

- Now lets see cases of splicing

I have table ofc_cables

```
create table public.ofc_cables (  
  id uuid not null default gen_random_uuid (),  
  route_name text not null,  
  sn_id uuid not null,  
  en_id uuid not null,  
  ofc_type_id uuid not null,  
  capacity integer not null,  
  ofc_owner_id uuid not null,  
  current_rkm numeric(10, 3) null,  
  transnet_id text null,  
  transnet_rkm numeric(10, 3) null,  
  asset_no text null,  
  maintenance_terminal_id uuid null,  
  commissioned_on date null,  
  remark text null,  
  status boolean null default true,  
  created_at timestamp with time zone null default now(),  
  updated_at timestamp with time zone null default now(),  
  constraint ofc_cables_pkey primary key (id),  
  constraint ofc_cables_en_id_fkey foreign KEY (en_id) references nodes  
(id),  
  constraint ofc_cables_maintenance_terminal_id_fkey foreign KEY  
(maintenance_terminal_id) references maintenance_areas (id),  
  constraint ofc_cables_ofc_owner_id_fkey foreign KEY (ofc_owner_id)  
references lookup_types (id),  
  constraint ofc_cables_ofc_type_id_fkey foreign KEY (ofc_type_id)  
references lookup_types (id),  
  constraint ofc_cables_sn_id_fkey foreign KEY (sn_id) references nodes (id)  
) TABLESPACE pg_default;  
  
create index IF not exists idx_ofc_cables_remark_fts on public.ofc_cables  
using gin (to_tsvector('english'::regconfig, remark)) TABLESPACE pg_default;  
  
create trigger ofc_cables_log_trigger  
after INSERT  
or DELETE  
or  
update on ofc_cables for EACH row  
execute FUNCTION log_data_changes ();  
  
create trigger trigger_ofc_cables_updated_at BEFORE  
update on ofc_cables for EACH row  
execute FUNCTION update_updated_at_column ();
```

and other table ofc_connections which gives connections/details for each ofc_cables

```

create table public.ofc_connections (
    id uuid not null default gen_random_uuid (),
    ofc_id uuid not null,
    fiber_no_sn integer not null,
    fiber_no_en integer not null,
    otdr_distance_sn_km numeric(10, 3) null,
    sn_dom date null,
    sn_power_dbm numeric(10, 3) null,
    system_id uuid null,
    otdr_distance_en_km numeric(10, 3) null,
    en_dom date null,
    en_power_dbm numeric(10, 3) null,
    route_loss_db numeric(10, 3) null,
    logical_path_id uuid null,
    fiber_role text null,
    path_segment_order integer null default 1,
    source_port text null,
    destination_port text null,
    connection_category text not null default 'OFC_JOINT_TYPES'::text,
    connection_type text not null default 'straight'::text,
    remark text null,
    status boolean null default true,
    created_at timestamp with time zone null default now(),
    updated_at timestamp with time zone null default now(),
    constraint ofc_connections_pkey primary key (id),
    constraint fk_connection_type foreign KEY (connection_category,
connection_type) references lookup_types (category, name),
    constraint fk_ofc_connections_logical_path foreign KEY (logical_path_id)
references logical_fiber_paths (id) on delete set null,
    constraint fk_ofc_connections_system foreign KEY (system_id) references
systems (id) on delete set null,
    constraint ofc_connections_ofc_id_fkey foreign KEY (ofc_id) references
ofc_cables (id),
    constraint ofc_connections_fiber_role_check check (
        (
            fiber_role = any (array['working'::text, 'protection'::text])
        )
    )
) TABLESPACE pg_default;

create index IF not exists idx_ofc_connections_ofc_id on
public.ofc_connections using btree (ofc_id) TABLESPACE pg_default;

create index IF not exists idx_ofc_connections_system_id on
public.ofc_connections using btree (system_id) TABLESPACE pg_default;

create index IF not exists idx_ofc_connections_logical_path_id on
public.ofc_connections using btree (logical_path_id) TABLESPACE pg_default;

create index IF not exists idx_ofc_connections_remark_fts on
public.ofc_connections using gin (to_tsvector('english'::regconfig, remark))
TABLESPACE pg_default;

create trigger ofc_connections_log_trigger
after INSERT
or DELETE

