}) \tag{5.1}$$

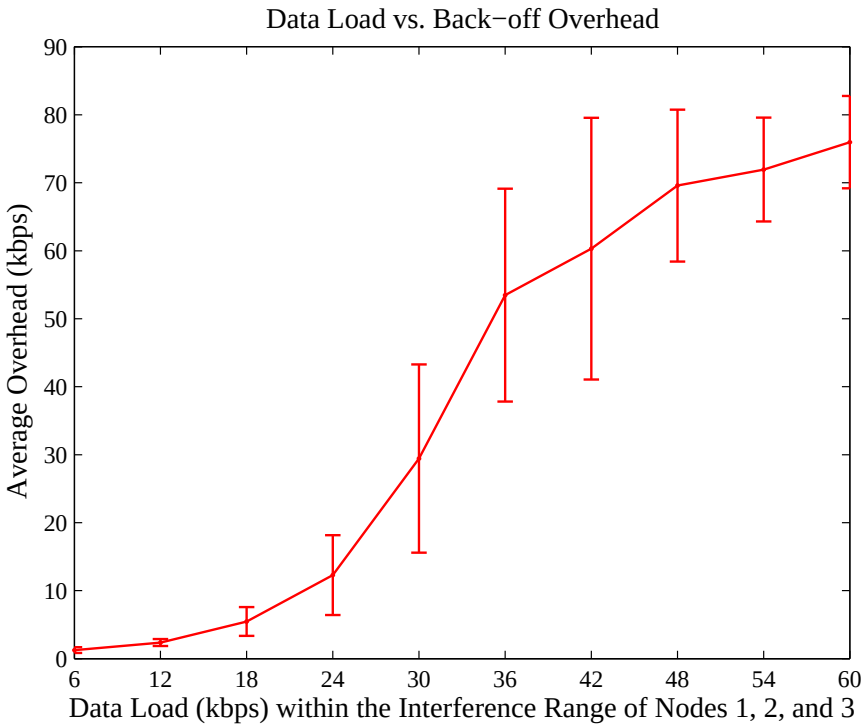


Figure 5.2.: Data Load Vs. Average Back-off Overhead

Figure 5.2 plots the average back-off and retransmission overhead w.r.t. the offered data traffic load, whereas Figure 5.2 only plots the average back-off overhead w.r.t. the offered data load. Both figures demonstrate that the average overhead is approximately the same. This shows that the back-off overhead is a dominant overhead among the two.

5.3. Flow Admission Control

Before admitting a new flow, the flow admission control algorithm of the proposed proactive bandwidth estimation mechanism determines the additional back-off overhead associated with the additional data load inside a network. This can be done using Figure 5.2. Afterwards, the flow admission control algorithm subtracts the back-off overhead in terms of bps from the available bandwidth of nodes within the interference range of the node and from the node's own available bandwidth. If the remaining available bandwidth is greater than the bandwidth requested by the new flow, the flow admission control algorithm proceeds, otherwise the admission request is rejected. For determining

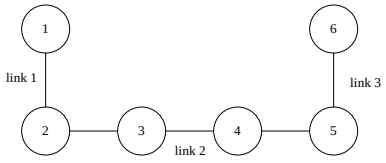


Figure 5.3.: Simulated Network Topology

the intra-flow contention, the flow admission control algorithm uses the same approach as used by ABE. If the available bandwidth at the node, after determining the intra-flow contention, is greater than the bandwidth requested by the flow, the flow admission control algorithm considers the overhead associated with the contention window using Equation 5.1. The flow admission control algorithm accepts the flow if, after considering the contention window overhead, the available bandwidth at the node is greater than or equal to the requested bandwidth. Otherwise, a flow's admission request is rejected.

5.4. Simulation Results

We use Cooja to evaluate the performance of the proactive bandwidth estimation method and its flow admission control algorithm. General simulation parameters are presented in Table 5.1. The simulated network topology is shown in Figure 5.3. We consider two simulation scenarios, which further contain sub-scenarios. In the first scenario, we use a total frame size of 127 bytes, and in the second scenario we use a total frame size of 107 bytes. Each node inside a network generates two control messages per second, and their collective size is approximately 254 bytes. Each simulation is repeated three times and average results are reported here. The averaging window size (α) is 5.

5.4.1. Scenario I

In the first simulation scenario, there is a Constant Bit Rate (CBR) data traffic of 14.88 kbps (15 frames) on links 1, 2, and 3. We vary data traffic generation rate on link 2, and study its effect on links 1, 2, and 3. Nodes 1, 3, and 5 start their transmission at 3, 6, and 9 seconds respectively. Nodes 1, 3, and 5 transmit a total of 1500 data frames. Figure 5.4 shows the average available bandwidth using ABE's modified bandwidth estimation algorithm. In this case, the overall average available bandwidth at links 1, 2, and 3 is approximately 71, 37, and 61 kbps respectively, given all nodes are transmitting.

Now let us consider that node 3 wants to admit another flow. The new flow will transmit 5 data frames per second, hence it requires 4.96 kbps bandwidth. If the new flow gains admission, node 3 will transfer 20 data frames in total, and its total data generation rate will be 19.84 kbps. The flow admission control algorithm of ABE checks the available bandwidth of nodes within the interference range of node 3. In this case, node 3 and all nodes within the interference

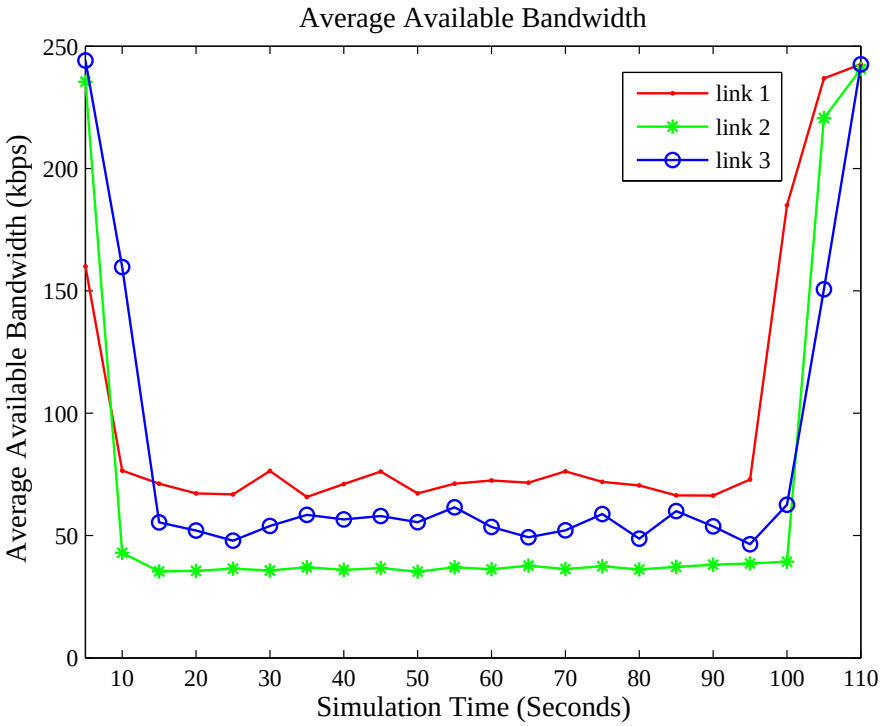


Figure 5.4.: Average Available Bandwidth Scenario I (link 2 Data Rate 14.88 kbps)

range of node 3 have available bandwidth in excess of 4.96 kbps, hence ABE's flow admission control admits the flow. On the other hand, the flow admission control algorithm of our proposed proactive bandwidth estimation method calculates the resulting additional MAC layer overhead. Node 3 wants to transmit 5 additional frames, therefore for Tmote sky motes, the contention window overhead is 39 kbps ($7812 \times 5 = 39$ kbps). Already there is approximately 45 kbps data load within the interference range of node 3. The additional data load is approximately 5 kbps, therefore total data load within the interference range of the node will be 50 kbps. The estimated additional back-off overhead will be the back-off overhead at a data load of 50 kbps minus the back-off overhead at a data load of 45 kbps. The additional estimated back-off overhead is approximately 3 kbps as per Figure 5.2. The estimated total MAC layer overhead is ($39 + 3 = 42$ kbps) with an additional data generation rate of approximately 5 kbps. The data geneartion rate of the flow will be approximately 5 kbps, therefore total bandwidth requirement is ($42 + 5 = 47$ kbps). The available bandwidth at link 2 is 37 kbps. Hence, the flow admission control algorithm of the proposed proactive bandwidth estimation method rejects the flow's admission request.

Figure 5.5 shows the average throughput on links 1, 2, and 3, assuming that the additional flow was admitted. Figure 5.5 shows that increasing node 3's data generation rate to 19.84 kbps results in a link 2 throughput of 17.89 kbps,

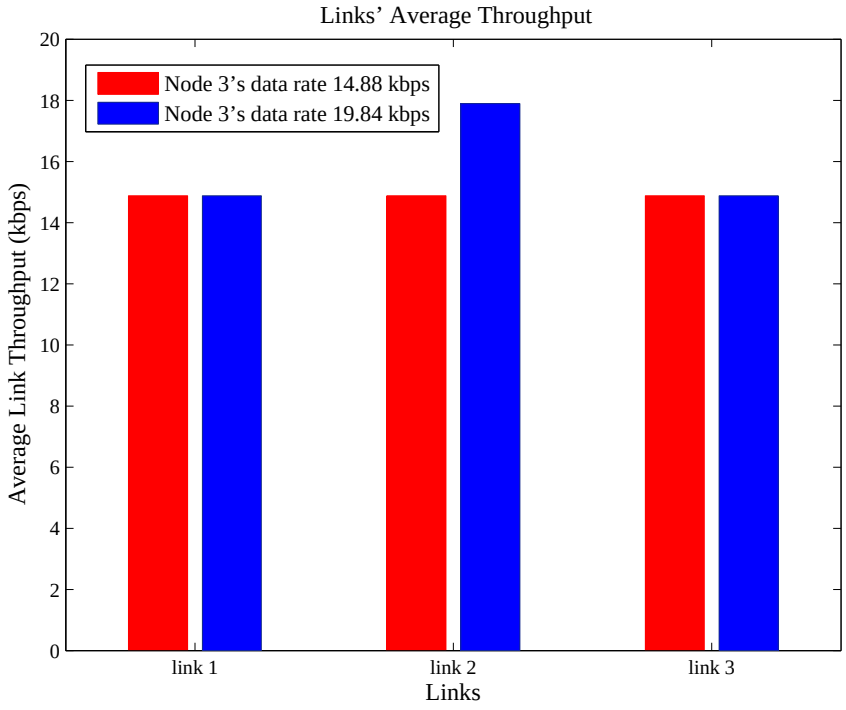


Figure 5.5.: Average Link Throughputs

below the requested rate, and hence the new flow should not have been accepted. Therefore, the proactive bandwidth estimation method works better than ABE.

5.4.2. Scenario II

In a second scenario, we use a frame size of 107 bytes instead of 127 bytes. Node 1 transmits 20 frames to node 2, and node 5 transfers 20 frames to node 6. Therefore, at links 1 and 3, CBR traffic is generated at the rate of 16.72 kbps. In all simulations, nodes 1 and 5 transmit 2000 frames. We vary the data traffic generation at link 2, and measure its effect on links 1, 2, and 3. Initially, node 3 transmits 12 frames to node 4, hence the initial data traffic load at link 2 is 10 kbps. Figure 5.6 shows the average available bandwidth as per ABE's modified bandwidth estimation algorithm. In this case, the overall average available bandwidth at links 1, 2, and 3 is approximately 21, 41, and 37 kbps respectively, given all nodes are transmitting.

Let us assume that node 3 wants to inject a new flow, transmitting 12 data frames per second. Hence, node 3 requires an additional 10 kbps. ABE's flow admission control algorithm admits the new flow, whereas, pro-actively considering the additional MAC layer overhead, our flow admission control algorithm rejects the flow's admission request. Figure 5.7 shows the average link throughputs, assuming that the additional flow is admitted over the link 2. It also shows that the proactive bandwidth estimation method works better compared

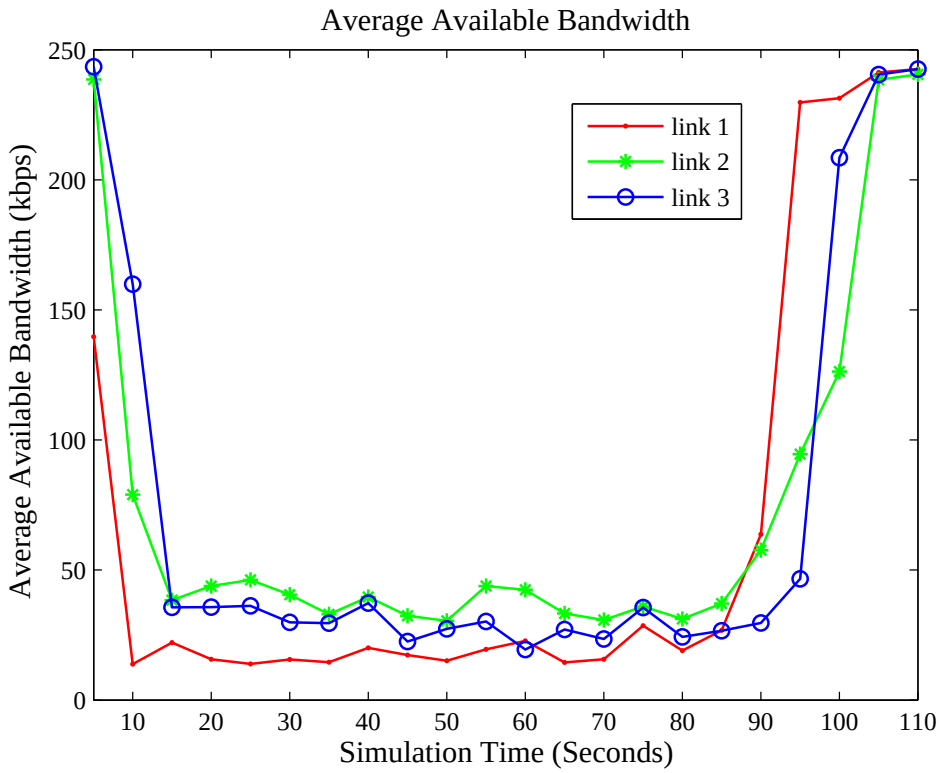


Figure 5.6.: Average Available Bandwidth (link 2 Data Rate 10 kbps)

to ABE, as it rejects an admission request whose bandwidth requirement can not be satisfied (by pro-actively considering the MAC layer overhead, and by capturing the actual MAC layer overhead). The additional flow admitted by ABE's bandwidth estimation method results in a transmission rate of 20 kbps at node 3, but the average throughput at link 2 is approximately 12.80 kbps. Furthermore, ABE's admission decision has resulted in decreased throughput at links 1 and 3. Hence, our proactive bandwidth estimation not only assures that a flow achieves the requested bandwidth, but also prevents performance degradation of already admitted flows.

