

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE
DO NORTE

GLADERSON PEREIRA BELIZIO DA SILVA

**TRAFFIC BEE: PROPOSTA DE APLICAÇÃO DA TECNOLOGIA ZIGBEE EM
SISTEMA INTELIGENTE DE SEMÁFOROS**

Natal (RN), Janeiro de 2018

GLADERSON PEREIRA BELIZIO DA SILVA

**TRAFFIC BEE: PROPOSTA DE APLICAÇÃO DA TECNOLOGIA ZIGBEE EM
SISTEMA INTELIGENTE DE SEMÁFOROS**

Trabalho de Conclusão do Curso Superior de Tecnologia em Redes de Computadores, apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, como requisito para a obtenção do título de Tecnólogo.

Orientador: Professor M. Sc. Allan Soares

Natal (RN), Janeiro de 2018

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS NÃO LUCRATIVOS, DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE E PRESERVADOS OS DEVIDOS DIREITOS DO AUTOR.

Silva, Gladerson Pereira Belizio da.

S586t Traffic Bee: proposta de aplicação da tecnologia Zigbee em sistema inteligente de semáforos / Gladerson Pereira Belizio da Silva. – Natal, 2018.

58 f : il. color.

Trabalho de Conclusão de Curso (Tecnologia em Redes de Computadores) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. Natal, 2018.

Orientador: M.e Allan Aminadab André Freire Soares.

1. Zigbee – Protocolo de comunicação. 2. Semáforo. 3. Tecnologia. I. Soares, Allan Aminadab André Freire. II. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. III. Título.

CDU 004.7

Catálogo na Publicação elaborada pela Bibliotecária Roberta Jerônimo da Silva CRB15: 761

Biblioteca Central Sebastião Fernandes (BCSF) - IFRN

GLADERSON PEREIRA BELIZIO DA SILVA

**TRAFFIC BEE: PROPOSTA DE APLICAÇÃO DA TECNOLOGIA ZIGBEE EM
SISTEMA INTELIGENTE DE SEMÁFOROS**

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Redes de Computadores, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Redes de Computadores.

Trabalho de Conclusão de Curso apresentado e aprovado em 31/01/2018, pela seguinte Banca Examinadora:

Allan Aminadab A. F. Soares.

Allan Aminadab André Freire Soares, M. Sc. – Presidente
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Felipe Sampaio Dantas da Silva

Felipe Sampaio Dantas da Silva, M. Sc. – Membro da Banca Examinadora
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Ronaldo Maia de Medeiros

Ronaldo Maia de Medeiros, M. Sc. – Membro da Banca Examinadora
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Dedico este trabalho ao meu amado Jesus, o Cristo.

*“O coração do homem planeja o seu caminho,
mas o SENHOR lhe dirige os passos.”*

(Provérbios 16:9)

AGRADECIMENTOS

Agradeço primeiramente ao meu Deus, em quem encontrei forças.

À minha esposa Jéssika, minha fiel ajudadora, companheira em todos os momentos da minha existência.

Aos meus pais, Djalma e Bernadete, que desde o início me incentivaram a concluir essa etapa da vida.

Ao meu amigo Pr. Fábio Cavalcante, que me ensinou a disciplina necessária para completar esse desafio.

Ao meu orientador Prof. Allan Soares, pela rara dedicação prestada aos seus alunos e o admirável conhecimento ministrado.

Enfim, à todos os que por algum motivo contribuíram para a realização desta pesquisa, sejam familiares, membros da Diretoria de Ensino ou colegas de sala.

RESUMO

Este trabalho aborda o uso dos sistemas inteligentes de semáforos como alternativa para a problemática do trânsito nas grandes cidades valendo-se do uso da tecnologia Zigbee como meio de comunicação sem fio entre dispositivos. O projeto denominado *Traffic Bee* consiste primordialmente em adequar os tempos de abertura e fechamentos dos semáforos à necessidade de fluidez das vias. O *Traffic Bee* propõe um cenário em que durante cada ciclo de funcionamento dos semáforos, sensores contam a quantidade de veículos a certa distância do cruzamento, enviando tais informações à um dispositivo controlador responsável por adequar os tempos de *siga e pare* de cada sinaleiro no próximo ciclo semafórico à necessidade de fluidez de cada uma das vias, sendo toda a comunicação de dados sem fio realizada por meio do protocolo Zigbee.

Palavras-chave: Zigbee. Semáforos. Cidades Inteligentes.

ABSTRACT

This work broach the use of intelligents traffic lights systems as an alternative to the traffic problems in large cities using Zigbee technology as a wireless communication between devices. The project called Traffic Bee consists primarily in adapting the opened and closed traffic lights to the need of the roadway flows. The Traffic Bee proposes a scenario where during each cycle of traffic lights, sensors will count the number of vehicles at a certain distance from the intersection and sending this information to a controlling device responsible for adjusting the going and stopping from each traffic signal in the next traffic light cycle for the need of fluidity of each one pathways, it will all be performed by wireless data communication using the Zigbee protocol.

Keywords: Zigbee. Traffic Lights. Smart Cities.

LISTA DE ILUSTRAÇÕES

Figura 1 – Setores de aplicação do Zigbee	22
Figura 2 – Estrutura do Superframe	26
Figura 3 – Transferência de dados de um dispositivo para o coordenador em rede não sinalizada	27
Figura 4 – Transferência de dados do coordenador para um dispositivo em rede não sinalizada	28
Figura 5 – Arduino UNO R3	32
Figura 6 – Shield XBee Pro	33
Figura 7 – Adaptador USB XBee Explorer	34
Figura 8 – Módulo RF XBee	34
Figura 9 – Controlador dos sinalizadores	36
Figura 10 – Funcionamento experimental do Traffic Bee	37
Figura 11 – Interface do programa XCTU	37
Figura 12 – Interface do programa XCTU	38
Figura 13 – Cenário real: cruzamento de vias	40
Figura 14 – Traffic Bee: diagrama de projeto para cenário real	41
Figura 15 – Comparação entre o necessário para um cenário real e o utilizado nos testes	42
Figura 16 – Erro de pouca memória disponível	43
Figura 17 – Linhas de retenção e área de conflito	45
Figura 18 – Console XCTU: fornecendo valores determinísticos	51
Figura 19 – Console XCTU: tempos de verdes	51

LISTA DE QUADROS

Quadro 1 – Tipos de dispositivo definidos pela IEEE 802.15.4	24
Quadro 2 – Relação de dispositivos entre IEEE e o protocolo Zigbee	24

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgment
AES	Advanced Encryption Standard
ANATEL	Agência Nacional de Telecomunicações
ANTP	Agência Nacional de Transportes Públicos
API	Application Programming Interface
APL	Application
AVR	Automatic Voltage Regulator
bps	Bits por segundo
CAP	Contention Access Period
CD	Compact Disc
CFP	Contention Free Period
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DENATRAN	Departamento Nacional de Trânsito
DIP	Dual In-Line Package
DSSS	Direct Sequence Spread Spectrum
DVD	Digital Video Disc
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EIRP	Effective Isotropic Radiated Power
FFD	Full Function Device
GND	Graduated Neutral Density Filter
GPS	Global Positioning System
GTS	Guaranteed Time Slot
IEEE	Institute of Electrical and Electronics Engineers
IOREF	Input/output Reference
Kbps	Kilobit por segundo
LED	Light Emitting Diode
LQ	Link Quality
MAC	Media Access Control
Mbps	Megabit por segundo
NWK	Network
OSI	Open Systems Interconnection

PAN	Personal Area Network
PC	Personal Computer
PCB	Printed Circuit Board
PDU	Protocol Data Unit
PHY	Physical
PIB	Produto Interno Bruto
Poli-USP	Escola Politécnica da Universidade de São Paulo
RAN	Random Access Memory
RF	Rádio Frequência
RFD	Reduce Function Device
RISC	Reduced Instruction Set Computer
TIC	Tecnologia da Informação e Computação
TV	Televisão
USB	Universal Serial Bus
VCR	Video Cassette Recorder
VIN	Voltage Input
WPAN	Wireless Personal Area Network
XCTU	XBee Configuration and Test Utility

LISTA DE SÍMBOLOS

μA	Microampere
Bit	Dígito binário (0 ou 1)
C++	Linguagem de programação de uso geral
dBm	Decibel Miliwatt
GHz	Giga-hertz
Hz	Hertz
KB	Kilobyte
km	Quilômetro
m	Metro
mA	Miliampere
MHz	Mega-hertz
mW	Miliwatt
mW	Miliwatt
°C	Grau Celsius
V	Volt
W	Watt
Ω	Ohm

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	17
1.2	TRABALHOS RELACIONADOS	18
1.3	ESTRUTURA DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	TRÂNSITO E SEMÁFOROS	20
2.2	O ZIGBEE	21
2.2.1	Arquitetura do Protocolo	22
2.2.2	Tipos de Dispositivos	23
2.2.3	Acesso ao Meio (Subcamada MAC)	24
2.2.4	Transferência de Dados	26
2.2.5	Associação e Desassociação de Dispositivos	28
2.3	O ARDUINO	29
3	A PROPOSTA TRAFFIC BEE: PROTÓTIPO E CENÁRIO REAL	31
3.1	TRAFFIC BEE PROTÓTIPO: COMPONENTES E CARACTERÍSTICAS	31
3.2	TRAFFIC BEE PROTÓTIPO: IMPLEMENTAÇÃO E CONFIGURAÇÃO	35
3.2.1	Módulos RF Xbee: Configuração e Modo de Funcionamento	37
3.3	TRAFFIC BEE: IMPLEMENTAÇÃO NUM CENÁRIO REAL	39
4	TRAFFIC BEE: DESAFIOS, TEMPORIZAÇÃO E RESULTADOS	42
4.1	DESAFIOS NA IMPLEMENTAÇÃO DO PROTÓTIPO	42
4.2	TEMPORIZAÇÃO: ENTREVERDES E VERDES	44
4.2.1	Tempo de Entreverdes: Amarelo e Vermelho Geral	44
4.2.2	Tempo de Verde	46
4.3	SIMULAÇÕES E RESULTADOS	48
5	CONCLUSÕES E PERSPECTIVAS FUTURAS	53
	REFERÊNCIAS	55

1 INTRODUÇÃO

Em companhia ao avanço do comércio, serviços e viagens, cresce também a demanda por mobilidade. Todas as riquezas de uma nação fluem pelas vias ao longo do seu território, chegam às fábricas, escoam para os centros urbanos das grandes cidades e chegam às casas, em forma de produtos ou serviços. Além dos caminhões, ônibus, vans e utilitários que fazem esse sistema de comércio acontecer, tem crescido também o número de carros de passeio nas ruas, tornando um espaço que seria suficiente para 15 pessoas em um transporte público, ocupado por um só cidadão em seu veículo particular (GRANDELLE, 2016). E a expectativa é de que esse fenômeno cresça, visto que ainda são poucos os investimentos no transporte público e cada vez maiores as possibilidades de crédito para compra de veículos.

Ao longo de décadas, as grandes cidades têm investido em soluções para esse problema. E a tecnologia tem ganhado espaço nesse contexto. Em 1914, na cidade de Cleveland, em Ohio, nos Estados Unidos, foi criado o primeiro semáforo luminoso, com a intenção de organizar e permitir a fluidez do trânsito, cada vez mais disputado por carros e pedestres. E hoje é difícil imaginar o cenário de um cruzamento de avenidas sem a presença de semáforos. Segundo o Departamento Nacional de Transito (DENATRAN, 1984 apud REGO; SEMENTE, 2017, p.1), “Em um cruzamento entre duas ou mais vias existem movimentos que não podem ser realizados simultaneamente, pois são conflitantes entre si”. Então, para evitar conflitos como esse, faz-se necessário estabelecer regras de prioridade de fluxo, ou seja, regras que definem quem tem o “direito de passar primeiro”.

Em companhia ao avanço organizacional que os semáforos trouxeram, surgiu um novo problema: o congestionamento. Segundo dados DENATRAN (2017), a frota brasileira de veículos passou de 29.722.950, em 2000, para 95.099.183, em 2017. Deste modo, o espaço nas ruas tem se tornado cada vez mais concorrido. Sendo assim, a tarefa de programar um semáforo tornou-se mais complexa, requerendo bastante cuidado e estudo, a fim de evitar a formação de filas, provocadas pela interrupção das vias.

Em sua maioria, a programação dos semáforos é determinada por **tempos fixos** – os departamentos de engenharia de trânsito dedicam-se à analisar o comportamento do tráfego nas cidades (suas variações de densidade de acordo com cada horário do dia), para assim elaborar os melhores planos semaforicos (tempos de abertura e fechamento) possíveis. Contudo, devido às variações de fluxo que uma via pode apresentar, os semáforos de tempo fixo podem não ser tão ideais, visto que o comportamento do trânsito quase sempre não obedece à oscilações padrões. Por exemplo, não há como prever a decisão humana de escolha pelo melhor caminho,

porém se a massa de motoristas escolher trafegar por uma avenida “x”, porque “y” encontra-se interrompida, os semáforos ao de longo de “x” deveriam adequar-se a esse aumento da demanda. Aos semáforos que possuem a capacidade de alternância de luzes, verde (*siga*) e vermelho (*pare*), com temporização adaptável à quantidade de veículos presentes, dá-se o nome de **semáforos inteligentes**.

O congestionamento é uma externalidade que causa diversos transtornos, inclusive financeiros. O custo dos congestionamentos é estimado entre 1 e 3% do PIB dos países por ano, isso é decorrência do tempo parado, improdutivo. Além do tempo improdutivo, os congestionamentos aumentam o consumo de energia e intensificam a poluição atmosférica. (ANTP, 2012 apud DA SILVEIRA, 2016, p. 42).

Além disso, as repedidas manobras de frenagem e aceleração, exigidas em um engarrafamento, provocam um maior desgaste de peças e, sucessivamente maior consumo de combustível, devido às trocas de marcha e ao tempo improdutivo (ESCOLA POLITÉCNICA DA USP, 2012).