```

```

or
update on ofc_connections for EACH row
execute FUNCTION log_data_changes ();

create trigger trigger_ofc_connections_updated_at BEFORE
update on ofc_connections for EACH row
execute FUNCTION update_updated_at_column ();

create trigger trigger_update_en_dom_on_otdr_change BEFORE
update on ofc_connections for EACH row
execute FUNCTION update_en_dom_on_otdr_change ();

create trigger trigger_update_sn_dom_on_otdr_change BEFORE
update on ofc_connections for EACH row
execute FUNCTION update_sn_dom_on_otdr_change ();

```

Navigation

- Home
- Employees
- Base Structure
- Ofc & Routes
 - Optical Fiber Cable
 - RouteManager
- Systems
- Diagrams
- BTS Map
- Quick Actions
 - Upload Data FOR: ofc_cables TABLE
 - Upload Excel

OFC Cable Management

Manage OFC cables and their related information.











182
Total OFC Cables

180
Active

2
Inactive

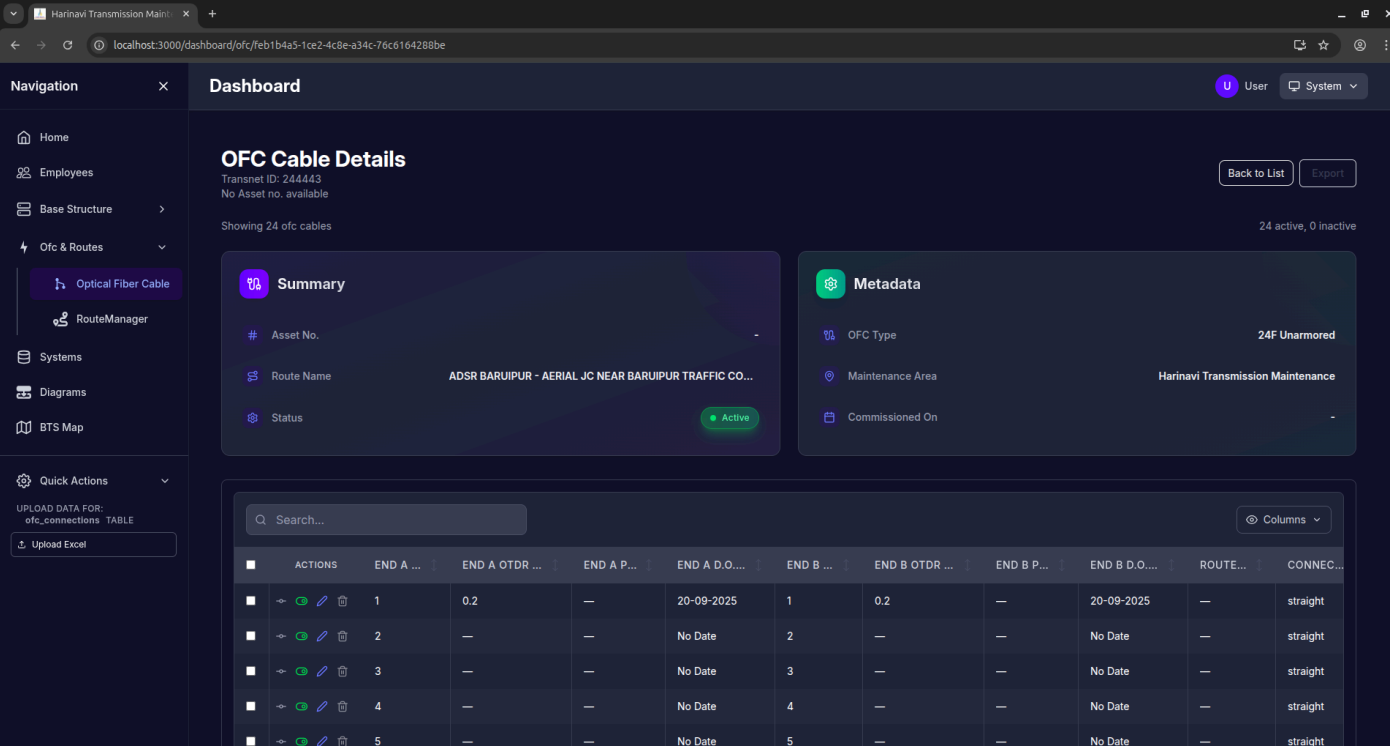
Search by Asset No or Route Name or Transnet ID...