5.5. Conclusions

In this chapter, we proposed a novel proactive bandwidth estimation method for IEEE 802.15.4-based networks. The proposed proactive bandwidth estimation method considers the additional unslotted CSMA-CA MAC layer overhead resulting from an increased data load inside a network. Our evaluations, carried out by simulations, show that the proactive bandwidth estimation method is more accurate compared to the state-of-the-art ABE method. While the input into the algorithm was derived experimentally using a fixed frame size of 127 bytes, the results in Section 5.4.2 show that we can also apply it to flows with

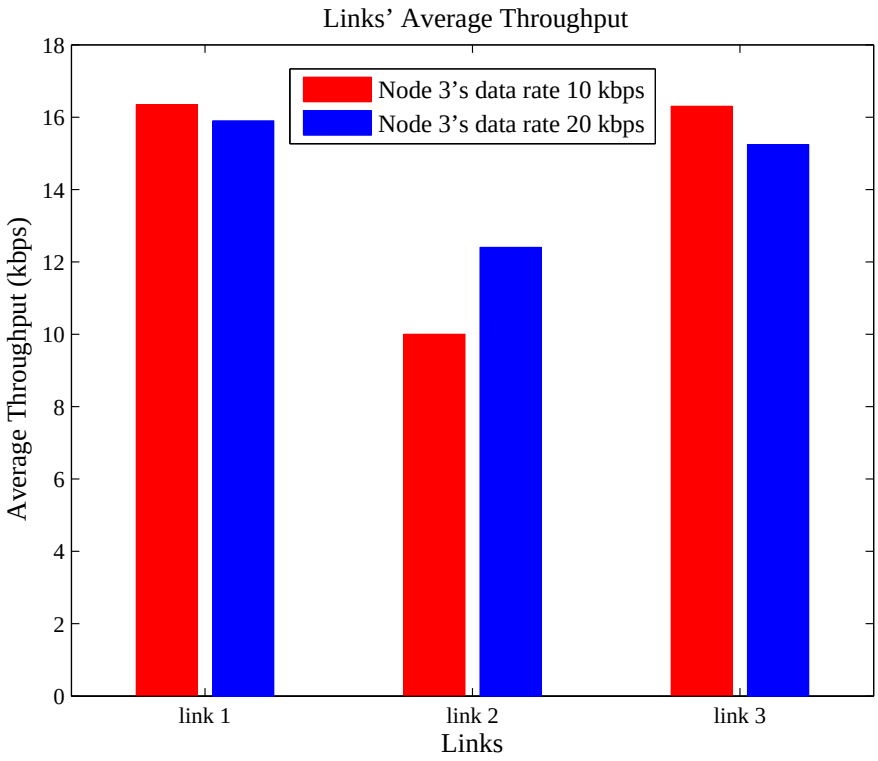


Figure 5.7.: Average Link Throughputs

different frame sizes, the dominant factor in both cases being the total offered load.

6. Available-Bandwidth-based Flow Admission Control

High-resolution and multi-dimensional sensing characteristics of IEEE 802.15.4-based WSNs have lead to their application in many real-time scenarios: visual surveillance, assisted living, and intelligent transportation to name but a few. These application areas suggest that WSNs have to deal with real-time multimedia flows. Real-time multimedia flows generate inelastic data requiring soft bandwidth guarantee and bounded delay (in this chapter, hereafter we shall refer to bandwidth and delay requirements of a flow as its QoS requirements). Excessive data (w.r.t. the available bandwidth) inside a network can cause congestion, and congestion increases packet drop rate and end-to-end packet delivery delay. Therefore, to restrict flows' data inside a network within a network's manageable limits (so that the QoS requirements of the real-time flows can be satisfied), available bandwidth-based flow admission control algorithms are used [68] [73] [58].

In wireless networks, bandwidth is a shared resource. The common assumption is that the bandwidth available to a node is shared within the interference range of the node, and nodes within a two hops distance can cause interference [12]. We also hold this assumption throughout the dissertation. The shared nature of the bandwidth in wireless networks results in the following phenomenon: (i) the data generation rate of nodes within the interference range of a node inside a network affects the available bandwidth [12], and (ii) intra-flow and inter-flow contention. Furthermore, a MAC layer protocol dictates the sharing of a communication medium, hence it limits the amount of bandwidth available to a node, e.g., in a CSMA-CA-based MAC layer protocol a node can not transmit in a back-off mode. Therefore, an effective available bandwidth-based flow admission control algorithm must consider the impact of the MAC layer on the available bandwidth before deciding about a flow's admission request.

This chapter presents BandEst; an available bandwidth-based flow admission control algorithm for IEEE 802.15.4-based networks. BandEst proposes novel algorithms to take into account the factors identified for a proper flow admission control in ad-hoc wireless networks, presented in Chapter 2. To estimate the available bandwidth, each node inside a network considers the transmission rate of nodes within the interference range of the node estimating the available bandwidth. Moreover, each node internally measures the MAC layer overhead and considers its actual impact on the available bandwidth. BandEst's flow admission control algorithm is a combination of novel algorithms that estimates additional MAC layer overhead with an increased data load inside a network,

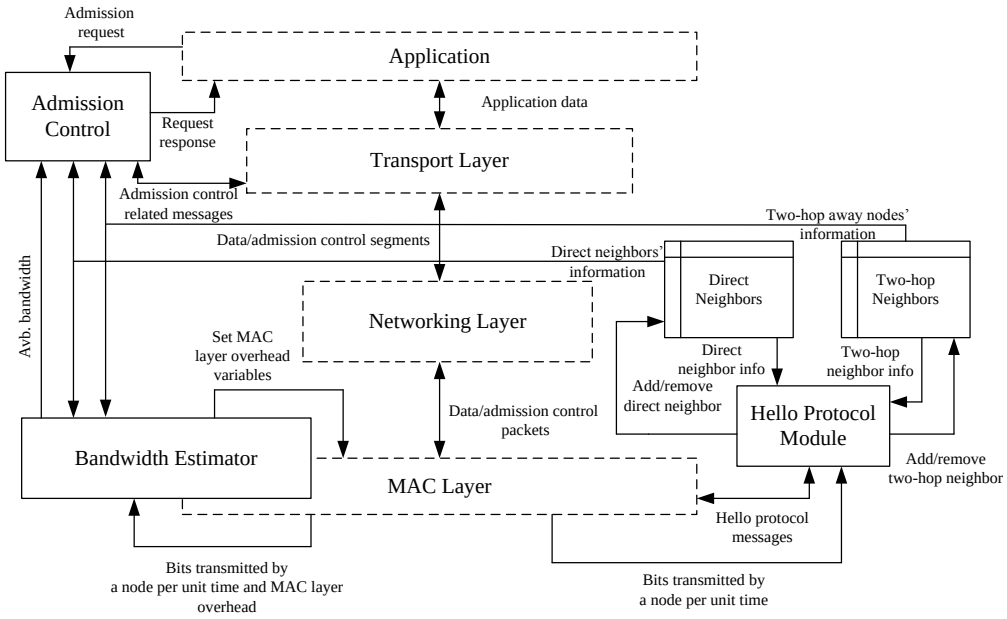


Figure 6.1.: BandEst Architecture

estimate intra-flow contention on nodes along the data forwarding path, and a new flow's contention on nodes which are within the interference range of the transmitters along the data forwarding path but are not on a flow's data forwarding path. Therefore, BandEst is a comprehensive available bandwidth-based flow admission control solution for ad-hoc IEEE 802.15.4-based networks. Extensive simulations are performed to compare BandEst with state-of-the-art available-bandwidth-based flow admission control algorithms for ad-hoc wireless networks. Our simulation results demonstrate that BandEst significantly outperforms the state-of-the-art available bandwidth-based flow admission control algorithms for ad-hoc wireless networks.

6.1. BandEst: Measurement-based Available Bandwidth Estimation and Flow Admission Control Algorithm

BandEst consists of a number of modules. The architecture of BandEst is shown in Figure 6.1 (BandEst modules are shown in solid lines):

- HELLO protocol module
- Available bandwidth estimation module
- Flow admission control module

6.1.1. HELLO Protocol Module

The HELLO protocol module regularly broadcasts a HELLO message after a predefined interval of time. The purposes of the HELLO messages are to discover direct neighbors (nodes within the transmission range of the node), learn direct neighbors' data generation rate, and the available bandwidth at direct neighbors. Hence, BandEst HELLO messages help to construct/maintain a node's direct neighbors table and the gathering of direct neighbors' information necessary for the available bandwidth estimation and a flow admission control algorithm.

To enable the discovery of two hop away neighbors (as the assumption is that two hop away nodes can cause interference), each node periodically broadcasts a neighbor information message apart from the HELLO message (a node can also piggyback its neighbors' information onto the HELLO message, but due to the small size of the IEEE 802.15.4 frame, we opted for a separate neighbor information message). In the neighbor information message, a node advertises its direct neighbors along with their data generation rates, and the available bandwidth. If a single neighbor information message is not enough for a node to advertise all of its direct neighbors and related information, the node broadcasts additional neighbor information message(s). The neighbor information message helps to construct/maintain a node's two hop away neighbors table and gathering of two hop away neighbors' information necessary for the available bandwidth estimation and a flow admission control algorithm.

From the above discussion it is evident that instead of monitoring the wireless channel to estimate the data activity within the interference range of a node, BandEst uses the HELLO and the neighbor information messages to learn about the data generation rate of nodes within the interference range of a node. The main reasons for using these messages are: (i) a data frame collision happens when data/control frames of two or more transmitters collide, hence multiple transmissions are inferred as a single transmission, this may result in an overestimate of the available bandwidth, (ii) a channel-monitoring-based mechanism requires the radio to always remain on, but to preserve energy invariably a radio duty cycling algorithm is used in IEEE 802.15.4-based networks, therefore channel-monitoring-based mechanism is not suitable for networks where radio duty cycling is used, and (iii) the HELLO and the neighbor information messages help to discover direct and two hop away neighbors. In this work we assume no radio duty cycling is used, but our work can easily be extended to incorporate effects of using a radio duty cycling algorithm on an available bandwidth-based flow admission control algorithm. The downside of exchanging data generation rate information through the HELLO and the neighbor information messages is that if a node A is not in the direct transmission range of another node B, but it is within its interference range, and none of the direct neighbors of node B have that node within their transmission range, the data generation rate information of node A does not reach node B, hence B overestimates the available bandwidth. But, in this research we assume that a network

is well connected, hence such occurrences are rare. However, if such occurrences are not rare, BandEst's performance will be impacted. Moreover, if a flow admission control algorithm uses the wireless-channel-monitoring mechanism to estimate data activity within the interference range of a node, the algorithm still requires control message(s) to learn the following: (i) available bandwidth at nodes which are within the interference range of a node and (ii) a node's two hop away neighbors.

6.1.2. Bandwidth Estimation Module

For estimating the available bandwidth, the bandwidth estimation module considers all of the IEEE 802.15.4's unslotted CSMA-CA MAC layer overheads, i.e., the impacts of back-off periods due to data frame collisions/busy channel, back-off periods before initiating a transmission of a new data frame (contention window), time spent in waiting for ACKs, bandwidth used by the IEEE 802.15.4 ACK frames, and retransmissions on the available bandwidth.

Periodically, the available bandwidth estimation module retrieves the total actual MAC layer overhead information from the MAC layer, and it keeps track of the changes over time. Similarly, the available bandwidth estimation module receives data generation rate information of nodes within the interference range of the node from its direct and two hop neighbors table. Moreover, the bandwidth estimation module considers additional bandwidth requirements of those flows which a node has admitted, but their admission reply is still outstanding (BandEst's flow admission control algorithm performs admission control on an end-to-end basis, therefore each node along the data forwarding path performs flow admission control. If a flow's admission request is successful at a particular node along the data forwarding path, the node sets aside the required bandwidth considering the intra-flow contention and additional MAC layer overhead, before forwarding the flow's admission request. This feature helps to deal with concurrent admission requests in a First-Come-First-Serve order). Each second, each node advertises its data generation rate information (including additional bandwidth of flows admitted by the node with pending admission replies) using the HELLO message.

Apart from the number of retransmitted bits and ACK frames bit, other MAC layer overheads are measured in time. The estimate of the available bandwidth is made in bits per second (bps). Therefore, the MAC layer overheads measured in time are converted to the number of bits per second by multiplying them with the channel rate. The random nature of the wireless communication medium can result in significant randomness in the available bandwidth per time unit. Such randomness can be caused by transient wireless channel impairments (shadowing, interference, multi-path fading, etc), hence estimating the available bandwidth by only considering the latest value pertaining to the overheads associated with the MAC layer and the data generation rate of nodes can be misleading. To address these issues, we propose to use a sliding-window-based averaging mechanism to estimate the available bandwidth. Let

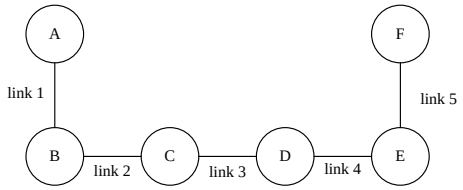


Figure 6.2.: Network Topology

us suppose that the size of the averaging window is α, a node stores the α most recent values corresponding to the MAC layer overheads, and data generation rates of nodes within the interference range of the node.

Let us assume that β represents the summation of data generation rate of a node and the nodes within the interference range of the node, and β_i represents β at the i^{th} index of the averaging window. Similarly γ_i represents the MAC layer overhead (in bps) at the i^{th} index of the averaging window. We further assume that ρ represents the channel rate. Therefore, at each bandwidth measurement instance, the available bandwidth is calculated by using average MAC layer overhead and average data generation rate of nodes' within the interference of the node using Equation 6.1.

$$\omega = \rho - \left(\frac{\sum_{\mu=1}^{\theta} (\beta_\mu + \gamma_\mu)}{\theta} \right) bps \tag{6.1}$$

In this equation, ω represents the average available bandwidth in bps, θ represents the current size of the averaging window (the maximum value of θ is α).

6.1.3. Choosing Averaging Window Size

The size of the averaging window has an impact on the available bandwidth, i.e., an averaging window having too small a size may not cope well with the transient wireless channel impairments. On the other hand, a large window size may not capture current network conditions accurately, as in this case insignificant past measurements are considered. Therefore, the size of the averaging window should be chosen carefully. To choose the averaging window size we ran a few simulations. We simulated the topology shown in Figure 6.2, and simulation parameters are shown in Table 6.1. Node A starts two flows: the first flow starts at the simulation time of 5 seconds and terminates at the simulation time of 105 seconds and the second flow starts at the simulation time of 30 seconds and terminates at the simulation time of 70 seconds. Node F is the destination for both flows and each simulation runs for 110 seconds. In different simulations, we vary the size of the averaging window. Figure 6.3 shows the available bandwidth at node C, we choose node C for reporting the available bandwidth because it is within the interference range of the maximum number of transmitters, i.e., nodes A, B, D, and E.