Buscando resolver o desafio de encontrar a melhor resposta para o fluxo ótimo de veículos nas cidades, evitando congestionamentos e tempos excessivos de paradas em semáforos, diversas soluções inteligentes que utilizam equipamentos eletrônicos que se comunicam por radiofrequência (RF) para produzir respostas em tempo real têm surgido e ganhado cada vez mais espaço no mercado. Nesse sentido, atualmente tornou-se comum o uso de aplicativos baseados em Sistemas de Posicionamento Global, *Global Positioning System* (GPS), que podem ser facilmente acessados por um motorista que tenha um *smartphone* à mão conectado a uma rede de dados. A título de exemplo, destacamos o *Google Maps*¹ e o *Waze*². Esses sistemas detectam em tempo real, baseados na coleta de informações, o comportamento do trânsito e avisam aos motoristas sobre congestionamentos, ruas interditadas, acidentes, etc., além de sugerir as melhores rotas para uma viagem. Em algumas grandes cidades existem também painéis eletrônicos de mensagens, que exibem constantemente avisos aos motoristas, permitindo a eles a melhor escolha de trajeto e alertando-os sobre perigos em vias específicas.

Essa tecnologia atual também trouxe à existência os semáforos inteligentes, que pouco à pouco deixam de ser um vislumbre do futuro e passam a ser uma realidade presente. Em maio de 2017, 88 semáforos inteligentes começaram a funcionar em Salvador, na Bahia, prometendo aumentar em até 30% a fluidez do trânsito na cidade, segundo CORREIO (2017). Os tempos de abertura e fechamento dos semáforos se adequam à necessidade de fluxo nas vias, permitindo que a população perca menos tempo no trânsito. Por meio de câmeras de alta resolução, o

¹ Serviço de pesquisa e visualização de mapas e imagens de satélite fornecido e desenvolvido pela Google

² Aplicação de trânsito para smartphones ou dispositivos similares baseada na navegação por satélite.

sistema contabiliza a quantidade de veículos nos cruzamentos e reprograma os semáforos em tempo real. A implementação do sistema exigiu um investimento de R\$12 milhões na época.

Este trabalho, o *Traffic Bee*, propõe um sistema inteligente de semáforos em que toda conexão é realizada via rede **Zigbee**, tecnologia que inspirou o nome do projeto: semáforos (*Traffic*) + Zigbee (*Bee*). O Zigbee é um padrão de rede sem fio definido pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), modelo 802.15.4, e desenvolvido pela *Zigbee Alliance* (SIEMEINTCOSKI, 2009), união de mais de 45 empresas que buscam desenvolver uma solução de controle seguro de dispositivos, com baixo custo e à baixa potência em redes sem fio de até 100 metros em suas versões mais comuns. No Brasil, a tecnologia Zigbee comercializada opera numa frequência de rádio de 2,4 *giga-hertz* (GHz) à uma taxa de transferência de 250 *kilobits* por segundo (Kbps).

Para que a implementação da tecnologia Zigbee fosse possível, os módulos RF (radiofrequência) utilizados neste projeto foram instalados em placas microcontroladoras Arduino (seção 2.3), que possibilitaram o controle do sistema via código computacional previamente desenvolvido para tal fim e nelas armazenado.

Ainda é escassa a presença de semáforos inteligentes nas ruas, especialmente do Brasil, mas seu crescimento tem ganhado ritmo noutros países. Estudos sobre esse tema têm atraído o interesse de estudantes, pesquisadores, empresas e governos no mundo inteiro.

1.1 OBJETIVOS

Este trabalho consiste na apresentação de uma alternativa tecnológica na área da segurança e otimização da mobilidade urbana, incluindo um breve estudo sobre o tema, além do desenvolvimento do protótipo de quatro semáforos para quatro vias que se intersectam, com todos os sinalizadores conectados e interagindo entre si via rede sem fio de médio alcance, permitindo a temporização automática dos dispositivos de semaforização de acordo com o volume de veículos nas vias, projeto esse intitulado como *Traffic Bee*.

O *Traffic Bee* propõe uma melhoria na circulação de veículos automotivos nas grandes cidades, além de auxiliar na exposição a situações de assalto em semáforos durante o período da noite e em áreas de risco. Isso porque, além da problemática que gira em torno da mobilidade, existe também a preocupação relacionada à segurança pública no que diz respeito aos semáforos. Durante o período da madrugada, os cruzamentos de vias têm se tornado zonas de risco para os motoristas, visto que obedecer ao sinal vermelho e parar em alguns horários pode facilitar a ação de criminosos; por outro lado, desobedecer às sinaleiras pode gerar acidentes

graves. O uso de semáforos inteligentes tem importante contribuição também nesse sentido da segurança, pois o sinal verde (*sigla*) é sempre garantido quando não há fluxo na via concorrente, minimizando os riscos para os motoristas. Podemos dizer que os semáforos inteligentes permitem aos motoristas não terem que decidir entre o risco de um abalroamento e a vulnerabilidade a um assalto.

1.2 TRABALHOS RELACIONADOS

O *Traffic Bee* segue a essência das cidades inteligentes (*Smart Cities*), propondo melhor qualidade de vida, uso inteligente dos recursos e redução da emissão de poluentes causadores do efeito estufa – menos tempo de viagem significa menos trabalho dos tubos de escape dos veículos. Semelhantemente, Daniel H. Stolfi e Enrique Alba, em artigo publicado no vigésimo quarto volume da revista internacional *Applied Soft Computing*, propuseram o intitulado *Red Swarm*, um sistema capaz de redirecionar veículos em movimento para rotas otimizadas, possibilitando que motoristas cheguem de forma mais rápida aos seus destinos (STOLFI e ALBA, 2014). O sistema permite a interação entre terminais ligados ao motorista (como smartphones) e dispositivos localizados nos semáforos – ao se aproximar de um semáforo, o motorista recebe por meio de uma conexão Wi-Fi a sugestão da melhor rota para o seu destino. Os pontos *Red Swarm* espalhados pela cidade são capazes de coletar informações sobre o trânsito, ajudando motoristas e autoridades.

Em 2012, o engenheiro eletrônico Bruno Sarno Mugnola, em sua dissertação de mestrado apresentada na Escola Politécnica da Universidade de São Paulo (Poli-USP), descreveu um interessante projeto sobre semáforos inteligentes intitulado “GenPolis” (ESCOLA POLITÉCNICA DA USP, 2012). Trata-se de um sistema que, baseado na teoria evolutiva das espécies, de Charles Darwin, possibilita a adaptabilidade dos planos semaforicos à necessidade do trânsito. Baseando-se na quantidade de veículos nas vias, o sistema constrói séries binárias, “cromossomos”, sobre os quais se aplicam “mutações e cruzamentos” que permitem a criação dos planos. Em um teste real, o protótipo foi instalado em semáforos de tempo fixo ao longo de seis cruzamentos de uma movimentada avenida de São Paulo (SP), e elaborou planos semaforicos tão bons que foram adotados pela Companhia de Engenharia de Tráfego da cidade.

1.3 ESTRUTURA DO TRABALHO

No capítulo seguinte serão abordados alguns conceitos sobre o funcionamento dos semáforos, vantagens e dificuldades da comunicação sem fio, o processo de comunicação de dados no padrão Zigbee e um breve estudo sobre o *hardware* da plataforma Arduino. No terceiro capítulo deste texto encontramos a descrição dos equipamentos utilizados, assim como das etapas de implementação do protótipo de semáforos inteligentes proposto. O quarto capítulo apresenta os desafios e os resultados dessa experimentação. Por fim, no capítulo 5 serão apresentadas as conclusões sobre o experimento e suas perspectivas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Neste tópico, serão abordados alguns conceitos sobre a semaforização inteligente, a tecnologia Zigbee e o Arduino, afim de compreender melhor a teoria voltada a esses assuntos, embasando-se na experiência de outros autores.

2.1 TRÂNSITO E SEMÁFOROS

O trânsito nas grandes cidades alcançou um quadro preocupante no século XX. Devido ao aumento do número de veículos e ao crescimento populacional dos centros urbanos, a demanda por espaço cresceu e a mobilidade se tornou um desafio. Os semáforos são um importante fator no que diz respeito a esse sentido.

Devido a variações no sentido do fluxo de veículos e pedestres nos horários de pico, o uso de semáforos com tempo fixo pode não ser o ideal. Este problema pode ser resolvido ou amenizado utilizando semáforos inteligentes. Já que o semáforo é um bom controlador de trânsito que permite diminuir os conflitos entre os veículos e aumentar a capacidade de escoamento, além de organizar o trânsito. (ARAÚJO, 2006 apud REGO; SEMENTE, 2017, p.1).

Segundo STOLFI e ALBA (2014, p. 181, tradução nossa), o acréscimo no número de acidentes e a maior frequência de congestionamentos dão-se devido ao rápido aumento do tráfego, em desarmonia com a infraestrutura destinada a contê-lo. Ainda segundo esses mesmos autores, tal problemática afeta diversos aspectos da nossa sociedade, entre eles: saúde, meio ambiente e desenvolvimento econômico.

Existem algumas limitações no que se diz respeito ao aumento da capacidade das estradas: custos de obra, vigilância e controle, manutenção e suporte, geografia, etc. É nesse contexto que surgem as *Smart Cities* (cidades inteligentes), uma iniciativa mundial definida por seis características: economia inteligente, pessoas inteligentes, governança inteligente, mobilidade inteligente, meio ambiente inteligente e vida inteligente (STOLFI e ALBA, 2014).

No âmbito mobilidade, uma cidade inteligente entende que toda limitação relacionada ao transporte e a acessibilidade pode ser vencida com o auxílio da Tecnologia da Informação e Comunicação (TIC). Ações que despendiam grandes esforços e exigiam altos investimentos passaram a dar lugar ao uso da tecnologia – de forma genérica. É mais simples, por exemplo, instalar semáforos inteligentes em uma cidade do que alargar suas vias ou construir mais viadutos; os resultados de ambas as ações podem ser positivos, mas a primeira solução gera menos estresse e conturbações.

Um outro aspecto que a utilização correta dos semáforos envolve é a segurança quanto à prevenção de acidentes em qualquer horário do dia. Segundo o Professor Cucci Neto (2015, p.1), em seu trabalho baseado no Manual Brasileiro de Sinalização de Trânsito (Volume 5): “O entreverdes é a parte do estágio programada após o fim do intervalo de verde, com o propósito de evitar acidentes entre os usuários que estão perdendo seu direito de passagem e aqueles que vão passar a adquiri-lo no estágio subsequente”. O entreverdes então é a soma do tempo de amarelo mais o tempo de vermelho geral.

Além de indicar “atenção”, o período de **amarelo** deve garantir que todos os veículos concluam sua passagem a tempo, sem que sejam surpreendidos pelo vermelho; o período em que todos os semáforos estiverem em vermelho (**vermelho geral**) deve garantir a “limpeza” do cruzamento para a próxima travessia de via transversa, anulando a possibilidade de conflitos entre veículos.

No Brasil, existe a tendência de sempre ajustar o tempo de amarelo dos semáforos para três, quatro, ou cinco segundos, ou acrescentar um ou dois segundos de vermelho geral (AKISHINO, 2004). Entretanto, visto a importância do período entreverdes, tais tempos devem ser escolhidos com cuidado, obedecendo à fatores particulares de cada cruzamento, tais como: largura das vias, limites de velocidade, relevo físico da via, entre outros. Mais adiante, no experimento realizado com o protótipo, a teoria criada por Gazis em 1959 será utilizada como base para a aplicação desses conceitos.

2.2 O ZIGBEE

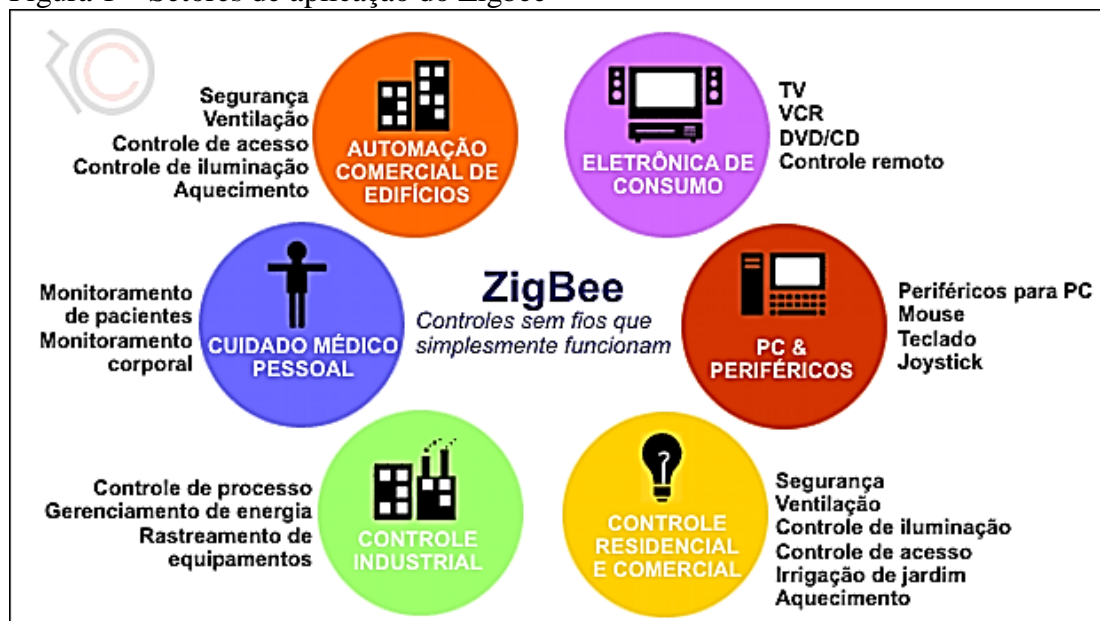
Inicialmente, o Zigbee surgiu como uma alternativa de comunicação sem fio para redes de área pessoal, *Wireless Personal Area Network* (WPAN). Porém, o padrão fez tanto sucesso que os seus desenvolvedores criaram novas versões da tecnologia. Entretanto, mesmo em suas primeiras versões o Zigbee foi experimentado em ambientes muito hostis.

Em Aakvaag, Mathiesen e Thonet (2005) são apresentados os resultados de um experimento feito com uma rede ZigBee, em uma instalação de automação industrial em uma companhia de mineração sueca, com o intuito de demonstrar que o novo padrão ZigBee se comporta bem mesmo em ambientes industriais pesados. Uma das principais observações feitas foi que a rede de sensores *wireless* aparentava estar funcionando satisfatoriamente, mesmo além da máxima distância especificada pelo padrão [...]. (MONSIGNORE, 2007, p.16).

Comparado a outros padrões mais populares como o Bluetooth e o Wi-Fi, o Zigbee apresenta menor complexidade de *hardware* e menor custo de implementação (TORRES DOS SANTOS, 2007).