Filters

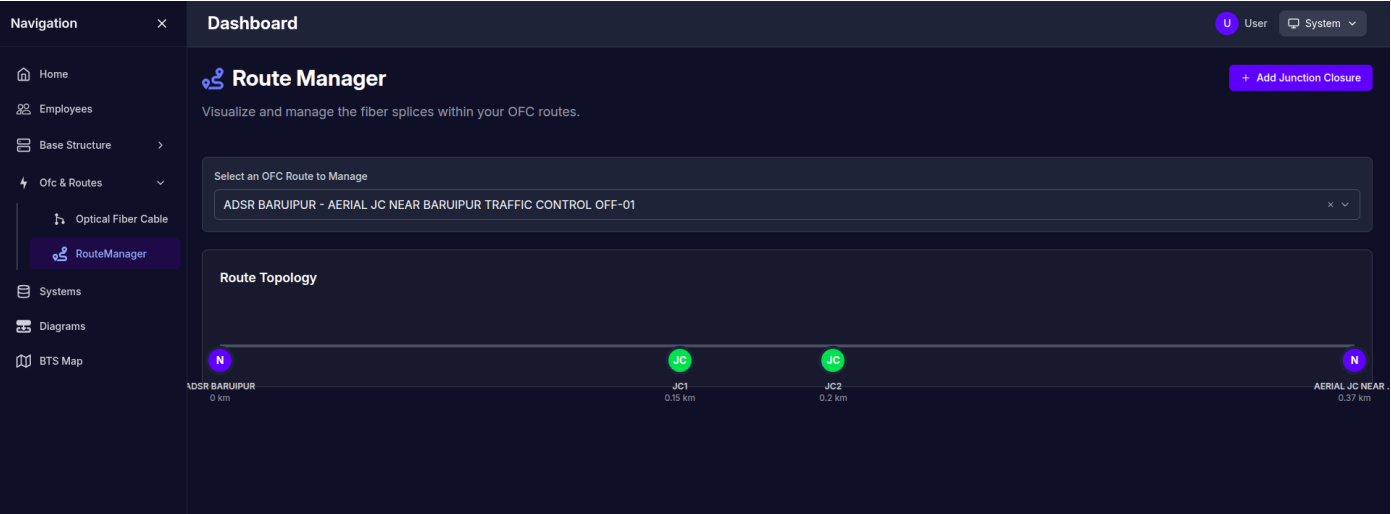
ACTIONS	ROUTE	CAPACITY	TRANSNE...	TRANSNE...	CURRENT...	ASSET NO	COMMISSIO...	REMARK	OFC TYPE...
	ADSR BARUIPUR - AERIAL JC NEAR BARUIP...	24	244443	0.37	0.37		No Date	—	24F-U
	AP NAGAR - JC 2 NEAR AP NAGAR	24	—	—	1.2		No Date	—	24F-U
	AP NAGAR - SNP GHASIARA BTS-01	24	154986	1.74	1.75		29-12-2010	—	24F-U
	AP NAGAR - SONARPUR ROAD BAZAR BTS-01	24	198606	0.93	0.91	9905006215...	31-08-2018	—	24F-U
	AP NAGAR - HNV PCM 1ST FLOOR-01	24	174474	9.52	21.21		09-02-2003	—	24F-U
	AP NAGAR - SITAKUNDU-01	24	170323	16.41	16.46		23-06-2003	—	24F-U
	AP NAGAR - SONARPUR-01	24	154982	1.85	1.85		09-02-2003	—	24F-U
	BANTALA - SONARPUR-01	24	175209	—	13.3		No Date	—	24F-U
	BARENDRA PARA BTS - SONARPUR	24	—	—	0.45	9905006385...	No Date	—	24F-U
	BARUIPUR - GOBINDAPUR BYPASS J/C	24	—	—	6.2		No Date	—	24F-U