Table 6.1.: General Simulation Parameters

Parameter	**Value**
MAC layer	Unslotted CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 frames
Channel rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Simulated node type	Tmote sky

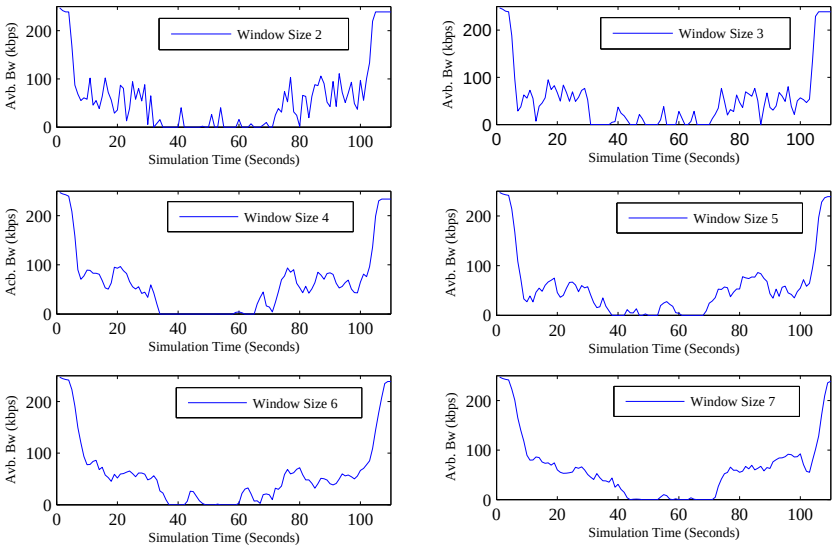
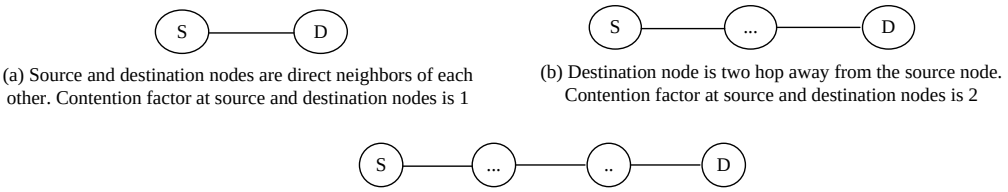


Figure 6.3.: Window Size Impact on the Available Bandwidth

Figure 6.3 demonstrates that a smaller window size results in higher degree of randomness in the available bandwidth readings at different simulation instances, but the change in the available bandwidth due to the new flow is detected earlier as compared to the lager window size (second flow starts at 30 seconds and terminates at 70 seconds). On the other hand, a larger window size decreases the amount of randomness in the available bandwidth readings, but time to detect the actual change in the available bandwidth increases. Therefore, Figure 6.3 demonstrates that choosing a window size involves a tradeoff between randomness in the available bandwidth and the time to detect the actual change in the available bandwidth. For the purpose of this work, we have chosen a window size of 5, as it demonstrates relative low randomness in the available bandwidth readings while limiting the amount of time it takes to detect the change in available bandwidth.



(a) Source and destination nodes are direct neighbors of each other. Contention factor at source and destination nodes is 1

(b) Destination node is two hop away from the source node. Contention factor at source and destination nodes is 2

(c) Destination node is more than two hops away from the source node. Contention factor at source and destination nodes is 3

Figure 6.4.: Contention Factor on Source and Destination Node

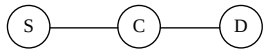
6.1.4. BandEst's Flow Admission Control Algorithm

BandEst's flow admission control algorithm takes into account the following: (i) intra-flow contention, (ii) contention on nodes that will not relay the new flow's data but are within the interference range of transmitters (nodes that will relay new flow's data) along the data forwarding path (contention on such nodes are only considered if they are relaying some other flows' data), (iii) increased MAC layer overhead with increased data traffic node at nodes that will relay new flow's data, and (iv) increased MAC layer overhead at nodes which are within the interference range of transmitters along the new flow's data forwarding path (if and only if the nodes are relaying some other flows' date). In the following subsections, we discuss different components of BandEst's flow admission control algorithm in detail.

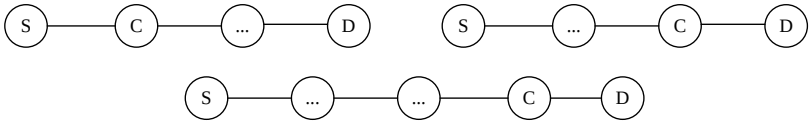
Intra-flow Contention Measurements

Determining the accurate intra-flow contention count depends on the interference range of a node. Assuming that the nodes within the two hops distance can cause interference, the interference count on any node along the data forwarding path mainly depends on the node's distance from the source and the destination nodes, as shown in Chapter 2 Section 2.3. For a new flow's admission request, BandEst's flow admission control algorithm determines the actual intra-flow contention counts on source, intermediate (data relaying nodes), and destination nodes.

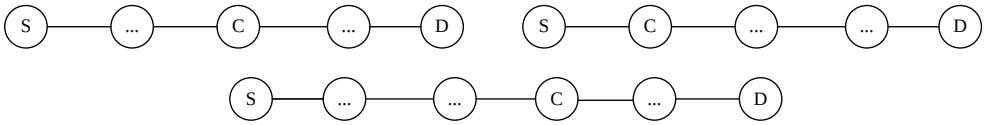
Figure 6.4 depicts the guidelines for determining the intra-flow contention count on the source and the destination nodes, and Figure 6.5 depicts the guidelines for determining the intra-flow contention count on intermediate nodes between a source-destination pair. The scenarios presented in Figure 6.4 and Figure 6.5 demonstrate that in order to determine the intra-flow contention count, a node must know its one hop and two hop neighbors. As in BandEst each node maintains lists of its one hop and two hop neighbors, BandEst's flow admission control algorithm can determine the intra-flow contention count without any additional overhead. Figure 6.5 demonstrates that the maximum intra-flow contention on an intermediate node along the data forwarding path is 5.



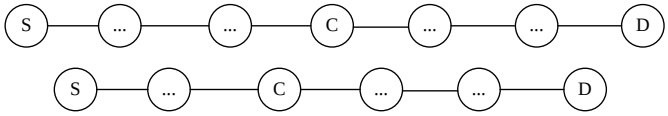
(a) Source and destination nodes are direct neighbors of the node calculating intra-flow contention factor, intra-flow contention factor at node C is 2



(b) Any one among the source or the destination node is direct neighbor of the node calculating intra-flow contention, and the other node is exactly two hops away from the node, or source node is more than two hops away, but the destination node is direct neighbor of the node then intra-flow contention factor is 3 at node C



(c) Node calculating intra-flow contention factor is exactly two hops away from the source and the destination node, or the node is direct neighbor of the source node, but the destination node is more than two hops away, or the source node is more than two hops away from the node, but the destination node is exactly two hops away from the node. In this case, the intra-flow contention factor at node C is 4



(d) Node calculating intra-flow contention is more than two hops away from both source and destination nodes, or source node is exactly two hops away from the node, and destination node is more than two hops away from the node. In this case intra-contention factor is 5

In all above scenario, node C is the node that is calculating the intra-flow contention

Figure 6.5.: Intra-flow Contention Factor Estimation at Intermediate Relaying Nodes

Considering Additional CSMA-CA MAC Layer Overhead

To consider the impact of the additional MAC layer overhead with an increased data traffic load and the number of transmitters, two methodologies can be used: analytical modeling and estimating the MAC layer overheads empirically. The analytical approach is used in [58] and [45], but the downside is that it requires simplified assumptions about collision probability, traffic patterns, error rates, and node synchronization. Therefore, analytical methods based on certain assumptions can only give a correct estimate if their assumptions hold true in real scenarios, which is not always the case. An empirical approach involves setting up a network and measuring the MAC layer overhead with different values of the offered data loads. One can run such experiments before the network becomes operational, and such experiments can be repeated multiple times to derive an average estimate of the MAC layer overheads as a function of the data load. The advantage associated with the empirical method is that one does not need to make assumptions, as the estimate is made considering the real network condition. The overhead associated with the empirical method is that a lookup table is stored on nodes that return estimated MAC layer overhead correspond-

ing to a certain value of the data load inside a network. It is not possible to store the MAC layer overhead in terms of bps corresponding to each possible offered data load, but an algorithm can estimate the MAC layer overheads for an offered data load not present in the lookup table by linear interpolation, using the two closest available data points. To consider the additional IEEE 802.15.4's unslotted CSMA-CA MAC layer overhead with an increase in the data traffic load, we used the results present in Chapter 2, Section 2.1.

Estimating Contention on Non-relaying Nodes and Concurrent Admission Requests

It is not enough to take into account the impact of intra-flow contention and additional MAC layer overhead to arrive at accurate admission decisions. An algorithm is required that can take into account the impact of a flow's contention on nodes that are not on the data forwarding path from the source node to the destination node, but those nodes are within the interference range of transmitter(s) on the data forwarding path.

We devised an algorithm to consider actual contention on non-relaying nodes which are within the interference range of transmitters along the forwarding path. Whenever a node receives an admission request message, it decides about the flow's admission request by considering intra-flow contention and additional MAC layer overhead. If, after considering the intra-flow contention and additional MAC layer overhead, the node decides to accept the flow, the node stores the information in the admission request message along with the bandwidth required by the flow, in an internal data structure. Afterwards, the node broadcasts a bandwidth increment message in which the node informs its direct neighbors about its increased bandwidth usage due to the new flow. After broadcasting the bandwidth increment message, the node waits for a small period of time before forwarding the admission request message to the next hop along the data forwarding path. Upon reception of the bandwidth increment message, direct neighbors of the node calculate their available bandwidth by considering the increased bandwidth usage information, and if the receiving node is already transmitting data, it also considers the additional MAC layer overhead. If after these checks, a node decides that it has enough bandwidth to bear the interference caused by the new flow, it updates bandwidth usage information in its one hop table corresponding to the broadcasting node. Afterwards, direct neighbors rebroadcast the bandwidth increment message so that the increased bandwidth usage information of the node (ideally) reaches all nodes within its interference range. Two hop neighbors estimate the available bandwidth considering the increased bandwidth usage information of the node, and additional MAC layer overhead (if the node is already transmitting data). If the two hop node decides that it has enough bandwidth to bear the new flow's contention, it updates bandwidth usage information in its two hop neighbors table corresponding to the bandwidth increment message originator node. If any of the nodes within the interference range of the node decides that

it does not have enough bandwidth to accommodate the interference caused by the new flow, it unicasts an admission reject message to the node. If the node receives an admission reject message while it was waiting to forward admission request message it clears the stored information and drops the admission request message. In case an admission is rejected, the bandwidth usage information of nodes are automatically adjusted in the neighbors' table when nodes re-advertise their bandwidth usage in the HELLO message. Moreover, it is possible that a node receives multiple copies of the same bandwidth increment message, hence a duplicate detection mechanism is in place to detect these scenarios, and drop duplicate bandwidth increment messages. Note that with this contention estimation algorithm the node only needs to considers the forward intra-flow contention (contention caused by the node and its downstream nodes) as due to the bandwidth increment message the nodes on the forwarding path have already considered upstream nodes contention. If the admission request fails at any node, reservations at preceding nodes time out.

In case of concurrent admission requests (assuming that nodes along the forwarding path can only accommodate a single flow), possible scenarios are: (a) One of the requests wins. If that request latter fails to be successful end-to-end, the resources will be freed, or (b) both requests may fail, even if one could have succeeded. In both cases, for unsuccessful flow(s), the best strategy is to retry after a random period of time, it is possible that some new resources have become available, or the failed requests have completed and a flow's renewed attempt to get admitted will not be concurrent to another, conflicting one, with high probability.

6.1.5. Simulation Results

Simulations were carried out with the Cooja WSN simulator. General simulation parameters are similar to the parameters shown in Table 6.1, but in these simulations we randomly select the data frame size. Total network area is $500 \times 500\ m^2$. In first scenario, we generated a random network topology of 100 wireless sensor nodes. In a second scenario, we generated a random network topology of 150 wireless sensor nodes. For BandEst's performance evaluation, we selected PABE and RABE as competitors because both are state-of-the-art flow admission control algorithms for ad-hoc wireless networks. It has been shown in [23] that proactively considering the MAC layer overhead using the anticipated future data load improves the performance of available bandwidth-based flow admission control algorithm. Therefore, in our RABE implementation, instead of using an analytical model with simplified assumptions to proactively consider the back-off and retransmission overhead, our implementation of RABE uses the empirical method to proactively consider the back-off and retransmissions overheads. Our implementation of PABE is exactly the same as described in [23]. Each simulation scenario was repeated 25 times with different random seeds. Each simulation ran for 100 seconds. Four different source-destination pairs are randomly selected. The throughput of each con-

nection is randomly distributed in the range [2-22] kbps. Nodes randomly select a frame size between 80 to 127 bytes. In the following scenarios, we consider that a flow admission control algorithm makes a wrong admission decision if accepting a new flow results in throughput degradation of the newly admitted and/or already admitted flows by more than 5% [58]. Similarly, a flow admission control algorithm makes a wrong admission decision if it unnecessarily rejects a flow, i.e., if the algorithm would have accepted the flow, this flow's and already existing flows' end-to-end bandwidth requirements could have been satisfied. It must be noticed here that ABE and RABE in their evaluations do not consider the fact that the algorithm can also make wrong decision by unnecessarily rejecting a flow, therefore our definition of effectiveness (η) as given in Equation 6.2 is more comprehensive. One may argue that unnecessarily rejecting a flow's admission request does not degrade the performance of already admitted flows, whereas wrongfully accepting a flow may degrade the performance of already admitted flows. Therefore, wrong accepts are worse compared to unnecessarily rejecting flows, hence we should only consider wrong admissions as a bad admission decision. The alternate argument is that one must use available resources efficiently, otherwise one has to deploy sufficient resources so that QoS requirements of flows can be satisfied during peak network utilization, but in this case most of the time the network resources are underutilized. Hence, to have a comprehensive evaluation, we give equal importance to both types of wrong decisions.

$$\eta = \frac{\textit{Number of correct admission decisions}}{\textit{Total number of admission requests}} \tag{6.2}$$

Figure 6.6 shows the mean effectiveness of the different evaluated methods over 25 repetitions, along with a 95% confidence interval. Figure 6.6 shows that for the network of 100 wireless sensor nodes the mean effectiveness of BandEst is higher than PABE, RABE, and no flow admission control techniques, and the difference is statistically significant. Figure 6.6 demonstrates that the performance of RABE and PABE in terms of the mean effectiveness is similar and their 95% confidence intervals overlap. We noticed that invariably the corruption of broadcasted bandwidth increment message due to the interference results in wrong admission decisions, as far as BandEst is concerned. Moreover, Figure 6.6 demonstrates that the mean effectiveness of BandEst is higher than the other techniques, and the difference is statistically significant in a denser network of 150 nodes. Furthermore, from Figure 6.6 we can conclude that the mean effectiveness of all the evaluated techniques decreases a bit as the network becomes denser. In the simulations, we have observed that a higher density lead to more broadcast messages being lost, hence a decreased effectiveness of BandEst. PABE and RABE do not consider the correct contention factor on relaying and non-relaying nodes, hence their inferior performance demonstrated compared to BandEst. Figure 6.6 also shows the mean effectiveness when no flow admission control algorithm is used. The mean effectiveness with no flow admission control algorithm is lower than BandEst, and the difference is statistically significant. Furthermore, the mean effectiveness with no flow admission

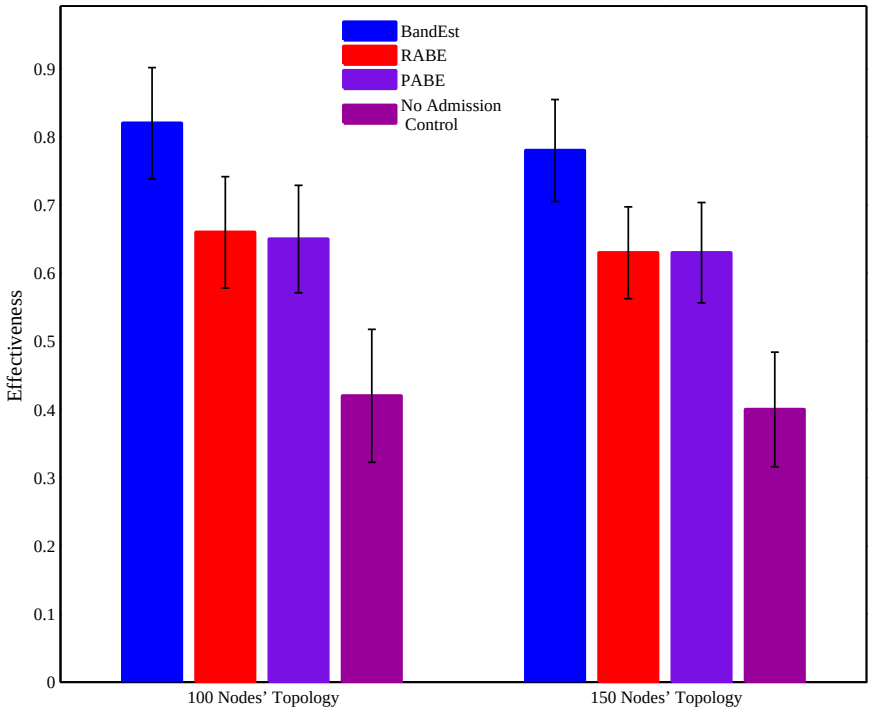


Figure 6.6.: Mean Effectiveness Comparison

control algorithm is lower compared to PABE and RABE, and the difference is statistically significant. The advantage in this case is that we do not have any control message overhead apart from the routing messages. If there are very few flows, we do not need any admission scheme as all flows can be accommodated. But that is unlikely to be the case, given the low bandwidth and shared nature of IEEE 802.15.4 networks. So most likely, the flows collectively would overwhelm the network, in which case the results here show that it worthwhile to pay the overhead.