Hoje, o uso da tecnologia Zigbee ganhou proporção e atende a diversos setores do mercado, o que testifica sua confiabilidade como padrão de comunicação sem fio. Abaixo, conforme ilustrado na Figura 1, seguem exemplos de aplicações da referida tecnologia.

Figura 1 – Setores de aplicação do Zigbee



Fonte: Repositório do Grupo de Teleinformática e Automação da Universidade Federal do Rio de Janeiro (2017)³

A utilização do padrão Zigbee neste projeto trouxe benefícios no que diz respeito à simplicidade de implementação e ao baixo consumo elétrico, além proporcionar uma nova aplicabilidade à tecnologia: a utilizada do Zigbee em semáforos inteligentes.

2.2.1 Arquitetura do Protocolo

O Zigbee possui uma metodologia padrão para suas funções baseada no modelo *Open System Interconnection* (OSI), incluindo a formação de redes, transmissão e recepção de dados e identificação de dispositivos (SIEMEINTCOSKI, 2009).

A arquitetura do protocolo Zigbee tem suas camadas iniciais regulamentadas pela norma IEEE 802.15.4, sendo elas a subcamada de acesso ao meio e a camada física, MAC e PHY, respectivamente. As camadas seguintes, rede (NWK) e aplicação (APL), estão sob o regimento da Zigbee Alliance.

³ Disponível em: <https://www.gta.ufrj.br/grad/10_1/zigbee/aplicacoes.html>. Acesso em Set. 2017.

A norma IEEE 802.15.4 estabelece a possibilidade do uso de três faixas específicas de frequência:

- a) Banda de 868 *mega-hertz* (MHz): utilizada na Europa, com 1 canal (0) de frequência;
- b) Banda de 915MHz: utilizada nos Estados Unidos, com 10 canais (1-10) de frequência;
- c) Banda de 2,4GHz: utilizada em outras partes do mundo, com 16 canais (11-26) de frequência.

De acordo com informações contidas no site da Agência Nacional de Telecomunicações (ANATEL), e na Resolução nº 680, de 27 de junho de 2017 (BRASIL, 2017), equipamentos que operam na faixa de frequência de 2,4GHz podem independender de licenciamento, porém existem regras à serem obedecidas, tais como:

- i. Não poder causar interferência em outros sistemas devidamente autorizados;
- ii. Não possuir mais de 5.000 acessos (conexões) à rede;
- iii. Emitir comunicação prévia de funcionamento à ANATEL;
- iv. Ter cadastro no sistema eletrônico da agência, além de outras exigências.

A velocidade do envio de *bits* (*baud rate*) do protocolo Zigbee está relacionada às frequências de transmissão em que opera: a faixa de 868MHz disponibiliza 20Kbps, a faixa de 915MHz disponibiliza 40Kbps e, por fim, a faixa de 2,4GHz disponibiliza 250Kbps.

2.2.2 Tipos de Dispositivos

A norma IEEE 802.15.4, por padrão, define dois tipos de dispositivos para uma rede LR-WPAN (*Wireless Personal Area Network - Low Rate*), conforme mostrado no Quadro 1: o dispositivo com funções completas (FFD⁴) e o com funções reduzidas (RFD⁵). A definição LR-WPAN emprega-se às redes de área pessoal com baixa taxa (*low rate*) de transmissão de dados, até 250 Kbps.

Por outro lado, o protocolo Zigbee dispõe de três tipos de dispositivos possíveis em uma rede, conforme mostrado no Quadro 2.

⁴ Full Function Device

⁵ Reduce Function Device

Quadro 1 – Tipos de dispositivo definidos pela IEEE 802.15.4

Tipos de dispositivos	Serviços Oferecidos	Fonte Típica	Configuração típica do receptor
Completo (FFD)	A maioria ou tudo	Rede elétrica	Ligado quando inativo
Reduzido (RFD)	Limitado	Bateria	Desligado quando inativo

Fonte: Siemeintcoski (2009, p. 23)

Quadro 2 – Relação de dispositivos entre IEEE e o protocolo Zigbee

Tipos de dispositivos do protocolo Zigbee	Tipo de dispositivo da IEEE	Função típica
Coordenador	FFD: dispositivo com funções completas	Liga através da rede, forma rede, aloca endereços de rede, armazena a ligação em tabela
Roteador	FFD: dispositivo com funções completas	Opcional, estende a capacidade física da rede, permite mais nós ligados na rede, pode executar também monitorando e/ou controle das funções
Dispositivo final	FFD: dispositivo com funções completas RFD: dispositivo com funções reduzidas	Executa monitorando e/ou controlando as funções de execução

Fonte: Siemeintcoski (2009, p. 23)

2.2.3 Acesso ao Meio (Subcamada MAC)

Redes sem fio que utilizam o protocolo Zigbee são multiacesso, ou seja, todos os dispositivos da rede têm igual acesso ao meio de comunicação. Sendo assim, existe a necessidade de uma organização, mecanismos de acesso aos canais, para controlar o tráfego de dados. Como exemplo, podemos imaginar o cenário de dois sensores que precisam trocar informações com um servidor, mas compartilham o mesmo canal. Como evitar colisões?

Em redes Zigbee, existem dois modos de funcionamento dos dispositivos, *active* e *sleep*. Neste último, os módulos entram em um estado de baixo consumo quando não estão enviando ou recebendo dados, reduzindo o consumo de energia e aumentando o tempo de vida útil das baterias, que, em alguns casos, resistem a anos de uso. Esses modos de funcionamento caracterizam-se em função dos métodos de multiacesso ao meio utilizados pelo protocolo, o *beacon* (sinalizado) e o *non-beacon* (não sinalizado).

No modo *non-beacon*, a camada MAC realiza a sincronia das transmissões através do algoritmo CSMA-CA⁶, que determina que cada nó deve verificar a ausência de tráfego nos

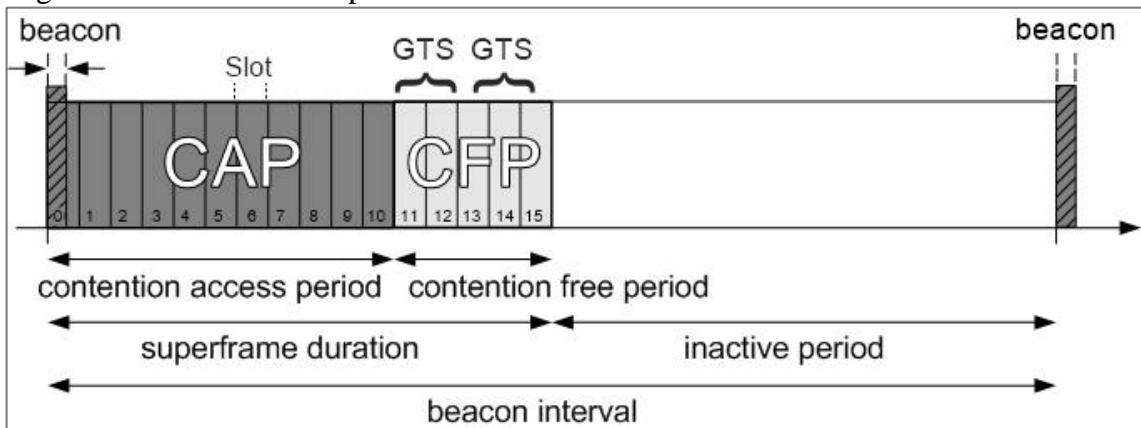
⁶ Carrier Sense Multiple Access with Collision Avoidance

canais antes de iniciar uma comunicação. Estando o canal livre, o dispositivo inicia sua transmissão; estando o canal ocupado, o dispositivo permanece em “escuta”, aguardando a liberação do meio. Quando um nó suspeita que uma colisão entre dados aconteceu, devido a ocorrência de transmissões simultâneas, ele interrompe imediatamente a comunicação e inicia-a novamente após um tempo aleatório.

Permanecer em “escuta” requer um consumo maior de energia, porém, dependendo da aplicação, o modo *beacon* pode ser utilizado. Em uma rede sinalizada, o coordenador encarrega-se de emitir periodicamente um superpacote, utilizado para sincronizar os dispositivos da rede. Inicialmente, a mensagem *beacon* é emitida pelo coordenador, definindo uma estrutura chamada **superframe**, dentro da qual 15 intervalos (*slots*) de tempo iguais permitem a transferência de dados entre os dispositivos. Os primeiros intervalos, no mínimo 8, formam o período de disputa de acesso (CAP), em que cada nó está autorizado à transmitir e receber dados, competindo uns como os outros por meio de um mecanismo semelhante ao CSMA-CA. Os intervalos (*slots*) seguintes, no máximo 7, são opcionais, e podem ser reservados à determinados dispositivos através de pedidos. Essas parcelas reservadas são chamadas de espaços de tempo garantido (GTS), em um período livre de disputa (CFP), que permite acesso (transmissão) exclusivo e sem a execução do CSMA-CA.

A Figura 2 exemplifica a estrutura do superframe. O dispositivo coordenador inicia emitindo um superpacote (*beacon*) no *slot* de tempo “0”, e todos os nós sincronizam-se de acordo com ele. Em seguida, inicia-se o período de disputa de acesso, onde todos os dispositivos poderão transmitir e receber dados de acordo com a sincronia estabelecida. Logo após encerrar-se o CAP, o tempo livre de disputa (CFP) é iniciado. Segundo Torres dos Santos (2017, p. 15), ao término de todos os *slots* de tempo, pode haver um período inativo (opcional), durante o qual os dispositivos podem desligar seus transceptores e assumir uma modalidade de baixo consumo de energia (*low-power*). Um novo superframe é iniciado após o coordenador emitir um novo *beacon* na rede.

Figura 2 – Estrutura do Superframe



Fonte: Adaptada do repositório do ResearchGate (2017)⁷

2.2.4 Transferência de Dados

Existe a possibilidade de três tipos de transferência de dados em uma rede Zigbee:

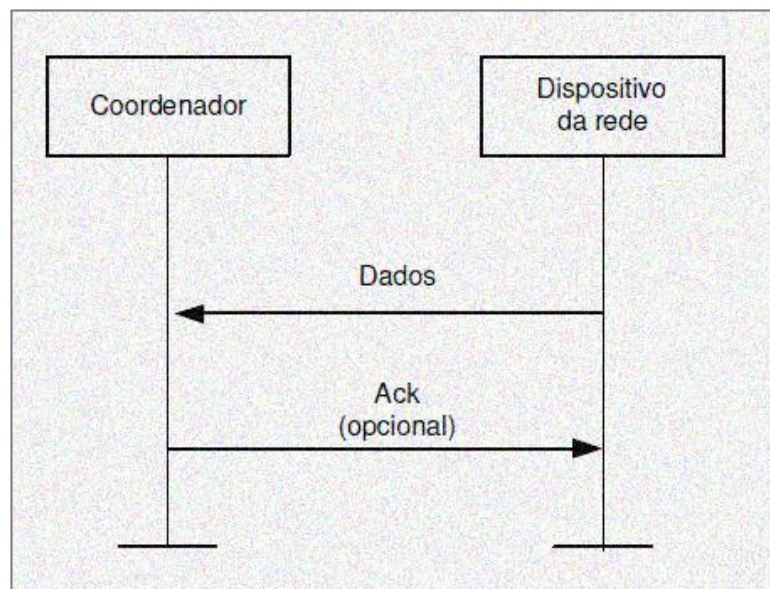
- a) De um dispositivo para o coordenador;
- b) Do coordenador para um dispositivo;
- c) De um dispositivo para outro dispositivo.

Na topologia ponto a ponto aplicada neste trabalho, em que um nó comunica-se diretamente com outro, todas as três formas de transferência podem ser utilizadas. Além disso, segundo Torres dos Santos (2007, p. 19), uma rede com *beacon* habilitado é usada para suportar dispositivos de baixa latência, como periféricos de um computador, por exemplo. Não havendo a necessidade de suportar tais dispositivos, caso do protótipo aqui desenvolvido, pode-se optar por não utilizar o *beacon* para transferências normais.

Em redes não sinalizadas, o dispositivo final, valendo-se do uso do algoritmo CSMA-CA, transmite seu quadro de dados para o coordenador, que por sua vez reconhece a recepção correta dos dados por meio de um quadro ACK (*Acknowledgment*), conforme Figura 3. A ausência de um ACK significa que o coordenador não recebeu corretamente os dados.

⁷ Disponível em: <https://www.researchgate.net/figure/253470553_fig1_Fig-1-Beacon-interval-and-superframe-concepts>. Acesso em Set. 2017.

Figura 3 – Transferência de dados de um dispositivo para o coordenador em rede não sinalizada

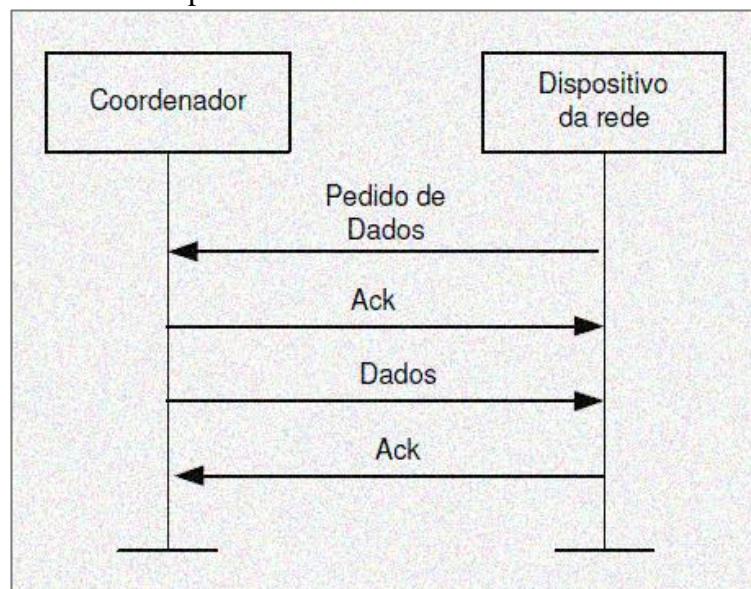


Fonte: Adaptada de Torres dos Santos (2007, p. 20)

O segundo tipo de transferência de dados, conhecido como *transmissão indireta*, e que também ocorre nesse trabalho, é o mecanismo para transferir dados **do coordenador para um dispositivo**. Em redes não sinalizadas, o processo de transmissão de dados do coordenador para o dispositivo acontece de forma diferente. O coordenador armazena os dados até que o dispositivo apropriado envie uma requisição. Um dispositivo final pode emitir requisições por meio de comandos MAC em uma taxa de *pooling*⁸ definida pela camada de aplicação. Ao receber o pedido de forma correta, o coordenador responde com um quadro de confirmação e, havendo mensagem pendente, realiza o envio para o dispositivo. Quando não existem dados pendentes, o coordenador transmite um quadro de dados com um cabeçalho de comprimento zero, para indicar que não há mensagem a ser enviada. A seguir, a Figura 4 ilustra a referida sequência de transmissões.