Showing 1 to 10 of 182 results | 10 per page

On clicking view button on ofc_cables page as shown above picture,we go to ofc connections/details page as in below picture. On clicking view and going to ofc connections/details page, we create fibers equal to capacity of cable(if already not created).



in route manager we can select any ofc_cables from dropdown. Then add as much JC as required. Example shown in below image



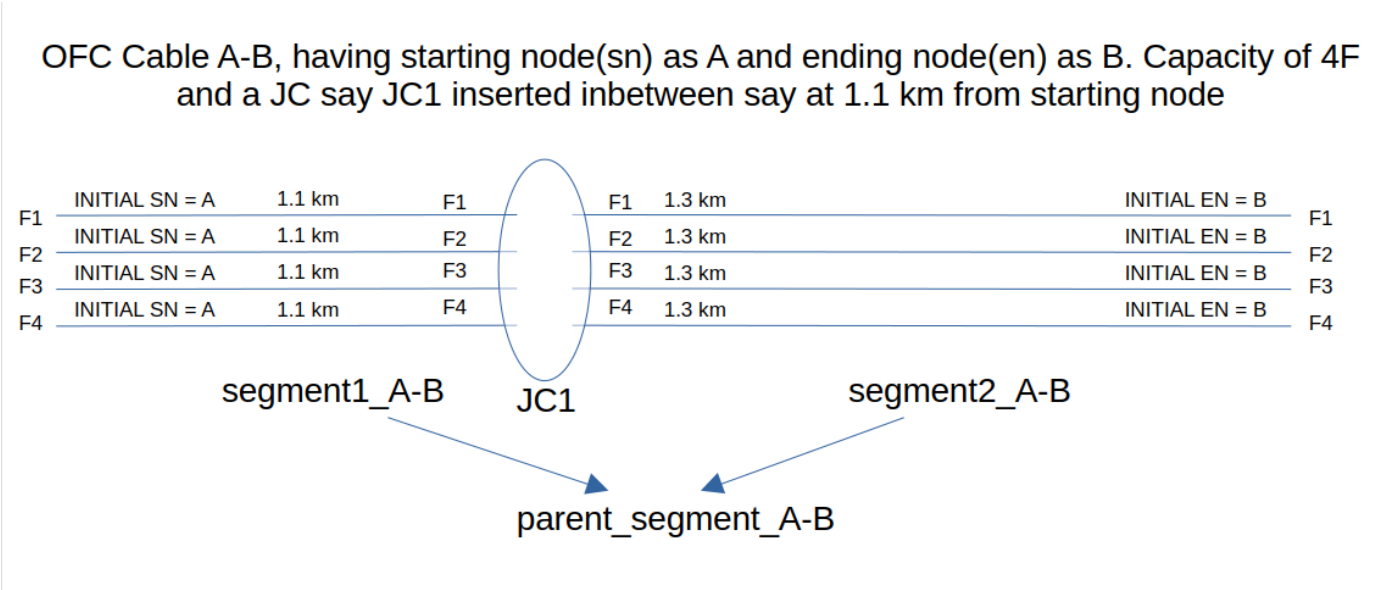
Above things are already implemented.

• Now, Let us see what we want to implement/achieve.

Suppose We are having a cable A-B of capacity 4 fiber and distance 2.4 Km. Then its ofc connections can be as in below picture.

OFC Cable A-B, having starting node(sn) as A and ending node(en) as B. Capacity of 4F			
F1	INITIAL SN = A	2.4 km	INITIAL EN = B
F2	INITIAL SN = A	2.4 km	INITIAL EN = B
F3	INITIAL SN = A	2.4 km	INITIAL EN = B
F4	INITIAL SN = A	2.4 km	INITIAL EN = B
parent_segment_A-B			