Table 6.2 shows the number of times different schemes make a wrong admission decision. It can be seen that BandEst made fewer wrong decisions compared to the other three schemes. Moreover, these results demonstrate that BandEst does not reject a single flow unnecessarily, whereas RABE and PABE do reject flows unnecessarily. Also, if we focus on only the wrong accept decisions, BandEst outperforms all other schemes as well.

Figure 6.7 and Figure 6.8 show mean admission response delay w.r.t. route length for cases when admission is granted. Admission response delay is simply measured by noting when the admission request message was sent and when the admission response was received. The average admission response delay for BandEst is higher compared to PABE and RABE. The average admission response delay for BandEst is high because it uses a distributed flow admission control method, i.e., before accepting a flow's admission request, the BandEst flow admission control algorithm broadcasts the bandwidth increment message

Table 6.2.: Number of Wrong Admission Decisions Comparison

Method	100 Nodes		150 Nodes	
	Wrong Accepts	Wrong Rejects	Wrong Accepts	Wrong Rejects
BandEst	18	0	22	0
RABE	27	7	28	9
PABE	30	5	31	6
No Flow Admission Control	58	0	60	0

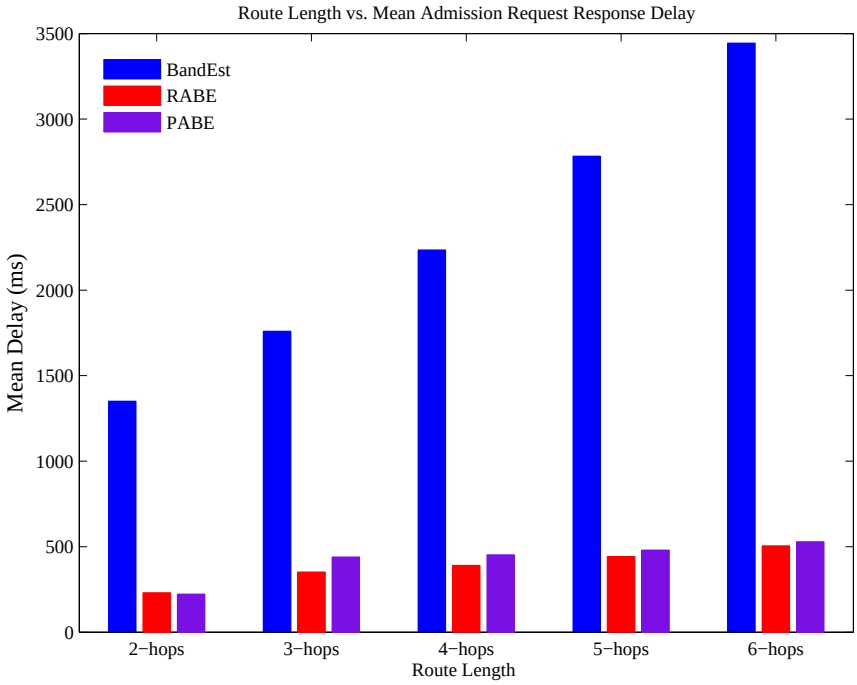


Figure 6.7.: Route Length vs. Mean Admission Request Response Delay (100 Nodes Scenario)

within the interference range of the node. This step is carried out to check whether all nodes within the interference range of the node can accommodate the existence of a new flow. As per the BandEst flow admission control algorithm, after broadcasting the bandwidth increment message, a node waits for a small amount of time, and if any node within the interference range of the node can not accommodate the flow, it unicasts the admission reject message to the originator of the bandwidth increment message. We performed some additional experiments and it was observed that after broadcasting the bandwidth increment message, a node typically receives an admission reject message (if required) within 400 ms. Therefore, the bandwidth increment message originating node waits for at least 400 ms before forwarding the admission request message. In our simulations, a node waits 500 ms before forwarding the admission request message. Figure 6.7 and Figure 6.8 demonstrates this, with the

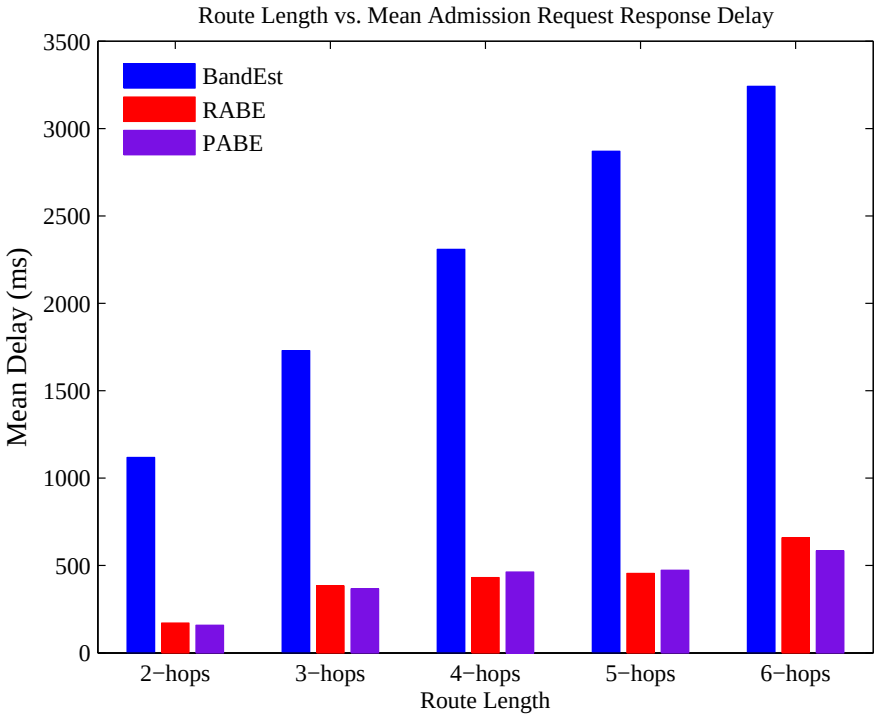


Figure 6.8.: Route Length vs. Mean Admission Request Response Delay (150 Nodes Scenario)

admission response delay increasing by about 500 ms for each additional hop.

The per-node overhead associated with BandEst is low, i.e., less than 1 kbps, and the additional overhead in BandEst is only 6% for the two network densities compared to RABE and PABE.

6.2. Conclusions

In this chapter, we presented BandEst; a novel available-bandwidth-based flow admission control algorithm for ad-hoc IEEE 802.15.4-based networks that takes into account the factors identified for a proper admission control (presented in Chapter 2). One of the main contributions of BandEst is that it proactively considers the complete IEEE 802.15.4 unslotted CSMA-CA MAC layer overhead considering the future data load. Other main contributions are: novel algorithms for estimating intra-flow contention, estimating contention on non-relaying nodes, additional MAC layer overhead associated with an increased data traffic load on non-relaying nodes, and an algorithm that deals with concurrent admission requests in a First-Come-First-Served scheme. Simulation results have demonstrated that taking into account the highlighted factors results in an effective available-bandwidth-based flow admission control algorithm for ad-hoc wireless networks. Moreover, BandEst shows significant improvements

compared to the state-of-the-art available bandwidth-based flow admission control algorithms.

7. Available-Bandwidth-based Proactive Routing Protocol

IEEE 802.15.4-based ad-hoc networks are capable of generating real-time multimedia flows. Real-time multimedia flows require bounded delay and soft bandwidth guarantee. A routing protocol forwards data from the source node to the destination node, therefore the state of the forwarding nodes (on a flow's data forwarding path) in terms of congestion and node traversal delay affects the performance of the real-time multimedia flows. In an attempt to satisfy QoS requirements of a real-time multimedia flow, a routing protocol must select the best data forwarding path (considering the QoS requirements of the real-time multimedia data) among different candidate data forwarding paths (if available). Estimating the available bandwidth and then selecting the best data forwarding path based on the available bandwidth may result in less end-to-end delay, thereby helps to satisfy real-time multimedia flows' QoS requirements. Therefore, in this chapter, we present end-to-end available-bandwidth-based proactive routing protocol for IEEE 802.15.4-based ad-hoc networks. Moreover, we integrate the routing protocol with BandEst.

An available-bandwidth-based proactive routing protocol maintains the best data forwarding path in terms of the end-to-end available bandwidth towards each sink node present in a network. Moreover, a node can maintain more than one data forwarding path towards the same sink node. We performed extensive simulations, and compared our proactive routing protocol with a state-of-the-art opportunistic routing protocol. The simulation results demonstrate that the opportunistic routing protocol can distribute data load unevenly (in case of multiple sink nodes), hence results in high end-to-end delay and low PDR. In case of our proactive routing protocol, selecting forwarding paths by only considering the end-to-end available bandwidth invariably results in lengthy data forwarding paths. Such a lengthy data forwarding paths results in higher intra-flow contention, hence PDR and end-to-end delay are impacted. One of the simulation scenarios, using multiple sink nodes, demonstrates that in case of our proactive routing protocol, carefully selecting the data forwarding path(s) that are not too long compared to the shortest available data forwarding path(s), but that have better end-to-end available bandwidth significantly improves the performance of the proactive routing protocol. Our results hint that, in general, trading off end-to-end available bandwidth and the length of a data forwarding path may improve end-to-end PDR and delay.

7.1. Proactive Routing Protocol Design

In this section, we elaborate on the design of the end-to-end available-bandwidth-based proactive routing protocol for IEEE 802.15.4-based ad-hoc networks.

7.1.1. Available Bandwidth Estimation Algorithm

The available bandwidth estimation algorithm is the same as the bandwidth estimation algorithm of BandEst, and the average available bandwidth at a particular node n inside a network is denoted by ω_n. The net bandwidth available at any node n inside a network is B_n, and it is equal to the minimum average available bandwidth at nodes within the interference range of a node, including the node n. Equation 6.1 uses the MAC layer overhead that is measured based on the current data traffic load. As shown in Chapter 2, Section 2.1, a considerable increase in the data traffic load results in an increased MAC layer overhead. In the design of the routing protocol, we do not consider the impact of the additional MAC layer overhead with an increased data traffic load because it requires information about the anticipated increase in the data traffic load. The role of a routing protocol is to find data forwarding path(s) that best suits the type of data being forwarded, e.g., if a data packet requires high reliability, it may select a path that offers higher reliability compared to the alternative available data forwarding path. Thus, at best, invariably a routing protocol considers data in aggregate rather than distinguishing between flows and finding data forwarding paths based on an individual flow's requirements.

7.1.2. Available-Bandwidth-based Routing Protocol

A network is represented by a set N of nodes. A node in set N is represented by n. The set of nodes within $n's$ interference range is denoted by I_n, and it is defined as: $I_n = \{m : hopcount_{n,m} \leqslant 2\}$, and $(n \in I_n)$. The net available bandwidth at node n is B_n, and it is defined as $B_n = \{min(\omega_m \forall m \in I_n)\}$. Set S presents sink node(s) in a network, and a sink node in S is represented by s. A possible list(s) of data forwarding path(s) to a particular sink node s at any node is K_s. Two forwarding paths are different if they at least have one distinct node. A single data forwarding path in K_s is denoted by $k_{s,j}$. A node in the data forwarding path $k_{s,j}$ is represented as r. The bandwidth available on a particular data forwarding path j towards a sink node s is $B_{k_{s,j}}$ and it is defined as $B_{k_{s,j}} = \{min(B_r \forall r \in k_{s,j})\}$. The goal of the available-bandwidth-based proactive routing protocol is for each node in a network, find a data forwarding path (if present) towards each sink node in a network such that the selected path has the highest available bandwidth, and it should not include loops. Moreover, when a routing protocol detects a new flow, it must select the sink node for the flow. The selected sink node is d, and $d = \{s : \arg\max_s (B_{k_{s,j}} \forall (s \in S \wedge j \in k_{s,j}))\}$.

For route discovery each sink node inside a network broadcasts its network layer address and sink sequence number along with the information required by the available bandwidth estimation algorithm in a HELLO message. A sink node broadcasts HELLO messages periodically with an incremented sequence number. Each node inside a network maintains a sink table, a single record in the sink table stores information about a sink node present inside a network. A record in a sink table stores the sink node's address, the sink sequence number, maximum available bandwidth on a selected data forwarding path towards the sink node, and the next hop address. On the reception of the HELLO message, direct neighbor of the sink node extracts the information about a sink node. It matches the sink node address with the sink nodes' addresses present in its sink table. If the sink node address does not match, the direct neighbor add a new record to its sink table. Otherwise, the direct neighbor compares the stored sink sequence number with the received sink sequence number. If the received sink sequence number is greater, the record is updated with the received information, otherwise the sink information present in the HELLO message is ignored.

Periodically, each node inside a network broadcasts HELLO messages (the same HELLO message as described in Chapter 6, Section 6.1.1). The additional information apart from the information required by the available bandwidth estimation module is about the sink nodes the node has discovered. With each discovered sink node's information the node advertises end-to-end minimum available bandwidth (minimum of B_n and the minimum available bandwidth stored in the node's sink table pertaining to the sink node) towards the sink node along with the sink sequence number extracted from the node's sink table.

When a node A receives a HELLO message from a non-sink node B, node A extracts the sink nodes' information for the HELLO message. Corresponding to each sink node's information, node A matches the received sink node address with the addresses present in the records of the sink table. If the address does not match, node A adds a new record in its sink table and stores the new sink node's address, sequence number, maximum available bandwidth on the path towards the sink node, and next hop address (node B address) in it. If the received sink node address matches, node A compares the received sink sequence number with the stored sink sequence number. If the received sequence number is less than or equal to the stored sink sequence number, the sink information is ignored. Otherwise, node A updates the sink sequence number in the corresponding record of the sink table. Afterwards, the node matches node B address with the next hop address stored in the corresponding record of the sink table. If both addresses match, node A updates the corresponding record in the sink table with the received sink information. If the addresses do not match, node A compares the received maximum available bandwidth value with the stored maximum available bandwidth field in the corresponding record. Node A updates the maximum available bandwidth and next hop fields of the corresponding record of the sink table with the received value of the maximum available bandwidth and node B's address respectively, if the received maximum available bandwidth is greater than the stored value. A record from the sink

table of node A is deleted in any one of the following cases: (i) next hop for the sink node is removed from the direct neighbor table (direct neighbor table is discuused in Chapter 6 and Section 6.1.1) and (ii) if the node does not receive the discovered sink's information from the node whose address is stored in the next hop address field of the sink table within a pre-defined time interval.