⁸ Taxa de verificação

Figura 4 – Transferência de dados do coordenador para um dispositivo em rede não sinalizada



Fonte: Adaptada de Torres dos Santos (2007, p. 21)

Para finalizar destacamos que a transferência de dados ponto à ponto em redes de área pessoal, *Personal Area Network* (PAN), permite que um dispositivo final possa comunicar-se diretamente com outro desde que esteja dentro do seu alcance de rádio.

2.2.5 Associação e Desassociação de Dispositivos

A associação de um dispositivo acontece após ele ter realizado uma exploração ativa ou passiva da rede. Na exploração passiva, o dispositivo final que deseja associar-se deverá localizar algum coordenador que esteja transmitindo *beacons* dentro de sua área de alcance; na exploração ativa, o próprio dispositivo final transmite um comando de pedido de *beacon*.

Após a exploração, os parâmetros da rede que influem um dispositivo a decidir associar-se à ela são: o canal físico (*phyCurrentChannel*), seu identificador (*macPANId*) e seu endereço curto (*CoordShortAddress*) ou longo (*CoordExtendedtAddress*).

Inicialmente, o dispositivo não associado enviará um pedido de associação ao coordenador da rede PAN existente. Se o coordenador receber a solicitação de forma correta, ele responderá com um quadro de reconhecimento (*Acknowledgmente*), que não significa a confirmação de que o nó final será associado. Somente após um determinado intervalo de tempo (*aResponseWaitTime*), durante o qual o coordenador verificará se existem recursos suficientes na rede para a inclusão de um novo dispositivo, que a associação acontecerá. Havendo recursos suficientes, o coordenador reservará um endereço curto (*CoordShortAddress*) para o novo

dispositivo e enviará à ele um comando de resposta indicando o sucesso da associação e o novo endereço. Não havendo recursos suficientes, o coordenador enviará ao nó um comando de resposta de associação indicando falha.

Da parte do dispositivo, após receber o quadro de reconhecimento (*Acknowledgment*), este aguarda o tempo de *aResponseWaitTime* e, havendo uma confirmação de associação, responderá com um quadro de reconhecimento (ACK). Sendo a associação positiva, o endereço (*CoordShortAddress*) será armazenado pelo dispositivo.

Quando o coordenador desejar desassociar um dispositivo da sua rede PAN, ele enviará por transmissão indireta uma notificação de desassociação ao nó. Ao receber tal informação, o nó final responderá com um quadro de reconhecimento (ACK). Mesmo não recebendo o ACK o coordenador considerará o dispositivo desassociado.

Já quando um dispositivo desejar desassociar-se de uma rede, enviará ao coordenador um comando de notificação de desassociação. Ao receber tal informação, o coordenador responderá com um quadro de reconhecimento (ACK). Mesmo não recebendo o ACK, o dispositivo considerar-se-á desassociado.

Na desassociação, o dispositivo eliminará de sua memória todas as referências daquela rede na qual estava inserido, assim como o coordenador eliminará todas as referências do dispositivo desassociado.

2.3 O ARDUINO

O Arduino tem ganhado o interesse de estudantes e pesquisadores no mundo inteiro desde o seu lançamento na Itália, em 2005, por ser uma plataforma de prototipagem simples, flexível e ter uma linguagem de programação típica. Baseado em *software* e *hardware* livres, o Arduino pode ser utilizado em projetos sem a necessidade de licença – o código Arduino é aberto à ser estudado, adaptado e distribuído. O design físico da sua placa (desenhos mecânicos, esquemas, lista de materiais, dados de layout da placa de circuito impresso (PCB⁹), etc.) também está totalmente liberado, isto é, com acesso livre (*Open Source*) para quaisquer interessados. Além disso, documentos e tutoriais são atualizados constantemente na página oficial **arduino.cc** na internet (HOEPERS, 2012, apud PORTANTE, FRESSATTI).

A placa Arduino não possui a função de conectividade, sendo necessária a utilização de um extensor (*shield*), que receberá o módulo RF Zigbee e incorporá-lo-á ao sistema. O

⁹ Printed Circuit Board

Arduino UNO, que será utilizado na construção do *Traffic Bee*, conta com um microcontrolador ATmega328, 14 entradas e saídas digitais, e 6 entradas analógicas que possibilitarão o desenvolvimentos do protótipo que será melhor detalhado em suas partes componentes e funções no próximo capítulo.

3 A PROPOSTA TRAFFIC BEE: PROTÓTIPO E CENÁRIO REAL

Traffic Bee é uma solução que, baseada na computação física¹⁰ (ou sistemas interativos físicos), possibilita a implementação de semáforos inteligentes que funcionam detectando o fluxo de trânsito num cruzamento de vias de rodagem respondendo digitalmente à ele através das temporizações semaforicas de cada via.

Toda a programação e organização de *hardware* necessários para o funcionamento otimizado do *Traffic Bee* em um cenário real não foram considerados relevantes para a execução deste trabalho, por fugir do seu escopo acadêmico de aplicação prática dos conhecimentos técnicos obtidos durante o curso de graduação, dando-se preferência a um cenário simulado via protótipo, cuja contribuição mais proeminente encontra-se na implementação da tecnologia de comunicação sem fio Zigbee e sua aplicação na lógica operacional dos semáforos inteligentes.

3.1 TRAFFIC BEE PROTÓTIPO: COMPONENTES E CARACTERÍSTICAS

Num cenário de laboratório o protótipo irá compor o que seria um semáforo de quatro estágios instalado em um cruzamento de 2 avenidas. Para esse fim os seguintes materiais foram utilizados na montagem do protótipo:

- a) Módulo controlador Arduino UNO R3;
- b) Adaptador USB XBee Explorer;
- c) Módulo de rádio frequência XBee S2;
- d) Extensor (*shield*) Xbee Pro;
- e) LED's coloridos (verde, amarelo e vermelho) de 5V;
- f) Placa de prototipagem (*protoboard*) de 60 furos;
- g) Resistores de 270Ohms (Ω).

O Arduino modelo UNO já está em sua terceira revisão, R3, e conta com diversas características interessantes em seu projeto (Figura 5). A placa possui tensão de funcionamento de 5 Volts (V) e tem duas opções de alimentação, por barramento serial universal, *Universal Serial Bus* (USB), ou por fonte externa que varie, por recomendação, entre as tensões de 7V à 12V, para evitar o sobreaquecimento do regulador de tensão responsável por proteger o

¹⁰ Solução que usa *softwares* e *hardwares* interligados para sensoriar e responder ao mundo analógico.

Arduino. Seus 14 pinos de entrada e saída digitais operam a 5V e podem receber correntes de no máximo de 40 *miliampères* (mA); suas 6 entradas analógicas têm uma resolução de 10 *bits*, cada uma, e conversor analógico/digital com referência ligada internamente à 5V.

Figura 5 – Arduino UNO R3



Fonte: Elaborado pelo autor em 2018

O transparente funcionamento do Arduino dá-se pela presença do microcontrolador ATmega16U2 entre seus componentes, ele funciona como interface USB para comunicação com computadores e possibilita o *upload* dos códigos binários gerados após a compilação de programas escritos pelo usuário.

Mas o principal componente da placa Arduino UNO R3 chama-se ATmega328, da Atmel Corporation, um dispositivo microcontrolador 8 *bits* da família AVR¹¹, arquitetura RISC¹² avançada e encapsulamento DIP28¹³. Ele ainda conta com uma memória flash de 32 *kilobytes* (KB), 2KB de memória de acesso aleatório, *Random Access Memory* (RAM), e 1KB de memória de leitura programável e apagável eletricamente, *Electrically-Erasable Programmable Read-Only Memory* (EEPROM). O ATmega328 consegue trabalhar com tensões reduzidas de até 1,8V, diminuindo sua frequência de operação neste caso para no máximo 4MHz.

Já a *shield* XBee Pro (Figura 6), assim como todas outras, é nada mais que uma placa de circuito impresso capaz de se conectar à parte superior do Arduino e estender as funções

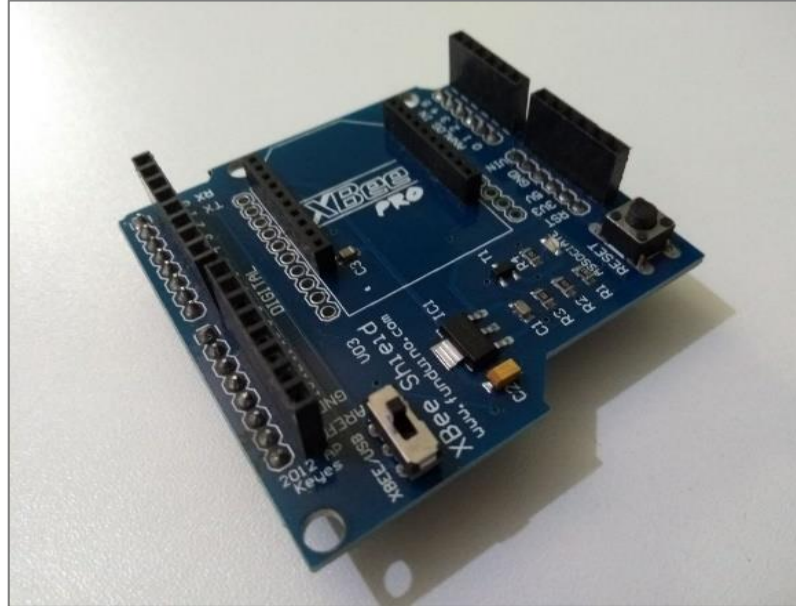
¹¹ *Automatic Voltage Regulator* – regulador automático de voltagem.

¹² *Reduced Instruction Set Computer* – computador com um conjunto reduzido de instruções.

¹³ *Dual In-Line Package* – pacote duplo em linha.

dele. Após ser acoplada ao Arduino, a *shield* XBee Pro permite que módulos de radiofrequência Zigbee sejam conectados ao microcontrolador e coordenados de acordo com as funções programadas pelo usuário.

Figura 6 – Shield XBee Pro



Fonte: Elaborado pelo autor em 2018

As *shields* são alimentadas por meio das conexões IOREF¹⁴, RESET, 3.3V, 5V, GND¹⁵ e VIN¹⁶ do Arduino. O pino IOREF fornece uma tensão de referência para que a *shield* possa selecionar a alimentação correta, 3.3V ou 5V; o pino RESET permite que o *reset* possa ser dado externamente no Arduino, por meio da *shield*; o pino GND fornece aterramento; o pino VIN permite alimentar o Arduino através da *shield*.

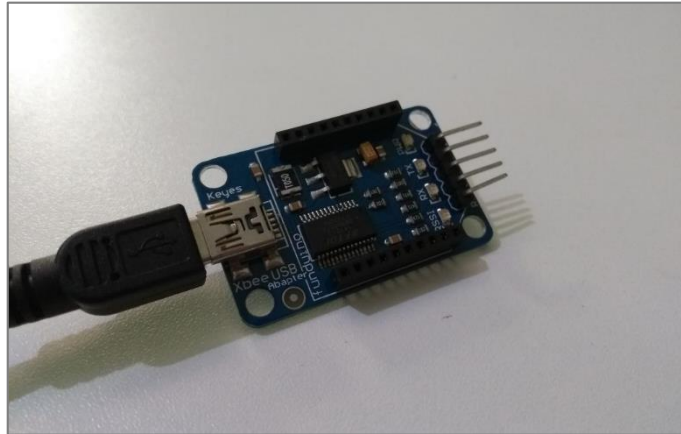
O adaptador XBee Explorer (Figura 7) é utilizado para configurar os módulos de radiofrequência Zigbee. Ele opera sob uma tenção de 5V, alimentado pela própria porta USB a qual se conectar. Com suporte ao programa *XBee Configuration and Test Utility (XCTU)*, da Digi International Inc., empresa fabricante dos módulos RF XBee, o Explorer permite que parâmetros como *phyCurrentChannel*, *macPANId*, *CoordShortAddress* e *CoordExtendedtAddress* sejam configurados em cada módulo RF, possibilitando a criação de redes.

¹⁴ *Input/output Reference* – referência de entrada/saída.

¹⁵ *Graduated Neutral Density Filter* – filtro graduado de densidade neutra.

¹⁶ *Voltage Input* – entrada de voltagem.

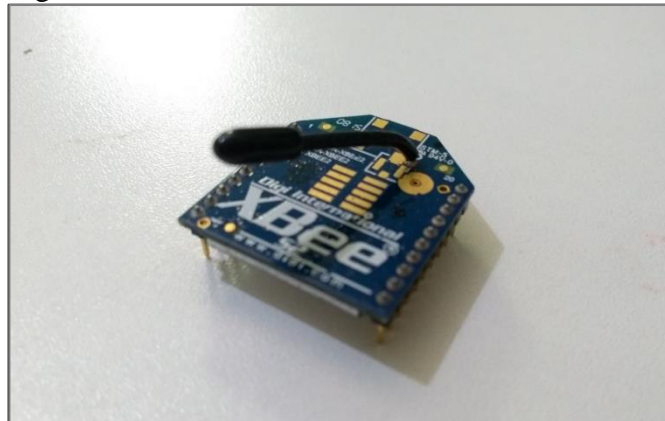
Figura 7 – Adaptador USB XBee Explorer



Fonte: Elaborado pelo autor em 2018

Os módulos RF Xbee (Figura 8) utilizados neste trabalho são da série 2, uma versão que possui melhora em respeito à potência de saída de sinais e protocolo de dados, permitindo uma comunicação mais simples e segura entre dispositivos que utilizam porta serial.

Figura 8 – Módulo RF XBee



Fonte: Elaborado pelo autor em 2018

É importante destacar que a comunicação entre os módulos Zigbee oferece excelente imunidade contra interferências, sendo ideal para diversos tipos de ambientes, incluindo aplicações industriais e até mesmo locais mais hostis (MONSIGNORE, 2007). Tal realidade é proporcionada por funcionalidades específicas encontradas nas camadas iniciais do protocolo: a Camada Física e Subcamada de Acesso ao Meio (AZEVEDO, 2010).