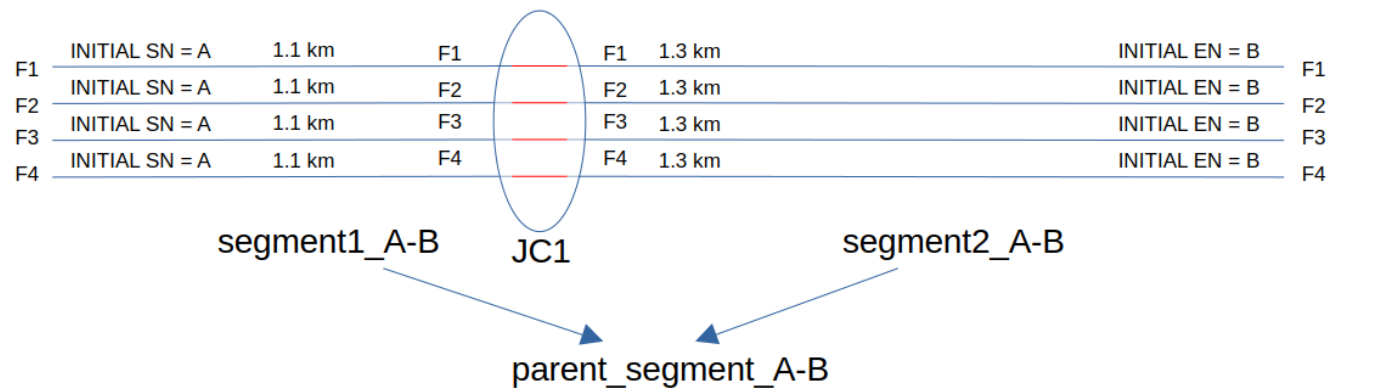
Initially ofc connections will have a default segment path and initial starting and ending node as ofc cable. Now if JC is added in-between, then fibers(ofc connections) will breaks at that JC and two new segments will be created with parent segment as initial one. Initial fiber length for the segments will also get updated



Now lets see cases of splicing

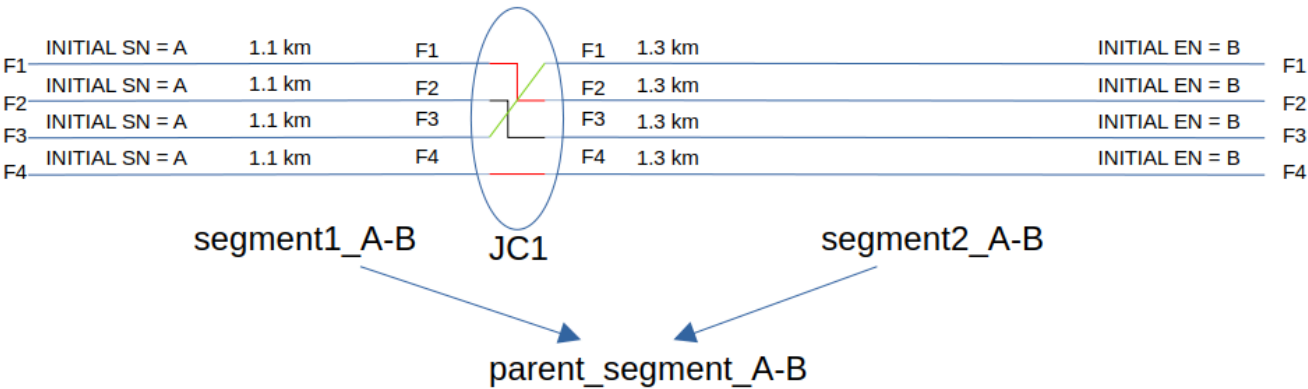
- Case 1: Straight joint to all fibers at JC. This will result in no change in sn,en and Fiber numbers of the ofc_connections as shown below

OFC Cable A-B, having starting node(sn) as A and ending node(en) as B. Capacity of 4F and a JC say JC1 inserted inbetween say at 1.1 km from starting node. All fibers are spliced as straight joint.



- Case 2: Cross joint between fibers at JC. This will result in no change in sn,en but change in Fiber numbers of the ofc_connections as shown below

OFC Cable A-B, having starting node(sn) as A and ending node(en) as B. Capacity of 4F and a JC say JC1 inserted inbetween say at 1.1 km from starting node. All fibers are spliced as crossed and straight joint.