After a flow starts to use a data forwarding path, the end-to-end available bandwidth on the data forwarding path decreases. Therefore, the routing protocol may discover an alternative data forwarding path to the same sink node that offers higher end-to-end available bandwidth, as compared to the existing data forwarding path. In such an event, the routing protocol updates the sink table so that a node starts to use the newly discovered better data forwarding path. In this case, all the data will be directed to the newly discovered data forwarding path. The following deficiencies are associated with this: (i) rerouting all data traffic on the newly selected data forwarding path can cause congestion on the data forwarding path, and (ii) inefficient utilization of different data forwarding paths (if present). Our solution to these deficiencies is that, once a flow starts to use a data forwarding path, it is not allowed to change the data forwarding path. The flow can only change the data forwarding path in case of a route failure. In this situation, it is possible that the same node is using different data forwarding paths to the same sink node for different flows. With this there arises a need to distinguish between the data forwarding paths in use by different flows. Our proposed solution to this problem is to use a separate forwarding table for established flows, and we assume that each flow has a unique identifier, and the identifier will not be re-used immediately.

Whenever a new flow starts, the node searches the best sink node for the flow (in terms of end-to-end available bandwidth on possible available data forwarding paths). Afterwards, the node stores the source address, source port, sink address, immediate upstream node's address, and next hop address (extracted from the sink table) in the forwarding table. Therefore, whenever a network layer receives the data packet, it matches the source address and source port with the same fields of the forwarding table records. If a match is found, the data packet is relayed to the next hop whose address is stored in the forwarding table. Otherwise, the data packet is dropped. A record in the forwarding table is removed if any one of the following cases happen: (i) a node does not receive a data packet for the flow for a pre-defined interval of time, (ii) a flow's next hop is removed from the direct neighbor table, and (iii) a record in the sink table containing the sink node and the next hop that is being used by the flow times out. The last two cases require route repairs.

In case of a route failure, a node tries to repair the route locally by searching an alternate data forwarding path in its sink table. If no alternate data forwarding path is found, the node informs the upstream node (upstream node's address is present in the forwarding table) about the route failure. The upstream node tries to repair the route, if unsuccessful it informs its upstream node about the route failure. This process continues until a node on the data forwarding path finds an alternate route, or the source node is informed about the route failure.

The proactive routing protocol uses the sink sequence number, normally sink/destination sequence numbers are not used in proactive routing protocols. The reason for using sink sequence number is because of the proactive routing protocol's goal to avoid loops. Loops (even if they are short-lived) in the proactive routing protocol can be harmful, as in the proactive routing protocol we fix the data forwarding paths for the flows. Therefore, if a new flow appears in a network, and at that time there was a loop on the best available data forwarding path, the new flow's data will not reach the sink node, because the data forwarding path is fixed. Lets consider the following example to illustrate how sink sequence numbers can help to avoid loops.

Let us consider a line topology of five nodes A $\leftrightarrow$ B $\leftrightarrow$ C $\leftrightarrow$ D $\leftrightarrow$ E. We further suppose that flow 1 starts from node B and terminates at node E. After some time flow 2 originates from node D and terminates at node E. As per the available-bandwidth-based proactive routing algorithm, nodes within the interference range of node D update the available bandwidth towards node E. We suppose that the updated information does not reach node A, and node A advertised old higher available bandwidth information towards node E. Therefore, node B thinks it discovered an alternate path towards node E with the higher available bandwidth end-to-end, hence it adds the new route in its sink table. If now node B wants to start flow 3 towards node E, it forwards data to node A and node A transfers data to node B, hence a loop is created. If the sink sequence number is in use, node A would have advertised the old information with a sink sequence number that node B already knows, as node A can only get node's E information from node B. Therefore in our available-bandwidth-based routing protocol node B only accepts the sink node information, if the received sink sequence number is greater than the one stored at node B, hence node B avoids the loop.

7.2. Simulation Results

Simulations were performed using the Cooja WSN simulator. One hundred wireless sensor nodes were placed in a 300×300 m^2 grid topology. General simulation parameters are shown in Table 7.1. Two set of experiments are performed. In the first set of experiments, we control the location of the sink and the source nodes, moreover the data generation rate of the source nodes are also controlled. In the second set of experiments, we increase the number of sink nodes inside a network, and randomize the location of the source and the sink nodes along with the data generation rate of the source nodes. Hereafter, we refer to the setup for the first set of experiments as the controlled setup, and to the setup for the second set of experiments as the random setup.

As our proactive routing protocol uses available bandwidth as a routing metric, therefore for a fair comparison we use available bandwidth as a routing metric for the opportunistic routing protocol as well. Following is the model of our available-bandwidth-based opportunistic routing protocol implementation. P_n

Table 7.1.: General Simulation Parameters

Parameter	Value
MAC layer	Unslotted CSMA-CA
MAC layer reliability	Enabled
Radio duty cycling algorithm	No duty cycling
Radio model	Unit disk graph model
MAC layer queue size	30 frames
Channel rate	250 kbps
Node transmission range	50 meters
Node carrier sensing range	100 meters
Total frame size	127 bytes
Simulated node type	Tmote sky

is the set containing the shortest hop-count value towards each sink node inside a network at a node n. An item in the set P_n is identified as $p_{n,s}$. The first goal of the available-bandwidth-based opportunistic routing protocol is to select a sink node $d = \{s : \arg\min_{s}(p_{n,s} \forall s \in S)\}$. This is because, invariably an opportunistic routing protocol selects a sink node that is closer to the source node. Afterwards, it selects the downstream node among the available candidate nodes (providing the shortest hop count towards the sink node) based on the node's available bandwidth. A set $C_{n,d}$ contains downstream nodes at node n that provide the same shortest hop-count towards the sink node d. A node in the set $C_{n,d}$ is identified as $c_{n,d,i}$. The second goal of the available-bandwidth-based opportunistic routing protocol is to select a downstream node r towards the sink node d at node n, and $r = \{c_{n,d,i} : \underset{c_{n,d,i}}{arg\,max}(B_{c_{n,d,i}} \forall c_{n,d,i} \in C_{n,d})\}$, B is defined in Section 7.1.2. This process is repeated before the transmission of each data packet. In the opportunistic routing protocol, the HELLO message also contains information about a node's hop count towards sink node(s) present in a network. We used the shortest hop-count to shortlist the candidate sink node(s) and downstream nodes, otherwise for example Geographical Position System (GPS) may be used to select the candidate downstream nodes that provide positive progress towards the selected sink node. GPS-based techniques require extra hardware, and a situation can arise in which a selected downstream node does not have a route to the sink node even though the network is fully connected (called a void, where a node does not have neighbors physically closer to the sink).

7.2.1. Controlled Setup Results

There were two sink nodes inside the simulated network and they were placed at coordinates (0, 0) and (150, 150). Three nodes were acting as source nodes. The three source nodes were placed closer to the second sink node, and their minimum hop-count distance towards the second sink node is 4, 4, and 3 hops. The minimum hop count distance of the source nodes towards the first sink

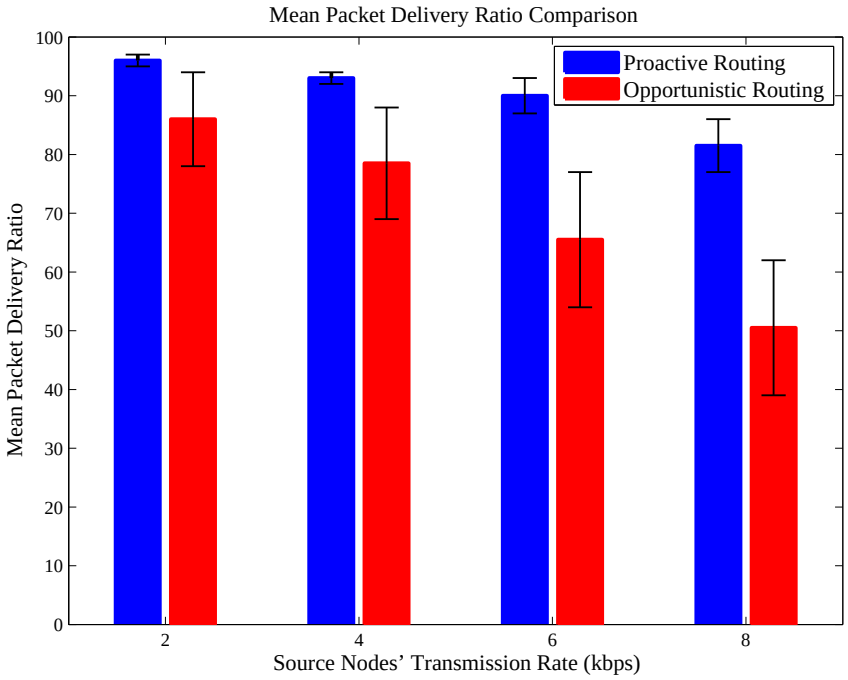


Figure 7.1.: Mean Packet Delivery Ratio

node is 8, 5, and 9 hops. The source nodes start their transmission at 10, 20, and 30 simulation seconds. The source nodes terminate their transmission at 100 simulation seconds. We vary the traffic generation rate in four different simulation scenarios, i.e., the source nodes transmit data at the rate of 2 kbps, 4 kbps, 6 kbps, and 8 kbps respectively. Each simulation scenario is repeated 10 times with different random seeds, and each simulation runs for 115 seconds. Hereafter (in this chapter), we refer to the available-bandwidth-based proactive and opportunistic routing protocols simply as proactive and opportunistic routing protocols respectively.

As the source nodes are closer to the second sink node compared to the first sink node, the opportunistic routing protocol directs all data traffic towards the second sink node, and it selects a next hop based on a downstream node's available bandwidth. On the contrary, the proactive routing protocol selects forwarding paths based on the end-to-end available bandwidth towards each sink node present inside the network, therefore in simulations the proactive routing protocol directs flows one and three towards the second sink node, and the data packets in the second flow are sent to the first sink node.

Figure 7.1 shows the mean PDR comparison of both routing protocols. Figure 7.1 demonstrates that the mean PDR of the proactive routing protocol is higher compared to the opportunistic routing protocol, and the 95% confidence intervals do not overlap. Hence, in the given scenario, the proactive routing protocol shows statistically significantly better PDR. The cause of inferior PDR in case of the opportunistic routing protocol is due to the fact that it directs all the flows'

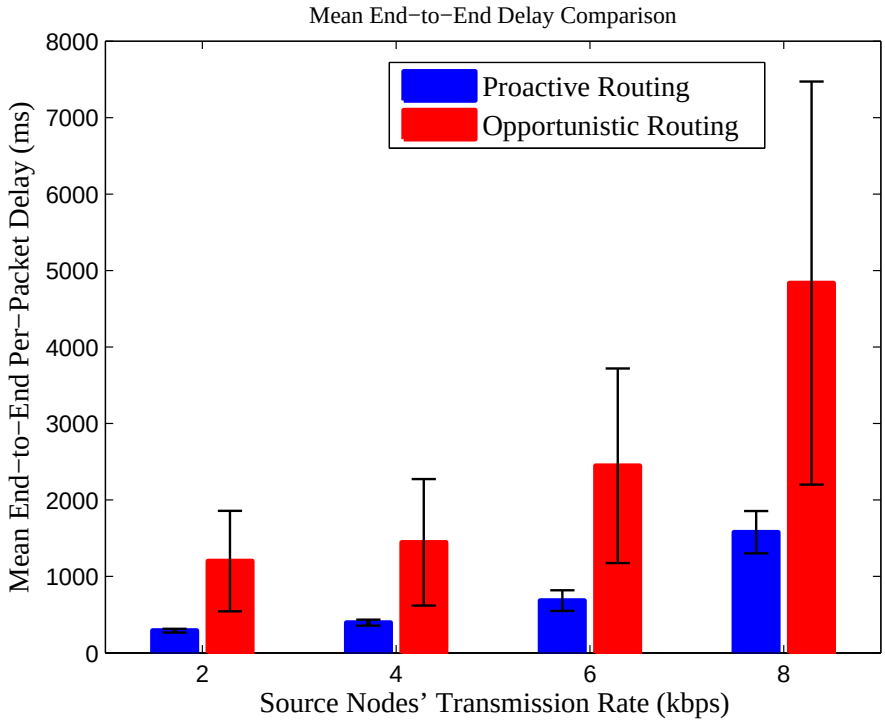


Figure 7.2.: Mean End-to-End Per-Packet Delay Comparison

data to the same sink node. Therefore, all the flows' data converge near the same sink node, resulting in increased contention near the sink node. Increased contention increases the packet delivery delay, and this can be verified from Figure 7.2, as the mean per-packet delay in case of the opportunistic routing protocol is statistically significantly higher compared to the proactive routing protocol. Figure 7.3 shows the mean number of retransmission comparison of both routing protocols. When source nodes were transmitting at the rate of 2 and 4 kbps, the 95% confidence intervals of both routing protocol overlap. But, when the source nodes transmit at the rate of 6 and 8 kbps, the mean number of retransmissions in case of the proactive routing protocol is statistically significantly lower compared to the opportunistic routing protocol.

In these set of simulations, the proactive routing protocol selected a slightly longer path for the second flow to an alternate sink node, and it showed significantly improved performance. But, in real situations, the data forwarding paths' length can vary a lot, moreover all source nodes may not generate data at a uniform rate. Hence, it is essential to evaluate the performance of both types of routing protocols using a random setup.

7.2.2. Random Setup Results

In the random setup, 6 nodes were randomly selected as sources nodes, and we incrementally increase the number of sink nodes inside a network from 1 to

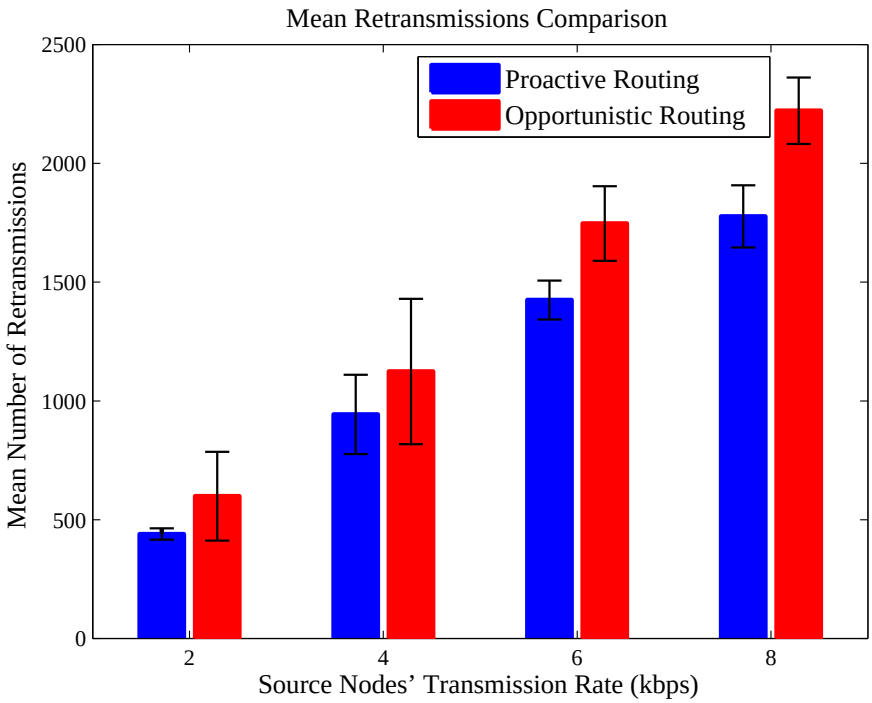


Figure 7.3.: Mean Retransmission Comparison

4. Furthermore, the sink node(s) are randomly selected from the total nodes (apart from the source nodes) present inside the network. The source nodes starts their transmission at 10, 20, 30, 40, 50, and 60 simulation seconds, and all flows terminate their transmission at 100 simulation seconds. The data generation rate of each flow is randomly distributed in the range [4-8] kbps. The total simulation time was 115 seconds, and each simulation scenario is repeated 10 times.