Os módulos RF XBee Série 2 possuem taxa de transmissão de dados seriais (*data rate*) de 1.200 *bits* por segundo (bps) a 1 *megabit* por segundo (Mbps), com rendimento de potência de saída de até 63mW, +18 *decibel-miliwatts* (dBm). O alcance do sinal de rádio é de 90 metros (m) em média em ambientes internos, podendo chegar até mais de 100m; já em ambientes externos, em linha de visada direta, esse alcance pode chegar a 1,5 quilômetros (Km).

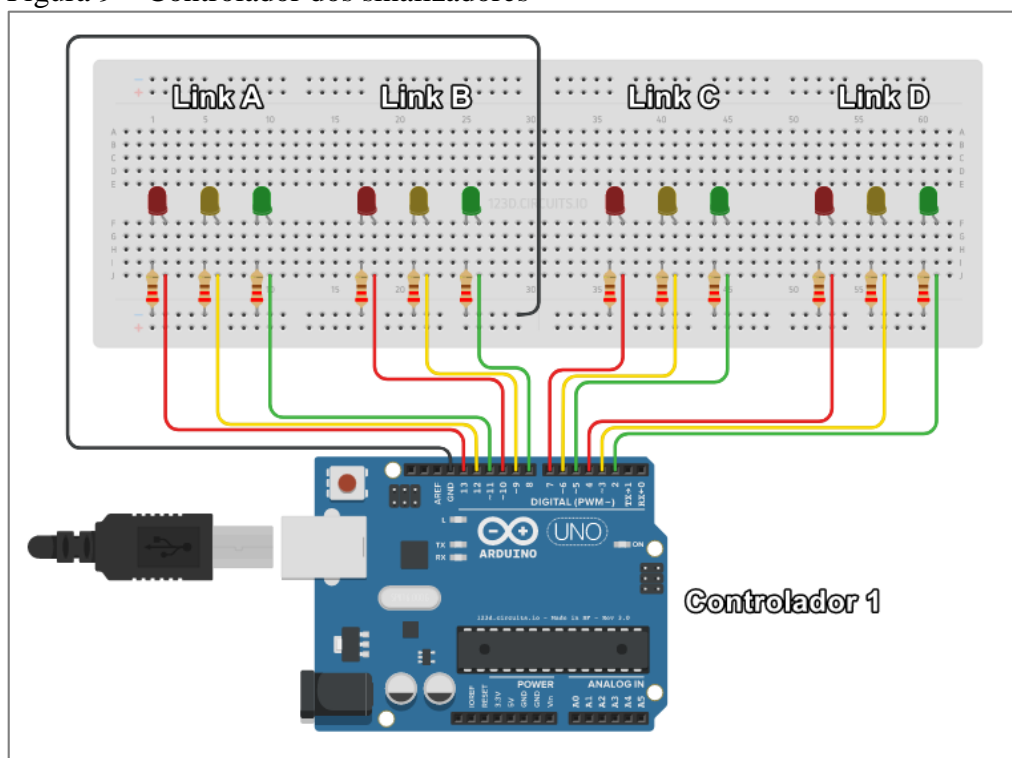
Operando na faixa dos 2,4GHz, frequência utilizada neste projeto, os módulos Xbee disponibilizam uma velocidade específica de transferência de dados em RF de 250Kbps. A alimentação é feita numa faixa de voltagem de 2,7 à 3,6V. Em repouso (*standby*), o consumo elétrico dos módulos é baixíssimo, apresentando uma corrente máxima de 3,5 *microamperes* (μA) numa temperatura ambiente de 25 graus Celsius ($^{\circ}\text{C}$).

Finalizando, os módulos RF Xbee também possuem sistema para detecção de erros de transmissão onde ao serem reconhecidos requisita-se a retransmissão dos dados afetados, criptografia de dados aplicando o padrão *Advanced Encryption Standard* (AES) de 128 *bits*, além de suporte às seguintes topologias de rede: par-a-par, ponto-a-ponto, ponto-a-multiponto e em malha, todas essas topologias com mais de 65.000 endereços de rede disponíveis para alocar logicamente os dispositivos.

3.2 TRAFFIC BEE PROTÓTIPO: IMPLEMENTAÇÃO E CONFIGURAÇÃO

Inicialmente, os diodos emissores de luz, *Light Emitting Diode* (LED), foram conectados ao Arduino por meio de uma placa de prototipagem (*proto-board*), aplicando-se na montagem resistores para limitação de corrente nos respectivos LEDs. Essa primeira parte do protótipo, conforme mostrada na Figura 9, foi elaborada para **simular** um quarteto de semáforos presentes no cruzamento de duas vias. A alimentação dessa etapa foi realizada através de uma fonte de tensão externa.

Figura 9 – Controlador dos sinalizadores



Fonte: Adaptado do repositório do Autodesk Circuits (2017)¹⁷

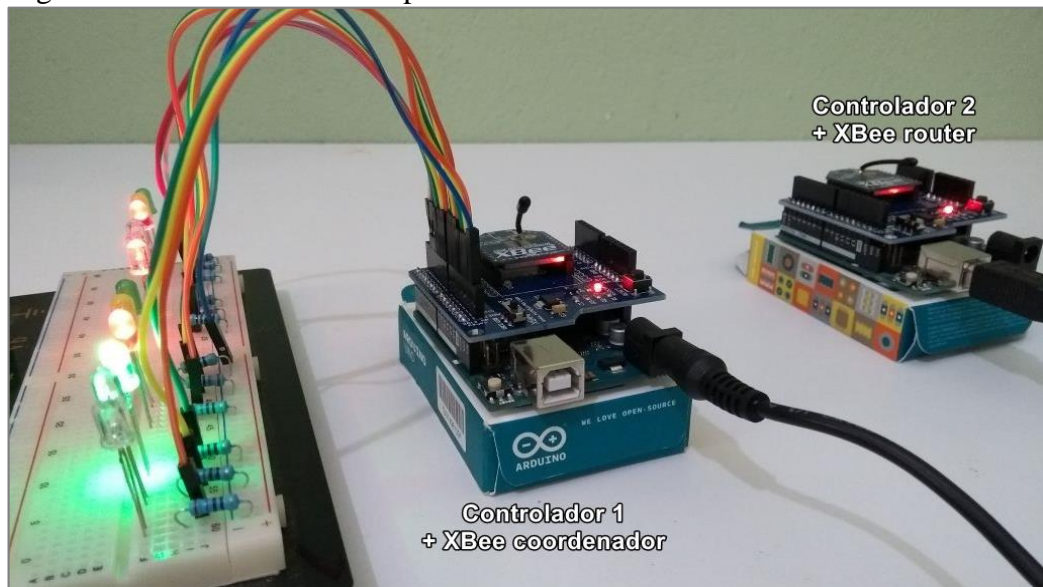
No protótipo, o módulo RF conectado ao **controlador 1** foi configurado como **coordenador** da rede Zigbee. Neste caso, o Arduino conectado à placa de prototipagem é o controlador 1, que controla os tempos de acendimento dos LEDs por meio de um algoritmo¹⁸ que será explicado mais adiante neste texto. Ao **controlador 2**, foi conectado um módulo RF configurado como **roteador** Zigbee. O controlador 2 é composto pelo Arduino responsável por informar ao controlador 1 a quantidade de veículos trafegando em cada uma das vias.

Um dos desafios deste projeto foi a quantidade de equipamentos disponíveis para a implementação da proposta. Contamos efetivamente com duas placas controladoras Arduino UNO R3 e dois módulos RF XBee Série 2, sendo que para uma reprodução fiel do cenário necessitaríamos de no mínimo 9 dispositivos de cada tipo (Arduino + Módulos RF). Nesse sentido, o controlador 2 simulou as informações que seriam enviadas por 8 dispositivos ao coordenador da rede. A Figura 10 mostra o controlador 1 (Arduino + XBee coordenador) e o controlador 2 (Arduino + XBee Roteador) em funcionamento.

¹⁷ Disponível em: <<https://circuits.io/circuits/1839665-semaforo-de-4-tempos#breadboard>>. Acesso em Dez. 2017.

¹⁸ <https://gist.github.com/Gladerson/08f70b7f7fa00c163e26564181487482>

Figura 10 – Funcionamento experimental do Traffic Bee

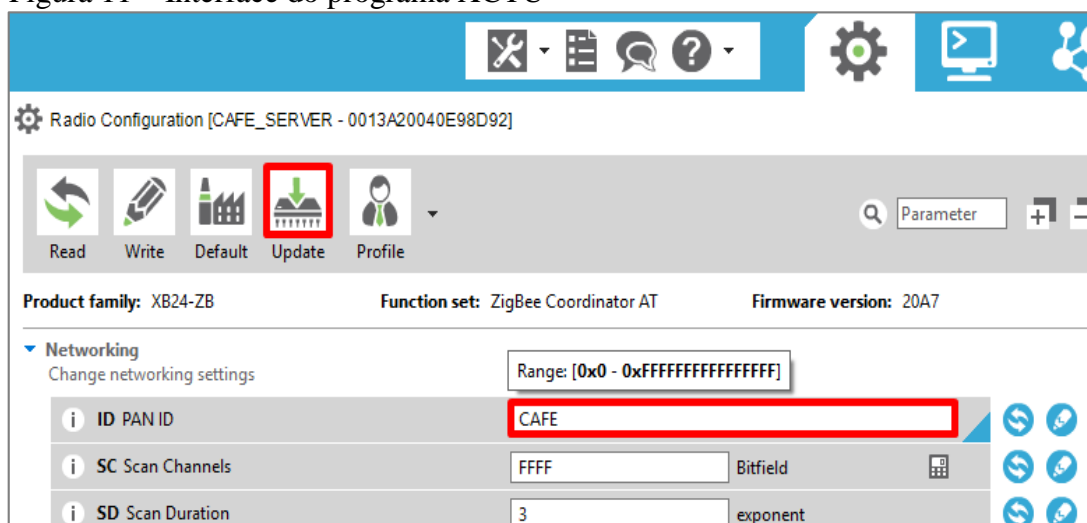


Fonte: Elaborado pelo autor em 2018

3.2.1 Módulos RF Xbee: Configuração e Modo de Funcionamento

Para configurar os módulos RF foi necessário conectá-los a um computador por meio do adaptador USB XBee Explorer, para que o programa XCTU pudesse acessar e configurar as propriedades de cada um deles (Figura 11).

Figura 11 – Interface do programa XCTU



Fonte: Elaborado pelo autor em 2018

Após ser reconhecido o primeiro módulo RF pelo XCTU, na guia “Update” pôde-se escolher o **coordenador** da rede. No parâmetro de identificação **ID PAN ID**, o endereço hexadecimal “CAFE” foi configurado como identificador da rede Zigbee a ser formada.

Para que uma conexão direta fosse estabelecida, o endereço físico do módulo RF roteador (*router*) foi declarado ao coordenador no parâmetro de configuração **DH Destination Address High** e em **DL Destination Address Low**. Por meio do campo **NI Node Identifier** o nome `CAFE_SERVER` foi associado ao coordenador (Figura 12).

Figura 12 – Interface do programa XCTU

Parameter	Value
SH Serial Number High	13A200
SL Serial Number Low	40E98D92
MY 16-bit Network Address	Range: [0x0 - 0xFFFFFFFF]
DH Destination Address High	13A200
DL Destination Address Low	40E98D85
NI Node Identifier	CAFE_SERVER
NH Maximum Hops	1E
BH Broadcast Radius	0

Fonte: Elaborado pelo autor em 2018

Já o módulo RF roteador, responsável por enviar ao coordenador os dados sobre a densidade de veículos em cada *link*, foi configurado no XCTU da seguinte forma: recebeu a função de roteador, o nome `CAFE_ROUTER` e as informações sobre nome da rede, identificador da rede e endereço físico do coordenador.

Concluindo, os módulos RF XBee podem operar em dois modos de funcionamento diferentes, **Transparente** ou **API** (*Application Programming Interface*). Como o protótipo *Traffic Bee* contou com somente dois dispositivos comunicando-se entre si, o modo transparente, mais simples, foi o utilizado e então será o único explanado nesse texto. Para começo de explanação, faz-se importante considerar algumas características básicas relacionadas a esse modo de operação:

a) Todos os dados que saem pela porta serial do dispositivo hospedeiro (neste caso o Arduino) são colocados em fila para transmissão via rádio, assim como todos os dados recebidos sem fio são encaminhados para a porta serial exatamente como foram recebidos sem nenhuma informação adicional;

b) Para ler ou escrever configurações nos módulos XBee, um procedimento especial deve ser executado, a alteração para o **modo comando**, pois no modo transparente os módulos RF apenas recebem e enviam dados pelo ar. Quando um deles recebe a sequência “1 segundo de silêncio, cadeia de caracteres especiais `+++`, mais 1 segundo de silêncio” logo entende que os próximos dados lerão ou escreverão alguma informação em suas configurações;

c) Se um XBee em modo transparente precisar enviar informações para vários módulos, antes precisará entrar no modo de comando para configurar cada novo destino antes de prosseguir com a transmissão;

d) Um XBee que opera em modo transparente não consegue identificar a origem da mensagem recebida. Para que o reconhecimento dos dados seja realizado no modo transparente o programador deve criar seu próprio protocolo de reconhecimento de origem de transmissão.

3.3 TRAFFIC BEE: IMPLEMENTAÇÃO NUM CENÁRIO REAL

Para auxiliar no entendimento do funcionamento da proposta *Traffic Bee* de forma mais ampla, explicaremos como aquela funciona num cenário real. Para isso, tomamos como exemplo duas grandes e movimentadas avenidas da cidade do Natal, Rio Grande do Norte, a Avenida Bernardo Vieira e a Avenida Senador Salgado Filho, ambas adjacentes ao IFRN Campus Natal Central.

Foi denominado *link* cada um dos quatro sentidos de tráfego possível nessas avenidas. Por convenção, foi adotado o comprimento de 260 metros para cada um desses quatro *links* – distância entre o cruzamento analisado e a interseção anterior (Avenida Senador Salgado Filho com Avenida Antônio Basílio). Desprezando-se as faixas reservadas para ônibus (aqui consideradas de tráfego geral), cada um desses *links* possui três faixas de velocidade e capacidade de comportar até 156 veículos de comprimento médio segundo os padrões comumente aplicados em estudos de tráfego, 5 metros. A Figura 13 ilustra como **ficaria** a configuração do sistema *Traffic Bee* no cruzamento dessas duas avenidas.