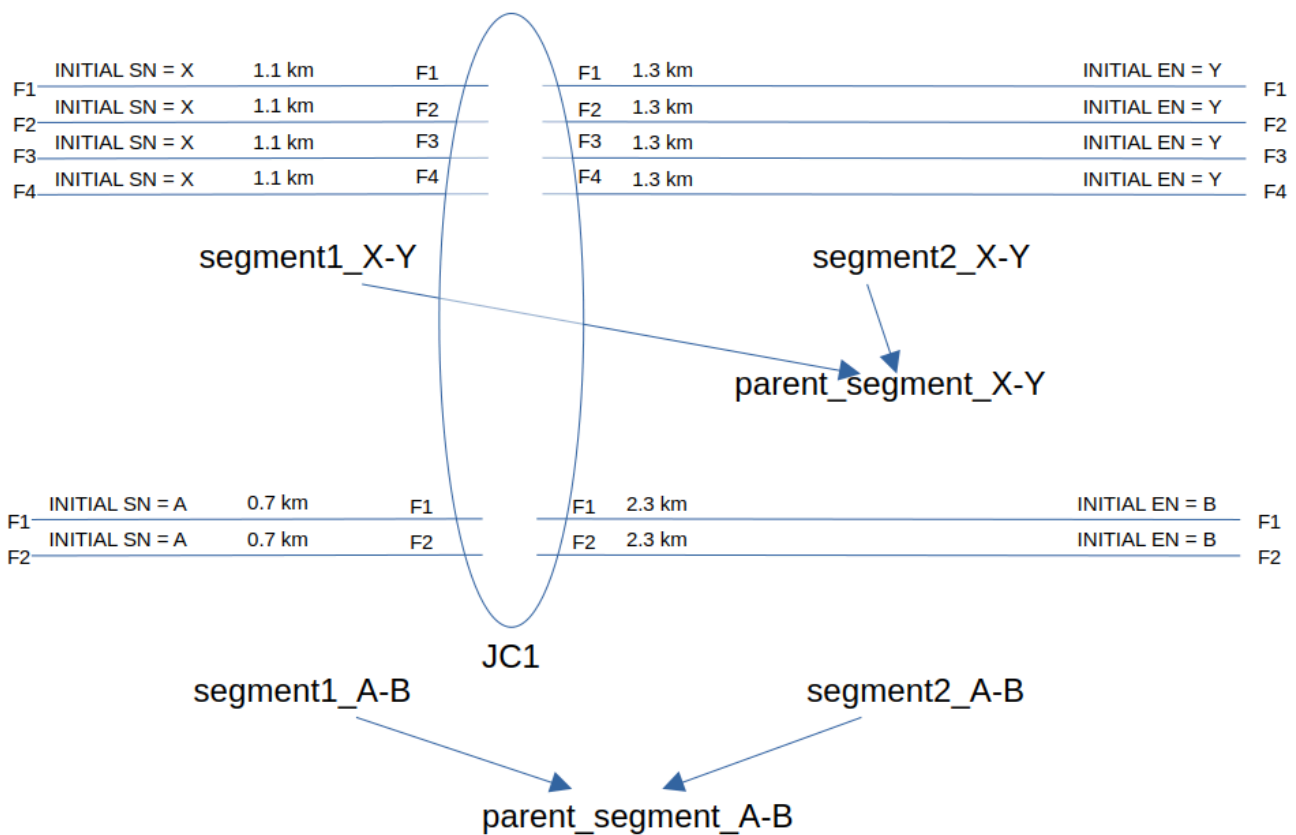


Here end to end fiber numbers in ofc_connections table will change as

sn Fiber Number	en fiber number
1	2
2	3
3	1
4	4

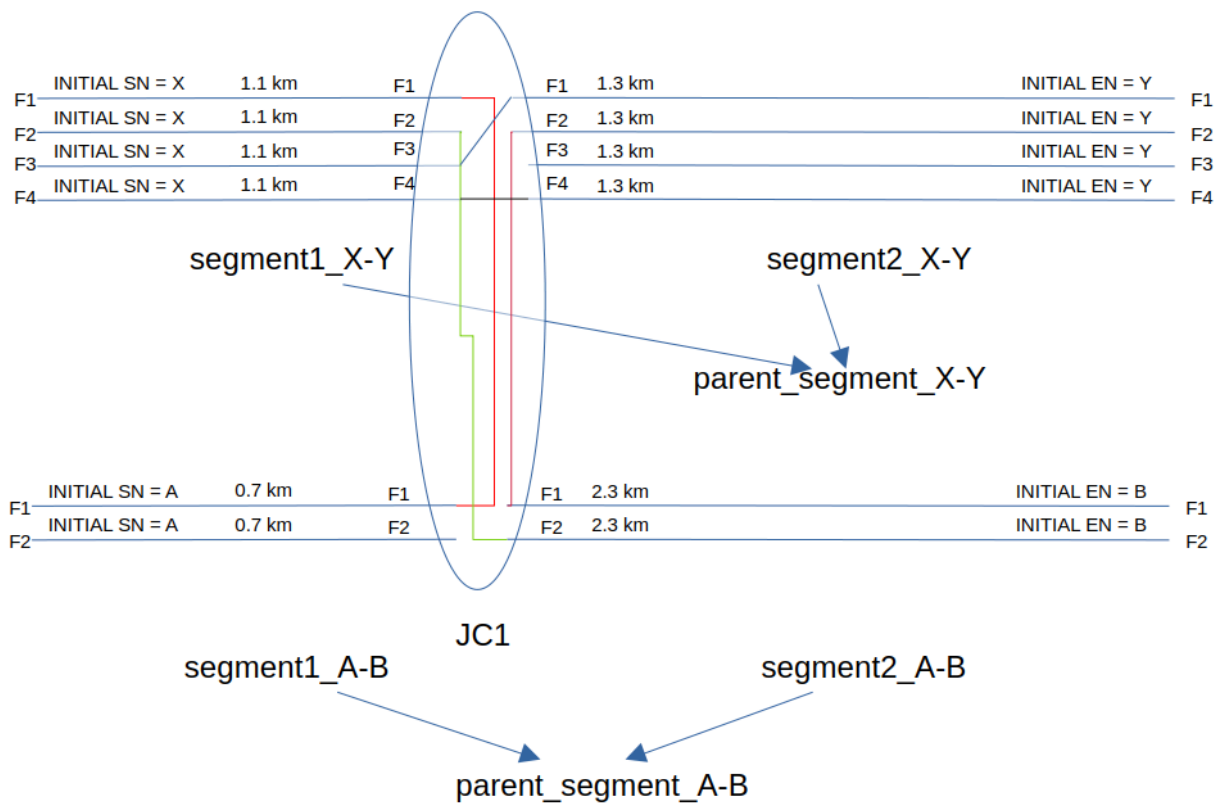
- Case 3: If same JC introduced on another cable, then two segments will be created for that also. For example, say there are two cables X-Y (capacity 4) and A-B (capacity 2). then segments will be

OFC Cable X-Y and A-B, having starting node(sn) as X and A & ending node(en) as Y and B respectively. Capacity of 4F and 2F respectively. A JC say JC1 inserted inbetween say at 1.1 and 0.7 km respectively from starting node. All fibers are spliced as crossed and straight joint.



Lets assume that splicing is as below:

F1 of segment 1 of X-Y is spliced with F1 of segment 1 of A-B. F2 of segment 1 of X-Y is spliced with F2 of segment 2 of A-B. F3 of segment 1 of X-Y is spliced with F1 of segment 2 of X-Y. F4 of segment 1 of X-Y is spliced with F4 of segment 2 of X-Y. F3 of segment 2 of X-Y remains unspliced. F2 of segment 1 of A-B remains unspliced.



so, here will be end to end fiber numbers in ofc_connections tables

ofc connections table for X-Y

sn	sn fiber number	en	en fiber number
X	1	A	1
X	2	B	2
X	3	Y	1
X	4	Y	4
		Y	3

ofc connections table for A-B

sn	sn fiber number	en	en fiber number
A	1	X	1
A	2		

sn	sn fiber number	en	en fiber number
Y	2	B	1
X	2	B	2

Now here I stuck as ofc connections table for X-Y has only 4 rows(equal fo fiber capacity) and ofc connections table for A-B have only two rows. How to solve/proceed.