Figure 7.4 (a) shows the comparison of the mean data forwarding path's length as the function of the number of sink nodes inside the network. As the number of sink nodes inside the network increases, the mean data forwarding path length of the flows decreases, in the case of opportunistic routing protocol, and the difference is statistically significant compared to the proactive routing protocol. In the case of proactive routing protocol, adding the sink nodes inside the network does not decrease the flows' mean forwarding path length, as the 95% confidence interval overlaps in all the cases. The results presented in Figure 7.4 (a) are logical as the opportunistic routing protocol selects the nearest sink node for a flow, and then selects the candidate downstream nodes based on the minimum hop-count, afterwards it selects the downstream node for transferring the data packet based on the candidate downstream nodes available bandwidth. On the other hand, the proactive routing protocol only considers a data forwarding path's end-to-end available bandwidth, and it does not consider the number of intermediate nodes along the selected data forwarding path, hence it may select longer data forwarding paths. Moreover, in the case of multiple sink

nodes, the proactive routing protocol selects the sink node for a flow, based on the data forwarding path that has the highest available bandwidth, hence the protocol does not select a sink node that is closer to the source node. Therefore, with an increase in the number of sink nodes inside the network the length of the data forwarding path does not decrease.

Figure 7.4 (b) shows the mean PDR of both routing protocols. Figure 7.4 (b) demonstrates that, in most cases, the observed difference in the PDR of both routing protocols is statistically not significant, as the 95% confidence intervals overlap, and the means are in the overlap regions. In the case of 4 sink nodes, the confidence intervals overlap, but the mean PDR of the opportunistic routing protocol is not in the overlap region. Hence, in this case, we cannot conclude that the observed difference in the performance is statistically significant. Hence, we conclude that in most of the cases, increasing the number of sink nodes inside the network does not yield any benefit in terms of the PDR. The 95% PDR confidence interval (in the case of opportunistic routing protocol) is wider especially when there were 2, 3, and 4 sink nodes inside the network. This demonstrates that the opportunistic routing protocol is inherently not that capable of balancing the traffic load well and its performance depends on the topology. The performance of the proactive routing protocol, on the other hand, is more predictable, it seems to be able to balance the traffic in a more topology-independent manner.

Figure 7.4 (c) shows the mean per-packet end-to-end delay comparison of both routing protocols. The figure demonstrates that the observed difference in the mean end-to-end delay of both routing protocols is statistically not significant, as the 95% confidence intervals overlap, and the means are in the overlap regions. If we only consider the proactive routing protocol, the 95% confidence intervals overlap in all the cases and the means are in the overlap regions, hence increasing the number of sink nodes inside a network does not provide benefit in terms of the mean end-to-end per-packet delay. For the opportunistic routing protocol, in most of the cases, the 95% confidence intervals overlap, and the means are in the overlap regions. But, if we compare the 1 sink node scenario result with the result corresponding to the 2 sink nodes scenario, the 95% confidence intervals overlap, but the mean end-to-end delay corresponding to the 2 sink node scenario is not in the overlap region. Comparison of the 1 sink node scenario result with the 4 sink nodes scenario results demonstrate that, the 95% confidence intervals overlap, but the mean end-to-end delay in the case of 4 sink nodes is not in the overlap region. Hence, in these cases, we cannot conclude that the observed difference in the performance is statistically significant. The 95% mean end-to-end per-packet delay confidence intervals are wider in the case of opportunistic routing protocol, and the reason is the same as we have given in the case of PDR comparison.

Figure 7.4 (d) shows the mean total number of retransmissions comparison. In a couple of cases, i.e., when there were 2 and 4 sink nodes inside the network, the mean total number of retransmissions in the case of opportunistic routing protocol is statistically significantly lower as compared to the proactive routing

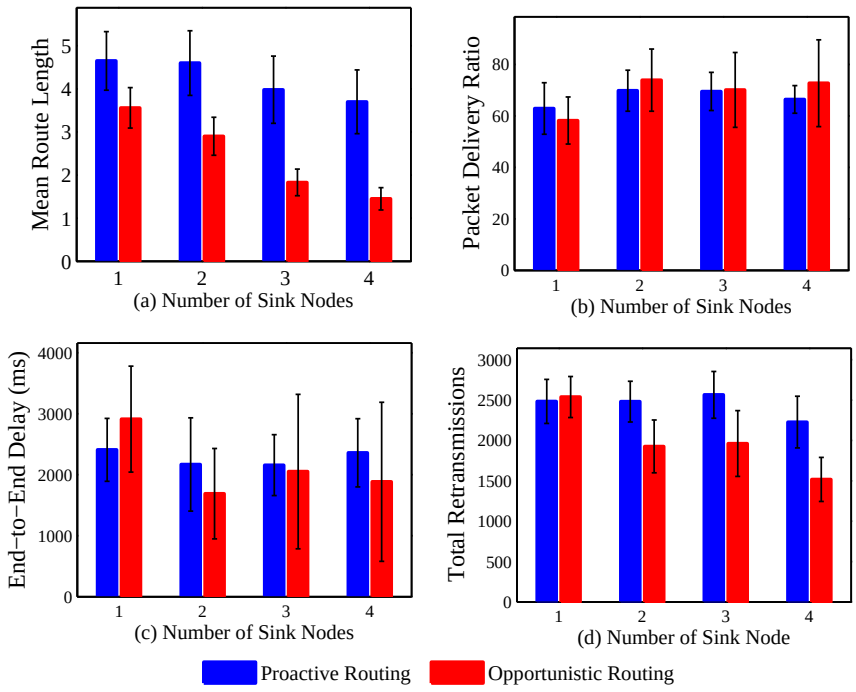


Figure 7.4.: Comparison w.r.t. Different Metrics

protocol. But in the case of 3 sink nodes, the 95% confidence intervals overlap, but the mean values are not in the overlap region, thereby we cannot conclude that the difference in the performance is statistically significant. In the fourth scenario, i.e., when there were 4 sink nodes inside the network the 95% confidence intervals do not overlap, and the mean number of retransmission in the case of opportunistic routing protocol is statistically significantly lower. This suggest that may be the number of retransmissions depends on the data forwarding path's length apart from the data traffic load (as in most of the cases both protocols' 95% confidence intervals pertaining to the route length do not overlap). In the case of proactive routing protocol, increasing the number of sink nodes inside the network does not result in fewer retransmissions, as the corresponding 95% confidence intervals overlap. If we individually consider the opportunistic routing protocol's mean number of retransmission as a function of the number of sink nodes, the confidence intervals overlap in the last three cases, i.e., when there were 2, 3, and 4 sink nodes inside a network. Increasing the number of sink nodes from 2 to 3 does not result in the performance difference that is statistically significant. But the mean number of retransmission in the case of 4 sink nodes is not in the overlap region when compared to the confidence intervals corresponding to 2 and 3 sink nodes scenarios. Hence, in these cases, we cannot state that the difference in the performance is statistically significant. Moreover, increasing the number of sink nodes from 1 to 2 results in the performance difference that is statistically significant.

The results presented here for the random setup in this section demonstrate that

both protocols perform similarly. The results in Section 7.2.1, the controlled setup, demonstrated that the proactive routing protocol outperforms the opportunistic routing protocol. In those simulations, the proactive protocol selected a different forwarding path for one of the flows, and the forwarding path only had one extra node. Therefore, looking at all the results (the controlled scenario simulation results and the random setup simulation results) suggests that a routing protocol should not only consider the end-to-end available bandwidth, but it should also consider a forwarding path's length. Therefore, a combined metric that trades-off end-to-end available bandwidth and a forwarding path's length may provide even better results.

7.3. BandEst Performance Over Proactive Routing Protocol

To evaluate the performance of BandEst using our proactive routing protocol, we run simulation experiments by running BandEst over the proactive routing protocol. General simulation parameters are shown in Table 7.1. Simulations were performed on the Cooja WSN simulator. Total network area is $500 \times 500\ m^2$. In a first scenario, we generated a random network topology of 100 wireless sensor nodes. In a second scenario, we generated a random network topology of 150 wireless sensor nodes. Each simulation scenario was repeated 25 times with different random seeds. Each simulation ran for 100 seconds. Four different source-destination pairs are randomly selected. The throughput of each connection is randomly distributed in the range [2-22] kbps. Nodes randomly select a frame size between 80 to 127 bytes.

Figure 7.5 shows the effectiveness comparison when BandEst is running over the shortest-hop count routing protocol (we do not choose the available-bandwidth-based opportunistic routing protocol, because the routing protocol can change the downstream relaying node depending on the downstream nodes' available bandwidth, and this is not allowed in BandEst) and when BandEst is running over the proactive routing protocol. Figure 7.5 demonstrates that in case of the 100 nodes network, statistically speaking, we cannot conclude anything about the difference in the mean effectiveness of BandEst using the two routing protocols, as the 95% confidence intervals overlap, but the mean value of BandEst's effectiveness using the shortest-hop count routing is not in the overlap region. But, in case of 150 nodes network, BandEst effectiveness is statistically significantly lower using the proactive routing protocol compared to the shortest-hop count routing. Thereby, the proactive routing protocol does not improve the BandEst performance.

Table 7.2 shows the number of wrong admission decisions of BandEst using both routing protocols. Table 7.2 shows that BandEst does not wrongly reject a single flow in both scenarios. Moreover, running BandEst over the proactive routing protocol results in a higher number of wrong accepts compared to running BandEst over the shortest-hop count routing protocol. The reason is that

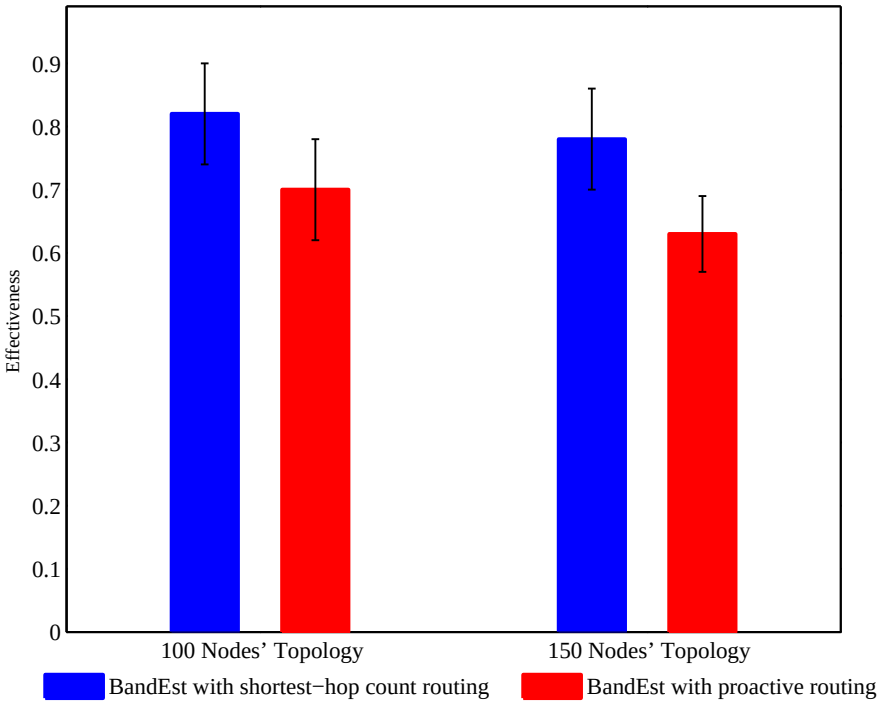


Figure 7.5.: Effectiveness Comparison

Table 7.2.: Number of Wrong Admission Decisions Comparison

Method	**100 Nodes**		**150 Nodes**	
	Wrong Accepts	**Wrong Rejects**	**Wrong Accepts**	**Wrong Rejects**
Shortest-hop routing	18	0	22	0
Proactive routing	30	0	37	0

the proactive routing protocol selects the data forwarding path based on the end-to-end available bandwidth, therefore in most cases, it selects lengthy data forwarding paths. Lengthy data forwarding paths result in higher number of broadcast control messages (bandwidth increment message of BandEst), and lost control messages results in inferior performance.

7.4. Conclusions

In this chapter, we presented an available-bandwidth-based proactive routing protocol for ad-hoc IEEE 802.15.4-based networks and its integration with BandEst. The comparison of the proactive routing protocol with the opportunistic routing protocol demonstrated that both protocols perform similarly in terms of the end-to-end delay and PDR. Moreover, for both protocols increasing the

number of sink nodes inside the network does not result in higher PDR and reduced end-to-end delay. The two routing techniques use two different and extreme approaches. The opportunistic routing protocol shortlists candidate downstream nodes that provide the same shortest hop-count towards the sink node, thereby it may have ignored slightly longer paths with better end-to-end available bandwidth. The proactive routing protocols selects the forwarding path based on the end-to-end available bandwidth, thereby it may have ignored shorter paths with slightly less available bandwidth. Therefore, a combined metric that trades-off end-to-end available bandwidth and a forwarding path's length may provide even better results.

8. Conclusions and Future Work

8.1. Conclusions

IEEE 802.15.4-based ad-hoc WSNs have their application in diverse domains, including application domains that can generate real-time multimedia data. Satisfying a real-time multimedia flow's QoS requirements considering the bandwidth supported by the IEEE 802.15.4 standard and the shared nature of the wireless communication medium is a challenging task. Therefore, the work presented in this dissertation focused on satisfying the real-time multimedia flows' QoS requirements in an IEEE 802.15.4-based ad-hoc WSN. In this context, we focused on the following: (i) the Contiki operating system's impact on an IEEE 802.15.4 channel throughput and node transmission and reception capabilities, (ii) suitability of the IEEE 802.15.4 unslotted CSMA-CA MAC layer protocol for real-time multimedia applications, (iii) the impact of the MAC layer on the available bandwidth, (iv) key factors of a proper flow admission control algorithm in ad-hoc wireless networks, and (v) available-bandwidth-based routing protocol. We have drawn the following conclusions from the work presented in this dissertation.