Figura 13 – Cenário real: cruzamento de vias



Fonte: Elaborado pelo autor em 2018

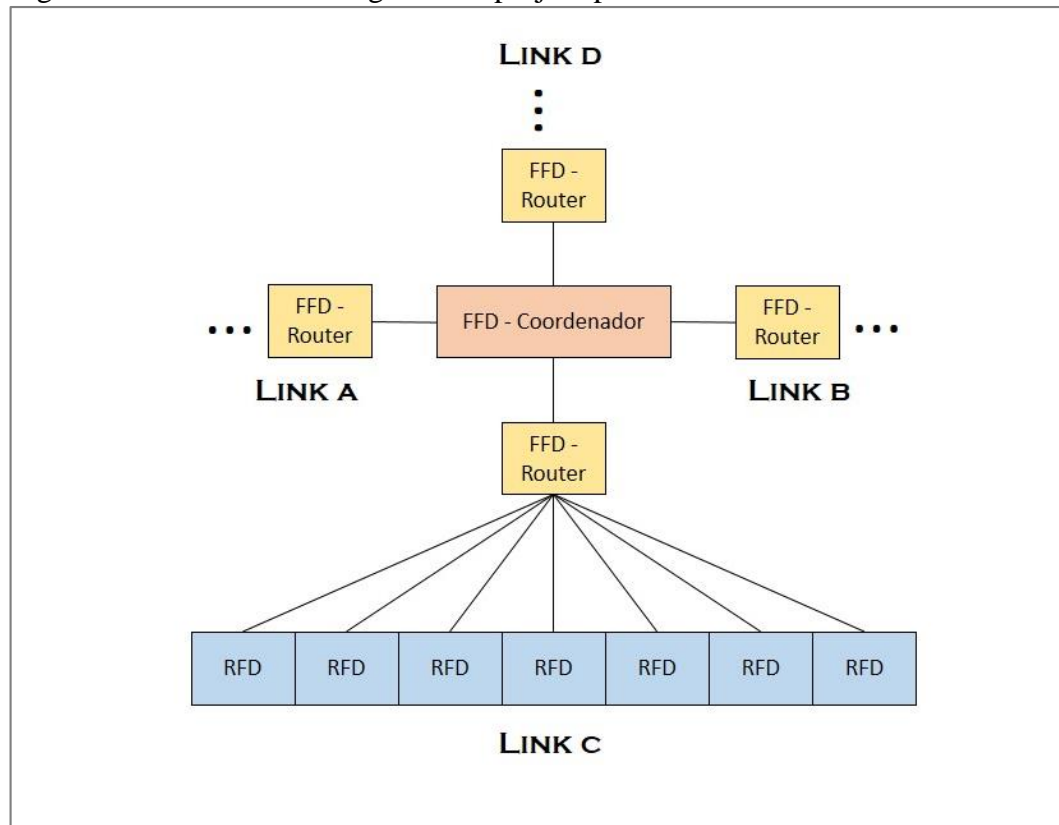
Num cenário real, ficará a critério da Engenharia de Trânsito ou Departamento responsável qual o tipo de equipamento que será utilizado para contar o número de veículos em cada *link*, que podem ir de sensores de metal instalados no chão à câmeras de alta resolução alocadas no postes da rede elétrica.

Responsáveis exclusivamente por contar o número de veículos por faixa de rodagem, propomos que esses sensores também estejam associados à plataformas microcontroladas capazes de transmitir usando a tecnologia Zigbee os dados sensorizados para seus respectivos nós roteadores.

Deste modo, podemos finalmente definir uma hierarquia para a aplicação real da proposta *Traffic Bee*: nela existem os dispositivos finais RFD acoplados aos sensores de tráfego,

seus respectivos roteadores e um coordenador geral, tudo isso estruturado numa topologia em árvores conforme o esquema de projeto para cenário real mostrado na Figura 14.

Figura 14 – Traffic Bee: diagrama de projeto para cenário real



Fonte: Elaborado pelo autor em 2018

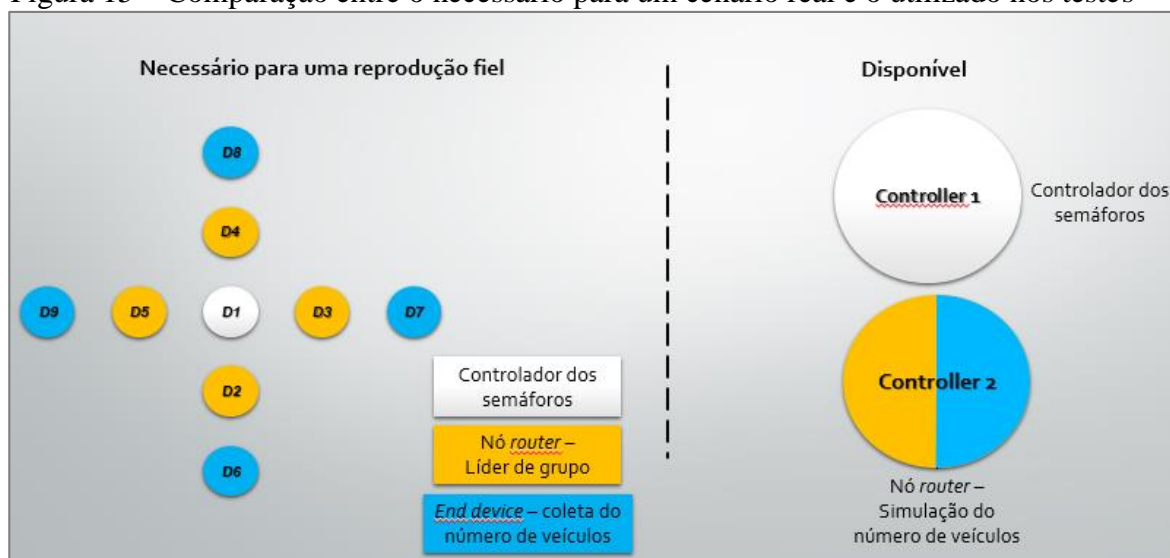
4 TRAFFIC BEE: DESAFIOS, TEMPORIZAÇÃO E RESULTADOS

Neste tópico, serão abordados os desafios enfrentados durante a execução deste projeto, os cálculos utilizados para a temporização dos semáforos inteligente e os resultados obtidos a partir dos experimentos.

4.1 DESAFIOS NA IMPLEMENTAÇÃO DO PROTÓTIPO

Como anteriormente citado, um dos desafios de execução do protótipo que testasse a proposta *Traffic Bee* foi a quantidade de equipamentos disponíveis para sua execução, onde existiriam quatro nós roteadores enviando informações ao coordenador houve apenas um nó roteador simulando essa transferência de dados. Além disso, como não haviam dispositivos finais RFD associados à sensores de tráfego disponíveis, onde existiriam “n” dispositivos finais enviados informações a seus líderes de agrupamento (roteadores), essas informações (quantidade de veículos) foram geradas de forma aleatória dentro do nó roteador. Exemplo conforme a ilustração da Figura 15:

Figura 15 – Comparação entre o necessário para um cenário real e o utilizado nos testes



Fonte: Elaborado pelo autor em 2018

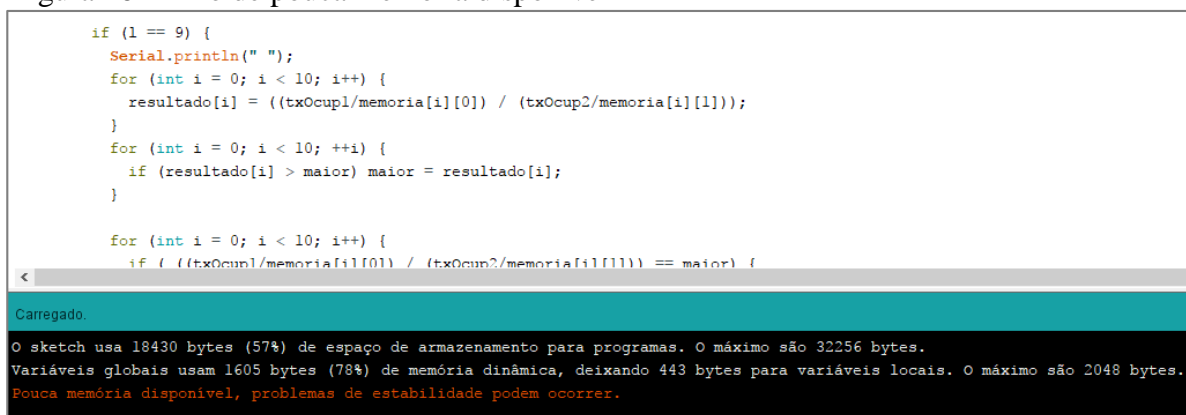
Um outro grande desafio encontrado foi a capacidade limitada de memória e de processamento do Arduino UNO. A versão UNO do Arduino possui apenas um núcleo de processamento, não sendo possível o uso de *Threads*¹⁹. Esse fator fez com que o *Traffic Bee*

¹⁹ Divisão de um processo computacional em duas ou mais tarefas que podem ser executadas concorrentemente.

gerasse um pequeno *delay* no final de cada ciclo de funcionamentos dos semáforos, por esse motivo o tempo de vermelho geral – todos semáforos em vermelho – no acionamento luminoso (verde-amarelo-vermelho geral) para o *link* D, configurado para 2 segundos, ficou entre 5 e 7 segundos, não estando fora do contexto atual da configuração do cruzamento real analisado, que hoje possui 7 segundos de vermelho geral para cada *link*. Mais especificamente esse atraso ocorre porque o processador ATmega328 não suporta a execução de duas funções ao mesmo tempo: controlar a temporização dos semáforos e comunicar-se com o dispositivo controlador 2.

O Arduino também possui uma baixa capacidade de memória dinâmica. No desenvolvimento deste trabalho, constantemente foram reportadas mensagens de erro pelo programa IDE referentes a tal situação. Para que fosse possível a implementação de todas as funções propostas neste projeto, o *Traffic Bee* teria que dispor de uma memória RAM maior no controlador principal (1). Conforme pode ser observado na Figura 16, 78% da memória dinâmica foi utilizada em um dos testes apenas com o carregamento das variáveis globais do código.

Figura 16 – Erro de pouca memória disponível



```

if (l == 9) {
  Serial.println(" ");
  for (int i = 0; i < 10; i++) {
    resultado[i] = ((txOcup1/memoria[i][0]) / (txOcup2/memoria[i][1]));
  }
  for (int i = 0; i < 10; ++i) {
    if (resultado[i] > maior) maior = resultado[i];
  }

  for (int i = 0; i < 10; i++) {
    if ( ((txOcup1/memoria[i][0]) / (txOcup2/memoria[i][1])) == maior) {

```

Carregado.

O sketch usa 18430 bytes (57%) de espaço de armazenamento para programas. O máximo são 32256 bytes.
 Variáveis globais usam 1605 bytes (78%) de memória dinâmica, deixando 443 bytes para variáveis locais. O máximo são 2048 bytes.
 Pouca memória disponível, problemas de estabilidade podem ocorrer.

Fonte: Elaborado pelo autor em 2018

Para contornar o problema da falta de memória RAM, o código foi otimizado em termos da redução do número de linhas, funções e variáveis de programação. Sendo assim, a implementação da função de garantir o sinal verde imediato para veículos que chegassem ao cruzamento no período da madrugada quando não existem outros veículos concorrendo à travessia não pôde ser realizada, pois mais funções e variáveis precisariam ser inseridas no código para executar essa lógica de sinalização.

4.2 TEMPORIZAÇÃO: ENTREVERDES E VERDES

Os tempos de verde, amarelo e vermelho geral foram calculados pelo *Traffic Bee* obedecendo às equações sugeridas pelo DENATRAN para um cruzamento com semaforização em dois tempos. Na próxima seção (4.3) essa temporização será extrapolada para uma semaforização em quatro tempos.

4.2.1 Tempo de Entreverdes: Amarelo e Vermelho Geral

É importante ressaltar que a definição dos tempos de amarelo e vermelho geral têm suma importância no que diz respeito à segurança: o período de amarelo deve garantir que os veículos concluam sua travessia a tempo sem serem surpreendidos pelo sinal vermelho; o período de vermelho geral, todos os semáforos em vermelho, deve garantir que o cruzamento esteja totalmente desocupado antes do início da próxima travessia veicular. A teoria de Gazis (CUCCI NETO, 2015) é a mais comumente usada para calcular esses períodos em semáforos de dois tempos. Suas principais equações são apresentadas a seguir.

Entreverdes é a parte da sinalização dos semáforos programada para acontecer após o fim do intervalo de verde de cada estágio (semáforo individual), com o propósito de evitar acidentes entre os usuários de veículos automotivos que estão perdendo seu direito de passagem no cruzamento e aqueles que vão passar a adquiri-lo no estágio subsequente do ciclo de semaforização. Especificamente, o tempo de entreverdes compõe-se do tempo de amarelo seguido do tempo de vermelho geral calculado **para cada uma das vias numa sinalização em dois tempos**.

Segundo Cucci Neto (2015), a seguinte equação (1) fornece o tempo total de **entreverdes** do sistema:

$$T_{ent} = t_{pr} + \frac{v}{2 \cdot (a_{ad} \pm i \cdot g)} + \frac{d_2 + c}{v} \quad (1)$$

Onde:

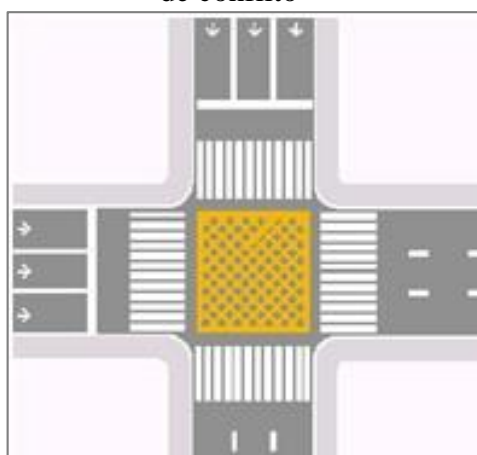
T_{ent}: Tempo de entreverdes para o grupo de veículos em questão, em segundos;

t_{pr}: Tempo de reação e percepção do condutor, em segundos;

- v:** Velocidade do veículo, em metros por segundo (m/s), geralmente tomada como a velocidade máxima permitida;
- a_{ad}:** Máxima taxa de frenagem admissível pela via plana, em m/s²;
- i:** Inclinação da via na aproximação, sendo ‘+’ em rampas ascendentes, e ‘-’ em rampas descendentes;
- g:** Aceleração da gravidade (9,8m/s²);
- d₂:** Extensão da trajetória do veículo entre a linha de retenção e o término da área de conflito, em metros;
- c:** Comprimento do veículo.