- The Contiki operating system limits a node's transmission capability, primarily due to its event handling mechanism and implementation of the networking protocol stack. The maximum rate at which a node can transmit data is approximately 40 kbps. Moreover, using the Contiki operating system the IEEE 802.15.4 channel throughput is approximately 180 kbps at maximum (the maximum bandwidth supported is 250 kbps).
- The unslotted CSMA-CA protocol without ACKs offers low end-to-end delay compared to the unslotted CSMA-CA protocol with ACKs. The unslotted CSMA-CA protocol with ACKs offers 0% packet loss rate, if the data traffic load within the interference range of a node is below 60 kbps. If the data traffic load within the interference range of a node is below 120 kbps, the unslotted CSMA-CA protocol with ACKs offers better throughput compared to the unslotted CSMA-CA protocol without ACKs. The unslotted CSMA-CA protocol without ACKs achieves better throughput if the data traffic load is in between 120 to 160 kbps. Typically, the IEEE 802.15.4 unslotted CSMA-CA protocol with ACKs offers better end-to-end throughput. But, in general, the choice of a suitable IEEE 802.15.4 unslotted CSMA-CA protocol depends on the requirements a real-time multimedia flow, data traffic load within the interference range of transmitters along the data forwarding path, and the length of the data

forwarding path.

- While estimating the available bandwidth in IEEE 802.15.4-based ad-hoc networks, the complete impact of the MAC layer overhead should be taken into account. Moreover, the available bandwidth estimator should proactively estimate the additional impact of the MAC layer overhead on the available bandwidth by considering the following: (i) the rate at which a new flow will generate data packet and (ii) the size of data packets. We demonstrated that proactively considering the additional MAC layer overhead results in a more effective available bandwidth estimator.
- A proper flow admission control algorithms for ad-hoc wireless networks should consider the following: (i) determine the correct intra-flow contention factor, (ii) determine the correct contention factor on nodes that are not on a data flow's forwarding path, but are within the interference range of transmitters along the data forwarding path, and (iii) when a new admission request is received at a node, the node must proactively take into account the additional complete CSMA-CA MAC layer overhead with an increased data traffic load (due to the flow's admission) not only at the node, but also at nodes which are within the interference range of the node (if those nodes are transmitting data). We designed BandEst that considers the identified factors listed above. Our results demonstrated that BandEst outperforms the state-of-the-art available-bandwidth-estimation-based flow admission control algorithms.
- Only using the end-to-end available bandwidth as a routing metric with a proactive routing protocol does not yield better performance compared to an available-bandwidth-based opportunistic routing protocol. Furthermore, considering both routing protocols, increasing the number of sink nodes inside a network does not result in higher PDR and reduced end-to-end delay. A combined routing metric that trades-off end-to-end available bandwidth and a forwarding path's length may provide better performance.

8.2. Future Work

The following are the possible future research opportunities.

- We noticed that having multiple transmitters make a difference on the additional MAC layer overhead. This can be incorporated into a future version of BandEst. This is not trivial as it requires an estimate of how many flows/transmitters there are within the interference range of a node estimating the MAC layer overhead.
- BandEst's flow admission control algorithm broadcasts control messages, the lost or corrupted control messages can impact the performance of BandEst. Therefore, in future it will be interesting to evaluate BandEst

performance using different values of the bit error rate.

- As the end-to-end available-bandwidth-based proactive routing protocol does not show a better performance compared to the available-bandwidth-based opportunistic routing protocol, there is a need to explore other metrics for a proactive routing protocol such as the end-to-end delay, or a routing metric that trades-off end-to-end available bandwidth and the length of a data forwarding path. The integration of BandEst with the modified routing protocol can also be of interest.

A. Appendix

A.1. Summary of Wireless Multimedia Sensing Nodes

Table A.1.: Summary of Wireless Multimedia Sensing Nodes

Name	**Processor**	**Memory**	**Multimedia Support**	**Wireless**
Stargate [42]	Intel PXA-55 Xscale processor at 40 MHz	32 MB flash and 64 MB SDRAM	High computational power. Embedded Linux OS. POSIX with Java and Pearl real-time environments	802.11 compact Flash, 802.15.4 through Mica2/z interface
Imote2 [42]	32-bit PXA271 Xscale processor at -400 MHz	256 KB SRAM, 32 MB Flash , and 32 MB SDRAM	Wireless MMX co-processor	Integrated 802.15.4, support for external radios through SDIO and UART
MeshEye [28]	32-bit ARM7TDMI RISC processor at 47.92 MHz	64 KB RAM and 256 KB Flash	Multiple resolution support	Integrated 802.15.4
Cyclops [52]	8-bit ATMEL ATmega128L micro-controller	512 KB Flash and 512 KByte RAM	On-board image processing, low-power, cost and size	Integrated 802.15.4
FireFly Mosaic [56]	60 MHz 32 bit LPC2106 ARM7TDMI MCU	128 KB Flash and 64 KB RAM	On board cc3 open source audio/video image processing library	Interfaced with Firefly mote, IEEE 802.15.4

FleckTM [32]	Atmega128 micro-controller running at 8MHz,	1 MB RAM	The DSP board holds a Texas Instruments 32bit 150 MHz DSP processor (TMS320F2812)	Nordic NRF905 radio transceiver with a bit rate of 76.8 kbps
WiSN [35]	32 bit ARM7TDMI based on Atmega AT91SAM7S, running at 48MHz	256 KB flash and 64 KB RAM	Camera based on Agilent ADCM-1670 352x288, 15 frames/sec. Agilent ADNS-3060 30x03, 30 frames/sec	Integrated 802.15.4

A.2. Comparison of WMSNs Testbeds

Table A.2.: Comparison of WMSNs Testbeds

WMSN Testbed Name	**Service Differentiation**	**Bulk Storage Capacity**	**Compression Algorithms**	**Separate DSP Processor**	**Multimedia Database**
BWN-Lab Testbed [2]	No	Yes	No	Yes	No
Explorebots [16]	No	Yes	No	Powerful single processor	No
A Lightweight Camera Network Operating on Symbolic Information [64]	No	No	Yes	No	No
IrisNet [57]	No	Yes	No	Yes	Yes
SenseEye [36]	No	Yes	Yes	Yes	No
Wireless Line Sensor Network for Distributed Visual Surveillance [15]	No	Yes	Yes	Yes	No
Hierarchical Character Oriented Wildlife Species Recognition through Heterogeneous WSNs [19]	No	Yes	Yes	Powerful single processor	No
TinyEARS [63]	No	Yes	No	Powerful single processor	No

Design and Implementation of Dual Camera WSN for Object Retrieval [67]	No	Yes	Yes	Yes	No
Design and Implementation of Sensor based Wireless Camera for Continuous Monitoring in Assistive Environments [39]	No	Yes	No	Powerful single processor	No
Distributed Image Search in Sensor Networks [70]	No	Yes	Yes	Powerful single processor	Yes
Fusion of Audio and Image Information for Efficient Object Detection and Capture [47]	No	Yes	No	Yes	No

A.3. Comparison of WSNs Testbeds

Table A.3.: Comparison of WSNs Testbeds

Testbed	**No. of Nodes**	**Hardware Hetero-geneity**	**Software Hetero-geneity**	**Availability**	**Deployment Scale**
WISEBED [11]	711	Yes	Supports multiple operating systems, provides implementation of network simulator, and software API library for application development	Public	Developed in multiple countries of Europe
SensLAB [59]	1000	No	Supports multiple operating systems, provides implementation of network simulator, and a software programming library	Through request	Four locations in France
moteLab [4]	190	Only Tmote	Only TinyOS	Public	Maxwell Dworkin Laboratory, Harvard University
CitySense [5]	100	No	Linux-based	Public	City-wide

Sensei [55]	20	Only TelosB	TinyOS and Contiki	Public	Lab. Level deployment at Uppsala University

Personal Publications

- Farooq, M. O., and Aziz, S. Qos based distributed multipath routing and admission control algorithm for ipv6. In *12th IEEE Multitopic Conference* (2008), pp. 323–328.

- Farooq, M. O., and Aziz, S. Stateless and controlled reservation based diffserv model for mobile ad-hoc networks. In *4th International Conference on Wireless and Mobile Communications* (2008), pp. 389–394.

- Farooq, M. O., and Aziz, S. Admission control and multipath routing algorithm for diffserv based networks. In *3rd International Conference on Complex, Intelligent and Software Intensive System* (2009), pp. 359–366.

- Farooq, M. O., and Aziz, S. Differentiated services based admission control and multi-path routing for ipv6. *Journal of Information Processing Systems 5*, 2 (2009), 97–104.

- Farooq, M. O., and Butt, A. Hybrid differentiated service architecture for qos routing in manet's. In *13th IEEE Multitopic Conference* (2009), pp. 1–6.

- Farooq, M. O., Dogar, A. B., and Aziz, S. State-of-the-art in wireless sensor networks operating systems: a survey. In *2nd International Conference on Future Generation in Information Technology* (2010), pp. 616–631.

- Farooq, M. O., Dogar, A. B., and Shah, G. Mr-leach: Multi-hop routing with low energy adaptive clustering hierarchy. In *4th International Conference on Sensor Technologies and Applications* (2010), pp. 262–268.

- Farooq, M. O., and Kunz, T. Operating systems for wireless sensor networks: A survey. *Sensors 11*, 6 (2011), 5900–5930.

- Farooq, M. O., and Kunz, T. Wireless multimedia sensor networks testbeds and state-of-the-art hardware: A survey. In *Future Generation in Communication and Networking* (2011), pp. 1–14.

- Farooq, M. O., and Kunz, T. Contiki-based ieee 802.15.4 node's throughput and wireless channel utilization analysis. In *5th IFIP Wireless Days* (2012), pp. 1–3.

- Farooq, M. O., and Kunz, T. BEAR: Bandwidth estimation-based admission control and routing for ieee 802.15.4-based networks. In *6th Joint IFIP Wireless and Mobile Networking Conference* (2013), pp. 1–4.

- Farooq, M. O., and Kunz, T. On determining bandwidth usage threshold to support real-time multimedia applications in wireless multimedia sensor

networks. In *27th International Conference on Advanced Information Networking and Applications Workshops* (2013), pp. 401–406.

- Farooq, M. O., and Kunz, T. Proactive bandwidth estimation for ieee 802.15.4-based networks. In *IEEE 77th Vehicular Technology Conference* (2013), pp. 1–5.

- Farooq, M. O., and Kunz, T. Available-bandwidth-based routing in ieee 802.15.4-based ad-hoc networks: Proactive vs. opportunistic technique. In *28th IEEE International Conference on Advanced Information Networking and Applications* (2014). to appear.

- Farooq, M. O., and Kunz, T. Contiki-based ieee 802.15.4 channel capacity estimation and suitability of its csma-ca mac layer protocol for real-time multimedia applications. *Mobile Information Systems* (2014). to appear.

- Farooq, M. O., and Kunz, T. Key factors for a proper available-bandwidth-based flow admission control in ad-hoc wireless sensor networks. In *8th International Workshop on Wireless Sensor, Actuator and Robot Networks* (2014). to appear.

- Farooq, M. O., and Kunz, T. Wireless sensor networks testbeds and state-of-the-art multimedia sensor nodes. *Advance Mathematics and Information Sciences Journal 8*, 3 (2014), 935–940.

- Farooq, M. O., Kunz, T., and St-Hilaire, M. Cross layer architecture for supporting multiple applications in wireless multimedia sensor networks. In *7th International Wireless Communication and Mobile Computing Conference* (2011), pp. 388–393.

- Farooq, M. O., Kunz, T., and St-Hilaire, M. Differentiated services architecture for qos provisioning in wireless multimedia sensor networks. In *4th IFIP Wireless Days* (2011), pp. 1–3.

- Farooq, M. O., Kunz, T., and St-Hilaire, M. Differentiated services based congestion control algorithm for wireless multimedia sensor networks. In *4th IFIP Wireless Days* (2011), pp. 1–6.

- Farooq, M. O., and Shah, S. A. Reactive qos routing protocol for mobile ad-hoc networks. *Ad-hoc and Sensor Wireless Networks 13*, 1-2 (2011), 13–38.

- Farooq, M. O., St-Hilaire, M., and Kunz, T. Cross-layer architecture for qos provisioning in wireless multimedia sensor networks. *KSII Transactions on Internet and Information Systems 6*, 1 (2012), 178–202.

List of Tables

List of Figures

List of Acronyms

Acronym	Explanation
ABE	Available Bandwidth Estimation
ACKs	Acknowledgments
BE	Back-off Exponent
BRuIT	Bandwidth Reservation Under Interference
CACP	Contention-Aware Admission Control Protocol
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CCI	Channel Check Interval
CSMA-CA	Carrier Sense Multiple Access Collision Avoidance
CTS	Clear to Send
DiffServ	Differentiated Services
FEC	Forward Error Correction
FIFO	First In First Out
GPS	Geographical Position System
GTSs	Guaranteed Time Slots
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IntServ	Integrated Services
ITU	International Telecommunication Union
KB	Kilo Byte
kbps	kilo bits per second
MAC	Medium Access Control
NB	Number of Back-offs
NS	Network Simulator
OS	Operating System
PAN	Personal Area Network
PDR	Packet Delivery Ratio
PHY	Physical
QoS	Quality of Service
RABE	Retransmission-based Available Bandwidth Estimation
RSVP	Resource Reservation Protocol
RTS	Request to Send
SFD	Start of Frame Delimiter
SLoPS	Self-Loading Periodic Stream
UDGM	Unit Disk Graph Model

WMSNs	Wireless Multimedia Sensor Networks
WSNs	Wireless Sensor Networks

List of Symbols

Symbol	Explanation
BE_{min}	Minimum value of a back-off exponent
BE_{max}	Maximum value of a back-off exponent
λ_{new}	New data arrival rate at a node
λ_{flow}	New flow's data arrival rate
$Delay_{total}$	Required end-to-end delay
Hop_{max}	Total number of hops between the source-destination pair
Hop_{rem}	Remaining number of hops to the destination node
MAC_{tx-max}	Maximum no. of bits that the MAC layer can transfer
FS_i	Total size in bytes of i^{th} frame in the MAC layer queue
T_{ub}	Theoretical upper-bound on the single hop throughput
α	Size of the averaging window
β	Data activity within the interference range of a node
β_i	The β value at the i^{th} index of the averaging window
γ_i	MAC layer overhead at the i^{th} index of the averaging window
ρ	Channel rate
CW_{size}	Size of the contention window
AN_{tx}	Average no. of packets transmitting by a node per second
$ACW_{overhead}$	Average contention window overhead
ω	Average available bandwidth
θ	Maximum size of the averaging window
η	Effectiveness of an admission control algorithm
ω_n	Average available bandwidth at node n
B_n	Net available bandwidth at node n
N	Set of nodes in a network
n	A node in a network
I_n	Set of nodes within the interference range of node n
S	Set of sink nodes in a network
s	A sink node in set S
K_s	List of data forwarding path(s) towards sink node s
$K_{s,j}$	A single data forwarding path in K_s
r	A relaying node on data forwarding path $K_{s,j}$
$B_{k_{s,j}}$	Available bandwidth on a data forwarding path $K_{s,j}$
d	Selected data forwarding path for a flow
P_n	A set of shortest-hop count values towards sink nodes
$P_{n,s}$	An item in set P_n

d	Selected sink node for a flow
$C_{n,d}$	Downstream nodes set at n with shortest-hop count towards sink d
$c_{n,d,i}$	A node in set $C_{n,d}$

Bibliography

[1] A.Dunkels, Gronvall, B., and Voigt, T. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th IEEE International Conference on Local Computer Networks* (2004), pp. 455–462.

[2] Akyildiz, I. F., Melodia, T., and Chowdhary, K. R. Wireless multimedia sensor networks: Applications and testbeds. *IEEE 96*, 10 (2008), 1588–1605.