Utilizadas antes das faixas de travessia de pedestres em cruzamentos rodociclovitários, as **linhas de retenção** (Figura 17) indicam o local limite onde o condutor deve parar o veículo em caso de parada obrigatória. Essa linha descrita em sinalização horizontal na cor branca não deve estar a menos que um metro da faixa de pedestres. Já a **área de conflito** (Figura 17) corresponde a marcação horizontal em amarelo localizada centralizada no cruzamento, que serve para reforçar a proibição de parada ou estacionamento de veículos na área de interseção das vias.

Figura 17 – Linhas de retenção e área de conflito



Fonte: Repositório do Transitoideal (2018)²⁰

A equação de entreverdes (1) pode ser dividida em três parcelas, sendo a soma das duas primeiras igual ao **tempo de amarelo**, conforme mostra a próxima equação (2):

²⁰ Disponível em: <http://www.transitoideal.com/upload/tiny_mce/noticias/aplicacao4.jpg>. Acesso em Maio de 2018.

$$t_{am} = t_{pr} + \frac{v}{2 \cdot (a_{ad} \pm i \cdot g)} \quad (2)$$

O tempo de **vermelho geral** é definido pela última parcela da equação de entreverdes, conforme a equação a seguir (3):

$$t_{vg} = \frac{d_2 + c}{v} \quad (3)$$

Calculados os tempos de amarelo (t_{am}) e vermelho geral (t_{vg}) para cada via, a soma desses tempos define o respectivo tempo de entreverdes (t_{ent}). Feito isso parte-se para o cálculo dos tempos de verde.

4.2.2 Tempo de Verde

Para seguirmos em frente nos cálculos dos tempos de verde, faz-se necessário definir o **grau de saturação** de cada *link* (x_{lc}): um parâmetro que indica qual a porcentagem de veículos parados ou chegando na fila de espera atravessará o sinal verde. Quando o grau de saturação de um *link* é igual a 1 significa que seu tempo de sinal verde será suficiente para a travessia de 100% dos veículos na fila de espera, ou chegando nessa fila, daquele *link*. Assim, para grau de saturação unitário, nenhum veículo ficará para trás e qualquer outro que chegue posteriormente à fila conseguirá atravessar o cruzamento. Graus de saturação menores que a unidade indicam que haverá tempo de verde ocioso no *link*, ou seja, sem a passagem de veículos pelo cruzamento. Graus maiores que a unidade indicam que veículos da fila de espera ou chegando a esta ficarão retidos no *link* ao término do tempo de verde do mesmo.

A escolha do grau de saturação é a parte fundamental do modelo aqui aplicado. Segundo Vilanova (2015), a teoria indica que se opere numa faixa que vai de 0,8 (80%) à 0,9 (90%). Nessa faixa obtêm-se os menores valores para o tempo de vermelho (atraso total) dos semáforos.

Definido o grau de saturação, o próximo passo consiste no estudo do *link* crítico **de cada via** do cruzamento. Ainda segundo Vilanova (2015), para determinar quais são esses *links* calcula-se a taxa de ocupação y_{lc} de cada *link* conforme a equação apresentada a seguir.

$$y_{lc} = \frac{F_{lc}}{FS_{lc}} \quad (4)$$

Onde:

F_{lc} : Fluxo do *link* (quantidade de veículos passando);

FS_{lc} : Fluxo de saturação do *link* (quantidade de veículos suportada).

O maior valor para y_{lc} **em cada via** identificará o *link* crítico dessa via, isto é, o *link* que apresenta o fluxo mais saturado de veículos chegando num dos seus dois semáforos.

A porcentagem de verde necessária para cada um dos *links* críticos identificados é fornecida pela seguinte equação (5):

$$p_{lc} = \frac{y_{lc}}{x_{lc}} \quad (5)$$

Para que cada tempo de verde seja encontrado, o tempo total do ciclo de funcionamento dos semáforos deve ser determinado conforme a próxima equação (6), abaixo onde o somatório das porcentagens de verde dos *links* críticos ($\sum p_{lc}$) é considerado:

$$T_{ciclo} = \frac{T_{morto}}{1 - \sum_{lc=1}^n p_{lc}} \quad (6)$$

Onde:

T_{morto} : Somatório dos tempos de entreverdes (T_{ent}) das vias.

O tempo de verde de um *link* crítico (T_{verde}) é o mesmo para **toda a via** na qual o *link* crítico considerado está inserido, pois devido à natureza da semaforização em dois tempos, quando temos sinal verde acionado para um *link* de uma determinada via, temos obrigatoriamente também a mesma situação de sinal verde acionado para o outro *link* componente dessa mesma via (semáforos abertos nos dois sentidos de tráfego).

O tempo de verde é obtido através da equação (7) abaixo:

$$T_{verde} = p_{lc} \cdot T_{ciclo} \quad (7)$$

As variáveis F_{lc} podem chegar a valores bem elevados, próximos ou até mesmo iguais aos parâmetros de saturação máxima (FS_{lc}) dos links. Quando isso acontece, o ciclo total de funcionamento dos semáforos pode exceder o tempo máximo determinado pelo engenheiro de trânsito responsável, provocando tempos de espera (sinal vermelho) excessivos.

Em casos assim, novas porcentagens de verde (p_{lc}) devem ser calculadas obedecendo às equações 8 e 9. A primeira garante que os tempos de verde caibam dentro do ciclo máximo e a segunda viabiliza a proporcionalidade dos tempos em relação aos graus de saturação (x_{lc}) desejados. Por exemplo, sendo os link A e C considerados críticos, essas equações ficariam assim:

$$p_A + p_C = 1 - T_{morto} \quad (8)$$

$$\frac{\frac{y_A}{p_A}}{\frac{y_C}{p_C}} = \frac{x_{A,desejado}}{x_{C,desejado}} \quad (9)$$

4.3 SIMULAÇÕES E RESULTADOS

Em sua maior parte, os resultados obtidos nas simulações com o protótipo *Traffic Bee* atenderam à expectativa de funcionamento da proposta. Sempre que solicitado pelo controlador 1, o controlador 2 executava o método aleatório (*random*) da linguagem de programação para Arduino, com fins de gerar uma quantidade randômica de veículos para cada faixa de rodagem de cada *link*. O valor total produzido aleatoriamente por *link* consistia no fluxo do *link* (F_{lc}).

De posse dessas quantidades, o controlador 1 (coordenador da rede) executava os cálculos das temporizações de entreverdes e verdes para os semáforos do cruzamento considerados como acionados em dois tempos. Em seguida essas temporizações eram extrapoladas para uma sinalização em quatro tempos, como será explanado mais adiante.

Para o cenário real (Figura 13, página 40), foi observado no horário de fluxo intenso de trânsito (horário de pico), das 12h às 13h, durante o mês de dezembro de 2017, a temporização para os quatro semáforos de dois tempos ali instalados. Dessas observações de campo os seguintes valores foram coletados para serem aplicados na simulação do *Traffic Bee* via protótipo:

- I. Velocidade máxima regulamentada, $v = 60$ quilômetros por hora (Km/h), ou 16,7m/s, para todos os *links* das duas avenidas;
- II. Distância entre a linha de retenção da Avenida Senador Salgado Filho e o término da área de conflito com a Avenida Bernardo Vieira, $d_2 = 33,5\text{m}$;
- III. Distância entre a linha de retenção da Avenida Bernardo Vieira e o término da área de conflito com a Avenida Senador Salgado Filho, $d_2 = 32,5\text{m}$;
- IV. 4 segundos de intervalo de amarelo para todos os sentidos do cruzamento;
- V. 7 segundos de vermelho geral ao final de todos os intervalos de amarelo;
- VI. 1 minuto e 48 segundos de sinal verde para o *link* C;
- VII. 1 minuto e 48 segundos de sinal verde para o *link* D;
- VIII. 48 segundos de sinal verde para o *link* A;
- IX. 48 segundos de sinal verde para o *link* B.

As distâncias d_2 (itens II e III) entre as linhas de retenção de cada avenida e os seus correspondentes terminos da área de conflito foram obtidos analisando-se as imagens dessas vias através do serviço *Google Maps*.

Além disso, aplicamos os valores usualmente adotados nos estudos de tráfego, e abaixo descritos, para cada dos seguintes parâmetros usados nos cálculos das temporizações:

- $t_{pr} = 1$ segundo;
- $a_{ad} = 3\text{m/s}^2$;
- $i = 0$ grau de inclinação;
- $c = 5$ metros.

A partir da extensão das vias (260 metros), obtida também por meio do *Google Maps*, do número de faixas de rodagem em cada uma delas (3) e do tamanho médio do veículo ($c = 5$ metros) calculamos uma capacidade total de 156 ($=260 \cdot 3/5$) veículos para cada via de tráfego. Esse valor corresponde ao fluxo de saturação (FS_{lc}) comum para as quatro vias.

Utilizamos graus de saturação (X_{lc}) de 0,85 (85%) e 0,90 (90%) para as Avenidas Bernardo Vieira e Senador Salgado Filho, respectivamente, valores esses escolhidos livremente de acordo com as recomendações apresentadas na seção anterior.

Com os dados v e d_2 extraídos do cenário real e os valores fixados para t_{pr} , a_{ad} , i , e c obtivemos na simulação via protótipo o valor de 4 segundos para o tempo de amarelo (t_{am}), acrescido de 2 segundos para o tempo de vermelho geral (t_{vg}), gerando-se um tempo de entreverdes (T_{ent}) igual a 6 segundos para cada uma das vias do cruzamento.

Os valores calculados para os tempos de entreverdes foram os mesmos devido ao fato de que o único parâmetro que realmente muda entre as vias na aplicação das equações de temporização é a distância entre as linhas de retenção e o término da área de conflito (d_2) para cada uma delas, onde a diferença entre os dois valores reais medidos é de apenas 1m.

De posse dos tempos de entreverdes, dos valores fixados para FS_{lc} e X_{lc} , e dos valores gerados aleatoriamente para F_{lc} , calculamos os tempos de verde para cada via. Partindo desse último cálculo, o *Traffic Bee* extrapola esses tempos de verde das vias para cada um dos *links* que as compõem, dividindo o tempo de verde da via em parcelas proporcionais à porcentagem de veículos daquela via presentes no *link* dela considerado.

Por exemplo, numa das simulações com geração aleatória das quantidades de veículos a via Avenida Senador Salgado Filho apresentou um tempo de verde igual a 1 minuto e 20 segundos (80 segundos), esse tempo foi dividido em duas parcelas proporcionais aos fluxos de *link*: o *link* D com 30% dos veículos presentes na via teve seu tempo de verde igual à 24 segundos ($0,3 \cdot 80$ segundos), já o *link* E com 70% dos veículos presentes na via teve seu tempo de verde igual à 56 segundos ($0,7 \cdot 80$ segundos).

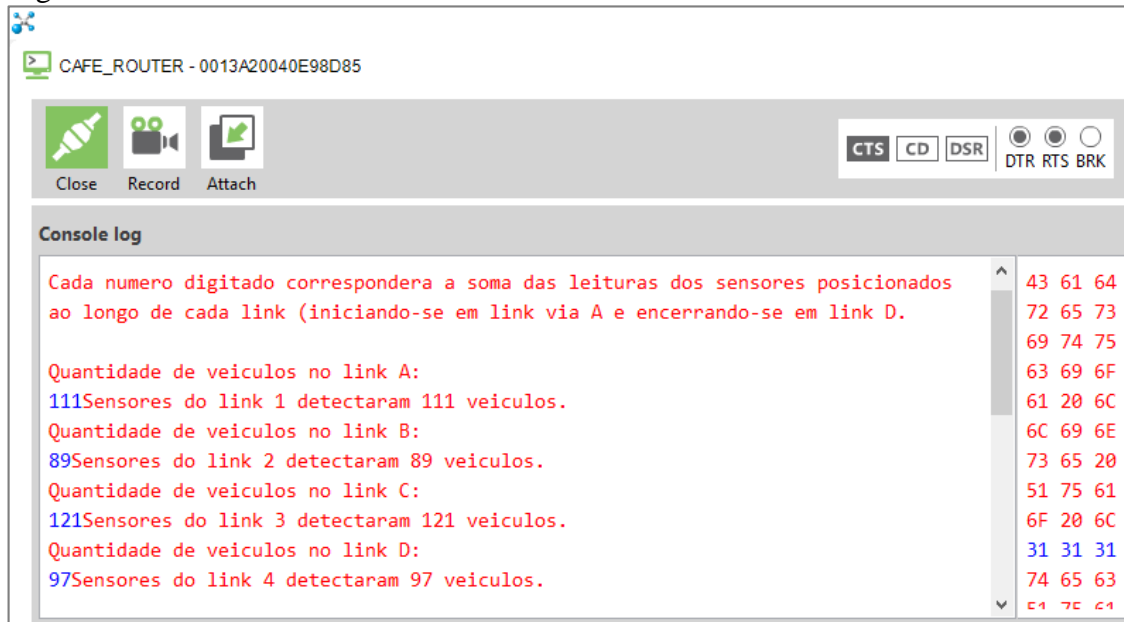
Ainda tomando a simulação anterior como exemplo, a via Avenida Bernardo Viera apresentou um tempo de verde igual a 1 minuto (60 segundos), esse tempo foi dividido em duas parcelas proporcionais aos fluxos de *link*: o *link* A com 67% dos veículos presentes na via teve seu tempo de verde aproximado para 40 segundos ($0,67 \cdot 60 = 40,2$ segundos), já o *link* B com 33% dos veículos presentes na via teve seu tempo de verde aproximado para 20 segundos ($0,33 \cdot 60 = 19,6$ segundos).