[3] Akyildiz, I. F., Melodia, T., and Chowdhury, K. R. A survey on wireless multimedia sensor networks. *Computer Networks 41*, 4 (2007), 921–960.

[4] Allen, G. W., Swieskowsli, P., and Welsh, M. MoteLab: a wireless sensor network testbed. In 4th *International Symposium on Information Processing in Sensor Networks* (2005), pp. 483–488.

[5] Allen, G. W., Swieskowsli, P., and Welsh, M. CitySense: an urban-scale wireless sensor network and testbed. In *IEEE Conference on Technologies for Homeland Security* (2008), pp. 583–588.

[6] Alwan, H., and Agarwal, A. Multi-objective reliable multipath routing for wireless sensor networks. In *Globecom workshops* (2010), pp. 1227–1231.

[7] Alwan, H., and Agarwal, A. Reliable fault-tolerant multipath routing protocol for wireless sensor networks. In 25th *Biennial Symposium on Communication* (2010), pp. 323–326.

[8] Baccour, N., Koubâa, A., Mottola, L., niga, M. A. Z., Youssef, H., Boano, C. A., and Alves, M. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transactions on Sensor Networks 8*, 4 (2012), 1–33.

[9] Blake, S., D. Black, M. C., Davies, E., Wang, Z., and Weiss, W. *An Architecture for Differentiated Services RFC 2475.* IETF, December 1998.

[10] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S. *Resource Reservation Protocol (RSVP) version 1, Functional Specifications RFC 2205.* IETF, September 1997.

[11] Chatzigiannakis, I., Koninis, C., Mylonas, G., Fischer, S., and Pfister, D. WISEBED: an open large-scale wireless sensor network testbed. In 1st *International Conference on Sensor Network Applications,*

Experimentation, and Logistics (2009), pp. 68–87.

[12] Chaudet, C., and Lassous, I. G. Bruit: Bandwidth reservation under interferences influence. In *European Wireless* (2002).

[13] Cheng, L., Cao, J., Chen, C., Ma, J., and Das, S. K. Exploiting geographic opportunistic routing for soft qos provisioning in wireless sensor networks. In *IEEE* 7^{th} *International Conference on Mobile Adhoc and Sensor Systems* (2010), pp. 292–301.

[14] Chitnis, M., Liang, Y., Zheng, J. Y., Pagano, P., , and Lipari, G. Wireless line sensor network for distributed visual surveillance. In 6^{th} *ACM Symposium on Performance Evaluation of Wireless Ad hoc, Sensor, and Ubiquitous Networks* (2009), pp. 71–78.

[15] Chitnis, M., Liang, Y., Zheng, J. Y., Pagano, P., and Lipari, G. Wireless line sensor network for distributed visual surveillance. In *ACM symposium on Performance evaluation of wireless ad-hoc, sensor, and ubiquitous networks* (2009), pp. 71–78.

[16] Dahlberg, T. A., Nasipuri, A., and Taylor, C. Explorebots: A mobile network experimentation testbeds. *ACM SIGOPS Operating Systems Review 35*, 2 (2001), 61–77.

[17] Djenouri, D., and Balasingham, I. Traffic-differentiation-based modular qos localized routing for wireless sensor networks. *IEEE Transactions on Mobile Computing 10* (2011), 797–809.

[18] Dunkels, A. Rime - a lightweight layered communication stack for sensor networks. In *European Conference on Wireless Sensor Networks* (2007), pp. 1–2.

[19] Duran, D., Peng, D., Sharif, H., Chen, B., and Armstrong, D. Hierarchical character oriented wildlife species recognition through heterogeneous wireless sensor networks. In 18^{th} *IEEE International Symposium on Personal, Indoor, and Mobile Radio Communication* (2007), pp. 1–5.

[20] Edwards, J., Demers, F., M. St-Hilaire, M., and Kunz, T. Comparison of ns2.34's zigbee/802.15.4 implementation to memsic's iris motes. In 7^{th} *International Wireless Communication and Mobile Computing Conference* (2011), pp. 986–991.

[21] Fan, S., Li, J., Sun, H., and Wang, R. Throughput analysis of gts allocation in beacon enabled ieee 802.15.4. In 3^{rd} *International Conference on Computer Science and Information Technology* (2010), pp. 561–565.

[22] Farooq, M. O., and Kunz, T. Operating systems wireless sensor networks: A survey. *Sensors 11*, 6 (2011), 5900–5930.

[23] Farooq, M. O., and Kunz, T. Proactive bandwidth estimation for ieee 802.15.4-based networks. In *IEEE* 77^{th} *Vehicular Technology Conference (VTC-Spring)* (2013), pp. 1–5.

[24] Felemban, E., Lee, C., and Ekici, E. MMSPEED: Multipath multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing 5*, 6 (2006), 738–754.

[25] Fonoage, M., Cardei, M., and Ambrose, A. A qos based routing protocol for wireless sensor networks. In 29^{th} *IEEE International Performance Computing and Communication Conference* (2010), pp. 1097–2641.

[26] Golomb, S. W. Mathematical models - uses and limitations. *Simulation 4*, 14 (1970), 197–198.

[27] He, T., Stankovic, J. A., Lu, C., and Abdelzaher, T. SPEED: A stateless protocol for real time communication in sensor networks. In *23^{r}d International Conference on Distributed Computing System* (2003), pp. 46–55.

[28] Hengstler, S., Prashanth, D., Fong, S., and Aghajan, H. MeshEye: a hybrid-resolution smart camera mote for application in distributed intelligent surveillance. In 6^{th} *International Symposium on Information Processing in Sensor Networks* (2007), pp. 360–369.

[29] Jain, M., and Dovrolis, C. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. *IEEE/ACM Transactions on Networking 11*, 4 (2003), 537–549.

[30] Jindal, A., Psounis, K., and Liu, M. Capest: A measurement-based approach to estimating link capacity in wireless networks. *IEEE Transactions on Mobile Computing 11*, 12 (2012), 2098–2108.

[31] Jurcik, P., Koubaa, A., Alves, M., Tovar, E., and Hanzalek, Z. A simulation model for the ieee 802.15.4 protocol: delay/throughput evaluation of the gts mechanism. In 15^{th} *International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems* (2007), pp. 109–116.

[32] Karlsson, J., Wark, T., Valencia, P., Ung, M., and Corke, P. Demonstration of image compression in low-bandwidth wireless camera network. In *International Conference on Information Processing in Sensor Networks* (2007), pp. 557–558.

[33] Khoukhi, L., Badis, H., Merghem-Boulahia, L., and Esseghir, M. Admission control in wireless ad hoc networks: A survey. *EURASIP Journal on Wireless Communications and Networking 2013*, 1 (2013), 1–13.

[34] Kim, J., Seo, H., and Kwak, J. Routing protocol for heterogeneous hierarchical wireless multimedia sensor networks. *Wireless Personal Communications* (2011), 1–11.

[35] Kleihorst, R., Abbo, A., Schueler, B., and Danilin, A. Camera mote with high-performance parallel processor for real-time frame-based

video processing. In *IEEE Conference on Advanced Video and Signal based Surveillance* (2007), pp. 69–74.

[36] Kulkarni, P., Ganesan, D., Shenoy, P., and Lu, Q. Senseeye: A multi-tier camera sensor network. In *ACM Multimedia* (2005), pp. 229–238.

[37] Kumar, A., Namboothiri, P. G., Deshpande, S., Vidhyadharan, S., Sivalingam, K. M., and Murty, S. A. V. Testbed based throughput analysis in a wireless sensor network. In *National Conference on Communication* (2012), pp. 1–5.

[38] Lee, T. J., Lee, H. R., and Chung, M. Y. Mac throughput limit analysis of slotted csma/ca in ieee 802.154. wpan. *IEEE Communication Letters 10*, 7 (2006), 561–563.

[39] Li, N., Yan, B., Chen, G., Govindaswamy, P., and Wang, J. Design and implementation of a sensor based wireless camera system for continuous monitoring in assistive environments. *Personal and Ubiquitous Computing Journal 14*, 6 (2010), 499–510.

[40] Liu, M., Xu, S., and Sun, S. An agent-assisted qos-based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications 35*, 1 (2012), 29 – 36.

[41] Mainwaring, A., Polastre, J., Culler, R. S. D., and Anderson, J. Wireless sensor networks for habitat monitoring. In *1st ACM International Workshop on Wireless Sensor Networks and Applications* (2002), pp. 88–97.

[42] MEMSIC. [online], `http://www.memsic.com`. (Last accessed on 13 March, 2014).

[43] Modeler, O. [online], `http://www.opnet.com/solutions/network_rd/modeler.html`. (Last accessed on 13 March, 2014).

[44] Mohideen, K., and Banu, V. Discussion on improving quality of service through available bandwidth estimation in mobile ad hoc networks. *International Journal of Computer Applications 11*, 8 (2010), 18–20.

[45] Nam, N. V., Guerin-Lassous, I., Victor, M., and Cheikh, S. Retransmission-based available bandwidth estimation in ieee 802.11-based multihop wireless networks. In *14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2011), pp. 377–384.

[46] of ITU, T. S. S. *ITU-T Recommendation G.114: Transmission Systems and Media : General Recommendations on the Transmission Quality for an Entire International Telephone Connection : One-Way Transmission Time*. International Telecommunication Union, 1994.

[47] O'Rourke, D., Moore, D., and Wark, T. Fusion of audio and image

information for efficient object detection and capture. In *International Conference on Information Processing in Sensor Networks* (2009), pp. 401–402.

[48] Osterlind, F., and Dunkels, A. Approaching the maximum 802.15.4 multi-hop throughput. In *5th ACM Workshop on Embedded Network Sensors* (2008), pp. 1–12.

[49] Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., and Voigt, T. Cross-level sensor network simulation with cooja. In *31st IEEE Conference on Local Computer Networks* (2006), pp. 641–648.

[50] Othman, J. B., and Yahya, B. An energy efficient and qos based routing protocol for wireless sensor networks. *Parallel and Distributed Computing 70* (2010), 849–857.

[51] Prasad, R. S., Murray, M., Dovrolis, C., and Claffy, K. Bandwidth estimation: Metrics, measurement techniques, and tools. *IEEE Networks 17*, 6 (2003), 27–35.

[52] Rahimi, M., Baer, R., Iroezi, O. I., Garcia, J. C., Warrior, J., Estrin, E., and Srivastav, M. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *the proc. of the 3rd International Conference on Embedded Network Sensor Systems* (2005), pp. 192–204.

[53] Rajkumar, R., Lee, I., Sha, L., and Stankovic, J. Cyber-physical systems: The next computing revolution. In *47th IEEE/ACM Design Automation Conference* (2010), pp. 731–736.

[54] Ramanathan, P., Dovrolis, C., and Moore, D. What do packet dispersion techniques measure? In *IEEE Computer and Communications conference* (2001), pp. 905–914.

[55] Rensfelt, O., Hermans, F., Ferm, C., Gunningberg, P., and Larzon, L. A. Sensei-UU: a nomadic sensor network testbed supporting mobile nodes. In *4th ACM International Workshop on Wireless Networks Testbeds, Experimental Evaluation, and Characterization* (2010), pp. 612–614.

[56] Rowe, A., Goel, D., and Rajkumar, R. FireFly Mosaic: a vision enabled wireless sensor networking system. In *28th International Real Time System Symposium* (2007), pp. 459–468.

[57] Rowe, A., Rosenberg, C., and Nourbakhsh, I. A low cost embedded color vision system. In *IEEE/RSJ International Conference on Intelligent Robotics System* (2002), pp. 208–213.

[58] Sarr, C., Chaudet, C., Chelius, G., and Lassous, I. G. Bandwidth estimation for ieee 802.11-based ad hoc networks. *IEEE Transactions on Mobile Computing 10*, 7 (2008), 1228–1241.

[59] Senslab. [online], `http://www.senslab.info`. (Last accessed on 13

March, 2014).

[60] Spachos, P., Song, L., and Hatzinakos, D. Performance comparison of opportunistic routing schemes in wireless sensor networks. In 9^{th} *Annual Communication Networks and Services Research Conference* (2011), pp. 271–277.

[61] Spachos, P., Song, L., and Hatzinakos, D. Energy aware opportunistic routing in wireless sensor networks. In *IEEE Global Telecommunications Conference (GLOBECOM)* (2012), pp. 405–409.

[62] Sun, B., and Makki, S. K. TORP: Tinyos opportunistic routing protocol for wireless sensor networks. In *IEEE International Conference on Consumer Communications and Networking* (2011), pp. 111–115.

[63] Taysi, Z. C., Guvensan, M. A., and Melodia, T. TinyEARS: spying on house appliances with audio sensor nodes. In 2^{nd} *ACM Workshop on Embedded Sensing Systems for Energy Efficiency in Building* (2010), pp. 31–36.

[64] Teixerira, T., Lymberopoulos, D., Culurciello, E., Aloimonos, Y., and Savvides, A. A lightweight camera sensor network operating on symbolic information. In *ACM Workshop on Distributed Smart Cameras* (2006), pp. 76–81.

[65] Venkataraman, M., Chatterjee, M., and Kwiat, K. Traffic based dynamic routing for wireless sensor networks. In *IEEE Conference on Wireless Communications & Networking Conference* (2009), pp. 2619–2624.

[66] Xie, D., Yan, T., Ganesan, D., and Hanson, A. Design and implementation of a dual-camera wireless sensor network for objective retrieval. In 7^{th} *International Conference on Information Processing in Sensor Networks* (2008), pp. 469–480.

[67] Xie, D., Yan, T., Ganesan, D., and Hanson, A. Design and implementation of a dual-camera wireless sensor network for objective retrieval. In 7^{th} *International Conference on Information Processing in Sensor Networks* (2008), pp. 469–480.

[68] Xu, Y., Deng, J. D., and Nowostawski, M. Quality of service for video streaming over multi-hop wireless networks: Admission control approach based on analytical capacity estimation. In *IEEE Conference on Intelligent Sensors, Sensor Networks and Information Processing* (2013), pp. 345–350.

[69] Yahya, B., and Othman, J. B. An energy efficient and qos aware routing multipath routing protocol for wireless sensor networks. In 34^{th} *IEEE Local Computer Networks Conference* (2009), pp. 93–100.

[70] Yan, T., Ganesan, D., and Manmatha, R. Distributed image search in sensor networks. In 6^{th} *ACM conference on embedded network sensor system* (2008), pp. 155–168.

[71] Yang, Y., and Kravets, R. Contention-aware admission control for ad hoc networks. *IEEE Transactions on Mobile Computing 4*, 4 (2005), 363–377.

[72] Yigitel, M. A., Incel, O. D., and Ersoy, C. Qos-aware mac protocols for wireless sensor networks: A survey. *Computer Networks 55* (2011), 1982–2004.

[73] Youn, J. S., Packl, S., and Hong, Y. G. Distributed admission control protocol for end-to-end qos assurance in ad hoc wireless networks. *Eurasip Journal of Wireless Communication and Networking 1*, 163 (2011), 1–18.

[74] Yuyan, X., Ramamurthy, B., and Vuran, M. C. A service differentiation real time communication scheme for wireless sensor networks. In 33^{rd} *IEEE Conference on Local Computer Networks* (2008), pp. 748–755.

[75] Zhou, H., Wang, Y., and Wang, Q. Measuring internet bottlenecks: Location, capacity, and available bandwidth. *Networking and Mobile Computing* (2005).