Finalizando a extrapolação da semaforização das vias para um sistema de quatro tempos, para **cada um** dos quatro semáforos do cruzamento associamos ao seu respectivo tempo de verde o valor do tempo de entreverdes anteriormente calculado para aquela via, da qual o semáforo considerado faz parte, fechando assim a temporização total de sinalização daquele estágio. No caso real estudado ficamos com um tempo morto (T_{morto}) de 24 segundos, igual a quatro vezes o valor comum de entreverdes (6 segundos) anteriormente calculado para as vias.

Num segundo momento das experimentações com o protótipo, simulando o mesmo cruzamento real, conectamos um dos módulos RF XBee ao programa XCTU por meio do adaptador USB XBee Explorer. O console de comando foi iniciado e os valores de fluxo de *link* (F_{lc}) puderam ser fornecidos de forma determinística ao controlador 1, responsável por controlar a temporização dos semáforos.

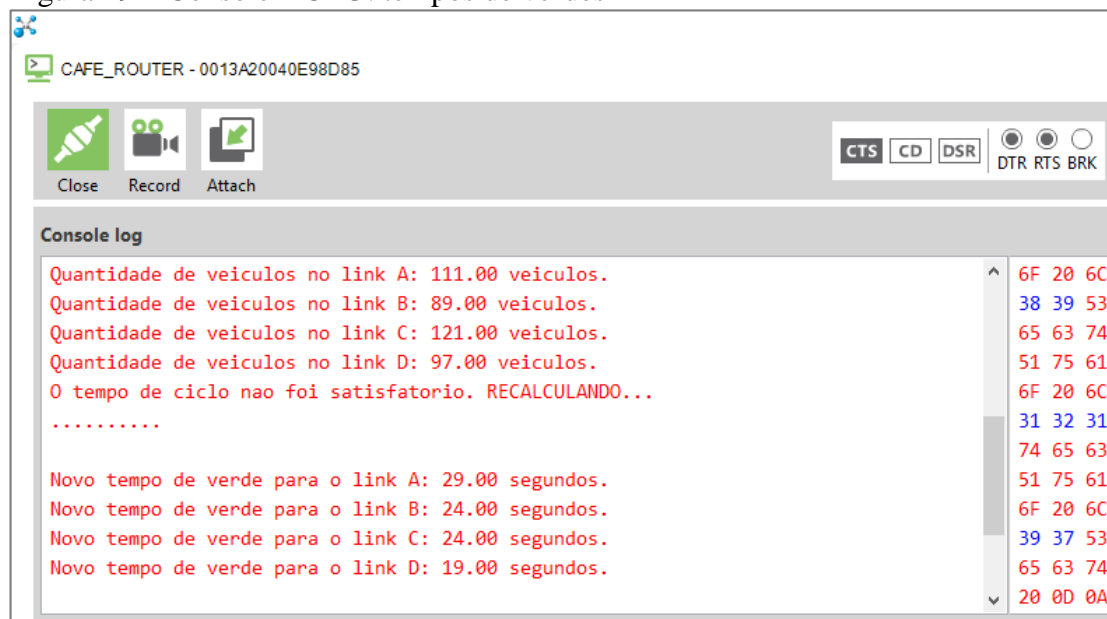
Na tela apresentada pela Figura 18, aparecem digitados no console do XCTU os seguintes valores para as quantidades de veículos (F_{1c}) de cada *link*: 111, 89, 121 e 97. Os resultados encontrados para os tempos de verde produzidos a partir dessa sequência de valores foram apresentados pelo programa conforme mostra a tela capturada na Figura 19.

Figura 18 – Console XCTU: fornecendo valores determinísticos



Fonte: Elaborado pelo autor em 2018

Figura 19 – Console XCTU: tempos de verdes



Fonte: Elaborado pelo autor em 2018

Para estas quantidades de veículos obteve-se um tempo total de 120 segundos (dois minutos) para a realização de um ciclo completo de sinalização no cruzamento simulado. Os

valores para os tempos de verde descritos na figura acima demonstram a correta operação do *Traffic Bee* no que diz respeito à implementação dos cálculos de temporização para cada semáforo: basta somar todos os tempos de verde, 96 segundos no total, que ao ser adicionado aos quatro tempos de entreverdes, tempo morto de 24 segundos, obtém-se exatamente valor esperado para o tempo total do ciclo de funcionamento dos semáforos, dois minutos.

Concluindo as experimentações com o protótipo fizemos, alguns teste de alcance do sinal propagado com a correta transmissão de dados via tecnologia Zigbee entre os módulos roteador e coordenador, controle 2 e 1 respectivamente.

Usando o método de visada em linha direta, denominado simplesmente de **visada direta**, entre as antenas (interfaces de propagação) dos módulos Zigbee, onde se garante a ausência de quaisquer obstáculos entre essas interfaces, encontramos uma distância máxima para correta transmissão de 95 metros em média, tendo como alcance mínimo o valor de 90 metros e máximo de 100 metros.

Em 90 metros todas as mensagens entre os módulos trafegaram sem perdas, demonstrando o perfeito funcionamento do canal de comunicação. Aos 95 metros de distância foi detectada a perda de 10% das mensagens transmitidas. Para 100 metros apenas 10% das transmissões realizadas ocorreram corretamente sem perda de dados.

5 CONCLUSÕES E PERSPECTIVAS FUTURAS

O *Traffic Bee* não consiste num produto final, mas sim numa proposta para o uso da tecnologia sem fio Zigbee em ambientes mais hostis como aqueles onde se encontram os semáforos urbanos, proporcionando novas aplicações tecnológicas a outras áreas de estudo como a Engenharia Urbana e a Análise de Sistemas de Tráfego.

O protocolo Zigbee ganhou notoriedade em relação aos seus concorrentes mediante sua simplicidade, baixo custo e eficiência, fazendo-o uma excelente alternativa para pôr em prática as novas ideias relativas ao conceito das cidades inteligentes, promovendo um uso mais eficiente dos tempos de sinalização semafórica. Desse modo, a proposta *Traffic Bee* corrobora com o fato da comunicação sem fio poder ser utilizada nas mais diversas áreas da vida humana sem que isso represente exagerado custo financeiro de implementação e manutenção, e sem abrir mão da confiabilidade.

Quanto aos resultados alcançados via protótipo, fosse alimentado a variável quantidade de veículos de forma aleatória ou determinística, todos se mantiveram dentro do comportamento matemático esperado para as equações implementadas, também demonstrando uma correta execução operacional no acendimento das luzes semafóricas (LEDs) de acordo com as temporizações calculadas.

As diferenças encontradas entre as temporizações obtidas no protótipo e as observadas no cenário real para a sinalização das vias em estudo também se situaram dentro do esperado, devido às considerações mais amplas de fatores relacionados ao tráfego de veículos e a aplicação de modelos mais complexos num espaço de trânsito real.

No campo das perspectivas futuras, o *Traffic Bee* pode ser implementado como protótipo ou num cenário real por meio de *hardwares* mais robustos, pois a versão UNO do Arduino não se mostrou suficiente para implementar todas as funções inicialmente propostas no projeto, tais como as temporizações destinadas à horários especiais do dia. Dispositivos mais robustos, como o Arduino DUE, que possui maior capacidade de memória e processamento, ou o Raspberry Pi, que pode ser considerado um computador de micro tamanho, robusto para esse tipo de aplicação, são dois bons candidatos.

Para determinar os tempos de abertura e fechamentos dos semáforos, outras alternativas de cálculo podem ser utilizadas, incluindo a aplicação de métodos mais complexos que contam com um número maior de variáveis e parâmetros. Além disso, métodos de decisão inteligente mais avançados como as redes neurais e a aplicação de estudos avaliativos fundamentados na Teoria de Filas também auxiliarão na otimização da proposta.

Finalizando as sugestões de aprimoramento do *Traffic Bee* acrescentamos a configuração da operação dos dispositivos Zigbee em modo API, que possibilitaria uma alcance maior da comunicação sem fio, segundo o fabricante podendo chegar a mais que um quilômetro de distância. Ou a manutenção do modo Transparente adicionando-se à rede dispositivos auxiliares destinados somente ao roteamento de mensagens.

REFERÊNCIAS

AKISHINO, Pedro. Semáforos. In: AKISHINO, Pedro. **Algumas Técnicas de Engenharia de Tráfego na Redução de Prevenção de Acidentes de Trânsito**. Paraná: Universidade Federal do Paraná, 2004. Cap. 5. p. 139-164.

AQUINO, Carol. Dos Barris à Paralela: 88 semáforos inteligentes vão diminuir tempo no trânsito. **CORREIO**, Bahia: maio de 2017. Disponível em: <<http://www.correio24horas.com.br/noticia/nid/dos-barris-a-paralela-88-semaforos-inteligentes-va-diminuir-tempo-no-transito/>>. Acesso em: 26 jul. 2017.

AZEVEDO, Carlos Manoel Oliveira. **Comando e Monitorização de Sistemas de Actuação via Redes Wireless: Zigbee**. 2010. 120 f. Dissertação (Mestrado) - Curso de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, 2012.

BAZZAN, Ana L. C. **Sistemas Inteligentes de Transporte e Tráfego**: Rio Grande do Sul: UFRGS, 2007. 67 slides, color. Disponível em: <http://www.inf.ufrgs.br/~bazzan/downloads/jai_intro.pdf.gz>. Acesso em: 2 jan. 2018.

BRASIL. Resolução nº 680, de 27 de junho de 2017. Aprova o Regulamento sobre Equipamentos de Radiocomunicação de Radiação Restrita, e dá outras providências. **Diário Oficial da União**. Brasília, DF: Poder Legislativo, 29 jun. 2017. n. 123, Seção 1, p. 4. Disponível em: <<http://pesquisa.in.gov.br/imprensa/jsp/visualiza/index.jsp?data=29/06/2017&jornal=1&pagina=4&totalAr>>. Acesso em: 3 jan. 2018.

CUCCI NETO, João. **O cálculo do entreverdes conforme o Manual Brasileiro de Sinalização de Trânsito**. 2015. Disponível em: <http://meusite.mackenzie.br/professor_cucci/leituras.htm>. Acesso em: 3 jan. 2018.

DA SILVEIRA, Igor Machado. **Avaliação da Mobilidade Urbana com Ênfase na Condição da Acidentalidade do Usuário Vulnerável**. 2016, 93 f. Dissertação (Mestrado em Engenharia Civil) – Universidade de São Paulo, São Paulo, 2016. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18144/tde-16062016-171651/pt-br.php>>. Acesso em: 26 jul. 2017.

FEITOSA JÚNIOR, Paulo; FRANCISCO, Lucas; TREVISANI, Kleber Manrique. **Desenvolvimento de um controlador de dispositivos para automação residencial utilizando tecnologia de rede Zigbee**. Colloquium Exactarum, SP, v. 5, n. 1, p.57-66, jun. 2013.

Frota de veículos. In: **DENATRAN**, 2016. Disponível em: <<http://www.denatran.gov.br/estatistica/237-frota-veiculos>>. Acesso em: 25 de jul. 2017.

GRANDELLE, Renato. Cidades buscam novas políticas para combater trânsito caótico e reduzir gases-estufa. **O Globo**, Rio de Janeiro: jan. 2015. Disponível em: <<https://oglobo.globo.com/sociedade/sustentabilidade/cidades-buscam-novas-politicas-para-combater-transito-caotico-reduzir-gases-estufa-15148344>>. Acesso em: 24 mar. 2018.

NAKAZIMA, Dario. **Zigbee**. 2014. 20 f. Dissertação (Pós-graduação) – Curso de Tecnologia em Redes de Computadores, Pontifícia Universidade Católica de Minas Gerais, Minas Gerais, 2014.

PORTANTE, André Luis Fadinho; FRESSATTI, Wyllian. Veículo Controlado por Arduino Através do Sistema Operacional Android. **SEINPAR**, Paraná: 6 f. Set. 2015. Disponível em: <http://web.unipar.br/~seinpar/2015/_include/artigos/André_Luis_Fadinho_Portante.pdf>. Acesso em: 3 jan. 2018.

REGO, Rosana Cibely B.; SEMENTE, Rodrigo S. Sistema de controle de semáforo baseado na densidade de tráfego. **Anais do Encontro de Computação do Oeste Potiguar**. Pau dos Ferros: Universidade Federal Rural do Semi-Árido, 2017. 8 f. Disponível em: <<https://periodicos.ufersa.edu.br/index.php/ecop/article/view/7073>>. Acesso em: 2 jan. 2018.

REIS, Fábio dos. **Enviando e recebendo dados com Serial Monitor no Arduino**. 2015. Disponível em: <<http://www.bosontreinamentos.com.br/electronica/arduino/enviando-e-recebendo-dados-com-serial-monitor-no-arduino/>>. Acesso em: 3 jan. 2018.

RF Kits Common. In: **DIGI INTERNATIONAL INC.**, 2015. Disponível em: <<http://docs.digi.com/display/RFKitsCommon/RF+Kits+Common>>. Acesso em: 3 jan. 2018.

Semáforos inteligentes. In: **ESCOLA POLITÉCNICA DA USP**. São Paulo, 2012. Disponível em: <<http://www.poli.usp.br/fr/comunicacao/noticias/arquivo-de-noticias/1088-semaforos-inteligentes.html>>. Acesso em: 3 jan. 2018.

SIEMEINTCOSKI, Michael Eberle. **Estudo e implementação de tecnologia sem fio usando protocolo Zigbee**. 2009. 148 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Centro de Ciências e Tecnologias, Universidade Regional de Blumenau, Blumenau - SC, 2009.

SILVEIRA, Igor Machado da. **Avaliação da mobilidade urbana com ênfase na condição da acidentalidade do usuário vulnerável**. 2016. 93 f. Dissertação (Mestrado) - Curso de Pós-graduação em Engenharia de Transportes, Departamento de Transportes (STT/EESC/USP), Universidade de São Paulo, São Carlos, 2016.

SOUZA, Fábio. **Arduino UNO**. 2013. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em: 3 jan. 2018.

STOLFI, H. Daniel; ALBA, Enrique. Red Swarm: Reducing travel times in smart cities by using bio-inspired algorithms. **Applied Soft Computing**, Spanha, v. 24, p. 181-195, 2014.

TORRES DOS SANTOS, Sérgio. **Redes de Sensores sem Fio em Monitoramento e Controle**. 2007. 67 f. Dissertação (Pós-graduação) – Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007.

VILANOVA, Luis. **Programação de um semáforo usando o método do grau de saturação**. 2013. Disponível em: <<http://www.sinaldetransito.com.br/artigos/saturacao.pdf>>. Acesso em: 3 jan. 2018.